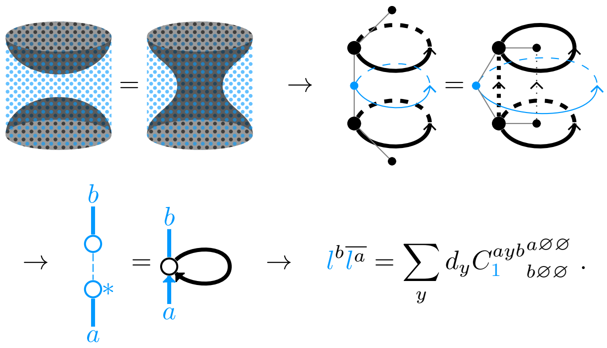
TOPOLOGICAL PHASES, FIXED-POINT MODELS, EXTENDED TQFT, AND FAULT-TOLERANT QUANTUM COMPUTATION – TENSORS IN SPACETIME



=
→
=
→
b
*
a
=
b
a
→
$l^{b}\overline{l^{a}} = \sum_{y} d_y C_{1}{}^{ayb}{}^{a\varnothing\varnothing}_{b\varnothing\varnothing}$.

Dissertation zur Erlangung des Grades eines Doktors der Naturwissenschaften (Dr. rer. nat.) am Fachbereich Physik der Freien Universität Berlin vorgelegt von

Andreas Bauer

Berlin, 2023

Erstgutachter: Jens Eisert
Zweitgutachter: Felix von Oppen
Datum der Disputation: 04.12.2024

Abstract

This thesis studies the classification of topological phases of matter in terms of tensors associated to discrete spacetimes. We show that all algebraic descriptions of topological order can be reduced to a single property, namely combinatorial topological invariance. This invariance yields equations for the tensors forming a description of a phase. We coin the notion of a topological tensor scheme (tTS) consisting of such a set of tensor-network equations. tTS formalize the whole spectrum of topological phase classifications: They can describe intrinsic bosonic topological, SPT and SET, symmetry-breaking, or fermionic topological orders, in any dimension. Further they can describe not only the bulk, but also superselection sectors of topological boundaries, anyons, twist defects, corners, and all other kinds of defects. In order to systematically talk about defects, we develop the notion of extended manifolds and cellulations. The majority of this thesis is focused on describing phases via microscopic fixed-point models, where the invariance is implemented as recellulation. However, we also show how to formulate extended TQFT as tTS, where the invariance is implemented by gluing, which can be understood as a generalization of recellulation.

In contrast to much of the established literature, we do not merely postulate that higher category theory describes topological order if we correctly implement a long list of technicalities. Instead, we derive all the structures without using any category theory, starting from a coarse and simple ansatz for topological invariance. More fine-tuned structures that closely resemble higher categorical structures are then obtained by a systematic process that we call block-diagonalization. We further investigate to what extent our ansatzes for topologically invariant fixed-point models are universal in the sense that they can emulate any other arbitrarily complicated ansatz for topological invariance. We find that the ansatzes discussed in most chapters of this thesis, which contain all established algebraic descriptions of topological order, are universal only under the condition that there exists a topological boundary. We find a new tTS, corresponding to a new fixed-point ansatz, that is universal independent of the existence of a topological boundary. This provides a promising route to reconcile microscopic fixed-point models with chiral phases of matter, which is one of the major unsettled questions of the field. An indication for this possibility is that the chiral anomaly of these models has the same geometric structure as our new tTS.

We illustrate many of the above ansatzes by concrete examples. Most notably, we look at the family of models arising from cohomology theory, including twisted gauge theories. In particular, we give an efficient systematic procedure to calculate arbitrary defects of spacetime dimensions 0, 1, and 2 in such models.

As an application, we show how to construct dynamic error-correcting codes from fixed-point path integrals, by measuring defects or 1-form symmetries of these path integrals. As an example, we demonstrate that the toric code, the subsystem toric code, as well as the recently developed CSS and honeycomb Floquet codes are secretly the same code up to microscopic equivalence. We also showcase this application by constructing two new codes, namely a Floquet version of the $3+1$-dimensional toric code, as well as a dynamic code for the double-semion phase.

Zusammenfassung (German abstract)

Diese Dissertation untersucht die Klassifikation topologischer Phasen der Materie unter Benutzung von Tensoren, die diskreten Raumzeiten zugeordnet sind. Wir zeigen, dass alle algebraischen Beschreibungen der topologischen Ordnung auf eine einzige Eigenschaft reduziert werden können, nämlich die kombinatorische topologische Invarianz. Diese Invarianz liefert Gleichungen für die Tensoren, die eine Beschreibung einer Phase verkörpern. Wir prägen den Begriff des topologischen Tensor-Schemas (tTS), das aus einem Satz von Tensor-Netzwerkgleichungen besteht. tTS formalisieren das gesamte Spektrum der Klassifikationen topologischer Phasen: Sie können intrinsische bosonische topologische, SPT- und SET-Ordnungen, symmetriebrechende oder fermionische topologische Ordnungen in jeder Dimension beschreiben. Darüber hinaus können sie nicht nur die Phase selbst, sondern auch Superselektionssektoren topologischer Randbedingungen, Anyonen, Twist-Defekte, Ecken und alle Arten von Defekten beschreiben. Um systematisch über Defekte zu sprechen, entwickeln wir einen Begriff erweiterter Mannigfaltigkeiten und Zellulierungen. Der Großteil dieser Arbeit konzentriert sich darauf, Phasen über mikroskopische Fixpunktmodelle zu beschreiben, bei denen die Invarianz als Rezellularisierung implementiert ist. Wir zeigen jedoch auch, wie man erweiterte TQFT als tTS formulieren kann, wobei die Invarianz durch Kleben implementiert wird, was als Verallgemeinerung der Rezellularisierung verstanden werden kann.

Im Gegensatz zur etablierten Literatur verlassen wir uns nicht auf die Idee, dass höhere Kategorientheorie grob zur topologischen Ordnung passt, wenn wir alle technischen Details richtig haben. Stattdessen leiten wir alle Strukturen ohne Verwendung von Kategorientheorie her, ausgehend von einem groben und einfachen Ansatz für topologische Invarianz. Fein abgestimmte Strukturen, die höheren kategorietheoretischen Strukturen ähneln, werden dann durch einen systematischen Prozess erhalten, den wir als Blockdiagonalisierung bezeichnen. Wir untersuchen weiterhin, inwieweit unsere Ansätze für topologisch invariante Fixpunktmodelle universell sind, d.h., ob sie jeden anderen beliebig komplizierten Ansatz für topologische Invarianz emulieren können. Wir stellen fest, dass die in den meisten Kapiteln dieser Arbeit diskutierten Ansätze, die alle etablierten algebraischen Beschreibungen der topologischen Ordnung enthalten, nur unter der Bedingung universal sind, dass eine topologische Randbedingung existiert. Wir finden ein neues tTS, das einem neuen Fixpunktansatz entspricht, der unabhängig von der Existenz einer topologischen Randbedingung universal ist. Dies bietet einen vielversprechenden Weg, mikroskopische Fixpunktmodelle mit chiralen Phasen der Materie in Einklang zu bringen, was eine der wichtigsten ungeklärten Fragen auf diesem Gebiet ist. Ein Hinweis darauf ist, dass die chirale Anomalie dieser Modelle die gleiche geometrische Struktur wie unser neues tTS hat.

Wir veranschaulichen viele der oben genannten Ansätze anhand konkreter Beispiele. Insbesondere betrachten wir die Familie von Modellen, die aus der Kohomologietheorie stammen, einschließlich getwisteter Eichtheorien. Insbesondere geben wir ein effizientes systematisches Verfahren zur Berechnung beliebiger Defekte der Dimensionen 0, 1 und 2 in solchen Modellen an.

Als Anwendung zeigen wir, wie man dynamische fehlerkorrigierende Codes aus Fixpunkt-Pfadintegralen konstruieren kann, indem man Defekte oder 1-Form-Symmetrien dieser Pfadintegrale misst. Als Beispiel zeigen wir, dass der “Toric Code”, der “Subsystem Code”, sowie die kürzlich entwickelten “CSS- und Honeycomb-Floquet-Codes” insgeheim der gleiche Code bis auf mikroskopische Äquivalenz sind. Wir illustrieren diese Anwendung auch durch Konstruktion von zwei neuen Codes, nämlich einer Floquet-Version des $3+1$-dimensionalen Toric Codes sowie eines dynamischen Codes für die gedoppelte Semion Phase.

Contents

Chapter 1

Introduction

1.1 Motivation

Phases of matter are one of the cornerstones of modern many-body physics. In classical statistical physics, phases of matter have been successfully classified via Landau's paradigm of *symmetry breaking* [104]. A symmetry-breaking phase for a symmetry group G is described by the subgroup H to which the symmetry is spontaneously broken. A paradigmatic example for this is the classical *Ising model* in 2 spatial dimensions, where $G = \mathbb{Z}_2$ is implemented as spin-flip symmetry, and H is the trivial subgroup. However, a few decades ago, so-called *topological phases* [136, 143] have been discovered, which go far beyond the principle of symmetry breaking. These phases have a much richer mathematical structure, which is related to cohomology and homotopy, higher category theory and higher algebra, as well as topology and algebraic geometry. One novel feature of topological phases is that there exist phases that are robust even in the absence of any symmetry, so-called *intrinsic topological order*. Adding symmetries not only restricts, but also enriches the classification of topological phases. Mathematically, symmetry-breaking phases can be regarded as a subfamily of topological phases with symmetries, but there is also a second family, namely *symmetry-protected topological (SPT) phases*: Whereas symmetry-protected phases become non-robust/degenerate when the symmetry is removed, SPT phases becomes trivial. One can also enforce symmetries on top of a model of intrinsic topological order, yielding what is known as *symmetry-enriched topological (SET) phases*. Finally, phases of matter can also be defined for models with *fermionic* degrees of freedom. Furthermore, we cannot only study topological phases of a model itself, but also of its different boundary conditions, domain walls, anyons, corners, and all other sorts of lower-dimensional defects.

There are two main approaches to classifying topological phases in the physics literature. The first approach is to restrict to quadratic fermionic Hamiltonians. Each such translation-invariant Hamiltonian with n modes per unit cell can be Fourier transformed into an n-mode Hamiltonian H_k depending on the momentum k. The below-zero subspace of H_k yields a vector bundle in k-space, and the isomorphism class of this vector bundle remains invariant under perturbations that do not close a gap of the Hamiltonian. So phases are classified by isomorphism classes of vector bundles, which can be partially characterized via their *Chern number*. Phases that can be classified with this approach are *invertible* fermionic phases, which become trivial if we stack them with a complex-conjugated copy. This includes fermionic SPT phases for different types of symmetries, known as the *10-fold way* [4, 123]. This classification of topological phases is *not* in the scope of this thesis. Instead, we will focus on a second approach, namely *fixed-point models*. Fixed-point models are microscopic models that represent the topological phase in a particularly pure manner, specifically, with *zero correlation length*. Fixed-point models are usually written down as commuting-projector Hamiltonians and are therefore exactly solvable. In contrast to a band-theoretic classification of topological phases, fixed-point models are capable of describing strongly interacting phases of matter, which includes non-invertible and intrinsic topological order. The perhaps most well-known family of fixed-point models are *Levin-Wen string-net models* [105] in $2+1$ dimensions, which include the *toric code* or more generally (twisted) *quantum double models* [95, 82]. Other examples include *Walker-Wang models* in $3+1$ dimensions [133], or the *Kitaev chain* in $1+1$ dimensions [94].

Mathematically, topological phases can be described via *topological quantum field theory* (TQFT) [6]. The modern *axiomatic* mathematical definition of a TQFT has little to do with "quantum field theories" in the sense of having field-like degrees of freedom. As we emphasize in this thesis, this notion of TQFT is closely linked to discrete rather than continuous spacetime. The name stems from its historic origin in high-energy physics, via the field of *axiomatic/algebraic quantum field theory* which attempts to make quantum field theory mathematically rigorous. The perhaps most notable example for an actual quantum field theory that can be described as a TQFT is *Chern-Simons theory* in $2+1$ dimensions [140], but we will not consider such actual field theories. It was later realized that for a more useful description of topological phases, one has to generalize from "ordinary" TQFT to *extended TQFT* [8]. More specifically, one can consider n-dimensional TQFTs that are *extended down to dimension* $n-i$, and ordinary TQFT corresponds to the first level of extension $i=1$. In this thesis, we take the point of view that only *almost-fully extended (*$i=n-1$*) TQFT* and *fully extended (*$i=n$*) TQFT* are useful descriptions of topological phases. Almost-fully extended TQFTs are also known as "extended down to the circle", whereas fully extended TQFT is "extended down to the point". Less extended TQFTs have too lax axioms, such that there is neither a finite set of generators and axioms, nor good reasons to believe that their equivalence classes are in one-to-one correspondence with topological phases. Roughly speaking, almost-fully extended TQFT describes the behavior of defects of dimension $\leq n-2$ of a hypothetical topologically ordered model. The most prominent example for this is *Reshetikhin-Turaev TQFT* [121], which by folklore corresponds to a $n=3$, $i=2$ almost-fully extended TQFT. Reshetikhin-Turaev TQFT is constructed from a *unitary modular tensor category*, which is well-known in physics as the structure that describes the fusion and braiding of anyons in models of $2+1$-dimensional intrinsic topological order [96]. On the other hand, fully extended TQFT corresponds to microscopic models by folklore. In this context, the microscopic models are *state-sum models*, such as the *Turaev-Viro-Barrett-Westbury* model [130, 14], the *Crane-Yetter* model [45], or the *Kuperberg invariant* [102]. Note that the Levin-Wen string-net models, Walker-Wang models, and (Hopf algebraic [37]) quantum double models from the previous paragraph are Hamiltonian reformulations of these state-sum models, respectively. The fully-extended TQFT corresponding to the Turaev-Viro-Barrett-Westbury state-sum has been explicitly described in Refs. [86, 9, 10]. The field of extended TQFT uses a heavy and complicated language of higher category theory, and is therefore not very accessible to a broader physics community.

An intriguing application of topological phases of matter comes from quantum computation: *Topological quantum computation* [114, 95] is one of the most promising candidates for implementing scalable fault-tolerant quantum computation. In topological quantum computation, the logical information is encoded into the ground state of a topologically ordered model on a topologically non-trivial spatial configuration. This configuration may also involve boundaries, anyons, domain walls, twist defects, and other sorts of defects. Note that we think of anyons and other sorts of defects not as excitations, but as local modifications in the Hamiltonian (or other descriptions of the model) that enforce the presence of an anyons in the modified ground state. A quantum gate (or a whole computation) is encoded in a process that adiabatically deforms the spatial configuration while remaining in the ground state. Due to the underlying topological order of the model, both the dimension of the logical space and the performed gate only depend on the topology of the spatial configuration or its deformation. The idea is that the robustness of topological order [31] carries over to topological quantum computation, and makes the implemented gates fault tolerant. In practice, topological quantum computation requires syndrome measurements and topological error correction [57]. When trying to achieve universal fault-tolerant quantum computation only using topological quantum computation, it is necessary to implement *non-Abelian* topological phases. In contrast, all established topological error-correcting protocols are for Abelian phases. For non-Abelian phases it is less clear how error correction works. Furthermore, the resulting protocols are complicated with a very large overhead. For a recent work demonstrating theoretical (but not practical) feasibility, see Refs. [126, 125]. The more established route is to use abelian topological error correction along with non-topological *magic state distillation*. Aside from loosing the elegance of the purely topological approach, magic state distillation again has a large overhead.

1.2 Scope and contents of the thesis

The main goal of this thesis is to establish a unified and systematic mathematical formalism describing topological order at zero correlation length. The two areas in physics and mathematics that this formalism covers are the middle two keywords in the title, namely *fixed-point models* and *extended TQFT*. The literature on these topics is vast and scattered, and suffers from a considerable language gap between the physics and mathematics side. Further, to motivate the resulting mathematical structures, it is common to make a large number of non-rigorous physical assumptions. This thesis aims to unify all algebraic descriptions of phases of matter using a single mathematical structure obeying a single rigorous assumption, namely *topological invariance*. To do this in the most concise and direct way, we represent physical models as amplitudes assigned to combinatorial triangulations of a Euclidean-signature spacetime manifold. Topological invariance is then imposed as a simple compatibility of the amplitudes with *bi-stellar flips* or *Pachner moves*, as well as *gluing*. All other known properties of topological models or their algebraic descriptions follow from this notion of topological invariance.

There are not only many different topological phases, but also many different families of topological phases, and of their fixed-point descriptions. Roughly, the family depends on the kind of matter we are considering, the dimension, the level on which we want to classify phases, and so on. Mathematically, each family corresponds to a fixed-point ansatz, that is, an algebraic structure subject to some constraints and equivalence relations. Equivalence classes of solutions to the constraints then correspond to the different topological phases in the family. In this sense, the topological phases of a family are classified by the corresponding fixed-point ansatz. More specifically, a fixed-point ansatz is given by a set of tensor-network equations that hold for a set of tensor-valued variables. The tensor-valued variables are just collections of amplitudes, and the equations correspond to their topological invariance.

The set of different fixed-point families itself is a multi-dimensional space (figuratively speaking), which is spanned by roughly seven different aspects in which fixed-point ansatzes can differ. Each configuration of these aspects, which we list in the following, roughly corresponds to one fixed-point ansatz.

- The first aspect is the spacetime dimension n. In this thesis we are only interested in the physically relevant cases with $n \leq 4$.

- The second aspect are different types of matter. The standard type of matter are qudit systems. Physically, these are either spins where we do not care about the $SU(2)$ symmetry representation they carry, or bosons where we disregard particle number conservation and only take a finite set of basis states inside the infinite-dimensional Fock space of a single mode. This is the type of matter which we are dealing with for most of this thesis. In Chapter 7, we discuss other types of matter. In Section 7.1, we discuss phases of matter with global symmetries, that is, restricted by an on-site representation of a finite group. In Section 7.2, we extend this to symmetry groups where some of the elements act by time-reversal, in which case they are represented by anti-unitary operators. Finally, in Section 7.3, we consider systems with fermionic degrees of freedom. For each of these types of matter, we have to equip our combinatorial spacetimes with some cohomological extra structure. For fermionic phases, we additionally need to interpret the tensor-network equations in terms of fermionic tensors instead of ordinary tensors.

- In additional to topological order in the bulk, we can consider topological order on the boundary or on domain walls, as we discuss in Chapter 5. We can also consider superselection sectors of anyons, twist defects, corners, fusion events between anyons, condensations of anyons, and many other sorts of "defects" in spacetime. All these lower-dimensional defects can be classified with our framework in a unified and systematic way, which is described in Chapter 6. To this end, we introduce the notion of *extended manifolds* in Section 2.4, which roughly speaking are composites of different manifolds of different dimensions. Lower-dimensional manifolds are attached to the boundary of higher-dimensional ones via a local neighborhood that we will refer to as *link*. The set of links defines the *type* of extended manifolds, and different types correspond to different sorts of defects. Even boundaries and domain walls can be viewed as sorts of defects this way.

- The first few chapters of this thesis concentrate on microscopic fixed-point path integrals for topological phases or their lower-dimensional boundaries or defects. In Chapter 8, we generalize this approach to also allow for descriptions of topological phases via higher-level invariants similar to almost-fully extended TQFT as mentioned in the previous section. Central to this is again our notion of extended manifolds and cellulations. We slightly change our perspective in Section 8.2, reinterpreting local combinatorial topological invariance as compatibility under *gluing*. After this change of perspective, microscopic fixed-point models and higher-level descriptions are treated on equal footing in a unified formalism. Almost-fully extended TQFTs describe the collection of defects of codimension at most 2 of the model. The most well-known example for such invariants is the description of $2+1$-dimensional topological phases via the fusion and braiding properties of their anyons. Less extended TQFTs do not appear useful since their axioms are too lax such that there is neither a finite set of generators and relations nor a reason to believe that their relation with microscopic is even approximately one-to-one. Note that we will not use the language of extended TQFT but resort to a definition based on our notion of extended manifolds that is simpler, at least for the categorical non-expert.

- There are different combinatorial implementations of topological invariance as discussed in Chapter 9.2. An example is a tensor-network path integral in $2+1$ dimensions with tensors on the edges and faces instead of tetrahedra of a triangulation, c.f. Section 9. Examples of this type yield the same classification of phases, but alternative microscopic models that might be useful for practical purposes. A third type of example is are the universal vertex-type models which we will discuss in Chapter 11.

- We can bring fixed-point ansatzes into a more refined *block-diagonal form*. In general, there is no hope that any of the structures resulting from our or any other framework of topological order can be classified in the sense of algorithmically efficient enumeration. This only seems to be possible in up to $1+1$ dimensions, where models correspond to (Frobenius) algebras that can be block-diagonalized. We can now use the classification in $1+1$ dimensions also in higher dimensions after a dimension-reduction, in order to “simplify” the resulting structures there. The resulting structures are actually more complex, but they are more fine-tuned in the sense that the same phases can be described with a smaller number of coefficients.

- Finally, we can also add extra axioms beyond topological invariance.

Aside from the systematic derivation of topological fixed-point ansatzes, there are three additional topics of this work. The first additional topic is related to the first keyword of the title: In Chapter 3, we explain the connection between the fixed-point classifications of topological order, and topological order in realistic non-fixed point models, such as ones arising in condensed-matter theory. We argue in Section 3.1 that in general, local models in physics can be described by discrete (tensor-network) path integrals. In Section 3.2, we demonstrate how path integrals can arbitrarily approximate the more established Hamiltonian formulation of quantum many-body physics via *Trotterization*. In Section 3.3 we briefly review established definitions of phases of matter, and suggest how they naturally transfer to the setting of discrete path integrals in Section 3.4. Finally, we motivate why phases possess fixed-point models by employing the idea of *renormalization group flow* in Section 3.5, and why it is natural for these models to possess topological invariance in Section 3.6.

The second additional topic is a family of concrete models that can be found and manipulated efficiently and are closely related to cellular cohomology and group cohomology. In the literature, these models are known as *discrete gauge theories*. In Sections 2.5, 2.6, and 2.7, we introduce the mathematical basics of cellular cohomology, cohomology operations, and characteristic classes in a way that is suitable for defining these spacetime lattice models. In Sections 4.16 and 5.1.3, we introduce cohomological models for the bulk and boundary. Furthermore, we show how to efficiently classify arbitrary defects of dimensions 0, 1, and 2 in cohomological models. We do this at hand of many examples for many sorts of defects in subsections of Chapter 6.

The third additional topic, discussed in Chapter 12, is the construction of dynamic quantum error-correcting codes from fixed-point path integrals, corresponding to the fourth keyword in the title. To this end, we use topological defects or 1-form symmetries to turn a path integral into a sequence of measurements implementing the topological phase as a fault-tolerant protocol. We refer to codes constructed in this way as *fixed-point path integral codes*. We find that essentially all topological fault-tolerant protocols are fixed-point path integral codes. As an example for this, we find in Section 12.2 that the traditional stabilizer toric code [57], the *subsystem toric code* [32], the *Hastings-Haah honeycomb Floquet code* [79], as well as the *CSS Floquet code* [92, 53, 2] all correspond to the same toric-code path integral on geometrically different lattices. The formalism can also be used to construct new codes which we do in Section 12.3. We first construct a Floquet version of the $3+1$-dimensional toric code consisting only 8 rounds of commuting 2-body XX and ZZ measurements. Second, we construct an example of a dynamic code that hosts an abelian phase beyond copies of the toric-code, namely the double-semion phase.

1.3 Contributions

This thesis aims to provide a rigorous, systematic, and unified approach to algebraic fixed-point descriptions of topological phases. Many of the contents of this thesis can thus be found in the large but scattered literature on the topic, and were rederived, or adapted, generalized, and fitted into the framework of this thesis. Apart from rederiving results from the literature in a systematic and accessible way, this thesis also contains numerous results that we believe go beyond of what is known in the literature. In this section, we will try to explain the degrees of novelty for different aspects of this thesis, and highlight some of the contributions. We will not attempt to describe every single novel aspect in detail, but briefly list some of them in the following, roughly in the order in which they appear in the thesis.

- We provide a versatile new mathematical tool, namely extended manifolds and cellulations, see Section 2.4. These feature prominently when we introduce defects into fixed-point models in Chapter 6, or use axiomatic-TQFT descriptions instead of microscopic descriptions in Chapter 8. Despite the apparent naturality of these structures, there seems to be nothing alike in the literature.

- We give a definition of phases of matter and renormalization-group flow directly in terms of tensor-network path integrals in Chapter 3.

- As far as we know, this is the only place in the literature that classifies phases of matter in a general setting without making any use of category theory. The mathematics used in our approach is very simple, and only consists of polynomial equations between arrays of complex numbers, which can be denoted as tensor-network equations, c.f. Section 2.3. As such we believe that our approach is more accessible to a large physics audience than approaches based on category theory, which suffers from an enormous vocabulary and tends to be formulated in a not computationally explicit way.

- In contrast to established approaches, our approach does not impose a "flow of time" at any stage. Algebraic or categorical structures are restricted by such a flow of time, as they are given by linear maps whose input space is distinguished from their output space. Instead of linear maps with inputs and outputs, we use tensors with indices, which is more natural since the underlying spacetimes are topological with Euclidean signature.

- We give a concrete reason for why the branching structure underlying the triangulations of fixed-point models is necessary, c.f. Section 4.2 and Chapter 10. Usually, the branching structure is just postulated without any motivation.

- In the literature for fixed-point models, it is common to postulate an algebraic or categorical structure. These structures are then motivated by a large number of assumptions based on non-rigorous physical arguments. In our approach, we make only one single assumption, namely topological invariance, and show how to derive algebraic or categorical structures from this single assumption.

- We give a simple geometric interpretation for unitary spherical fusion categories in terms of recellulations, c.f. Sections 4.8 and 4.14. While it is well-known that the F-symbol corresponds to a tetrahedron and the pentagon equation corresponds to a 2-3 Pachner move, we show that also the pivotal structure and Frobenius-Schur indicator can be understood as 3-cells.

- We give an explicit way to deal with the branching structure when demanding retriangulation invariance of topological state-sum constructions, c.f. Sections 4.7, 4.8, and 4.15. In the physics literature, it is well known that, for example, the pentagon equation for fusion categories imposes invariance of the path integral under Pachner moves. However, it was not known why and how invariance under changes in the branching structure follows in the general case. Here, we show that this is the case by introducing a few simple extra generators and axioms.

- We introduce a versatile mathematical tool that we call block-diagonalization, which allows us to systematically derive more fine-tuned structures from the coarse-grained fixed-point ansatzes that we start with, c.f. Sections 4.13, 4.14, 4.15, and 8.4.2. For example, we can use this to derive unitary fusion categories from a coarse-grained topologically invariant path integral in $2+1$ dimensions.
- We provide a systematic formalism to classify lower-dimensional defects in cohomology models in Section 5.1.3 and many subsections of Chapter 6. We give a unified algorithm to compute explicit microscopic representations of all defects of spacetime dimensions 0, 1, and 2 in these cohomology models. This applies to any kind of defects, as described by our notion of an extended manifold type.
- We show how to calculate and classify defects in $3+1$-dimensional twisted gauge theories, in particular standalone defects that are line or membrane-like in spacetime, as well as line-like intersections of three different membrane defects in Sections 6.10, 6.9, and 6.11. We are not aware of any references in the literature that systematically classify membrane defects and membrane-fusion line defects in $3+1$-dimensional twisted gauge theories.
- In Ref. [70] it is argued that in order to systematically define fermionic fixed-point models, we need to introduce a combinatorial representation of a spin structure. However, this approach appears not to be widely understood and adopted in the physics literature on fixed-point models. In Sections 7.3, 7.4, and 7.5, we show how to systematically incorporate combinatorial spin structures into fixed-point models. We also extend this approach by showing how symmetries that include the fermion parity as well as time-reversing elements can be described by some cohomological extra structure in Section 7.6.
- In Chapter 8, we introduce a simple, fully explicit, mathematical structure, which we refer to as axiomatic topological tensor scheme, and which appears to deliver the things that one would hope to get from extended TQFT. Note it is notoriously hard to spell out explicit definitions for extended TQFT in terms of concrete computational data. Also there are many different ways in which extended TQFT can be defined, differing in many technical details, and it is hard to tell which definition is the natural one. In contrast, there is one natural way to define axiomatic topological tensor schemes, and its data is given by arrays of complex numbers satisfying polynomial equations. They are also more flexible and can for example include arbitrary types of defects, harnessing our notion of extended manifold types. One feature of our approach is that we do not need to postulate that we map to a target higher category as in extended TQFT. For axiomatic topological tensor schemes, the “target” are just arrays of complex numbers, and the higher category emerges by equipping the source “cobordisms” with a cellulation.

- The fact that braided fusion categories describing anyons must be modular, and that algebra objects describing topological boundaries via anyon condensation must be Lagrangian is usually motivated by postulating the physical principle of remote detectability. In our formalism, an additional physical principle is not necessary, and the modular and Lagrangian conditions are direct consequences of topological invariance as we show in Sections 8.4.3 and 8.5.1. We also find a general formulation for the Lagrangian condition in terms of the S-matrix that does not seem to exist in the literature yet.

- In Sections 9.4 and 9.4, we provide a state-sum TQFT picture for weak-Hopf-algebra quantum double models that was not known in the literature. We show that the weak-Hopf-algebra axioms follow immediately from combinatorial topological invariance. In fact, we find that the weak-Hopf-algebra axioms can be greatly simplified, and can be essentially reduced to having a nice-enough algebra and co-algebra that fulfill the bi-algebra law.

- We give a simple geometric picture for the correspondence between weak Hopf algebras and unitary fusion categories in Section 9.6.

- In Chapter 10, we introduce a notion of universality, and prove that the fixed-point ansatzes we study are universal if we assume the existence of a topological boundary. This provides a very clear explanation why established fixed-point models only exist in the presence of topological boundaries. It also shows precisely why a branching structure is necessary to incorporate the most general phases with topological boundaries.

- In Chapter 11, we find a new fixed-point ansatz for topological order that is universal without the limitation of having a topological/gapped boundary. This opens up a route towards finding fixed-point models for chiral topological phases, which is one of the most important open questions in the classification of topological phases.

- In Chapter 12, we propose a new paradigm for thinking about topological error-correcting protocols. Namely, we show that they can be understood as fixed-point path integrals that are turned into circuits of instruments by measuring defects. This provides a unified framework to analyze many distinct codes, such as the standard toric code, measurement-based topological quantum computation, subsystem codes, or recently discovered Floquet codes.

- In Section 12.2.4, we give an explicit microscopic way to map between the honeycomb Floquet code and the CSS Floquet code, which was previously unknown. It turns out that in the sector of trivial measurement outcomes these two codes are directly related via an on-site change of basis, but non-trivial measurement outcomes correspond to different locations of defects in the path integral.

- In Section 12.3.1, we develop a Floquet version of the $3+1$-dimensional toric code using only 2-body measurements.

- In Section 12.3.2, we discover a dynamic "Floquet" error correcting code that is non-Pauli and corresponds to a phase that is not (copies of) the toric code. Note that the vast majority of the topological error correction literature only deals with Pauli measurements and the toric code phase.

1.4 Author publications and preprints

In this Section, I list the publications and preprints that I have (co-)authored during my time as a PhD student, and explain to which extent they are related and have contributed to this thesis. Readers who are not on my PhD committee are invited to skip this section. I will go through the preprints and publications chronologically.

- A. Bauer, J. Eisert, C. Wille, *Towards a mathematical formalism for classifying phases of matter*, preprint Ref. [21]. This preprint already contains many of the main ideas in this thesis, including the simplified tTS, 3D face-edge tTS, and early incarnations of extended manifolds and cellulations. A subset of the ideas was rewritten and published in Ref. [20].

- A. Bauer, *Generalized topological state-sum constructions and their universality*, preprint Ref. [18]. This preprint contains the main ideas behind universality and beyond-gappable-boundary tTS discussed in Chapters 10 and 11.1. An extended and rewritten version of this was published as Ref. [25].

- A. Bauer, *Quantum mechanics is *-algebras and tensor networks*, preprint Ref. [19]. In this preprint, I give a simple interpretation of (qudit) quantum mechanics in terms of tensor networks. I show that channels, unitaries, POVMs, state preparations, instruments, or classically controlled operations can be understood in a unified way as "positive tensors". This work is not part of this thesis, but the formulation of physical models in terms of tensor networks also plays an important role here.

- A. Bauer, J. Eisert, C. Wille, *A unified diagrammatic approach to topological fixed point models*, Ref. [20]. This is a condensed and more digestible version of the preprint [21]. The materials of this paper are the basis of Chapters 4, 9, and Section 7.4.

- A. Bauer, J. Eisert, C. Wille, *Towards topological fixed-point models beyond gappable boundaries*, Ref. [25]. This is a reworked version of the preprint Ref. [18]. This is the basis of Chapters 10 and 11.

- A. Bauer, A. Nietner, *Tensor types and their use in physics*, preprint Ref. [22]. This was a project on which I spent basically a whole year of my PhD. On a general level, it is a program to extend monoidal category theory beyond a flow of time, introducing so-called *2-schemes*. More specifically, we look at one kind of 2-scheme called *tensor types*, which intuitively can be understood as mathematical structures obey the graphical calculus of tensor networks. Even more specifically, we give a long list of specific tensor types that correspond to specific types of models or matter in physics. Although this project originated from the idea of extending the notion of tensor-network path integrals to other types of models, the framework of tensor types is overkill for description of topological phases in this thesis. Therefore, this work is not included in the thesis. The only occurrence of a tensor type are *fermionic tensors* used in Section 7.3, which we introduce in a more pedestrian way in Section 2.2.

- A. Bauer, *Disentangling modular Walker-Wang models via fermionic invertible boundaries*, Ref. [23]. In Appendix B of this paper, I propose so-called *tensorial TQFT* as an alternative to extended TQFT that naturally comes by without using any category theory. I show that Walker-Wang models have a simple interpretation in terms of tensorial TQFT, and use this simple interpretation to prove an open conjecture that Walker-Wang models based on Drinfeld-center UMTCs can be disentangled by generalized local unitary circuits. I also show that this is possible for a larger class of UMTCs if we allow for fermionic auxiliary degrees of freedom. Appendix B of this paper is the basis for Chapter 8, where tensorial TQFT was renamed to *axiomatic topological tensor scheme*. The main result of Ref. [23] is not included in this thesis.

- E. Anschuetz, A. Bauer, B. Kiani, S. Lloyd, *Efficient classical algorithms for simulating symmetric quantum systems*, Ref. [5]. This is a collaboration with colleagues at MIT that I had the pleasure to be part of. We find polynomial-time algorithms to solve general quantum systems restricted by a S_n qubit permutation symmetry. Here I employed my tensor-network understanding of (bi-)algebras and representation theory to efficiently compute the map from a symmetrized Pauli operator basis to the basis of irreducible S_n blocks. Since these blocks are low-dimensional, the ground state and dynamics can then be found by exact diagonalization inside them. This work has no immediate connection to topological phases of matter, so it is not included in this thesis. However, the methods and notation I use in the appendices of Ref. [5] are closely related to the tensor-network formulations of block-diagonalization in Section 4.13 and bi-algebras in Section 9.4.

- J. Magdalena de la Fuente, J. Eisert, A. Bauer, *Bulk-to-boundary anyon fusion from microscopic models*, Ref. [66]. This was a project with Julio Magdalena de la Fuente in the lead, which I had the pleasure to supervise. We calculate the fusion multiplicities of bulk anyons and boundary anyons inside the boundary of $2+1$-dimensional group-cohomology models of topological order. I contributed my general state-sum understanding of the bulk, boundary, anyons, and fusion events in these models, which I used to rederive the general formalism of the paper in Appendices D and E. These appendices are the base of the cohomology models discussed as concrete examples in this thesis in Sections 4.16, 5.1.3, and the cohomology subsections of Chapter 6. The main result of Ref. [66], namely the calculation of bulk-to-boundary fusion multiplicities is not included in this thesis, but can be computed straight-forwardly using the presented methods.

- A. Bauer, *Topological error correcting processes from fixed-point path integrals*, preprint Ref. [24]. In this preprint, I look at quantum error correction through the lens of tensor-network path integrals. I discover that essentially all protocols of topological quantum error correction can be interpreted as so-called *fixed-point path integral codes*. Most notably, this includes recently discovered Floquet codes. This work forms Chapter 12 of this thesis, with only little alterations, additions, and removal of contents that were already covered by other parts of the thesis.

With the exception of Ref. [24] in Chapter 12, all of the publications and preprints were significantly rewritten, generalized, filtered, and extended for this thesis. Let me point out some new material in this thesis that was not covered by any of my previous publications and preprints.

- The formalism of extended cellulations in Section 2.4 is new.
- The block-diagonal form and block-diagonalization procedure in Sections 4.8, 5.1.2 and subsections of Chapter 6 is not explicitly contained in any of my publications, though the similar block-diagonalization in Section 8.4.2 was sketched in Appendix B of Ref. [23].
- I have improved and extended the methods in Appendices D and E of Ref. [66]. I am now using the formalism of extended cellulations, and have generalized the method to map between ordinary and G-set cohomology to arbitrary degrees/spacetime dimensions. This way, the methods can efficiently and systematically compute explicit microscopic representations of arbitrary defects of spacetime dimensions 0, 1, and 2. I have also added how to use these methods to calculate three different types of defects in $3+1$-dimensional cohomology models in Sections 6.9, 6.10, and 6.11.
- Most of the defects in Chapter 6 are new, even though similarly anyons and boundaries of Hopf-algebraic models were already discussed in Ref. [21], though in a very different language.
- The discussion on how to mix time-reversal and fermions with ordinary symmetry in Chapter 7 is new. Also the discussion of fermionic models in $2+1$ dimensions is new.
- The discussion of atTS models for boundaries and defects in Section 8.5 is new.

Chapter 2

Mathematical preliminaries

In this chapter, we will discuss some basic mathematical structures that will be used in the main text. Let us quickly mention the degree of novelty of the different structures and terminology and how they are used in this thesis. Fermionic tensors have been introduced in the literature in many incarnations, however, the language we use is a little different. We also introduce the notion of fermionic conjugation. TS and tTS mappings are nothing but a simple language for algebraic structures that can be represented by tensor-network diagrams. Extended manifolds and cellulations are central to our formalism. They provide a systematic way to represent the large zoo of different families of TQFT-like descriptions of phases of matter. Cohomology operations and characteristic classes are textbook topics, but we present them with an emphasis on cellulations which is useful for physics applications. We use them to define cohomology models, symmetric or fermionic phases, as well as to better understand the chiral anomaly.

2.1 Tensor networks

At the heart of our approach are tensor networks. Let us start by defining tensors and their basic operations.

Definition 1. *By an n-index* tensor*, we mean a complex-valued multi-dimensional array $A_{i_0,\ldots,i_{n-1}}$, that is, a map*

$$\begin{gathered}\bigtimes_{0\leq x<n} \{0,\ldots,d_x-1\} \rightarrow \mathbb{C}\;, \\ (i_0,\ldots,i_{n-1}) \mapsto A_{i_0,\ldots,i_{n-1}}\;.\end{gathered} \tag{2.1}$$

Here, d_x is the bond dimension *of the xth index. The* transposition *of the xth and yth index of an n-index tensor A is the n-index tensor $\tau_{xy}(A)$ with these two indices exchanged,*

$$\tau_{xy}(A)_{i_0,\ldots i_{n-1}} = A_{i_0,\ldots,i_{x-1},i_y,i_{x+1},\ldots,i_{y-1},i_x,i_{y+1},\ldots,i_{n-1}}\;, \tag{2.2}$$

assuming $x < y$. The tensor product *of a n-index tensor A and a m-index tensor B is an $n+m$-index tensor $A \otimes B$ given by the entry-wise product,*

$$(A \otimes B)_{i,j,\ldots,k,l,\ldots} = A_{i,j,\ldots} \cdot B_{k,l,\ldots} \ . \tag{2.3}$$

The contraction *of the xth and yth index of a n-index tensor A is the $n-2$-index tensor $[A]^{x,y}$ given by restricting to the subspace where these two indices take the same value, and then summing over this value,*

$$([A]^{x,y})_{i_0,\ldots i_{n-3}} = \sum_i A_{i_0,\ldots,i_{x-1},i,i_x,\ldots,i_{y-2},i,i_{y-1},\ldots,i_{n-3}} \ , \tag{2.4}$$

assuming $x < y$. For this to be possible, the bond dimensions of the two indices need to be the same, $d_x = d_y$.

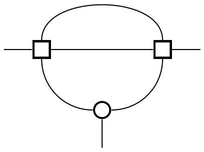
With this, we are ready to define tensor networks.

Definition 2. *A* tensor-network diagram *(or sometimes just* diagram *or* network*) is a computation whose input is a finite set of tensors, and whose output is a single tensor. By computation we mean a finite sequence of index transpositions, tensor products, contractions, and taking copies of tensors. Up to equivalences, such a computation is fully specified by 1) how many copies of which tensor we take and 2) which index of which tensor copy we contract with which other index. The equivalences are, for example, that the order in which we contract two index pairs does not matter. Or, that $\otimes$ is associative, and $A \otimes B$ is the same as $B \otimes A$ after permuting the indices accordingly. The contractions will be called* bonds, *and the indices of the resulting tensor* open indices.

The term *tensor network* loosely refers to a tensor-network diagram together with a set of tensors as its inputs. As the name says, tensor-network diagrams have a neat diagrammatic representation, which is also known as *Penrose diagrams*: We choose a shape for each input tensor, like a circle or a square. Then we draw one copy of the shape for each copy of the corresponding input tensor. Most importantly, we connect a pair of shapes with a line if the corresponding tensor copies share a contracted index pair. Finally, we draw a line from a shape (ending somewhere in the empty space) for every index of the tensor copy that contributes to the final tensor. For example, consider the computation

$$\sum_{x,y,z,w} A_{x,y,i,z} A_{j,x,y,w} B_{w,z,k} \ , \tag{2.5}$$

taking two tensors A and B as input. Choosing a small square for A and a circle for B, the actual diagram for this is

 . (2.6)

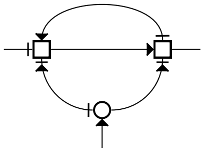
There is two contracted index pairs between the two copies of A, and an additional one between B and every copy of A. Each tensor copy has one index that is part of the resulting 3-index tensor. In addition, we will denote the identity matrix as a "free bond",

$$(\mathbb{1})_{a,b} = \delta_{a,b} \quad \rightarrow \quad \text{——} \; . \tag{2.7}$$

The fact that connecting one end of this line with a shape in a tensor-network diagram has no effect is compatible with the fact that contracting $\mathbb{1}$ with any index of a tensor leaves this tensor invariant,

$$\sum_x \delta_{x,y} T_{x,\ldots} = T_{y,\ldots} \; . \tag{2.8}$$

In fact, the notation above is not fully precise yet: The diagram does not specify *which* index of the respective tensor copy is contracted. To specify which indices a bond in the diagram corresponds to, we add little markings where the line connects to each shape. For example, we mark the second index of A with an arrow, the third with a tick, the fourth with both an arrow and a tick, and the first with nothing. We also mark the second index of B with a tick, the third with an arrow, and the first with nothing. Then, the diagram representing Eq. (2.5) looks like



. (2.9)

Sometimes when we are lazy and it does not matter that much, we will drop the index markings. Often, we also use different line styles (like thick, or dashed) for indices with different bond dimensions, which reduces the amount of markings we need to put. Note that in the literature which index is which is usually implicit by the position of the index, but this would not be practical for the "topological" tensor-network diagrams in this thesis.

2.2 Fermionic tensors

Tensor network path integrals can describe many-body quantum models. The degrees of freedom in these models are qudits, and they are also known as *quantum spin systems*. If we want to model *fermions*, we have to use a different flavor of tensor networks. Namely, we need to use *fermionic tensor networks* which we will define in this section.

Definition 3. *A* fermionic tensor *is an array*

$$\begin{gathered} \bigtimes_{0\leq x<n} \{0,\ldots,d_x^+ + d_x^- - 1\} \rightarrow \mathbb{C} \; , \\ (i_0,\ldots,i_{n-1}) \mapsto A_{i_0,\ldots,i_{n-1}} \; . \end{gathered} \tag{2.10}$$

d_x^+ *will be called the* even *bond dimension of the xth index,* d_x^- *the* odd *bond dimension j, and both together the* fermionic bond dimension*. A is subject the following* parity constraint. *Let us define the* parity $|i_x| \in \mathbb{Z}_2$ *of a configuration* $0 \leq i_x < d_x^+ + d_x^-$ *of the* x*th index,*

$$|i_x| = \begin{cases} 0 & \text{if} \quad 0 \leq i_x < d_x^+ \\ 1 & \text{if} \quad d_x^+ \leq i_x < d_x^+ + d_x^- \end{cases} . \tag{2.11}$$

Now the parity constraint is

$$A_{i_0 \ldots, i_{n-1}} = 0 \quad \text{if} \quad \sum_{0 \leq x < n} |i_x| \neq 0 \mod 2 . \tag{2.12}$$

Furthermore, the x*th index of a fermionic tensor can either be an* input index *or an* output index.

The tensor product is defined as for ordinary tensors. The fermionic index transposition *is where the crucial difference to ordinary tensors is: The transposition of the* x*th with the* $x+1$*th index is given by*

$$\tau^{x,x+1}(A)_{i_0,\ldots,i_{n-1}} = (-1)^{|i_x||i_{x+1}|} A_{i_0,\ldots,i_{x-1},i_{x+1},i_x,i_{x+2},\ldots,i_{n-1}} . \tag{2.13}$$

In other words, we collect a factor of -1 *if both the* x*th and* $x+1$*th index have parity* 1*. If the* x*th index is an output index and the* $x+1$*th index is an input index, the their contraction is defined as for ordinary tensors. We may never contract two input indices or two output indices. Contractions between input-output pairs with other locations can be performed by fermionic index transpositions together with the contraction described above.*

For any fermionic bond dimension (d^+, d^-)*, the* fermionic identity matrix *is the ordinary identity matrix where the* 0*th index is an input and the* 1*st index is an output index. The* fermion parity matrix *is the same tensor with input and output exchanged.*

Computations with fermionic tensors, their tensor products, contractions, and fermionic index transpositions, are subject to the same equivalences as for ordinary tensors. For example, it is still true that $A \otimes B$ is the same as $B \otimes A$ followed by permutation of all the B-indices with all the A-indices. This is because the total parity of all the A indices as well as all the B indices is 0 via the parity constraint in Eq. (2.12). Thus tensor-network diagrams for fermionic tensors look essentially the same as for ordinary tensors. The only difference is that we are more constrained in drawing the diagrams, as we may only connect input with output indices by bonds. The example diagram in Eq. (2.9) representing the computation in Eq. (2.5) happens to also make sense as fermionic tensor network: If the first and third index of A, as well as the first and second index of B are input indices, then we only contract input with output indices. If, for example, we had contracted the second index of the first copy of A with the second index of the second, then there would be no attribution of input/output under which this diagram would be a valid fermionic tensor network.

The fermionic identity matrix will still be denoted by a free bond as in Eq. (2.7). Note that as such, it is not clear which ending of the free bond is the input and which the output index. However, in concrete usage, tensor-networks always appear in an equation, and the input/output distinction is clear from how the indices are matched with the other side of the equation. The fermionic identity matrix will be denoted by a little diamond shape,

$$-\!\!\blacklozenge\!\!- \; . \tag{2.14}$$

Again, which of the indices is input and which output is determined from how they are either contracted with indices of another tensor or equated with another index in an equation.

Let us also define an additional operation on fermionic tensors, which will be necessary to define what it means for a fermionic Hamiltonian to be Hermitian.

Definition 4. *The* fermionic conjugate *of a fermionic tensor A is the following tensor A^*,*

$$A^*_{i_0,\ldots,i_{n-1}} = (-1)^{\frac{1}{2}\sum_x \widehat{|i_x|}} A^*_{i_0,\ldots,i_{n-1}} \; , \tag{2.15}$$

where $a \to \widehat{a}$ denotes the map that takes $0, 1 \in \mathbb{Z}_2$ to $0, 1 \in \mathbb{Z}$. Thereby, every input index of A becomes an output index of A^*, and vice versa. Note that the exponent in the sign prefactor is always an integer due to the parity constraint of A. Also note that this sign prefactor is the prefactor we would get from inverting the index ordering, and is alternatively expressed as

$$(-1)^{\sum_x i_x (\sum_{y>x} i_y)} \; . \tag{2.16}$$

We will denote the fermionic conjugate of a tensor copy in a fermionic tensor network by placing a $*$ symbol next to the corresponding shape.

It can easily be seen that taking the fermionic conjugate twice gives back the original fermionic tensor. Also, fermionic conjugation commutes with taking tensor products since each individual tensor satisfies the parity constraint. Further, fermionic conjugation commutes with contractions, since the ordering of the two contracted indices changes, and at the same time input and output are swapped. Combining all of the above, fermionically conjugating the result of a tensor-network diagram is the same as toggling the fermionic conjugate of every individual tensor.

We will now briefly explain how the evaluation of a fermionic tensor networks differs from that of an ordinary tensor network in practice. The difference lies in a ± 1 prefactor coming from the index transpositions we need to make to perform index contractions in the right order. To find these signs, we write down the tensor copies in some order, and label all their indices with letters. We use the same letter for contracted index pairs and put a line over the input index. Then we write down the indices of all tensors in one large sequence. For example, for Eq. (2.5) we get,

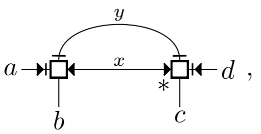
$$A_{x,y,i,z} A_{j,x,y,w} B_{w,z,k} \quad \to \quad \bar{x} y \bar{i} z \bar{j} x \bar{y} w \bar{w} \bar{z} k \; . \tag{2.17}$$

We now perform all the contractions and transpositions, giving rise to a sign $(-1)^P$, where P is a degree-2 polynomial whose variables are the (parities of) the different index labels. A contraction corresponds to removing $x\bar{x}$ from the sequence for some label x, not changing P A fermionic index permutation permutes two letters $ab \to ba$ and also changes $P \to P + ab$. Finally, we might use the parity constraint of tensor copies to simplify P, for example we have $x + y + i + z = 0$ from the first A copy in the example above, so we can change $P \to P + (x+y+i+z)f$ for any label f. Now, start from some sequence of labels S and $P = 0$ and apply the above operations until every label occurs only once in S, then P determines the overall sign. For the above example, we get the following $S|P$,

$$\begin{aligned} &\bar{x}y\overline{i}z\overline{j}x\bar{y}w\bar{w}\bar{z}k|0 \to \bar{x}y\overline{i}z\overline{j}x\bar{y}\bar{z}k|0 \to \bar{x}y\overline{i}z\bar{z}\overline{j}x\bar{y}k|zw \\ &\to \bar{x}y\overline{\overline{i}}\overline{\overline{j}}x\bar{y}k|zw \to x\bar{x}y\overline{\overline{i}}\overline{\overline{j}}\bar{y}k|zw + x(j+z) \to y\overline{\overline{i}}\overline{\overline{j}}\bar{y}k|zw + x(j+z) \\ &\to y\bar{y}\overline{\overline{i}}\overline{\overline{j}}k|zw + x(j+z) + y(i+j) \to \overline{\overline{i}}\overline{\overline{j}}k|zw + x(j+z) + y(i+j) \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \to \overline{\overline{j}}\overline{\overline{i}}k|y + xy \ . \end{aligned} \tag{2.18}$$

Thus, the overall sign of this fermionic tensor network compared to an ordinary one is $(-1)^{|y|+|x||y|}$. This sign depends on the chosen final index ordering, like jik above.

If a copy of a tensor variable is fermionically conjugated, we write down the inverse ordering of indices, and exchange input and output indices. For example, consider the diagram



, (2.19)

where one of the two copies of the tensor variable A is fermionically conjugated. The fermionic reordering sign we get We find

$$\begin{aligned} &b\bar{x}y\bar{a}d\bar{y}x\bar{c}|0 \to b\bar{x}\bar{a}dy\bar{y}x\bar{c}|y(a+d) \\ &\quad \to bx\bar{x}\bar{a}d\bar{c}|y(a+d) + x(a+d+x) \to b\bar{a}d\bar{c}|(a+b)(a+d) + x \ . \end{aligned} \tag{2.20}$$

Let us briefly relate our definitions with notions of fermionic tensors in the literature. One common way to deal with fermions is to use *Grassmann variables* $\theta_a, \theta_b, \ldots$. For example, these have been used for a long time in perturbation theory for evaluating fermionic Feynman diagrams, which secretly are fermionic tensor networks. Using the anti-commutativity $\theta_a\theta_b = -\theta_b\theta_a$ of Grassmann variables in an expression like

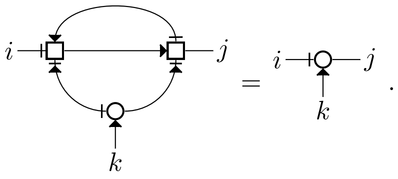
$$A_{a,b,c}(\overline{\theta}_a)^a(\theta_b)^b(\theta_c)^c \ , \tag{2.21}$$

is equivalent to fermionic index transposition of the array A. All in all, manipulating terms with Grassmann variables is equivalent to the computations described in the previous paragraphs. An example for Grassmann variables being used in a context similar to Chapter 7 of this theses is Ref. [70]. The sign factors obtained from reordering fermionic indices or Grassmann variables is also known as *Kozul sign*. Since around 2009, there are a few application-specific tensor-network formulations of fermionic sign bookkeeping in the literature, for example Ref. [15] for unitary circuits, Ref. [99] for projected entangled-pair states, or Ref. [44] for the multi-scale renormalization group ansatz.

2.3 Tensor schemes (TS) and their mappings

In this section, we introduce some basic vocabulary necessary to talk about different types of algebraic structures that can be expressed in terms of tensor-network diagrams.

Since there is no established name for this notion in the literature and it is central to our formalism, we will introduce the term *tensor schemes*. The most important ingredient in a tensor scheme are tensor-network equations, that is, equations between two tensor networks. If we want to unambiguously write down such an equation using Penrose diagrams, we have to match up all the uncontracted *open indices* on the left and right-hand side. We do this by putting matching labels at the corresponding line ends, for example,



(2.22)

When written out in terms of letters and subscripts, this equation becomes

$$\sum_{x,y,z,w} A_{x,y,i,z} A_{j,x,y,w} B_{w,z,k} = B_{j,i,k} \; . \tag{2.23}$$

We will sometimes omit the open index labels when the matching is either not that important in the current discussion or clear from the context or from the approximate positioning of the corresponding line ends. With this, we are ready for the general definition.

Definition 5. *A* tensor scheme (TS) *consists of the following.*

- *A set of* bond dimension variables.
- *A set of* tensor variables*, each of which has a* shape*. The shape consists in the number of indices, and associates a bond dimension variable to every index.*
- *A set of* moves*, which are tensor-network equations. Each tensor in such an equation is associated with a tensor variable. For every bond and every matched pair of open indices, the bond dimension variables have to agree.*

Note that usually, we demand the sets of bond dimension variables, tensor variables, and moves to be finite. Sometimes, it is useful to relax this condition at least temporarily, which we do for example in Chapter 8.

As such, tensor schemes are a purely diagrammatic notion. An actual choice of tensors is captured by the following notion of a TS model.

Definition 6. *A* model *of a TS (or in short, a* TS model*)*

- *associates to every bond dimension variable a bond dimension,*
- *associates to every tensor variable a tensor,*
- *such that all the moves hold, as tensor-network equations for the tensors.*

Many linear-algebraic structures can be formulated as TS, and concrete instances of these structures are models of these TS. For example, finite-dimensional associative algebras form a TS, and a concrete associative algebra is a model of this TS. There is one tensor variable with three indices, namely the structure coefficients of the multiplication. Associativity can be expressed as a tensor-network equation that holds for this tensor variable. We will discuss this example explicitly in Section 4.12. Note that algebraic structures are usually formulated in terms of (multi-)linear maps and their compositions. So they obey a global "flow of time", which general TS do not. For the application in this thesis, *topological TS (tTS)* are TS that describe families of fixed-point models or other descriptions of topological phases. Here, the notion of TS becomes very handy since there is a great variety of such tTS.

Next, we will introduce a powerful concept that allows us to study relations between (models of) different TS.

Definition 7. *A* TS mapping *from a TS A to a TS B*

- *associates to every bond dimension variable of A a collection of bond dimension variables of B,*
- *associates to every tensor variable of A a tensor network formed by the tensor variables of B,*
- *such that every mapped move of A can be derived from the moves of B.*

Here, by mapped move *we mean a move obtained by applying the mapping to every (copy of a) tensor variable in a move of A. A new move is* derived *from the moves of a TS B if its left-hand side is transformed into the right-hand side by applying a sequence of the moves of B.*

Again, a TS mapping is a purely diagrammatic notion, but it can be applied to concrete tensors as follows.

Definition 8. *Consider a TS mapping from a TS A to a TS B, and a model of B. To each A tensor variable, assign the evaluation of the associated B tensor-network for the B-model. This assignment of tensors to A tensor variables defines an A model due to the condition that the mapped moves of A are derived from the moves of B. We say that this A model is obtained from* applying *the mapping to the B model. Note that this application to models goes in the opposite direction from B models to A models, whereas the mapping itself goes "from A to B".*

For the application in this thesis, TS mappings allow us to relate models of different fixed-point families. While there is a rich variety of TS describing fixed-point families, the possibilities for TS mappings are even richer.

2.4 Extended manifolds and cellulations

In this section, we will introduce some basic definitions regarding cellulations and triangulations. Cellulations are of great importance for this thesis, since they are the discrete representations of spacetime to which we will associate tensors. We will further introduce the notion of an extended manifold and extended cellulation. This notion is central to our formalism if we want to either introduce any sort of topological defects, or describe phases of matter via more high-level invariant data.

By a n*-cellulation*, we mean a representation of a topological n-manifold as a cell complex M. To be precise, we work with *piece-wise linear* manifolds. For all applications in this thesis, we need to equip these cellulations with the following structure analogous to a local choice of coordinate system for a manifold.

Definition 9. *A* branching *for a n-cellulation M consists of*

- *for every $0 \leq x \leq n$, a set $\mathcal{R}_x$ of $x-1$-cellulations, which we refer to as x-*cell representatives*, and*
- *a for every x-cell, an isomorphism of cellulations between its boundary and one x-cell representative.*

We will denote the set of x-cells of M by $S_x[M]$, and those with representative S by $S_x^S[M]$.

Note that a branching is does *not* describe any global topological properties like an orientation or spin structure would. It is really just an arbitrary local choice of coordinate system, and all branchings are equivalent to each other.

The above definition of cellulations is easy to understand and gives the right intuition. However, the mathematically more elegant way is to define cellulations purely combinatorially, in which case a branching is naturally and automatically part of it. Such a definition would proceed inductively in n: The standard representatives are themselves branched $n-1$-cellulations, and the isomorphism is an isomorphism of branched cellulations. A n-cellulation is fully combinatorially determined by a *pairing* ψ between the different $n-1$-subcells for any representative $T \in \mathcal{R}_{n-1}$ of all the n-cells,

$$\begin{gathered}\psi_T : \bigcup_{S\in\mathcal{R}_n} S^T_{n-1}[S] \times S^S_n[M] \rightarrow \bigcup_{S\in\mathcal{R}_n} S^T_{n-1}[S] \times S^S_n[M]\\ \psi_T \circ \psi_T = \mathrm{id}\ , \quad \psi_T(x) \neq (x)\ .\end{gathered} \tag{2.24}$$

This paring maps between the two $n-1$-subcells corresponding to the same $n-1$-cell. For any x-cell C in a n-cellulation, we can define its *link*, which is an $n-x-1$-cellulation, whose y-cells are the $n-y$-cells adjacent to C. In other words, the link is the boundary of the $n-x$-cell corresponding to C in the Poincaré dual cellulation. The link can be determined purely combinatorially from the pairing. For a cellulation of a (piecewise linear) n-manifold, the link of each x-cell C has the topology of an $n-x-1$-sphere. Thus, combinatorially, we can define a cell complex as a set of x-cells and a pairing such that every link has sphere topology.

In many cases, it is convenient and sufficient to restrict ourselves to *triangulations*.

Definition 10. *A* triangulation *is a cellulation where all x-cells are x-simplices. When equipping a triangulation with a branching, we can take the set of x-cell representatives to only consist of one standard x-simplex. The identification of an edge with the standard 1-simplex can be indicated by giving the edge a direction. In a drawing, this direction may be indicated by an arrow on the edge. In the standard x-cell* *representative, we can choose the edge directions are non-cyclic around every of its triangles. Such a choice of non-cyclic edge orientations in a triangulation is known as a* branching structure.

After defining cellulations and triangulations of manifolds, we will now generalize this to *extended manifolds*, which are composites of different manifolds attached to each other in different ways. These will be essential for defining topological boundaries, or defects such as anyons. They will also be important if we want to describe phases in terms of higher-level invariants instead of microscopic lattice models. Let us start by a continuum definition, and then show how extended manifolds can be cellulated.

Definition 11. *Extended manifolds come in different types. An* extended manifold type *of (maximal) dimension n consists of a set of* regions*, and for each region r a* dimension $0 \leq d_r \leq n$ *and a* link L_r*, which is itself an extended manifold. The type of L_r has maximal dimension $n-d_r-1$, so the definition of extended manifolds and their types is inductive in the maximal dimension n. A region of dimension d_r will also be called a d_r-region. The type of L_r is obtained from the original type by changing the dimension or each region r' from $d_{r'}$ to $d_{r'}-d_r-1$, and then discarding regions with negative dimension. An extended manifold M of a given type consists of*

- *for every region r, a (compact piece-wise linear) d_r-manifold M_r with boundary ∂M_r, and*
- *for every pair of regions r and r', a (piece-wise linear) map*

$$\psi^M_{r,r'} : M_r \times (L_r)_{r'} \rightarrow \partial M_{r'} \ . \tag{2.25}$$

ψ is subject to two types of conditions. First, we have

$$\psi^M_{r_1,r_2}(\psi^M_{r_0,r_1}(x,p_0),p_1) = \psi^M_{r_0,r_2}(x,\psi^{L_{r_0}}_{r_1,r_2}(p_0,p_1)) \tag{2.26}$$

for all regions r_0, r_1, r_2, and $x \in r_0$, $p_0 \in (L_{r_0})_{r_1}$, $p_1 \in (L_{r_1})_{r_2}$. Additionally, we demand that

$$(\bigcup_r \psi^M_{r,r'})/[\textit{Eq.}\ (2.26)] : \quad (\bigcup_r M_r \times (L_r)_{r'})/[\textit{Eq.}\ (2.26)] \rightarrow \partial M_{r'} \tag{2.27}$$

is a (piece-wise linear) homeomorphism for each region r'. The induction in the definition is started by the fact that an extended manifold type of maximal dimension -1 has an empty set of regions. Two extended manifolds of the same type are considered equivalent if all of their regions are homeomorphic and the homeomorphisms commute with ψ.

While the above definition is precise, the following informal *collapsed picture* might be simpler to understand or draw: Start with the "topological space" formed by the union $\bigcup_r M_r$. For every region r and every point $x \in M_r$, identify the image $\psi_{r,r'}(x \times L_r) \subset \partial M_{r'}$ for each r' with the point x. We do this starting with the highest-dimensional regions. This way, we obtain one common topological space (which is not necessarily a manifold), into which $M_r \setminus \partial M_r$ is embedded for every region r. We do this for the extended manifolds themselves as well as for their links. Now, look at a point x of M_r embedded into the common topological space. Consider the space normal to M_r at x, which is of dimension $n - d_r$ if we imagine the overall space being embedded in $\mathbb{R}^n$. The distance-ϵ neighborhood within the extended manifold around the point restricted to this normal space is L_r.

As a trivial example, n-manifolds are just extended manifolds with a single n-region, whose link is the empty extended manifold without regions. Next, consider an n-dimensional extended manifold type with a n-region a and a $n-1$-region b whose link is a single point,

$$a: \ (n, \varnothing), \quad b: (n-1, \overset{a}{\bullet}) \, . \tag{2.28}$$

Here and in the following, we list regions r in the format $r : (d_r, L_r)$. Extended manifolds of this type consist of an n-manifold M_a with boundary, and a $n-1$-manifold M_b that is identified with the boundary. So in the collapsed picture, these are just manifolds with boundary. Vice versa, if we take a point x inside the boundary, its normal space is a line normal to the boundary, and the ϵ-neighborhood is a single point in M_a, which is the link of b. Next, consider the case where the link of b consists of two points,

$$a: \quad (n, \varnothing), \qquad b: (n-1, \overset{a}{\bullet} \quad \overset{a}{\bullet}) \, . \tag{2.29}$$

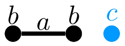
Extended manifolds of this type consist of an n-manifold M_a with boundary, where each connected component is either identified with $M_b \times 0$, or $M_b \times 1$, if we label the two points in L_b by 0 and 1. In the collapsed picture, the two points $\psi_{b,a}(x \times 0) \in \partial M_a$ and $\psi_{b,a}(x \times 1) \in \partial M_a$ are identified. After this identification, we obtain an n-manifold without boundary, but with an embedded $n-1$-manifold M_b. A slight variation is to take two different n-regions a and c, and let the link of b contain one point of each,

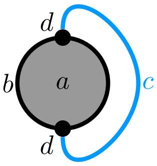
$$a: \ (n, \varnothing) \, , \quad c: \ (n, \varnothing) \, , \quad b: (n-1, \overset{a}{\bullet} \quad \overset{c}{\bullet}) \, . \tag{2.30}$$

As shown, we will color regions differently if they are not distinguishable by their dimension. Extended manifolds in the collapsed picture are now n-manifolds that are divided into a-regions and c-regions, and these regions are separated by embedded $n-1$-dimensional "domain walls" M_b. The normal space of a point x in M_b is a line, and its ϵ-neighborhood inside the line consists of two points, one inside M_a, and one inside M_c. Next, consider the extended manifold type of maximal dimension 3 given by

$$a: \ (3, \varnothing), \quad b: \ (1, \bigcirc a) \, . \tag{2.31}$$

Extended manifolds consist of a 3-manifold M_a whose boundary ∂M_a is identified with $M_b \times S_1$ for 1-manifold M_b and S_1 the circle. In the collapsed picture, we identify each circle in ∂M_a with a single point. So then extended manifolds are just 3-manifolds (without boundary) with an embedded 1-manifold. Last, let us consider a slightly more involved example,

$a:\ (3,\varnothing)\ ,\quad b:(2,$ $)\ ,\quad c:\ (2,\varnothing)\ ,\quad d:\ (1,$  $)\ ,$

$e:\ (0,$  $)$ (2.32)

Extended manifolds of this type consist of 1) a 3-manifold M_a with boundary (M_b), 2) a 2-manifold (M_c) which is attached to the boundary $\partial M_a \sim M_b$ along a line (M_d), and 3) points M_c along this line. If we take the normal plane at a point x of the line M_d where M_c is attached to the boundary M_b of M_a, then a circle inside this normal plane overlaps with M_a at an interval (whose endpoints are in M_b, and with M_c at a point. This is the link L_d. The normal space of one of the points of M_e is all of the 3-dimensional space into which we locally embed our extended manifold. The intersection of a 2-sphere around this point with M_a is a half-sphere/disk, and with M_c is an interval whose endpoints are attached to the boundary of the disk. Note that the link of d inside L_e is the same as L_d itself, and analogously for every other pair of regions.

The collapsed picture looses one important aspect of the formal definition, namely a normal framing of the various submanifolds. For example, in an extended manifold of type Eq. (2.29), the embedded $n-1$-manifold M_b has a consistent "favorite side", which can be pictured as a little arrow pointing perpendicular to the manifold everywhere. This also has global consequences: In $n=2$, consider the real projective plane $\mathbb{R}P(2)$ with an embedded non-contractible M_b circle. This does *not* define an extended manifold since we cannot consistently choose a normal direction of the line (which is reversed when we travel around the loop once). Equivalently, if we cut $\mathbb{R}P(2)$ along the line to obtain M_a we get a disk with a single circle as boundary. There is no homeomorphism between a single circle (∂M_a) and two circles ($M_b\times\{0,1\}$), so we cannot find a ψ that combines M_a and M_b to an extended manifold. As a second example, consider the type in Eq. (2.31). We have a 3-manifold M_a whose boundary is identified with $M_b\times S_1$. If we imagine the image of S_1 to be "small", then M_a is a 3-manifold with "tube-like" holes. If we mark a "base point" of the link S_1, then this base point becomes a line inside the "tube wall" under ψ. When we now shrink S_1 to a point in the collapsed picture, then the location of the base point becomes a normal vector at each point of M_b, that is, we have a normal framing of M_b. Now, imagine M_b forming a contractible loop inside some 3-manifold. When going along this loop, the normal framing might twist n times around M_b, and all these framings correspond to inequivalent extended manifolds.

After discussing the continuum picture for extended manifolds, it is now time to cellulate them.

Definition 12. *An* extended n-cellulation *is defined inductively with respect to the maximal dimension n, and with respect to an extended manifold type. For each region r,* *an extended cellulation M is defined with respect to a set $\mathcal{R}^r_x$ of x-cell representatives (for $0 \leq x \leq d_r$), which are extended $n - d_r + x - 1$-cellulations. The topology of each x-cell representative is given by*

$$\{M_{r'}, \psi_{r',r''}\}_{r',r''\neq r} = L_r \times B_x, \quad M_r = S_{x-1}, \quad \psi_{r,r'} = \mathrm{id}_{L_r \times S_{x-1} \sim \partial(L_r \times B_x)}\ . \tag{2.33}$$

An extended cellulation consists of

- *a set $S^{r,S}_{d_r}$ of extended d_r-cells for each region r and each d_r-cell representative $S \in \mathcal{R}^r_{d_r}$, and*
- *a pairing $\psi_{r,T}$ for each region r and each $d_r - 1$-cell representative $T \in \mathcal{R}^r_{d_r-1}$,*

$$\psi_{r,T} : \bigcup_{r', S \in \mathcal{R}_{r', d_{r'}}} S^{r,T}_{d_r-1}[S] \times S^{r',S}_{d_{r'}}[M] \rightarrow \bigcup_{r', S \in \mathcal{R}_{r', d_{r'}}} S^{r,T}_{d_r-1}[S] \times S^{r',S}_{d_{r'}}[M]\ . \tag{2.34}$$

We would like to stress that it is possible to pair faces of cells of different regions r'.

From the pairing and the pairing of all the x-cell representatives, we can combinatorially define the link *of an x-cell of a region r, which is a extended $n - x - 1$-cellulation. The topology of each link has to be the same as for the $d_r - x$-cell representatives in Eq.* (2.33)*.*

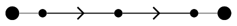
Instead of cellulations of manifolds, it is often more convenient to work with triangulations which are more restricted. The same is possible with extended cellulations.

Definition 13. Extended triangulations *are extended cellulations with only one single x-cell representative for every x and every region r, defined inductively. An extended triangulation is defined with respect to a choice of extended triangulation $C[L_r]$ of the link L_r of each region r. Then the single x-cell representative obtained via (extendedly) cellulating $L_r \times B_x$ in Eq* (2.33) *as $C[L_r] \times \mathrm{Sx}_x$ and S_{x-1} as $\partial\,\mathrm{Sx}_x$. Here,* Sx_x *denotes the (branching-structure) x-simplex. Then, $C[L_r] \times \mathrm{Sx}_x$ consists of cells that are cartesian products of simplices. We then use a standard triangulation of* $\mathrm{Sx}_x \times \mathrm{Sx}_y$ *by $\binom{x+y}{y}$ $x + y$-simplices in order to refine $C[L_r] \times \mathrm{Sx}_x$ into a triangulation.*

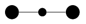
We will now give some examples for extended triangulations and cellulations. Let us start with the example in Eq. (2.28), namely manifolds with boundary. For $n = 1$, there is only one a 1-cell representative, and one b 0-cell representative,

$$a : \bullet\!\!-\!\!\!\longrightarrow\!\!\!-\!\!\bullet\ , \qquad b : \bullet\!\!-\!\!\!\!\bullet\ . \tag{2.35}$$

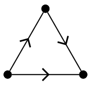
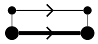
The arrow on the 1-cell denotes the branching structure, and is used to to identify a copy of this 1-cell with its standard representative. Note that according to the above definitions, the cell representatives are 0-cellulations, which would be two vertices for a, and a single vertex for b. In our drawings, we are also adding a "bulk" to these 0-cellulations, which is an edge for a, and a vertex (in fat, connected to the a vertex via a gray line) for b. An extended cellulation is essentially a cellulation with boundary, but we explicitly draw the boundary vertices, for example,

 . (2.36)

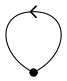
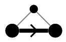
One important technical difference to cellulations with boundary is that cellulations like the following are allowed,

 . (2.37)

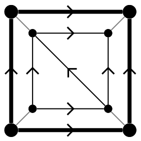
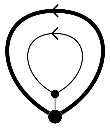
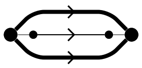
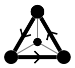
It consists of only two boundary (b) vertices, whose bulk (a) vertices are paired, without a bulk edge in between. For $n = 2$, the simplicial standard cell representatives are

$a :$  , $b :$  . (2.38)

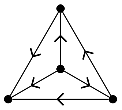
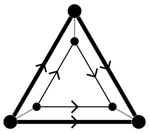
Again, the a cell representatives according to the definition above are a 1-cellulation with three edges for a, but we also imagine the "bulk" triangle it encloses. The b cell representative is an extended cellulation like in Eq. (2.36) but with only one a edge, but we also draw the "bulk" b edge. For general cellulations, other cell representatives could look like

$a :$  , or $b :$  . (2.39)

Possible extended cellulations could look like

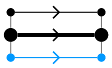
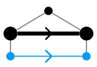
 , or  , or  , or  . (2.40)

In $n = 3$, the standard simplicial cell representatives are

$a :$  , $b :$  , (2.41)

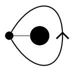
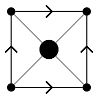
a tetrahedron, and a "triangle prism volume". Note that usually in (extended) cellulations, we only draw the 1-skeleton of the cellulation, that is, the vertices and edges. Thus the standard simplicial cell representatives above just look like extended cellulations for $n = 2$. In fact, all the example cellulations in Eq. (2.40) define possible b cell representatives, since they happen to all have disk topology.

As a next example, for the type in Eq. (2.30) with $n = 2$, possible c 1-cell representatives look like

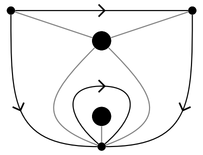
 , or  . (2.42)

Without the c edge (in fat), these drawings represent examples of extended cellulations of the type in Eq. (2.30) with $n = 1$.

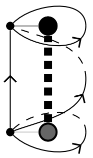
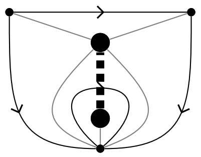
Our next example is the extended manifold type in Eq. (2.31). Possible b 0-cell representatives are given by cellulations of the (b, circle) link,

 , or  . (2.43)

As usual, we draw not only the link, but also the b vertex itself, connected to the a vertices via gray lines. As a convention, the larger the difference in dimension of a region and the maximal dimension n (here $2 = 3 - 1$ for region b), the fatter we draw the vertices and lines of this region. An extended 2-cellulation with these cell representatives could look like

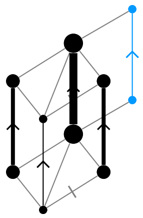
 . (2.44)

This example consists of one a triangle (at the back), and two b vertices, one of each representative in Eq. (2.43). With this, possible b 1-cell representatives are

 , or  . (2.45)

The first example is the standard simplicial representative, whereas the right example is the same as the extended cellulation in Eq. (2.44). As usual, we include the b edge in the drawing, and we will think of the representatives as volumes with the b edge in the middle (and therefore represented by a dashed line).

Finally, let us look at the extended manifold type in Eq. (2.32). The simplicial d 1-cell representative with a minimal cellulation of the link looks like,

 . (2.46)

2.5 Cellular cohomology

Cohomology is important for fixed-point descriptions of topological phases for two reasons. First, it is necessary for describing phases restricted by symmetries, or containing fermionic degrees of freedom. Second, it can be used to construct explicit examples for fixed-point models or descriptions. In this section, we will introduce the basic theory of cellular (co-)homology, which includes group cohomology.

We start by giving some intuition for cohomology on manifolds in the continuum. The central entities of (co-)homology are *i-cocycles* (or, cocycles of *degree* i). Intuitively, these can be thought of embedded sets of points for $i = n$, closed loops for $i = n - 1$, closed membranes for $i = n - 2$, and so on. Two cocycles are equivalent if they can be continuously deformed into each other. More precisely (and this is roughly the difference between homology and homotopy), the embedded closed manifolds are dressed by elements of an Abelian *coefficient group* A. So cocycles are more like networks of submanifolds, with branching rules according to the A group labels.

Let us now start with the technical definitions. The most basic objects in cellular cohomology are *chains*.

Definition 14. *For an n-cellulation M, an A-valued* i-chain *for* $0 \leq i \leq n$ *is a map*

$$c : S_i[M] \to A \;, \tag{2.47}$$

where $S_i[M]$ is the set of i-cells of M.

For technical reasons, we need to start by defining the boundary for $\mathbb{Z}_2$-valued chains.

Definition 15. *For $A = \mathbb{Z}_2$, the* boundary *of an i-chain is the $i - 1$-chain δc given by*

$$(\delta c)(s) = \sum_{t \in S_{i+1}[s]} c(t) \;. \tag{2.48}$$

We need the boundary operation to define a *local orientation* on general cellulations, which in turn is needed to define δ for general A. Intuitively, a local orientation assigns to every $x - 1$-cell of an x-cell whether it is "left-handed" or "right-handed", in a consistent way. The formal definition is inductive in the dimension n.

Definition 16. *A* local orientation *for a collection of x-cell representatives is a choice of* orientation $\sigma[C]$ *for every representative C. An* orientation $\sigma[M]$ *for an n-cellulation M assigns a $\mathbb{Z}_2$-valued n-chain, such that*

$$\delta\sigma[M](c) = 1 + \sum_{X \in S_n[c]} \sigma[X] \;, \tag{2.49}$$

for every $n - 1$-cell $c \in S_{n-1}[M]$. Here, $S_n[c]$ denotes the set of n-cells containing c, and X denotes an n-cell as well as its representative.

Intuitively, the $n - 1$-cycle on the right-hand side of Eq. (2.49) describes the "domain walls" where the orientation should change. At a $n - 1$-cell c, the orientation should change if $\sigma[X_1](c) = \sigma[X_2](c)$ for the two n-cells X_1 and X_2 adjacent to c.

The full definition of (co-)boundaries depends on a local orientation. Note that we do not need an orientation, and the local orientation does not imply that our cellulations are orientable.

Definition 17. *The* boundary *of an i-chain c is the $i-1$-chain δc given by*

$$(\delta c)(s) = \sum_{t \in S_i[s]} (-1)^{\sigma[s](t)} c(t) \; , \tag{2.50}$$

for every $i-1$-cell $s \in S_{i-1}[M]$. The coboundary *of c is the $i+1$-chain dc given by*

$$(dc)(s) = \sum_{t \in S_i[s]} (-1)^{\sigma[t](s)} c(t) \; , \tag{2.51}$$

for every $i+1$-cell $s \in S_{i+1}[M]$. An i-cycle *is an i-chain c with $\delta c = 0$. An i*-boundary *is a i-chain c such that there exists an $i+1$-chain x with $c = \delta x$. An i*-cocycle *is an i-chain c with $dc = 0$. An i*-coboundary *is a i-chain c such that there exists a $i-1$-chain x with $c = dx$. i-chains of which we take the coboundary are usually called i-*cochains *instead. We will denote the sets of i-chains, i-cochains, i-cycles, i-cocycles, i-boundaries, and i-coboundaries as $C_i(M,A)$, $C^i(M,A)$, $Z_i(M,A)$, $Z^i(M,A)$, $B_i(M,A)$, and $B^i(M,A)$, respectively. All of these sets form groups under cell-wise A-addition. The (co-)boundary maps on the sets of (co-)chains of different degree define* chain complexes*, meaning for any (co-)chain c, $d^2c = 0$ ($\delta^2 c = 0$). Thus we have $B_i \subset Z_i$ and $B^i \subset Z^i$. The quotient group $H_i := B_i/Z_i$ ($H^i := B^i/Z^i$) is called the ith* (co-)homology group. *Its elements, which are equivalence classes of a (co-)cycles modulo (co-)boundaries, are known as* (co-)homology classes.

For $A = \mathbb{Z}_2$, or more generally $A = \mathbb{Z}_2^k$, the factors of $(-1)^\sigma$ can be removed. In this case, all definitions are symmetric under Poincaré duality: i-cycles are the same as $n-i$-cocycles on the Poincaré dual cellulation. For A with elements of order larger than 2, the definitions are not dual since both involve the local orientation σ of the primal lattice and not the dual one. There are thus some differences between i-homology and $n-i$-cohomology. For example, a 0-cocycle is any constant function on the vertices of a cellulation, and the 0-cohomology group is just A for a connected cellulation. In contrast, for $A = \mathbb{Z}_l$ with $l > 2$ written additively, n-cycles c are of the form

$$c(X) = (-1)^{\sigma(X)} a \; , \tag{2.52}$$

for any element $a \in A$. So if M is non-orientable and l is odd, then the only n-cycle is the trivial one with $a = 0$.

Finally, note that A must be abelian since there is no canonical order how to sum the group elements in the definition of the (co-)boundary. However, for 1-cocycles ($n-1$-cycles), there is a canonical ordering of the edges around a face ($n-1$-cells around a $n-2$-cell) up to cyclic permutations. If for an edge, the product of group elements at the surrounding faces equals 1, then the same is true for any cyclic permutation. Thus, 1-cocycles ($n-1$-cocycles) are also defined for non-abelian groups, even though the (co-)boundary is not defined.

2.6 Cellular cohomology operations and group cohomology

In this section, we will discuss how *cohomology operations*, which roughly are operations mapping (co-)cycles to (co-)cycles of a smaller (larger) degree. Cohomology operations are important since they will define the *action* that defines cohomology-based models together with the coefficient group A. Roughly, a cohomology operation ω is a function

$$\omega : Z^i(M,A) \rightarrow Z^j(M,B) , \tag{2.53}$$

for fixed degrees $j > i$ and abelian groups A and B, and arbitrary cellulations M. More precisely, we actually do not mean any such function, but merely a local formula defining such a function. That is, a prescription that computes the value of $\omega(a)$ on a i-cell c depending only on the value of a in a constant-size neighborhood of c. One might straight-forwardly generalize the notion to k-ary cohomology operations with k cycles of different degree as input.

In the mathematics literature, cohomology operations are usually defined quite differently from "locally computable functions on cellular cocycles". There, cohomology operations are defined for *singular* cocycles on continuum n-manifolds, and in fact also on more general topological spaces. A particularly important object in the study of A-valued cohomology operations is the *classifying space* $B^i(A)$, together with its *universal* i*-cocycle* on it. The classifying space is an infinite-dimensional topological space with the following property: For every A-valued i-cocycle a on a manifold or topological space M, there is a continuous map $M \rightarrow B^i(A)$, the *classifying map*, such that a is the pullback of a the universal cocycle onto M. A cohomology operation then corresponds to a j-cocycle in $Z^j(B^i(A), B)$, which is pulled back to an i-cocycle on $Z^i(M,A)$ via the classifying map. For discrete A, $B^i(A)$ is equal to the *Eilenberg-MacLane space* $K(A,i)$, and can be explicitly constructed as a *simplicial Abelian group* via the *Dold-Kahn correspondence*. As a result, cohomology operations possess simple local combinatorial formulas in simplicial homology, where the output j-cochain of a cohomology operation on a j-simplex can be obtained from the input cochains on its sub-simplices.

Having sketched how some keywords from the math literature are related to our setting, let us now go back to cohomology operations as local formulas for functions mapping cocycles to cocycles. Cohomology operations correspond to local formulas (which we will call *cellular cocycle operation* below), where the dependence of $\omega(a)$ on a is particularly simple: $\omega(a)(c)$ only depends on the value of a on the cells contained in c. The reason for this is that cohomology operations are defined for arbitrary topological spaces or simplicial complexes, rather than manifolds or triangulations.

Definition 18. *For $j \geq i$ and two abelian groups A and B,* cellular chain operation ω *from $H^i(A) \rightarrow H^j(B)$ consists of a map*

$$\omega[X] : Z^i(X,A) \rightarrow B , \tag{2.54}$$

for each j-cell representative X. The coboundary *of ω is the cohomology operation*

$d\omega$ from $H^i(A) \to H^{j+1}(B)$ defined by

$$(d\omega)[R](C) := \sum_{x\in S_j[R]} (-1)^{\sigma[R](x)} \omega[x](C|_x) = 0 \;, \tag{2.55}$$

for every $j+1$-cell representative R and every $C \in Z^i(R,A)$. Cellular chain operations ω with $d\omega = 0$ will be called cellular cocycle operations. *Cellular chain operations ω for which there is η with $d\eta = \omega$ will be called* cellular coboundary operations. *Two cellular cocycle operations are equivalent if they differ by a cellular coboundary operation.*

Above we have defined cellular cocycle operations by the data needed to compute them locally. These local formulas are applied to a cellular cocycle a as follows.

Definition 19. *Given a cellular chain operation ω from $H^i(A) \to H^j(B)$ and a cocycle $a \in Z^i(M,A)$, we can define a cocycle $\omega(a) \in Z^j(M,B)$ by*

$$\omega(a)(X) = \omega[X](a|_X) \;. \tag{2.56}$$

Per construction of the boundary map in Eq. (2.55)*, we have*

$$(d\omega)(a) = d(\omega(a)) \;. \tag{2.57}$$

Thus, if ω is a cellular cocycle operation, then $\omega(a)$ is a cocycle. If ω is a cellular coboundary operation, then $\omega(a)$ is a coboundary.

For many use-cases, we can limit ourselves to branching-structure triangulations, where the only x-cell representative is the x-simplex Sx_x. Then, a cellular cocycle operation is given by a function $\omega := \omega[\mathrm{Sx}_j]$, subject to one condition related to Sx_{j+1}. i-cochains on Sx_j are determined by $\binom{j+1}{i+1}$ A-elements associated to all the i-subsimplices of Sx_j. For a i-cocycle a, there are $\binom{j+1}{i+2}$ relations corresponding to $da = 0$ on every $i+1$-simplex, and $\binom{j+1}{i+3}$ dependencies among these relations corresponding to all the $i+2$-simplices on which we must have $dda = 0$. Thus, a i-cocycle is determined by $\binom{j+1}{i+1} - \binom{j+1}{i+2} + \binom{j+1}{i+3}$ A-elements, and a cellular cocycle operation is determined by a function

$$\omega : A^{\binom{j+1}{i+1} - \binom{j+1}{i+2} + \binom{j+1}{i+3}} \to B \;. \tag{2.58}$$

This function is subject to one constraint for every i-cocycle on a $j+1$-simplex, that is, it is subject to $\binom{j+2}{i+1} - \binom{j+2}{i+2} + \binom{j+2}{i+3}$ equations.

Let us consider some special cases for i and j. First, let us look at the case of $i=1$ and arbitrary j. Cellular cocycle operations from $H^1(A) \to H^j(B)$ (or equivalently in $Z^j(B^1A, B)$) are known as *group cocycles*, often denoted in short by $Z^j(A,B)$. Accordingly, the corresponding cohomology group is called *group cohomology*. A 1-cocycle a on Sx_j is determined by the j A-elements $g_x := a(x, x+1)$ of a on the edge with vertices $x, x+1$ for $0 \leq x \leq j$. The value of a on the other edges is determined by

$$a(x,y) = \sum_{x\leq i<y} a(i, i+1) \;. \tag{2.59}$$

Thus, a cellular cocycle operation $H^1(A) \to H^j(B)$ is defined by a function

$$\omega : A^j \to B \; . \tag{2.60}$$

There is one constraint for every 1-cocycle on a $j+1$-simplex, given by $j+1$ group elements g_x. The xth sub j-simplices of a $j+1$ simplex is spanned by the edges

$$(0,1),(1,2)\ldots(x-2,x-1),(x-1,x+1),(x+1,x+2)\ldots(j,j+1) \; . \tag{2.61}$$

So the constraint is given by

$$\begin{aligned}&(d\omega)(g_0,\ldots,g_j) = \omega(g_1,\ldots,g_j)\\ &\quad\prod_{0\leq x<j} \omega(g_0,\ldots g_{x-1}, g_x g_{x+1}, g_{x+2},\ldots,g_j)^{(-1)^x} \omega(g_0,\ldots,g_{j-1})^{(-1)^j} \; ,\end{aligned} \tag{2.62}$$

where we set $g_{-1}=g_{j+1}=1$ for some limiting x. For example, for $j=2$, we have

$$(d\omega)(g_0,g_1,g_2) = \omega(g_1,g_2)\omega(g_0 g_1,g_2)^{-1}\omega(g_0,g_1 g_2)\omega(g_0,g_1)^{-1} \; . \tag{2.63}$$

Next, let us consider the case $i=j$, that is, cohomology operations $H^i(A) \to H^i(B)$. There is only one i-subsimplex of the i-simplex, so such a cohomology operation is determined by a function

$$\omega : A \to B \; . \tag{2.64}$$

This function has to fulfill the constraint

$$\prod_x \omega(g_x) = 1 \quad \forall \{g_x \in G\}_{0\leq x\leq i} : \prod_x g_x = 1 \; . \tag{2.65}$$

This is precisely the case if ω is a group homomorphism $A \to B$.

As a next case, let us consider $j=i+1$, so, cohomology operations $H^i(A) \to H^{i+1}(B)$. It is known that all such cohomology operations are given by the *Bockstein homomorphism* for some short exact sequence

$$B \xrightarrow{f} C \xrightarrow{g} A \; . \tag{2.66}$$

Here, C is another Abelian group, f is an injective and g a surjective group homomorphism, and $g \circ f$ is the trivial homomorphism sending everything to the identity of A. The Bockstein homomorphism β has a simple combinatorial formula in terms of triangulations, and maps an i-cocycle c to an $i+1$-cocycle

$$\beta(c) = f^{-1}(d(g^{-1}(c))) \; , \quad g \circ g^{-1} = \mathrm{id}_A \; , \quad f^{-1} \circ f = \mathrm{id}_B \; , \tag{2.67}$$

where both g^{-1} and f^{-1} are applied simplex-wise. $d(g^{-1}(c))$ is in the image of f, so the choice of f^{-1} does not matter, and the result is indeed an $i+1$-cocycle. Different choices for the right inverse g^{-1} of g yield different, but equivalent cocycle operations. One example is the Bockstein homomorphism for the short exact sequence

$$\mathbb{Z} \xrightarrow{\cdot 2} \mathbb{Z} \xrightarrow{\text{mod } 2} \mathbb{Z}_2 \; , \tag{2.68}$$

defining a cohomology operation $H^i(\mathbb{Z}_2) \to H^{i+1}(\mathbb{Z})$.

Next, we can also define cocycle operations with multiple inputs.

Definition 20. *An mary cochain operation ω from $H^{i_0}(A_0)\times H^{i_1}(A_1)\times\ldots\to H^j(B)$ is a map*

$$\omega[X] : Z^{i_0}(X, A_0) \times Z^{i_1}(X, A_1) \times \ldots \to B \ , \tag{2.69}$$

for every j-cell representative X. The definition of coboundary, cocycle operations, etc., are analogous to the unary case.

A rather trivial first example for a binary cocycle operation $H^j(A) \times H^j(A) \to H^j(A)$ is the simplex-wise product of group elements.

A particularly important family of interesting binary chain operations are the *higher order cup products* $\cup_x$ from $H^i(\mathbb{Z}) \times H^j(\mathbb{Z}) \to H^{i+j-x}(\mathbb{Z})$. For the application of the higher cup product to a pair of cocycles a and b, we write $\cup_x(a, b) := a \cup_x b$ as common in the literature. Intuitively, the (ordinary) cup product $\cup := \cup_0$ of two cocycles is given by the intersections of the corresponding closed loops, membranes, etc. So the ordinary cup product is in fact a cocycle operation. The cocycle operation $A \cup B$ is equivalent to $B \cup A$, and their difference is the boundary of the first order cup product $\cup_1$. Next, the equivalence between $A \cup_1 B$ and $B \cup_1 A$ is given by $\cup_2$, and so on. Note that $\cup_x$ for $x \neq 0$ is itself not a cocycle operation but only a chain operation, but it can be used to build cocycle operations.

One crucial property of the (higher order) cup product is that it is $\mathbb{Z}$-bilinear, that is, it is of the form,

$$\cup_x^{i,j}[X](x, y) := \sum_{r \in S_i[X], s \in S_j[X]} \Cup_x^{i,j}[X](r, s) x(r) y(s) \ , \tag{2.70}$$

for $\Cup_x^{i,j}[X] : S_i[X] \times S_j[X] \to \mathbb{Z}$ defined below. For simplicity, we restrict ourselves to branching-structure triangulations where the only i-cell representative X is the i-simplex Sx_i. In this case, an explicit formula for the cup product can be found in Ref. [129], namely

$$\Cup_x^{i,j}[\mathrm{Sx}_{i+j-x}](r, s) := \begin{cases} 1 & \text{if} \quad r = [0r_0][r_1 r_2] \ldots [r_{x_e-1} r_{x_e}] \ , \\ & \text{and} \quad s = [r_0 r_1][r_2 r_3] \ldots [r_{x_e} r_{x_e+1}] \ , \\ 0 & \text{otherwise} \end{cases} \tag{2.71}$$

for some $0 \leq r_0 < r_1 \ldots \leq r_{x+1} = n$. Here, $[ab]$ with $a < b$ integers denotes the sequence of integers from a to b, and x_e is x for even x and $x - 1$ for odd x. The ordinary 0th order cup product has an especially simple formula,

$$\Cup_0^{i,j}[\mathrm{Sx}_{i+j}](r, s) := \begin{cases} (-1)^r & \text{if} \quad r = [0r] \ , \quad s = [r, i+j] \\ 0 & \text{otherwise} \end{cases} \ . \tag{2.72}$$

The (higher order) cup products can be used to construct a variety of cellular cocycle operations by composing them in different ways, or also composing them with group cohomology or the Bockstein homomorphism. Different compositions might yield equivalent cocycle operations. For example, we can combine two cup products to a 3ary cohomology operation

$$(X, Y, Z) \to (X \cup Y) \cup Z \ . \tag{2.73}$$

The cup product is associative, which means that we have an equivalence

$$(X \cup Y) \cup Z = X \cup (Y \cup Z)\ . \tag{2.74}$$

In fact, this is not only a cohomological equivalence, but a direct equality of cocycles if we use the cup product formula in Eq. (2.71). Another direct equality for this cup product formula is given by its bilinearity,

$$X \cup (Y + Z) = X \cup Y + X \cup Z\ . \tag{2.75}$$

A cohomology equivalence that is not a direct equality is the graded-commutativity of the cup product,

$$X \cup Y = (-1)^{ij} Y \cup X\ , \tag{2.76}$$

where X is an i-cocycle and Y a j-cocycle. This local equivalence is not an equality of cocycles since the cup product formula in Eq. (2.71) is not symmetric in A and B. However, using the formula in theorem 5.1 of Ref. [129] we see that the difference of the left and right-hand side is given by the following coboundary,

$$X \cup Y - (-1)^{ij} Y \cup X = d((-1)^{i+j+1} X \cup_1 Y)\ . \tag{2.77}$$

All in all, different orderings, placing brackets, or expansions of cup products and sums yield locally equivalent formulas. We may thus think of expressions involving addition and cup products as polynomials.

Let us look at some examples for cellular cocycle operations that are defined via (higher order) cup products. For the usage in the main text, we are mostly interested in the case where A is a finite group B is either another finite group, or $B = U(1)$. We start with the kth *Steenrod square* $\mathrm{Sq}^k(x)$, for $1 \leq k \leq i$, from $H^i(\mathbb{Z}_2) \to H^{i+k}(\mathbb{Z}_2)$. An explicit combinatorial formula for cocycles can be found in Ref. [129], see also Ref. [70],

$$\mathrm{Sq}^k(x) = (\bar{x} \cup_{i-k} \bar{x}) \bmod 2\ . \tag{2.78}$$

Here, $\bar{x} \in Z^k(\mathbb{Z})$ is any $\mathbb{Z}$-valued cocycle such that $\bar{x} \bmod 2 = x$.

Also many group cocycles can be expressed in terms of cup products. For example, there are cellular cocycle operations from $H^1(\mathbb{Z}_l) \to H^2(U(1))$, or group 2-cocycles in $H^2(\mathbb{Z}_l, U(1))$ given by

$$\omega(x) = e^{\frac{2\pi i p}{l} \bar{x} \cup \bar{x}}\ . \tag{2.79}$$

However, this cocycle operation is equivalent to the trivial one. Another example is the group cocycle $H^1(\mathbb{Z}_l \times \mathbb{Z}_k) \to H^2(U(1))$ given by

$$\omega((x, y)) = e^{\frac{2\pi i p}{\gcd(l,k)} \bar{x} \cup \bar{y}}\ , \tag{2.80}$$

for $0 \leq p < \gcd(l, k)$. Plugging in the formula for the cup product in Eq. (2.72), we get

$$\omega((g_{01}, h_{01}), (g_{12}, h_{12})) = e^{\frac{2\pi i p}{\gcd(l,k)} \overline{g_{01}}\, \overline{h_{12}}}\ . \tag{2.81}$$

In fact every cocycle operation $H^1(\mathbb{Z}_l \times \mathbb{Z}_k) \to H^2(U(1))$ is equivalent to one of this form. As a next example, let us consider group 3-cocycles. Every group 3-cocycle in $H^3(\mathbb{Z}_l, U(1))$ is equivalent to one of the form

$$\omega(x) = e^{\frac{2\pi i p}{l}\bar{x}\cup\beta x} = e^{\frac{2\pi i p}{l^2}\bar{x}\cup d\bar{x}} \ , \tag{2.82}$$

where β denotes the Bockstein homomorphism from Eq. (2.67) for the short exact sequence

$$\mathbb{Z} \xrightarrow{\cdot l} \mathbb{Z} \xrightarrow{\mod l} \mathbb{Z}_l \ . \tag{2.83}$$

Using the simplicial formulas for the boundary and cup product, we get

$$\omega(g_{01}, g_{12}, g_{23}) = e^{\frac{2\pi i p}{l^2}\overline{g_{01}}(\overline{g_{12}+g_{23}} - \overline{g_{12}} - \overline{g_{23}})} \ . \tag{2.84}$$

An example for a group 3-cocycle in $H^3(\mathbb{Z}_l \times \mathbb{Z}_k, U(1))$ given by

$$\omega((x,y)) = e^{\frac{2\pi i p}{lk}\bar{x}\cup d\bar{y}} \ , \quad 0 \leq p < \gcd(l,k) \ . \tag{2.85}$$

Finally, an example for a group 3-cocycle in $H^3(\mathbb{Z}_l \times \mathbb{Z}_k \times \mathbb{Z}_m, U(1))$ is given by

$$\omega((x,y,z)) = e^{\frac{2\pi i p}{\gcd(l,k,m)}\bar{x}\cup\bar{y}\cup\bar{z}} \ . \tag{2.86}$$

The three examples above generate all group 3-cocycles for abelian groups. Last, consider some group 4-cocycles. A set of group 4-cocycles in $Z^4(\mathbb{Z}_l \times \mathbb{Z}_l, U(1))$ is given by

$$\omega((x,y)) = e^{\frac{2\pi i}{l^2}(p_x \bar{x}\cup\bar{y}\cup d\bar{y} + p_y \bar{y}\cup\bar{x}\cup d\bar{x})} \ . \tag{2.87}$$

We believe that every such 4-cocycle is equivalent to one of the above form.

Finally, let us consider a family of examples for a cocycle operation $H^2(\mathbb{Z}_{2l}) \to H^4(U(1))$, namely

$$\omega(x) = e^{\frac{2\pi i p}{4l}(\bar{x}\cup\bar{x} + \bar{x}\cup_1 d\bar{x})} \ . \tag{2.88}$$

This corresponds to a cohomology operation known as the *Pontryagin square*.

2.7 Cellular characteristic classes

In this section, we will make a cellular-cohomology definition for *characteristic classes*, or more precisely, characteristic classes of the tangent bundle. We will think of characteristic classes as local formulas that compute an i-cocycle ω from the combinatorics of the surrounding cellulation alone. That is, it is a prescription to compute the value of ω on an i-cell c depending on the structure of the cellulation surrounding c. So in this respect, characteristic classes are very similar to cohomology operations, or better, 0ary cohomology operations.

Again, this perspective is rather different from the common math literature [113], of which we will sketch a few important notions in the following. There, a characteristic class can roughly be understood as cycles that can be locally computed from the metric of a continuum manifold. More generally, characteristic classes are defined for manifolds equipped with a *fiber bundle*, in which case they can be computed from a *connection* of the bundle. A fiber bundle with *fiber* i-manifold F and *base* n-manifold B is a in-manifold which locally looks like $B \times F$, but where the copies of F at different points of B are not identified trivially but via an element of a *structure group* G that is represented as homeomorphisms over F. For our purposes, we are interested in characteristic classes for the *tangent bundle* of an n-manifold, which is a fiber bundle with fiber $\mathbb{R}^n$ and structure group $O(n)$ (or $SO(n)$ in the oriented case). A central tool to analyze characteristic classes is the *classifying space* BG of the structure group G (in our case $BO(n)$ or $BSO(n)$), which is some infinite-dimensional topological space, and a *universal bundle* over this classifying space, with the following property: Isomorphism classes of bundles with base B are in one-to-one correspondence with homotopy classes of *classifying maps* (maps up to continuous deformations) from B to BG, and are obtained as a pullback of the universal bundle of BG via the classifying map. Characteristic classes are then just cohomology classes of BG which give rise to cohomology classes of B, also by pullback via the same classifying map. Like cohomology operations, characteristic classes are known to be computable via local formulas on triangulations to a certain extent. Unfortunately, the contemporary understanding here is much less complete than for cohomology operations. In Ref. [107], the authors explicitly construct simplicial combinatorial representations for $\mathbb{R}^i$-bundles, the classifying space, and classifying map, and show that for every degree-i characteristic class there exist local formulas that determine the value of a representing $n-i$-cycle on an $n-i$-simplex depending only on its *star*. However, explicit combinatorial formulas are only known for some of the characteristic classes.

Let us now get back to characteristic classes as locally computable cycles on cellulations. As we mentioned above, characteristic classes are defined only for manifolds, and not for more general topological spaces. As a consequence, ω for an i-cell c depends not only on the i-cell representative of c, but on the cellulation in some larger neighborhood. This makes sense since in a triangulation where there is only one i-cell representative, such a local formula would be immediately trivial. Let us start by introducing further vocabulary to describe the larger neighborhood.

Definition 21. *Consider a branching-structure triangulation. The* star *of an x-simplex is the n-triangulation of an n-ball consisting of all n-simplices containing the x-simplex. If the x-simplex occurs multiple times in an adjacent n-simplex, we take multiple copies of this n-simplex. A* dually branched *triangulation is one where the star of every x-simplex is identified with one of a finite set of x-*star representatives. *As usual, this definition is actually inductive in x: The star representatives themselves are dually branched triangulations.*

Using the methods of Chapter 11.1 (specifically the universality proofs in the case where all tensors are scalars), one can show that any local formula computing a characteristic class can be brought into a form where its value on a x-simplex only depends on its x-star representative This motivates the following definition.

Definition 22. *An A-valued* cellular characteristic chain ω *of degree x associates to every x-star representative R a value*

$$\omega[R] \in A \,. \tag{2.89}$$

The boundary *of ω is the A-valued cellular characteristic chain $\delta\omega$ of degree $x-1$ given by*

$$(\delta\omega)[S] = \sum_{R\in S_x[S]} (-1)^{\sigma[X](R)}\omega[R] \ . \tag{2.90}$$

Here, R denotes at the same time an x-simplex of S as well its x-star representative. Also, $\sigma[X](R)$ encodes whether R is positively or negatively oriented inside S. We call ω a cellular characteristic cycle *if $\delta\omega = 0$, and a* cellular characteristic boundary *if $\omega = \delta\eta$ for some η. We will say that two cellular characteristic cycles ω_1 and ω_2 are equivalent if $\omega_1 - \omega_2$ is a cellular characteristic boundary.*

Above we defined cellular characteristic cycles via their local formulas. Given a concrete dually branched triangulation, we can evaluate these local formulas to get an actual cycle.

Definition 23. *Given a dually branched triangulation M and a cellular characteristic chain ω, we can define the x-cycle $\omega(M)$ by*

$$\omega(M)(S) = \omega[S] \ , \tag{2.91}$$

again using S for both the x-cell and its x-star representative. If ω is a cellular characteristic cycle (boundary), then $\omega(M)$ is a cycle (boundary).

Note that in contrast to cellular cocycle operations, the formulas for cellular characteristic chains can only be applied in a fixed dimension n. Also note that cellular characteristic cycles of degree i correspond to characteristic classes of degree $n-i$. Many characteristic classes are *stable*, which for us means that they are families characteristic cycles of degree $n-i$ in n dimensions for a fixed i.

Let us now consider three examples for cellular characteristic cycles that correspond to well-known characteristic classes. Let us start with the (non-stable) *Euler characteristic class* e, giving rise to a $\mathbb{Z}$-valued degree-0 characteristic class in n dimensions. A local formula for the value of e on a vertex v is given by

$$e(v) = \sum_{0\leq x\leq n} (-1)^x \sum_{t\in S_x[v]} \delta_{v|t,0} \ , \tag{2.92}$$

where $v|t$ is the vertex number of v in t. In other words, $e(v)$ is the number of x-simplices that contain v as their 0-vertex (including v itself), weighted by $(-1)^x$. It is easy to see that the summation of e over all vertices yields the *Euler characteristic*.

Next, the (stable) ith *Stiefel-Whitney characteristic class* ω_i corresponds to a $\mathbb{Z}_2$-valued degree-$n-i$ cellular characteristic cycle in n dimensions. Following Ref. [72], a local formula computing its value on an $n-i$-simplex s is given by

$$\omega_i(s) = \sum_{n-i\leq x\leq n} \sum_{t\in S_x[s]} E(s|t) \ . \tag{2.93}$$

Here, $s|t$ denotes the sequence of vertex numbers $0 \leq \{(s|t)_l\}_{0\leq l\leq n-i} \leq x$ of the t-vertices that span the subsimplex s, and $E(s|t)\in\mathbb{Z}_2$ is given by

$$E(s|t) = \begin{cases} 1 & \text{if} \quad (s|t)_{2l} = (s|t)_{2l-1}+1 \quad \forall \quad 1\leq l < (n-i)/2 \ , \\ & \text{and} \quad (s|t)_0 = 0 \ , \quad \text{and} \quad (s|t)_{n-i} = x \quad \text{if} \quad n-i \quad \text{even} \\ 0 & \text{otherwise} \end{cases} . \tag{2.94}$$

For example, for $i = n$ we have $E(s|t) = \delta_{s|t,0}$, so the formula coincides with the $\mathrm{mod}\,2$ reduction of the Euler class in Eq. (2.92). For $i = 1$, $E(s|t) = 1$ if $s = t$, and $E(s|t) = \sigma[t](s)$ if t is an n-simplex, that is, $E(s|t) = i \bmod 2$ if s equals t with the ith vertex missing. Thus, ω_1 precisely consists of the $n-1$-simplices where the orientation of the n-simplices would change if we choose an orientation locally.

The (stable) ith *Pontryagin class* P_i corresponds to a $\mathbb{Z}$-valued degree-$n-4i$ cellular characteristic cycle in any spacetime dimension n. While they must exist due to Ref. [107], there are to date no known local simplicial formulas for the Pontryagin classes that are comparable to the ones above in simplicity. Nonetheless, it has been shown in Ref. [68] that there exist formulas for computing a $\mathbb{Q}$-valued $n-4i$-cycle P_i representing the ith Pontryagin class locally from a triangulation. The value of P_i on an $n-4i$-simplex depends only on its link and the orientation on the latter. The formula depends neither on a branching structure nor any other decoration of the triangulation. Ref. [69] presents a more or less explicit formula computing the $\mathbb{Q}$-value of P_1 from the link of a $n-4$-simplex. It is conceivable that a formula for a $\mathbb{Z}$-valued instead of $\mathbb{Q}$-valued $n-4i$-cycle is possible when it is allowed to depend on a branching and dual branching. Note that unlike the two characteristic classes above, the Pontryagin classes are defined on *oriented* manifolds, and reversing the orientation corresponds to a $\mathbb{Z}$-inversion of the Pontryagin class.

Last, the cellular characteristic cycles above can be combined with cohomology operations. To be precise, since cellular characteristic cycles are cycles and not cocycles, we need to formulate the cohomology operations as acting on cycles instead of cocycles. Since cycles are essentially cocycles on the Poincaré dual cellulation, this boils down to defining these cocycle operations on arbitrary cellulations instead of just triangulations, which can be done. For example, combining a Stiefel-Whitney class with the Bockstein homomorphism for $\mathbb{Z}_2 \rightarrow \mathbb{Z}_4 \rightarrow \mathbb{Z}_2$, yields a degree-$n-i-1$ cellular characteristic cycle $\beta\omega_i$. Or, $\omega_i \cup \omega_j$ yields a degree-$n-i-j$ cellular characteristic cycle.

Characteristic classes of the above kinds have been fully classified in the mathematics literature. For example, the $\mathbb{Z}_2$-valued characteristic classes are elements of the $\mathbb{Z}_2$-valued cohomology group of $BO(n)$, which is computed, for example, in Ref. [113]. These characteristic classes are precisely polynomials of Stiefel-Whitney classes, where the product is the cup product. Accordingly, the corresponding characteristic cycles are compositions of the Stiefel-Whitney classes with the cup product. Equivalence classes of $\mathbb{Z}_2$-valued *oriented* characteristic classes, corresponding to cohomology classes of $BSO(n)$, are the same except that ω_1 is trivial. The $\mathbb{Z}$-valued cohomology of $BO(n)$ or $BSO(n)$ is more complicated to describe [36]. It is generated via cup-product polynomials by the Euler class, Pontryagin classes, but also of $\mathbb{Z} \rightarrow \mathbb{Z} \rightarrow \mathbb{Z}_2$ Bockstein homomorphisms of products of different even-degree Stiefel-Whitney classes ω_{2i}. These polynomials are subject to several equivalence relations.

Since every $\mathbb{Z}_2$-valued characteristic cycle is equivalent to a polynomial of Stiefel-Whitney classes, we can expand every "new" characteristic cycle as such a polynomial. For example, the Steenrod squares of a Stiefel-Whitney class is a new characteristic cycle, which expands as

$$Sq^i(\omega_j) = \sum_{t=0}^{i} \binom{j-i-1+t}{t} \omega_{i-t} \cup \omega_{j+t} \ . \tag{2.95}$$

This expansion is known as the *Wu formula* (cf. page 197 in Ref. [112]). The $\mathrm{mod}\, 2$ reduction of a $\mathbb{Z} \to \mathbb{Z} \to \mathbb{Z}_2$ Bockstein homomorphism is equivalent to the Steenrod square Sq^1, which applied to a product of Stiefel-Whitney classes can be obtained from Eq. (2.95). The $\mathrm{mod}\, 2$ reduction of a Pontryagin class can be expanded as

$$P_i \bmod 2 = \omega_{2i} \cup \omega_{2i} \ , \tag{2.96}$$

cf. page 181 in Ref. [113].

Chapter 3

Phases in tensor-network path integrals

In this chapter, we introduce (tensor-network) path integrals, and motivate why we use them. We review established notions of phases of matter and discuss how to adapt them to the setting of tensor-network path integrals. Finally, we will introduce fixed-point models and motivate why they obey a notion of spacetime topological invariance.

3.1 Physical systems as tensor networks

Phases of matter describe the collective behavior a large number of degrees of freedom distributed over some space or spacetime. In quantum condensed-matter physics, such models with many degrees of freedom are usually described by a Hamiltonian. In this thesis we break with this tradition. Instead, we describe physical models in terms of *discrete path integrals*. We prefer discrete path integrals over Hamiltonians for a number of reasons.

1. They are simpler: When we calculate expectation values from a discrete path integral all we ever have to do is matrix multiplication. When we calculate expectation values from a Hamiltonian we have to first exponentiate it. Since rigorous statements about phases of matter are notoriously hard to make, we highly prefer the simpler definition.

2. They are as good as Hamiltonians: Any Hamiltonian can be Trotterized to give a discrete path integral, as we will see in Section 3.2. Even though this process is approximate, it shows that using discrete path integrals as first-principle models is as powerful as using Hamiltonians.

3. They treat space and time on equal footing. In a Hamiltonian description, time is continuous but space is (usually) discrete. In discrete path integrals, all directions in spacetime are discrete, which is more aesthetically pleasing. To be fair, one may conversely argue that the explicit continuous time translation symmetry of

the Hamiltonian description fits the physical reality in many condensed-matter systems.

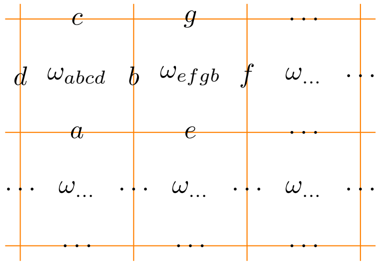
4. They are the predominant language of classical statistical physics: Classical statistical models like the Ising model are discrete path integrals. Even though topological phases seem less relevant in classical physics, our formalism also applies there.

5. They are the predominant language of (topological) quantum error correction and computation: Circuits of unitaries, channels, or measurements are real-time discrete path integrals. The discrete path integrals describing topological phases in this thesis are not in real-time but imaginary-time, but they are closely related to real-time fault tolerant error-correcting circuits.

6. Discrete path integrals are what TQFT is secretly about. As we will show in Chapter 8, (extended axiomatic) TQFT arises from simply assigning tensors to discretized spacetime manifolds.

7. Discrete path integrals might provide an elegant solution to the chiral fixed-point problem, as we describe in Chapter 11.

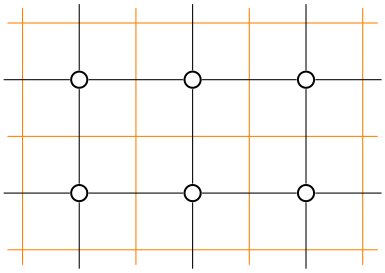
After motivating why we like to use discrete path integrals, it is about time to define what they are. Just like a path integral is an integral over continuous field configurations, a discrete path integral is a sum over discrete configurations of local variables. More precisely, discrete path integrals are defined on a discrete spacetime, such as for example a 2-dimensional square lattice. They assign variables to certain places on this lattice, such as for example one at every edge of the lattice. The variables take values from a discrete set, for example, one out of $\{0,1\}$. They also assign weights to certain places, for example one at every plaquette. These weights are $\mathbb{C}$-valued (or $\mathbb{R}$-valued), and they depend on the configuration of the variables in a constant-size neighborhood. For example, each weight at a plaquette might depend on the configuration of the variables on its four edges. The path integral is then given by taking the product of all the weights, and summing over all configurations of the variables. The following picture shows the example described above,



$$\Rightarrow \quad \begin{aligned} &Z[d,c,g,\dots] \\ &= \sum_{a,b,e,f,\dots\in\{0,1\}} \omega_{abcd}\omega_{efgb}\cdots\ . \end{aligned} \tag{3.1}$$

The weights are called ω, and carry the configuration of the surrounding variables on which they depend as a subscript. That is, ω consists of 16 numbers, one for each of the 16 configurations of the four surrounding variables. Note that for talking about phases of matter, it is important that the models are uniform, which here means translation invariant. That is, a model is determined by a single unit cell, and all the weights ω are the same. Evaluating the discrete path integral yields a number Z, if we compute it for a finite lattice using periodic boundary conditions on a torus. We might also define *physical boundary* conditions where the path integral terminates, with potential new variables and weights near this boundary. Then we can compute the partition function Z on a $n \times m$ lattice with open boundary condition. Note that for this type of boundary we still sum over all variables. Very different in character are *space boundaries*: Introducing a space boundary means evaluating the path integral conditioned on the values of the variables in the vicinity of the boundary. That is, we only sum over the variables in the interior. This summation is shown in Eq. (3.1), where the variables at the boundary edges are fixed, whereas those at interior edges are summed over. The value of Z for a configuration of boundary variables can then be interpreted as the amplitude (in quantum physics) or probability (in classical physics) for this configuration. In other words, the evaluation can be interpreted as a state living on the space boundary.

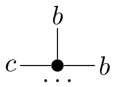
The weights ω of a path integral are just tensors with one index for each variable that the weight depends on. When we look at the summation in Eq. (3.1), we see that each tensor index occurs twice and is summed over. Thus, the evaluation of Z is just an Einstein summation, consisting of contractions between index pairs. Thus, the above summation is the same as the evaluation of a tensor network (c.f. Section 2.1 with one tensor at each plaquette, and adjacent tensors sharing a bond,



. (3.2)

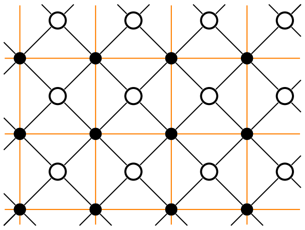
Here, the orange lattice is just in the background, and the black drawing is the tensor-network diagram, which each circle representing a copy of the tensor ω. This gives us a second way of defining discrete path integrals: They are tensor-networks living in spacetime. When we terminate the tensor network by cutting off some of the indices (as shown above), this corresponds to a space boundary. Contracting this tensor-network with open indices yields a tensor that can be interpreted as a state living on this space boundary.

Note that this direct translation from discrete path integrals to tensor networks only works when each variable occurs in exactly two weights. If a variable occurs in x weights, the translation can be fixed easily by including an x-index δ-*tensor* in order to make x copies of the value of the variable. The δ-tensor is a tensor that forces all of its indices to have equal value (assuming some preferred basis). We will denote it with a small dot,



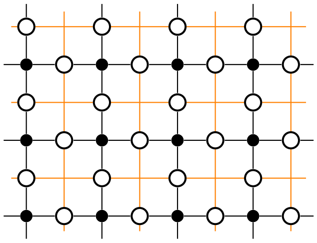
$$= \begin{cases} 1 & \text{if } a = b = c = \dots \\ 0 & \text{otherwise} \end{cases} \quad . \tag{3.3}$$

The tensor-network path integral now consists of one such δ-tensor for every variable and one weight tensor for every weight. For example, consider a discrete path integral with one variable at every vertex of the square lattice, and one weight at every plaquette depending on the four corner variables. The corresponding tensor-network diagram would look like



. (3.4)

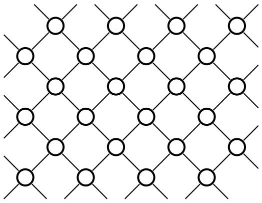
As another example, consider one variable at every plaquette, and one weight at every edge depending on the two adjacent plaquettes. This yields a tensor-network path integral,



. (3.5)

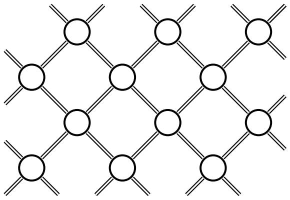
To avoid potential confusion, we want to stress that this is *not* the usual way in which tensor networks are used to describe many-body systems in the condensed-matter literature: There, tensor networks known as *PEPS* (or *MPS* for $d = 1$) [42] are used to parameterize (ground) states of a local Hamiltonian. These tensor networks live in space, and are therefore one dimension lower than our tensor-network path integrals. Also, these tensor networks have open (uncontracted) *physical indices* distributed over space, in addition to *virtual indices* contracted between neighboring tensors. In contrast, our tensor-network path integrals do not have a distinction between virtual and physical indices, and open indices only occur where we cut the network at a space boundary. In fact, as we will see later, MPS or PEPS are formally the same as tensor-network path integrals for physical boundaries.

After introducing discrete, or tensor-network path integrals, we now want to discuss in which way they can describe physical many-body systems. Readers familiar with classical statistical physics might already have recognized that discrete path integrals as we have defined them in the beginning equivalent the computation of the *partition function* Z of a classical statistical model: The lattice is a spatial lattice, the variables are degrees of freedom, and ω is given by *Boltzmann weights*, which for classical statistical physics need to be real and non-negative. For example, the tensor-network diagram shown in Eq. (3.5) could describe a classical *Ising model* on a 2-dimensional square lattice. Next, we also mentioned in the beginning that (geometrically local) quantum circuits are products of linear operators, and therefore tensor-network path integrals. For example, a brick-layer circuit of 2-qudit unitaries acting on a chain of qudits is a tensor-network path integral in the $1+1$-dimensional spacetime,



$$. \qquad (3.6)$$

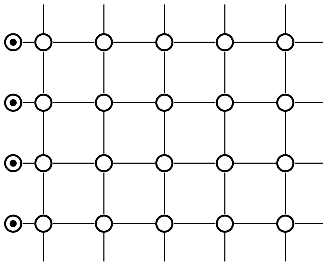
Here, we imagine time going from bottom to top, and the 4-index tensor is a 2-qudit linear operator acting from the bottom to the top indices. The tensor-network diagram is the same as the circuit diagram. The same holds for circuits of quantum channels instead of unitaries,



$$. \qquad (3.7)$$

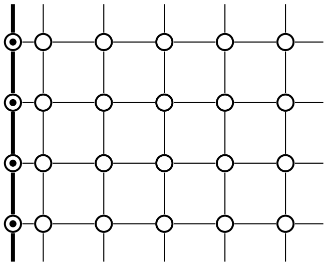
Here, the circles represent 2-qudit channels as linear (super-)operators acting from bottom to top. We are using double lines for the indices, since each index is a composite of a ket, and a bra index. Tensor-network path integrals can also represent an imaginary time evolution under a Hamiltonian, or a finite-temperature Gibbs state. In the former case, the path integral lives in (Euclidean) spacetime, while in the latter case it lives in space only. In these cases, discrete path integrals are not the established language in the literature, but they can be built from a Hamiltonian by Trotterization as we sill see in the next section.

Finally, let us describe how to enrich tensor-network path integrals with lower-dimensional defects. One type of defect (even though not usually called this) is given by physical boundaries, which we have already mentioned above. In tensor-network language, a physical boundary is a way to terminate the tensor-network path integral in such a way that there are no open indices at this boundary. To this end, we need to necessarily introduce additional tensors at the boundary. The simplest way to do this is to close off each open index with a vector. The following shows a patch of 2-dimensional tensor-network path integral with such a physical boundary on the left,



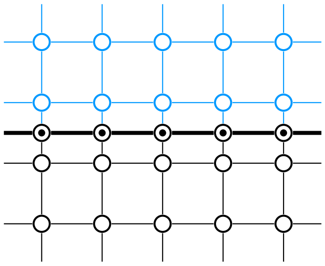
$$. \qquad (3.8)$$

Again, uniformity means that all the boundary vectors, denoted as a circle with a dot in the above, are the same. We could have equally defined a boundary at the top, bottom, or right. We can also introduce additional index pairs contracted between the boundary tensors, such as



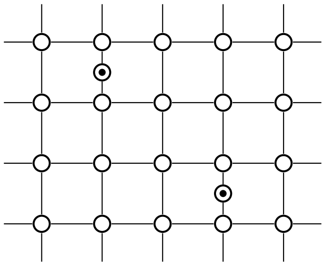
. (3.9)

These indices might have a different bond dimension, so we use a thicker line style. Note that boundaries can have a different physical interpretation depending on the direction in which they extend. For example, if the path integral represents a real-time evolution like a quantum circuit, with time going from bottom to top, then a boundary on the left or right is a physical boundary. A boundary at the bottom on the other hand corresponds to a constant-depth circuit preparing an initial state. Similarly, we can represent domain walls between two different tensor-network path integrals,



. (3.10)

This time, we have depicted a horizontal domain wall, separating one bulk model (in black) at the bottom from another (in blue) at the top. The blue and black tensor are different. We cannot only enrich (2-dimensional) tensor-network path integrals by 1-dimensional boundaries or interfaces, but also by including special points, for example,



. (3.11)

Physically, contracting such a tensor network (with physical open, or periodic boundary condition) with two inserted special tensors corresponds to evaluating a 2-point function.

3.2 Trotterization

As mentioned in the previous section, tensor-network path integrals can approximate the time evolution under a local quantum many-body Hamiltonian. This can be done via *Trotterization*, which is a standard technique in the field of numerical tensor-network methods using PEPS or MPS. Here we do not use Trotterization to calculate the time evolution applied to a state, but to obtain a tensor-network path integral for the time evolution itself. In this section, we will explain Trotterization focusing on the simplest non-trivial example, namely a nearest-neighbor 1-dimensional qudit chain. The generalization of the presented methods to higher dimensions or other local geometries of Hamiltonian terms is straight-forward. The Hamiltonian for the 1-dimensional chain is given by

$$H = \sum_i h_i \; , \tag{3.12}$$

where h is a 2-qudit Hamiltonian, and h_i denotes h applied to the qudits i and $i+1$. The goal is to turn a unit cell of the Hamiltonian model, determined by h, into a unit cell of tensor-network path integral, given by a square-lattice-path-integral tensor.

We can divide the Hamiltonian terms into the ones acting on even-odd site pairs and those acting on odd-even site pairs as

$$H = \sum_i h_{2i} + \sum_i h_{2i+1} = H_1 + H_2. \tag{3.13}$$

All terms within H_1 act non-trivially only on non-overlapping sets of spins and therefore commute. The same holds for H_2. H_1 and H_2 however do not commute, and therefore we have

$$e^{it(H_1+H_2)} \neq e^{itH_1} e^{itH_2} \tag{3.14}$$

in general. We can still use the following *Suzuki-Trotter expansion*,

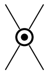
$$e^{it(H_1+H_2+\ldots)} = \lim_{n\to\infty} \left(e^{i\frac{t}{n}H_1} e^{i\frac{t}{n}H_2} \ldots \right)^n \; , \tag{3.15}$$

to obtain

$$e^{itH} = \lim_{n\to\infty} \left(\prod_i e^{i\frac{t}{n}h_{2i}} \prod_i e^{i\frac{t}{n}h_{2i+1}} \ldots \right)^n \; . \tag{3.16}$$

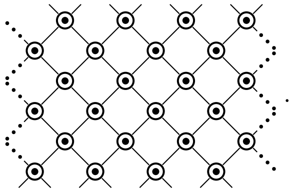
Consider the expression on the right-hand side for a fixed n. It is a product of local operators acting on a the qudit chain, or in other words, a brick-layer circuit. $e^{ith/n}$ is a linear operator acting on two spins, so it is a tensor with 4 indices, two for both input (bottom) and output (top) of the operator,

$$e^{i\frac{t}{n}h} \quad \rightarrow$$



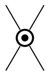
. (3.17)

With this interpretation, the product of operators becomes a tensor network. For example, for $n = 3$, we get the network

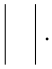


. (3.18)

The tensor network we are looking for should have the same notion of locality structure as the continuum time evolution. That is, every tensor should correspond to a finite space-time volume $\Delta x \times \Delta t$. So it does not make sense to directly take the above tensor network, as the time interval corresponding to a tensor scales like $1/n$. In addition, this tensor network has a trivial limit for $n \to \infty$

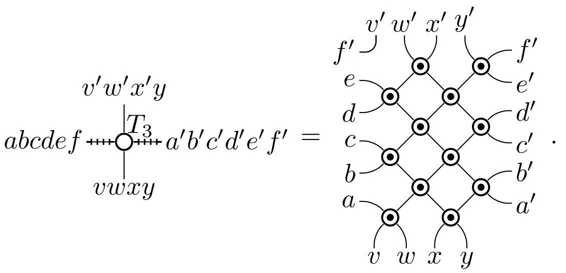


$$= e^{i\frac{t}{n}h} \xrightarrow{n\to\infty} \mathbb{1} =$$



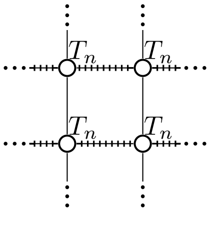
. (3.19)

Instead, we have to pick a fixed Δt, Trotterize $e^{i\Delta t H}$ for some n and divide the resulting tensor networks into spatial unit cells. Evaluating the whole tensor-network patch inside the unit cell yields a tensor T_n forming a square-lattice tensor-network path integral that approximates the time evolution. For example, for $n = 3$ and Δx consisting of 4 sites, we can choose



. (3.20)

For larger and larger n, the square lattice tensor network



(3.21)

then better and better approximates the time evolution of the local Hamiltonian. However, T_n does not have a sensible large-n limit either: If we let $n \to \infty$, the number of indices we need to block on the right and left grows linearly with n. Therefore, the horizontal indices of T_n, drawn as ticked lines, have a bond dimension that grows exponentially with n.

Luckily, experience from numerical algorithms performing time evolution with MPS/PEPS suggests that this exponentially growing bond dimension can be truncated to a much smaller bond dimension with only a very small approximation error. Specifically, in numerical algorithms like iTEBD, the Trotterized time evolution is applied to an MPS which are then truncated. There one finds that the state after a *fixed* time Δt is well approximated by an MPS with bond dimension essentially independent of the system size, although the necessary bond dimension might grow with Δt. The more crucial observation is that the truncated bond dimension is essentially independent on the number n of Trotter steps, even though the bond dimension would grow exponentially in n without truncation. This suggests that also the MPO given by the Trotterized Δt time evolution itself can be truncated from something growing exponentially in n to something essentially independent of n. In particular, if we apply the time evolution to each first qudit of a product of bell pairs, then the result is the MPO describing the time evolution itself.

To avoid potential confusion, we would like to stress that this consideration is completely different from the question whether ground states are well approximated by MPS, which has been famously conjectured for *gapped* Hamiltonians, and proven in some formulation in $1+1$ dimensions. Ground states are obtained by applying the imaginary time evolution for a time t and then taking the limit $t \to \infty$. In contrast, we only consider a fixed time interval Δt which we do not ever think of scaling. For such a constant interval, we believe that the time evolution can be truncated independent from whether the Hamiltonian has a spectral gap and whether we consider the imaginary or real time evolution.

In order to make Trotterization into a precise conjecture, we need to take the continuum limit $n \to \infty$. The tensors T_n for different n have different bond dimensions for the indices on the left and right, and should be considered as vectors in different vector spaces. In order to take the limit, we need to embed all of those vector spaces into one common infinite-dimensional vector space. To define convergence, this vector space needs to be equipped with a norm. In order to make sense of tensor networks formed from those infinite-dimensional tensors, we the norm needs to be defined for tensors with multiple infinite-bond-dimension indices, and the contraction and tensor product should be both well-defined and continuous as a (bi-)linear functions. In other words, the infinite-dimensional tensors need to form a tensor type [22]. Possible norms that define a tensor type are, for example, the entry-wise 1-norm,

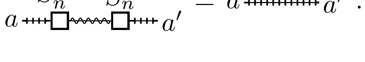
$$\|T\| = \sum_{a,b,c,\dots} |T_{abc\dots}| \,, \tag{3.22}$$

or a norm enforcing a $\frac{1}{n}$ decay in every individual index,

$$\|T\| = \max_{a,b,c,\dots} \quad a \cdot b \cdot c \cdots |T_{abc\dots}| \,. \tag{3.23}$$

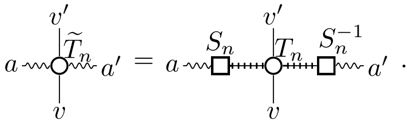
Valid tensors are those infinite-dimensional arrays for which the norm is finite.

In order to perform the embedding of the different T_n into one shared infinite-dimensional normed vector space, we could use a left-invertible matrix S_n,



(3.24)

Here, the shared infinite-dimensional vector space is drawn as a zigzag line. By conjugating with S_n at all horizontal indices, we obtain a new tensor $\widetilde{T}_n$ whose horizontal indices correspond to the shared infinite-dimensional vector space,



(3.25)

The indices of $\widetilde{T}_n$ now live in the same vector space for all n. With this, our Trotterization conjecture can be formalized as follows. There exist invertible matrices S_n such that the according series of tensors $\widetilde{T}_n$ converges to some tensor $\widetilde{T}$,

$$\widetilde{T}_n \xrightarrow{n\rightarrow\infty} \widetilde{T} \; . \tag{3.26}$$

The square-lattice infinite-bond-dimension tensor network $\widetilde{T}$ then *exactly* represents the time evolution, in the sense that evaluating one row of it yields the exact time evolution operator. A finite bond dimension tensor network approximating the time evolution can be obtained by simply truncating $\widetilde{T}$ after a certain bond dimension.

Note that the above conjecture is likely not true since the relation between T_n and $\widetilde{T}_n$ is too restrictive. We believe that this can be fixed by generalizing the matrix S_n to an *invertible domain walls* between the T_n and $\widetilde{T}_n$ path integrals, as we introduce in Section 3.7. Roughly, such an invertible domain wall is a way to transform the T_n path integral into the $\widetilde{T}_n$ path integral by locally applying tensor-network equations. In the case of an isometry, these equations are given by Eqs. (3.24) and (3.25), but there are more general invertible domain walls as discussed in Section 3.7.

We have described Trotterization for the real time evolution of a Hamiltonian. The procedure works in the exact same way for the imaginary time evolution, by just replacing e^{itH} with $e^{-\beta H}$. Note that even though real and imaginary-time evolution have very different truncation behavior when scaling $\Delta t \rightarrow \infty$, there is not reason why real versus imaginary time should affect the behavior for a fixed Δt and $n \rightarrow \infty$. Note that the topological path integrals that we will study in this thesis live in Euclidean spacetime, and thus represent an *imaginary* time evolution.

Finally, it is easy to see how to generalize Trotterization to other geometric setups, such as higher dimensions, higher spatial support of the Hamiltonian terms, or presence of boundaries or defects of any kind. First we divide the Hamiltonian terms into a constant (system-size independent) number of subsets, such that the terms within one subset all commute with each other. Then we proceed using the Suzuki-Trotter expansion applied to the division into subsets, resulting in a tensor network, which we block and truncate into finite unit cells.

3.3 Gapped-path and local-unitary phases of matter for Hamiltonians

In this section, we briefly review conventional notions of phases for local quantum many-body Hamiltonians. *Quantum phases of matter* are defined as equivalence classes of local translation-invariant gapped Hamiltonians $H \in \mathcal{H}$. By *gapped*, we mean that there is an integer $g \geq 0$ called *ground state degeneracy* and a real number $\epsilon > 0$ called the *gap*, such that for every system size n (greater than some n_0), the g lowest eigenvalues of H are separated from the rest of the spectrum by at least ϵ, and among each other by β_n such that $\beta_n \rightarrow 0$ for $n \rightarrow \infty$. Two gapped Hamiltonians $H_1, H_2 \in \mathcal{H}$ are said to be *in the same phase* if there is a continuous path of gapped Hamiltonians connecting H_1 and H_2 [143],

$$\begin{gathered} \widetilde{H} : [0,1] \rightarrow \mathcal{H} \;, \\ \widetilde{H}(0) = H_1, \quad \widetilde{H}(1) = H_2 \;. \end{gathered} \tag{3.27}$$

Recall that $\mathcal{H}$ contains only gapped Hamiltonians, so all $\widetilde{H}(s)$ for $s \in [0,1]$ must be gapped, otherwise one speaks of a “gap closing” inducing a “phase transition”. If we aim at comparing two Hamiltonians with different local Hilbert spaces, we can arbitrarily embed both into a shared local Hilbert space and use the same definition. A *phase of matter* is an equivalence class of gapped Hamiltonians under being in the same phase.

The importance of models with a spectral gap comes from the fact that they are “generic” in the following sense. Under certain additional *local topological order* conditions, the property of having a gap is robust to perturbations. That is, for any perturbation direction there is a non-zero perturbation strength such that the perturbed model remains gapped for all perturbations of smaller strength. This has been proven so far only for perturbations around fixed-point models of topological order [31]. Consequently, the set of gapped models is an open subset of the space of all models, and in any few-parameter family of models (i.e., in any phase diagram), there are either no gapped models at all or they have a non-zero volume.

For generic local many-body Hamiltonians it is very hard to tell whether they possess a spectral gap, and the general question of the existence of a gap of a certain size has been shown to be algorithmically undecidable for certain families of Hamiltonians [47]. However, there is a very simple family of Hamiltonians for which it is very simple to verify the spectral gap, namely *commuting-projector Hamiltonians*, where the Hamiltonian is -1 times the a sum of geometrically local projectors that mutually commute. Such Hamiltonians can be solved exactly analytically and have zero *correlation length*. They are also fixed points under a blocking/renormalization-group operation, and are therefore often referred to as *fixed-point models*. Many topological phases (presumably all with gapped boundary) can be represented by fixed-point models.

The gapped-path definition is rather unconstructive which makes it very hard to classify phases of matter. To naively show that two Hamiltonians are in a different phase, we would have to look at all different continuous paths connecting them inside the infinite-dimensional space of local Hamiltonians, and assert whether an algorithmically undecidable property holds for the models on each path. There is an alternative definition which is much more constructive: Two Hamiltonians are in the same phase if their ground states are related via a finite-depth *local (generalized) unitary circuit* [40]. Note that this definition is usually used to define phases of states on their own, but one needs to be careful to define what a “state” even means in the thermodynamic limit. This definition replaces the continuous path over tensors by a single set of tensors (forming the circuit). Furthermore, it makes the physical meaning of phases very clear: Two Hamiltonians are in the same phase if their ground states are equivalent under locally restructuring their degrees of freedom.

It is important to point out that the two presented definitions are *not* equivalent. In one direction, it is easy to see that a local unitary circuit mapping ground states gives rise to a continuous gapped path of Hamiltonians: First we conjugate H with e^{itV} from $t = 0$ to $t = 1$ where $e^{iV} = U$ for each layer U of the local unitary circuit. Then, we linearly interpolate between two Hamiltonians with the same ground state. However, conversely, a local unitary circuit cannot change the *correlation length* of a model, whereas a gapped continuous deformation can. So local unitaries are strictly less powerful than gapped paths. One might try to fix this by going from exact to approximate local unitary equivalence, with an approximation error decreasing with the depth of the circuit or the support size of the individual local unitaries. Alternatively, we can directly use a "fuzzy circuit", corresponding to the time evolution under a local time-dependent Hamiltonian [88]. In this case, the equivalence has been shown using the notion of quasi-adiabatic evolution [78].

Even though the gapped-path definition is more general than the exact local-unitary definition, the latter seems to be working if we restrict ourselves to fixed-point models such as commuting-projector models. There, two models in the same gapped-path phase are usually observed to be equivalent via a local unitary circuit, and giving the latter is the most common way of proving phase equivalence of models.

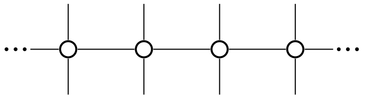
3.4 Spectral gap and phases in tensor-network path integrals

In this section, we will find natural definitions of phases of matter in terms of tensor-network path integrals, by transferring the definitions from the previous section to this setting. Before we start, let us argue why tensor-network path integrals are a particularly natural representation of quantum systems for the study of quantum phases of matter. The latter describe the ground state properties of Hamiltonians. Such ground states can be obtained directly from the Hamiltonian by applying the imaginary time evolution to some initial state vector $|x\rangle$ as

$$\lim_{\beta\rightarrow\infty} e^{-\beta H} |x\rangle \,. \tag{3.28}$$

However, for any finite system size, the lowest eigenvalue will not generically have a g-fold degeneracy, but the "ground states" will have slightly different energies. So, at a particular system size, $e^{-\beta H}$ will not converge to a ground state projector with g-dimensional support, but to the projector on the lowest eigenstate only. We see that in order to talk about ground states, we do not only have to scale the imaginary time β, but also simultaneously the system size n. This simultaneous scaling of both imaginary time and space is elegantly featured by the imaginary-time evolution tensor-network path integral.

The gapped-path definition of phases carries over to the tensor-network path integrals as follows. A gapped Hamiltonian H yields a gapped imaginary-time evolution operator $e^{-\beta H}$, just that now the gap separates the *largest*-magnitude values from the rest of the spectrum. In n spacetime dimensions, consider an $n-1$-dimensional constant-time layer of a tensor-network path integral, such as

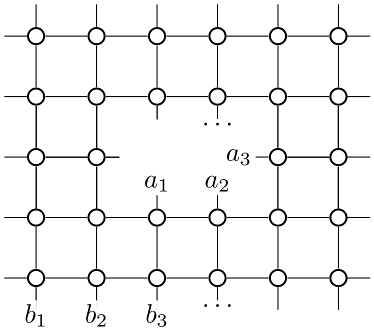


(3.29)

for a square lattice tensor network in $1+1$ dimensions. Suppose the tensor-network path integral comes from Trotterizing an imaginary-time evolution. Then this layer, as an operator from the bottom indices to the top indices, with N tensors and with periodic boundary conditions connecting left to right, approximates $e^{-\beta H}$ where β is the chosen discretization step in time direction. Tensor networks of this form are known as *projected entangled pair operators (PEPO)*, or *matrix product operators (MPO)* [42] in $n-1=1$ dimension, and are often called *transfer operator*.

From the tensor-network path integral point of view, this notion of gap seems a bit unnatural as it specifically involves the transfer operator in imaginary time direction. The path integral lives in Euclidean spacetime, so space and imaginary time should be treated on equal footing. Instead of demanding a spectral gap of the imaginary-time transfer operator, it is natural to demand a spectral gap for the tensor network along any n-dimensional curve in spacetime, interpreted as an operator from one to the other side. Unfortunately, this idea suffers from the technical problem that the indices (and thus the Hilbert spaces) on the two sides of an arbitrarily curved transfer operator can be different, and hence we cannot talk about eigenvalues. In the following, we give a definition which we believe is the natural tensor-network path integral analogue to the spectral gap of Hamiltonians. It is based on the observation that for a gapped Hamiltonian for intrinsic robust topological order on a sphere, the normalised operator $e^{-\beta H}$ gets exponentially close to becoming a rank-1 operator in the "width" β of the transfer operator.

For a more formal definition in n spacetime dimensions, we consider an *annulus network* A, that is, a patch of a translation-invariant tensor network whose topology is $S_{n-1}\times[0,1]$. We then look at its behavior when its *width* d_A, given by the minimum number of bonds it takes to get from some open index at the inside- to some open index at the outside boundary, is increased. Let I_A and E_A denote the *inside boundary* and *outside boundary* of an annulus network A and let $X[A]$ denote a tensor-network path integral X evaluated on A. We can interpret $X[A]$ as a linear operator from the open indices on the inside boundary to those of the outside boundary. As an example, consider the following network annulus and the according operator in $1+1$ dimensions,



$$= X[A]_{\vec{a},\vec{b}} \, . \tag{3.30}$$

Equipped with this terminology, we say that a tensor-network path integral X has a *robust gap* with *correlation length* bounded by some $\xi > 0$, if the following holds.

- For any interior boundary I_A, there exists a prefactor C_{I_A} and two tensors $\langle V_{I_A}|$ and $|W_{I_A}\rangle$ whose open indices match those of I_A.
- For every annulus network A, and every tensor $\langle T|$ with indices matching those of E_A, we have

$$\| \langle T| \, X[A] - \langle T| \, X[A] \, |W_{I_A}\rangle \, \langle V_{I_A}| \| < \| \langle T| \, X[A] \| C_{I_A} e^{-\frac{d_A}{\xi}} \, . \tag{3.31}$$

Note that the choice of norm in Eq. (3.31) does not matter as the corresponding vector space has a finite dimension only depending on I_A, so we can make up for possible changes of the norm by adapting C_{I_A}. *The* correlation length of X is the smallest ξ for which X has a robust gap. For tensor-network path integrals that hypothetically exactly represent an imaginary time evolution via Trotterization as discussed in Section 3.2, a robust gap of the path integral implies a gap of the Hamiltonian. It is unclear whether the opposite implication would be true.

As we announced, the definition says that $X[A]$ is approximately rank-1. However, we have formulated it in such a way that it also applies to symmetry-breaking phases: In this case, the tensor $\langle T|$ is restricted to be a symmetric tensor, so the condition above only states that $X[A]$ is (approximately) rank-1 within the symmetric subspace.

Equipped with an appropriate notion of a robust gap, the definition of phases on the level of tensor-network path integrals is analogue to the Hamiltonian case: Two tensor-network path integrals X_0 and X_1 with a robust gap are *in the same phase* if there is a continuous family of tensors $X(s)$, $s \in [0,1]$, $X(0) = X_0$, $X(1) = X_1$, such that

- $X(s)$ has a robust gap for all s, and
- there is a choice of $s \mapsto \langle V_{I_A}(s)|$ that is continuous.

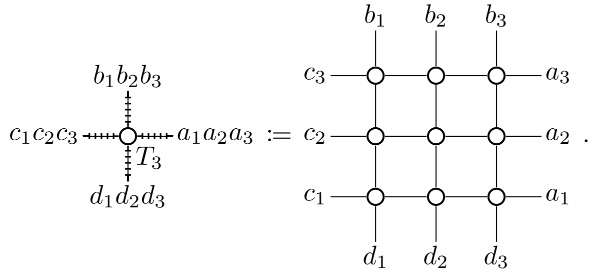
Note that the second condition is necessary to detect first order phase transitions, by which we mean transitions between two symmetry-broken sectors via a non-trivial symmetry-breaking phase.

Finally, for a classification of topological phases, we assume that the correlation length is zero. This is a very restrictive condition, and the tensors of such models are often found to satisfy exact algebraic relations, which enable us to analytically calculate some of their properties.

Definition 24. *By a* fixed-point model*, we mean a tensor-network path integral such that the correlation length ξ is zero. That is, there is some d_{min}, such that the left-hand side of Eq.* (3.31) *is exactly zero if $d_A \geq d_{\mathrm{min}}$. In this case, $X[A]$ is a rank-1 operator. For symmetry-breaking phases, $X[A]$ is still rank-1 if we restrict ourselves to the symmetric subspace.*

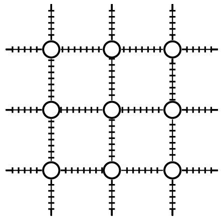
3.5 Fine-graining/renormalization of tensor-network path integrals

In the previous section, we have defined fixed-point path integrals. A priori, it is unclear whether every phase can be represented such a fixed-point path integral. An argument of why this is the case is provided by the vague idea of *renormalization group (RG) flow*. Roughly speaking, this is a procedure that coarse-grains a model, and therefore reduces the correlation length, until we hopefully converge to a model with zero correlation length. This is also the origin behind the name "fixed-point model", since applying further coarse graining leaves a fixed-point model invariant. In this section, we will give a precise definition of RG flow for tensor-network path integrals. Since RG flow is a vague idea that comes in many technically very distinct formulations, we will just refer to our version as *fine-graining mapping*. Given a tensor-network path integral T, we can construct a new path integral T_λ by taking blocks whose linear size λ is called the *fine-graining scale*, and grouping them together into a single tensor each. For example, for $\lambda = 3$ and a square-lattice tensor network in $1 + 1$ dimensions with correlation length ξ, we can define a new tensor by



(3.32)

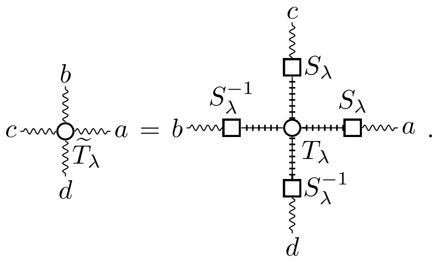
The tensor-network path integral formed by the new tensor then has a correlation length $\xi/3$, when measured in the new lattice distance,


. (3.33)

This comes at the expense of the bond dimension of the new tensor network increasing exponentially in λ.

Two points in the new tensor-network path integral with a combinatorial distance d have a distance λd in the original network. Thus, if the correlation length of the old tensor-network path integral was ξ, then the new correlation length is ξ/λ. The idea is that by choosing larger and larger λ, we will eventually arrive at a fixed-point model with $\xi = 0$. Often, renormalization is thought of as an iterative procedure where in each step we block by a factor of 2 (or another small number), corresponding to only considering the subsequence T_{2^x} instead of all T_λ. A "fixed-point model" is then literally a fixed point of this iteration.

There are, however, two major difficulties with this idea. First, to arrive at a fixed-point model, we would need the sequence T_λ to converge. As such, this does not make any sense, as the different T_λ have different bond dimensions corresponding to different vector spaces. To make the definition work, we use the same formulation as for the Trotterization limit in Section 3.2. That is, we choose a norm for the space of infinite-bond-dimension tensors which is compatible with tensor product and contraction, such that the normalizable tensors form a tensor type. Then, for every λ, we conjugate T_λ on both the horizontal and vertical bonds with an invertible matrix S_λ. This yields a tensor $\widetilde{T}_\lambda$ all of whose indices correspond to the shared infinite-dimensional vector space (denoted as a zigzag line),



. (3.34)

Convergence can then be defined as follows: There exists a sequence of S_λ such that the according sequence of $\widetilde{T}_\lambda$ converges,

$$\lim_{\lambda\to\infty} \widetilde{T}_\lambda = \widetilde{T} \ . \tag{3.35}$$

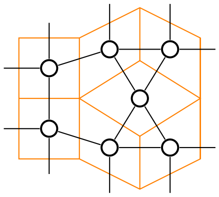
The limit $\widetilde{T}$ is then an exact renormalization fixed-point. Note that even though the sequence is defined in an infinite-dimensional normed vector space of tensors, it is possible that the limit $\widetilde{T}$ has only a finite number of non-zero entries. In this case, the resulting fixed-point model can be truncated without approximation to a finite bond dimension.

It is unlikely that generic models have converging fine-graining mappings in the above sense. This can potentially be fixed by allowing for a more general relation between T_λ and $\widetilde{T}_\lambda$. In general, we may allow T_λ and $\widetilde{T}_\lambda$ to be related via an arbitrary invertible domain wall, as discussed in Section 3.7. Note that on-site S_λ transformations resemble a tensor-network renormalization scheme called *tree tensor networks*. In contrast, general invertible domain walls would allow things like applying a collective transformation to two neighboring indices, which resemble the nearest-neighbour "disentanglers" in a the *multi-scale entanglement renormalization ansatz (MERA)*. All in all, however, it is an open question if and under what circumstances there exist invertible domain walls such that the sequence $\widetilde{T}_\lambda$ converges.

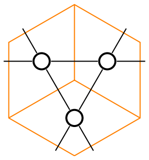
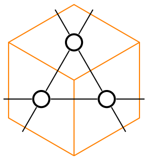
3.6 Topological invariance and tTS models

In this section, we will motivate the single central property that we use to classify gapped phases of matter throughout all of this thesis. This property is *topological invariance*. While for generic models, topological invariance is an approximate thermodynamic-limit property, for fixed-point models, it turns into *exact combinatorial topological invariance*. Let us note here already that strictly speaking not all gapped phases of matter have this property, which we discuss in more detail at the end of this section.

Let us start by defining what we mean by topological invariance. First, imagine a model defined in some continuous spacetime, like a quantum field theory. In this case, topological invariance would mean that the model can be defined on arbitrary topological manifolds and is independent of a metric, and thus invariant under arbitrary homeomorphisms. However, we define models in terms of discrete path integrals, which live in a discrete spacetime. The natural discrete-spacetime analog of defining a model on arbitrary manifolds is to put the discrete path integral on arbitrary irregular lattices instead of regular lattices like the square or cubic lattice. For example, we could take the square-lattice tensor network in Eq. (3.2) and also allow for vertices where 3 or 5 instead of 4 plaquettes meet, like

 . (3.36)

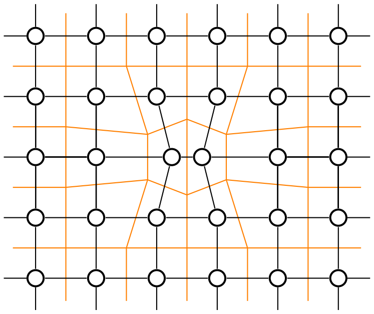
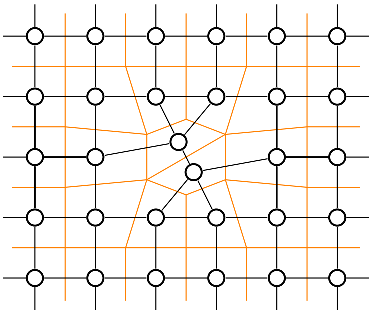
Now, the extended model has topological invariance if it is locally invariant under deformations of the lattice that do not change the topology. For a fixed-point model, this topological invariance is exact, and can be elegantly defined using tensor-networks: If we deform the lattice locally, also the associated tensor-network diagram changes locally. The local change consists of cutting out a local patch of the tensor network, and replacing it by another local patch. Topological invariance is then imposed by demanding that the cut-out and substituted tensor networks evaluate to the same tensor. For example, consider the following recellulation and the according equation of tensor networks,

 $= \alpha \cdot$  . (3.37)

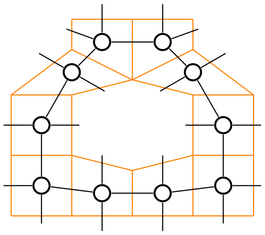
In orange we depicted two different cellulations consisting of four plaquettes each, with the same boundary. When replacing the left-hand cellulation with the right-hand cellulation in some spacetime lattice, the tensor-network patch drawn in black on the left-hand side is replaced by another patch on the right-hand side. For topological invariance, we demand that the two tensor-netowrk patches evaluate to the same tensor. In fact, since our tensor-network path integrals have a quantum mechanical or classical statistical interpretation, it is fine if this equation holds up to a global prefactor α. In many cases, however, the prefactor α can be normalized to 1. We demand that an analogous equation also holds for all other kinds of local lattice deformations. Topological invariance then corresponds to a set of tensor-network equations that need to hold for the 4-index tensor defining the model. These equations form a tensor scheme (TS), as defined in Section 2.3. Since models of this TS are topological fixed-point models, we refer to it as a *topological tensor scheme (tTS)*.

Note that tensor-network path integrals coming from condensed-matter models or similar are only defined on regular lattices, and it does not necessarily make sense to put these path integrals onto arbitrary irregular lattice. Indeed, we do not claim that a generic square-lattice fixed-point model will fulfill topological invariance as defined above. The statement is rather that for every phase, there exist fixed-point models that can be put on arbitrary irregular lattices and that obey topological invariance.

Let us now give some motivation for why it is natural to expect topological invariance, and how this is related to the notion of a robust gap. The basic idea is simple: For a fixed-point model, the operator corresponding to an annulus is rank-1. So no matter how we fill the interior with tensor network, we will obtain the same tensor on the exterior boundary, up to a global prefactor. For example, for a fixed-point model with $d_{\min} = 2$, we have:

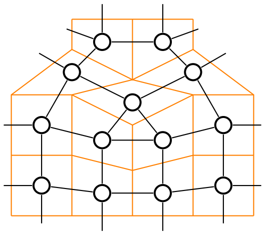
 $= \alpha \cdot$  . (3.38)

Thus, the underlying spacetime lattice for a topological fixed-point model can be arbitrarily deformed using tensor-network equations. There is, however, one little caveat in this argument. Namely, the rank-1 condition defining a fixed-point model in Section 3.4 is only required to hold for annuli that form a subset of a regular square lattice. Thus, we need to make the non-trivial additional assumption that the rank-1 condition also holds for arbitrary irregular-lattice annuli of width larger than d_{min}. For example, for a fixed-point model with $d_{\text{min}} = 1$, we demand the following annulus operator to be rank-1,

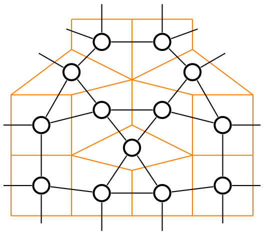


$$. \tag{3.39}$$

Thus, the model is invariant under, for example, the following topological lattice deformation,



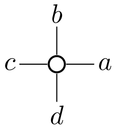
$$= \alpha \cdot$$



$$. \tag{3.40}$$

So we see that if the rank-1 condition holds for arbitrary irregular annuli, then this implies that the lattice can be arbitrarily deformed by locally applying tensor-network equations, and so the fixed-point model is topologically invariant.

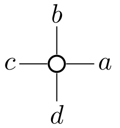
Let us now give a simple example for such a topological fixed-point model, namely the path integral describing the (classical) Ising model at zero temperature and magnetic field. On a 2-dimensional square lattice, it is given by a tensor with bond dimension 2,



$$= \begin{cases} 1 & \text{if} \quad a = b = c = d \\ 0 & \text{otherwise} \end{cases} . \tag{3.41}$$

Since this tensor forces all of its indices to have the same value, the evaluation of any connected tensor-network diagram yields an equal-weight superposition of the all-0 and the all-1 configuration. Any annulus operator has thus rank 2, and its image and coimage is the 2-dimensional space spanned by the all-0 and the all-1 configuration. As such, the path integral does not have a robust gap. However, if we impose the $\mathbb{Z}_2$ symmetry acting by swapping the 0 and 1 configuration, then the annulus operator is rank-1 inside the symmetric subspace. The corresponding phase is not commonly called "topological", but rather known as a *symmetry-breaking* phase. However, it still has the topological invariance property. For proper topological phases that possess a robust gap also without imposing a symmetry, we would need to go to $2 + 1$ spacetime dimensions.

Last, let us discuss for which kinds of gapped phases of matter there exists a fixed-point model with topological invariance. As we mentioned in the beginning, this is not always the case. A simple paradigmatic example for this is the 2-dimensional (classical) anti-ferromagnetic Ising model at zero temperature, which is the following variant of the Ising model given above,



$$= \begin{cases} 1 & \text{if} \quad a = b \neq c = d \\ 0 & \text{otherwise} \end{cases} . \tag{3.42}$$

This is a fixed-point model with a robust gap if we impose the $\mathbb{Z}_2$ symmetry. However, the annulus operator it is not rank-1 inside the symmetric subspace of any irregular-lattice annulus. Since the configuration is forced to swap between 0 to 1 when going from the indices a or b to c or d, a tensor network evaluates to 0 if it contains a cycle where the configuration changes an odd number of times. This happens, for example, if we put the path integral on a ring as depicted in Eq. (3.29), with periodic boundary conditions of odd length. The annulus operator on such configuration is rank-0 instead of rank-1.

An example for a proper topological phase without topological invariance is given by the phenomenon of *weak breaking of translation symmetry* in the abelian topological phase of the honeycomb model [96]. Since this property singles out one specific spatial direction, the underlying models are strictly speaking not topologically invariant. The most interesting class of examples, however, is given by so-called *fracton phases* in $3 + 1$ dimensions [76]. These are gapped phases of matter that host excitations which cannot move freely but are confined to certain rigid geometric submanifolds or subspaces, which indicates that there is no notion of topological invariance.

One attempt to resolve the discrepancy between topological invariance and a robust (fixed-point) gap is to slightly loosen the notion of a phase by also allowing for changes of the unit cell. Note that this is common in the physics literature anyways. Equivalently, one could allow explicitly breaking the translation symmetry of the models when deforming them along a gapped path. Another point of view is to compare two models after a fine-graining step at some scale λ as described in Section 3.5. For example, if we fine-grain the model in Eq. (3.42) with $\lambda = 2$, we get a bond-dimension 4 tensor, which can be mapped back to bond dimension 2 via an on-site isometry. The result is the model in Eq. (3.41). So, up to changes in the definition of the unit cell, the Ising ferromagnet and Ising anti-ferromagnet are in the same phase. Also the weak breaking of translation symmetry in the honeycomb model disappears if we choose the unit cell by a factor $\lambda = 2$ larger than necessary, and we get an ordinary toric-code phase with topological invariance. However, this is (by definition, in fact) *not* the case for fracton phases, which is the reasons why fracton phases are so interesting in the first place: No matter how we fine-grain a fracton path integral, we will never get a model with topological invariance. The fine-grained model $\widetilde{T}_\lambda$ can even consist of a tensor product of many copies of the same model, whose number scales with λ, which is known as *entanglement bifurcation* [73].

There is one other, more promising attempt to reconcile gapped phases and topological invariance. This is to represent general gapped phases as topologically invariant phases decorated by topologically invariant defects placed on a rigid superlattices foliating the spacetime. That is, we consider a n-dimensional hypercubic superlattice, fill the n-cubes with a topologically invariant model, the $n-1$-cubes with some topological domain wall, the $n-2$-cubes with codimension-2 defects interfacing the four adjacent domain walls, and so on. The unit cell of the model is given by one n-cube of the superlattice. We will refer to this construction as a *defect network*. The anti-ferromagnet in Eq. (3.42) is such a defect network: It is given by taking the ferromagnet in Eq. (3.41), and inserting spin-flip domain walls (in other words, symmetry defects of the $\mathbb{Z}_2$ symmetry), along all horizontal and vertical lines in the lattice. The weak breaking of translation symmetry in the honeycomb model corresponds to a defect network based on the $2+1$-dimensional toric code path integral, where we insert duality defect domain walls on all xt planes of a cubic superlattice. More interestingly, Ref. [3] constructs defect networks for many examples of fracton phases in $3+1$ dimensions, and conjectures that any gapped phase of matter can be represented by a defect network.

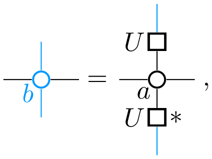
Note that tensor-network path integrals with a robust gap are analogous to gapped Hamiltonians with further conditions that are referred to as *topological quantum order conditions* [31]. Accordingly, it is common in the physics literature to call all robust gapped phases of matter "topological", even if they do not obey the central property of topological invariance described in this section. Clearly, the name "topological" may be justified from different perspectives, for example from the fact that phases are path-connected regions in the "topological" space of models, or from the use of vector bundles describing the band structure of non-interacting free-fermion Hamiltonians in momentum space. However, we feel that these other notions of topology are not nearly as significant as the property of topological invariance, and from this point view calling general gapped phases "topological" would be a misnomer.

To summarize, in this section, we have motivated why fixed-point models of gapped phases of matter obey a property of topological invariance. We have conjectured that fixed-point models without topological invariance can still be described by ones that do via defect networks. In this thesis, we will classify and describe phases of matter via topologically invariant fixed-point models, namely so-called tTS models.

3.7 Invertible domain walls and exact phases

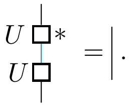
In Section 3.3 we have discussed how local unitary circuits provide a natural definition of phases alternative to gapped paths. However, without any approximations, this definition makes sense only if we restrict ourselves to fixed-point models. In this section, we will give an analogue of the local unitary definition for tensor-network path integrals. We will do so by defining a notion of an *invertible domain wall* which generalizes local unitary circuits, and we will call equivalence classes under such invertible domain walls *exact phases*.

To start with, let us consider a particularly simple local unitary circuit between two ground states of Hamiltonians, namely applying an on-site unitary $U^{\otimes N}$. If we conjugate Hamiltonian terms by $U^{\otimes N}$, the tensors representing the (Trotterized) imaginary-time evolution $e^{-\beta H}$ get conjugated in the same way. So we see that the analogue of applying U for a tensor-network path integral is to conjugate the two "time-like" indices of a tensor a by U, obtaining another tensor b. For example, for a 2-dimensional square-lattice path integral, this corresponds to,



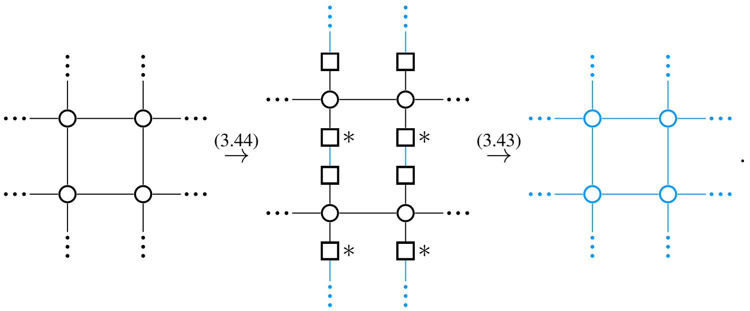
(3.43)

where the $*$ denotes complex conjugation. The two path integrals formed by the tensors a and b are then in the same phase. The fact that U is unitary can be denoted in tensor-network notation by



(3.44)

One can also argue from a purely path-integral point of view that the two path integrals a and b should be considered "equivalent". To this end, we start with the a path integral, and apply Eq. (3.44) backwards at every vertical bond, inserting a resolution of the identity $U^\dagger U = \mathbb{1}$. After this, every a tensor is adjacent to one copy of $U^\dagger$ below and one copy of U above. Applying Eq. (3.43) backwards replaces the a tensor with the two adjacent U tensors by a b tensor. All in all, we have rewritten the a path integral as the b path integral by only locally applying tensor-network equations,



(3.45)

So we see that a and b are really the same path integral, just locally rewritten in a different form. More concretely, imagine performing measurement in a, by contracting the path integral with some extra "measurement" tensors inserted like in Eq. (3.11). Then, there is an equivalent measurement in b that yields the same result: To this end, we just conjugate the measurement tensor by U as well. Furthermore, when enriching a with boundaries or other defects, equivalent defects can be defined for b by just conjugating them by U. In this sense, a and b are equivalent for all practical purposes.

One can now think of more general ways or rewriting path integrals using tensor-network equations, which motivates the following definition.

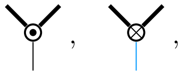
Definition 25. *An* invertible domain wall *between two tensor-network path integrals a and b is defined by the following.*

- *There is a set of* domain wall tensors.

- *The tensors of a and b together with the domain wall tensors satisfy a set of tensor-network equations. In other words, invertible domain walls are TS models (c.f. Section 2.3).*
- *Using the tensor-network equations we can transform the networks of a into the networks of b and vice versa. As usual, we do not think of this as a global transformation at a certain system size, but as a prescription per unit cell.*

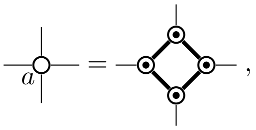
If such an invertible domain wall exists, then a and b are said to be in the same exact phase.

Let us give some examples for invertible domain walls beyond applying an on-site unitary. First, we notice that inserting $U^\dagger U$ where U is a constant-depth local unitary circuit still defines an invertible domain wall. The tensor-network equations correspond to 1) unitarity of each local unitary and 2) conjugation of the a tensor with a lightcone of the circuit yielding the b tensor. We could even go beyond unitaries, and allow for arbitrary left-invertible matrices. Insertions of unitaries or invertible matrices single out the time direction, and are thus not the most natural invertible-domain-wall TS. Consider the following example for an invertible-domain-wall TS for 2-dimensional square-lattice tensor-network path integrals that treats space and time on equal footing. It is given by two additional tensor variables,

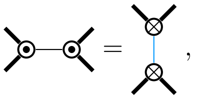


(3.46)

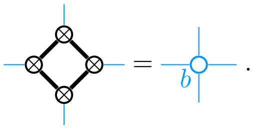
that satisfy the following three moves,



(3.47)

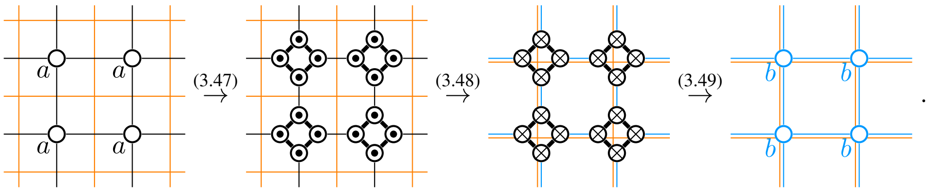


(3.48)



(3.49)

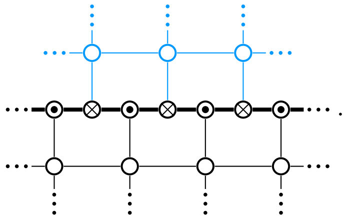
By locally applying the moves of this invertible-domain-wall TS, we can transform the a path integral into a b path integral,



(3.50)

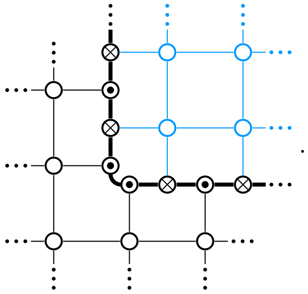
The fact that this also shifts the path integral by half a lattice site is not a problem.

There is another interpretation of invertible domain walls, which will turn out to be more useful in the context of topologically invariant fixed-point models: Imagine starting with the a path integral, and applying the moves of the invertible-domain-wall TS only in one half of spacetime. For example, applying the moves of the TS above only inside the upper half plane of a square lattice path integral yields the following,



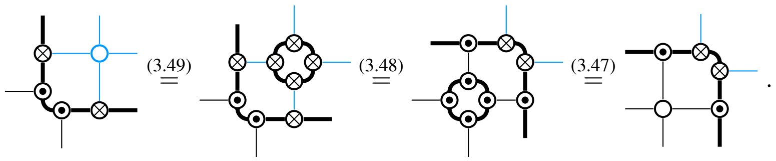
(3.51)

This is just a domain wall between two tensor-network path integrals, similar to the one depicted in Eq. (3.10). So this interpretation justifies the "domain wall" part of the name. However, the so-obtained domain wall has some extra properties that make it "invertible". First of all, instead of the upper half plane, we can apply the tensor-network equations inside any arbitrarily-shaped b subregion of the plane. Thus, the domain wall can be arbitrarily curved, for example it can go around corners,



(3.52)

The domain wall can be moved around arbitrarily by locally applying the tensor-network equations. For example, we can "cap" the bottom left corner of the b region by



(3.53)

This ability to arbitrarily change the geometry of the domain wall makes it a *topological* domain wall. However, we can not only change the geometry, but also the topology of the domain wall, which makes it an invertible domain wall. For example, by transforming a to b in the middle of an a region, we generate an isolated b island within a. In fact, a single application of the move in Eq. (3.47) can be interpreted as such an island generation. Or, we can connect two a regions separated by a thin bridge of b region, which the move in Eq. (3.48) implements in a minimalistic form.

As for Hamiltonians, the gapped-path and invertible-domain-wall definitions for phases in tensor-network path integrals are not equivalent. Again, invertible domain walls of tensor-network path integrals cannot change the correlation length in contrast to gapped paths. So, two models in the same gapped-path phase need not be related by an invertible domain wall, and in fact generically they are not. However, the definition via invertible domain walls turns out to work well for comparing fixed-point models, which all have zero correlation length.

Interestingly, invertible domain walls are more general than local unitary circuits, and therefore, in contrast to the Hamiltonian definitions, not even the converse might be true. Namely, it is not a priori clear whether an invertible domain wall between two path integrals a and b can be turned into a continuous gapped path. For some invertible-domain-wall TS, it is possible to find a "standard" model of this TS with $b = a$, which we can continuously connect to the desired invertible-domain-wall TS model between a and b. For example, for the invertible-domain-wall TS given by an on-site unitary U, we can take $U = \mathbb{1}$ as a standard TS model with $b = a$. Then for the desired U model corresponding to an invertible domain wall between a and b, we can find a Hermitian H such that $U = e^{iH}$. $\hat{U}(s) = e^{isH}$ then defines a continuous family of invertible domain walls interpolating between $\hat{U}(0) = \mathbb{1}$ and $\hat{U}(1) = U$. It is unclear whether such a continuous family exists for the invertible-domain-wall TS in Eq. (3.46) though.

Chapter 4

Topological phases via fixed-point models

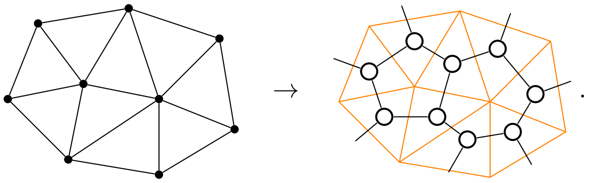
In this chapter, we explain how topological phases of $(n-1)+1$-dimensional qudit systems can be classified via fixed-point tensor-network path integrals. Thereby, we restrict to the physically relevant cases of spacetime dimension $n=2,3,4$. Specifically, for every n, we construct a fixed-point ansatz in the form of a tensor scheme, c.f. Section 2.3. We then show how to fine-tune these fixed-point ansatzes, and give some examples for models. We also discuss a general class of models related to cellular cohomology.

4.1 Basic idea

As discussed in Section 3.6, we aim to classify topological phases assuming that they can be represented by fixed-point models with exact combinatorial topological invariance. Instead of rather complicated tensor-network equations like in Eqs. (3.37) or (3.40), we aim to impose topological invariance in the simplest possible way. To this end, we think of the tensor-network diagrams themselves as combinatorial representations of topological n-manifolds, and the tensor-network equations as discrete versions of homeomorphisms. The most straight-forward way to combinatorially represent a manifold is via triangulations, and the simplest way to impose topological invariance is via *bi-stellar flips* or *Pachner moves* [116].

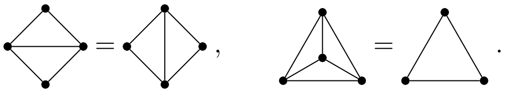
In this section, we describe how to construct TS (c.f. Sections 2.3 and 3.6) representing fixed-point ansatzes for topological order based on triangulations and Pachner moves. For simplicity, we will postpone some technical details to the following Sections 4.2 and 4.3. We will start with the smallest non-trivial dimension, namely $1+1$, and then briefly sketch how things are done in $2+1$ and higher dimensions.

In $1+1$ dimensions, a triangulation can be represented as a tensor-network diagram by associating one copy of a 3-index tensor to every triangle, and contracting indices between tensors at adjacent triangles,



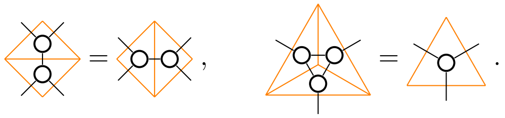
(4.1)

The drawing on the left is a patch of a triangulation, and the drawing on the right is the associated tensor-network diagram, where we still indicate the underlying triangulation using orange lines. In other words, the tensor-network diagram looks like the graph Poincaré dual to the triangulation. In $1+1$ dimensions, there are two Pachner moves, acting on the triangulation by



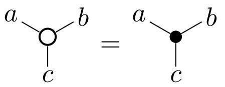
(4.2)

In order to apply these moves to a triangulation, we identify a patch of one, two, or three triangles like on the left (right) side of either of the two moves, cut it out, and replace it by the right (left) side. It is known that any two (piece-wise linear) triangulations of the same (piece-wise linear) manifold are related by Pachner moves [116]. Conversely, it is easy to see that Pachner moves preserve the topology of a triangulation. The Pachner moves yield tensor-network equations for the 3-index tensor, which are the moves of the tTS. These tensor-network equations consist in equating the tensor-network patches assigned to the left and the right hand side,



(4.3)

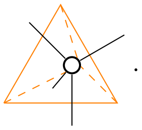
We will refer to this (t)TS consisting of the 3-index tensor variable and the two equations above as the *2D tTS*. A model of the 2D tTS thus is given by a 3-index tensor fulfilling the two equations above. An example for such a model is given by the δ-*tensor* of bond dimension d,



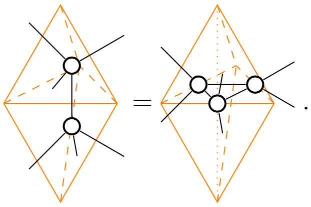
$$= \begin{cases} 1 & \text{if } a=b=c \\ 0 & \text{otherwise} \end{cases} \quad , \qquad (4.4)$$

for $0 \leq a,b,c < d$, which we will also denote by a small dot. δ-tensors can be defined for an arbitrary number of indices, with entry 1 if all the index values are equal and 0 otherwise. Any connected tensor network of δ-tensors yields another δ-tensor, which is why the equations in Eq. (4.3) hold. Physically, this model does *not* represent a topological phase, but a degenerate model that is not robust to perturbations, but it still obeys topological invariance. In particular, this model violates the "robust gap" property discussed in Section 3.4 since the rank of the corresponding annulus operator is d and not 1. We will look into this in more detail in Section 4.6, where we define a robustness condition specific to tTS models. In fact, there is no robust topological order in $1+1$ dimensions, so we cannot expect find any robust tTS models. Despite not describing a topological phase, non-robust models are still relevant to us for the following two reasons. First, such models can arise from compactification of higher-dimensional models, which can be used to classify anyons or other lower-dimensional defects, see Chapter 6. Second, if we equip these models with a symmetry, they become models for non-trivial *symmetry-breaking phases*, as we will see in Section 7.1.

The generalization to higher dimensions is straight-forward. In $2+1$ dimensions, we put the same 4-index tensor (in black) onto every tetrahedron of a 3-dimensional triangulation (in orange),

 (4.5)

One Pachner move is given by the following 2-3 move,

 (4.6)

On the left-hand side, we have two tetrahedra stacked on top of each other, whereas on the right there are three tetrahedra surrounding the vertical dotted edge connecting the bottom to the top vertex through the middle. The corresponding tensor-network equation is an equation between two 6-index tensors. On the left-hand side, the 6-index tensor is obtained by contracting one index between two copies of the 4-index tensor. On the right-hand side, it is obtained by contracting three index pairs among three copies of the 4-index tensor. There also is a 1-4 Pachner move that we do not show here. The 4-index tensor variable together with these moves forms a tTS, the *3D tTS*. Again, a δ-tensor will provide a solution to this equation, corresponding to a $2+1$-dimensional non-robust model. 3D-tTS models also do include robust topological phases. However, we will have to wait for some further technical details in Section 4.2, before we can give a concrete example.

Generally, in n dimensions, Pachner moves arise from decompositions of the boundary of the $n+1$-simplex into two parts. Indeed, the two sides of each of Eq. (4.3) glued together form the boundary of a tetrahedron, and the two sides of Eq. (4.6) together form the boundary of a 4-simplex. So the Pachner moves in $3+1$ dimensions correspond to decompositions of the boundary of the 5-simplex into two parts. Thus, there is a 3-3, a 2-4, and a 1-5 Pachner move depending on how many 4-simplices are on each side. The 3-3 Pachner move has three 4-simplices on each side, such that all three are adjacent to one common triangle, and each pair of them is adjacent to one common 3-simplex.

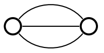
Note that 2D-tTS models are very similar to $1+1$-dimensional *lattice TQFT* as introduced in Ref. [67]. The 3D and 4D tTS are not commonly discussed in the literature in this form, though the thesis Ref. [124] uses a similar language of “Pachner-move invariant simplex tensors”.

4.2 Distinguishing indices

In the previous section we have ignored one important technical detail, namely the necessity to distinguish all indices of a tensor in our graphical notation, as explained in Section 2.1. In this section, we will fill in this detail. We will see that on the level of triangulation, this corresponds to adding directions to all edges, which makes the tensor-network equations slightly more involved.

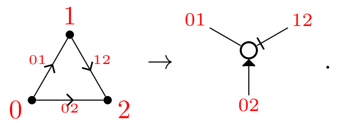
We will start by explaining the general principles focusing on the case of $1+1$ dimensions. Most commonly in physics and mathematics, a tensor or array is denoted with subscripts representing the indices, like T_{abc}. The three indices of this tensor are distinguished by the fact that they correspond to the first, second, and third subscript, respectively. For the tensor-network notation we are using currently, there is no way to distinguish the three indices of a tensor. Thus, the tensor-network diagram does not tell us precisely which index is contracted with which other.

An analogous problem arises on the geometric side: A tensor-network diagram does not uniquely encode the full combinatorial information of the triangulation. To see this, imagine rebuilding the triangulation from the diagram by taking one triangle for each tensor, and gluing a pair of triangle edges for each bond in the diagram. We encounter two problems: (1) The combinatorial structure of the diagram does not distinguish between the three adjacent bonds, so we cannot tell which edges of the triangles we have to glue together. (2) Two edges can be glued in two opposite ways. For example, consider the following graph that corresponds to two triangles with all edges glued together pairwise,

 . (4.7)

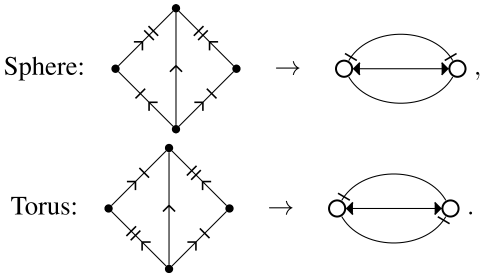
This graph does not determine the topology of the resulting manifold. If we glue one of the three edge pairs, we obtain a 4-gon. Depending on how we glue the remaining edges of the 4-gon, we can obtain a sphere, a real projective plane, a torus, or a Klein bottle.

The second problem can be solved by giving each edge a direction and demanding that these directions match when we glue two edges of two triangles. Then the first problem is solved by using only triangles where the edges have non-cyclic directions. This is also known as a triangulation with a *branching structure*. Note that this branching (structure) is present naturally when combinatorially defining cellulations/triangulations as explained in Section 2.4. For a fixed triangle, the non-cyclic edge directions allow us to order its vertices along the edge directions from 0 to 2. Using this, we can label the three edges uniquely by their endpoints as 01, 12, or 02. Simultaneously, we distinguish the three indices of a tensor variable, which correspond to the three edges, using markings as described in Section 2.1. Concretely, we mark the 12 index with a tick, and the 02 index with an ingoing arrow,



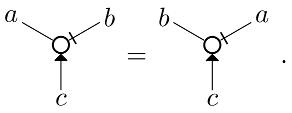
(4.8)

Note that here and throughout this document, red labels like on the right are not part of the formal graphical notation, but serve as an aid to identify the network notation with its geometric interpretation in terms of cell complexes. After adding the index markings, tensor-network diagrams do uniquely specify the triangulation and thus the topology. For example, a sphere and a torus then correspond to distinct tensor-network diagrams,



(4.9)

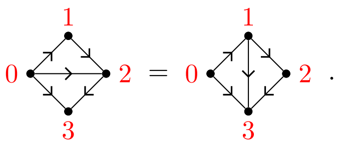
From now on, we will by default distinguish the indices of every tensor by markings. If two indices have the same marking, this implicitly means that we impose invariance under permutation of these two indices. For example, the tensor depicted in Eq. (4.1) would be invariant under arbitrary permutations of its three indices. This invariance is equivalent to adding moves to the 2D tTS, such as



(4.10)

If we interpret these moves in terms of triangulations, they correspond to cutting out a triangle and gluing it back in a different way. Such an operation generally changes the topology of the triangulation. So these additional permutation moves restrict the potential models even stronger then by topological invariance. One can therefore expect that models of this tTS are too restricted and do not contain general fixed point models for topological order.

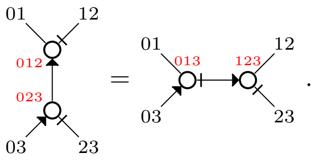
The branching structure also needs to be incorporated into the moves of the tTS. There are many different ways in which a branching structure can be added to a Pachner move. Keeping in mind moves are not actually different if they are just rotated/reflected or we exchanged the left and right side, we count 3 different versions of the 2-2 Pachner move. One of them is



(4.11)

Another one can obtained by, for example, changing the direction of the 23 edge. Note that there is a unique tetrahedron with branching structure, whose vertices can be ordered according to the edge directions. 2-2 Pachner moves correspond to decompositions of this tetrahedron, and the 3 versions of the 2-2 Pachner move correspond to the 3 different decompositions of the tetrahedron boundary with two faces on each side.

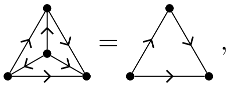
In terms of tensor-network diagrams, the move in Eq. (4.11) becomes



(4.12)

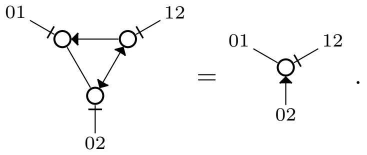
The red labels identify the tensors in the network with the triangles in the geometric interpretation. For example, 023 refers to the triangle in Eq. (4.11) whose 0-vertex is the vertex 0, 1-vertex is 2, and 2-vertex is 3. Note again that the red labels are only hints for the reader and not part of the actual notation. Also, the open index labels were chosen in accordance with the names of the corresponding edges.

Analogously, there are now 4 different versions of the 1-3 Pachner move, corresponding to the 4 decompositions of the branching-structure tetrahedron into two patches with 1 and 3 triangles each. One of them is



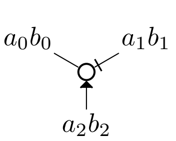
(4.13)

or in terms of tensor-network diagrams,

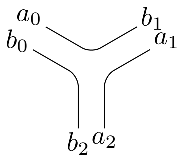


(4.14)

As we have seen, distinguishing different indices indirectly amounts to removing index permutation symmetries. Thus, the 2D tTS with index markings is less restricted by equations, and potentially allows for more general models. One new model we get is the following tensor product of three different identity tensors of bond dimension d,

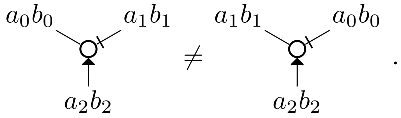


$$= (d^{-1/2})$$



$$= d^{-1/2}\delta_{b_0,b_2}\delta_{a_0,b_1}\delta_{a_1,a_2}\ . \qquad (4.15)$$

It is easy to see that plugging the diagram above into Eq. (4.12) yields identical diagrams on both sides. This model clearly violates Eq. (4.10),

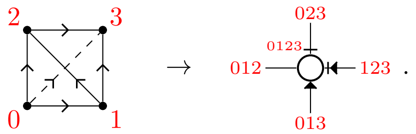


(4.16)

On the left-hand side, b_0 is connected via an identity matrix to b_2, but on the right-hand side, b_1 is connected to b_2 instead. Note that we will find in Section 5.4 that this model is in a trivial phase and are therefore not of immediate physical interest. It is still relevant to us since we will see in Section 7.1 that equipping this model with symmetries yields non-trivial *symmetry-protected topological* phases.

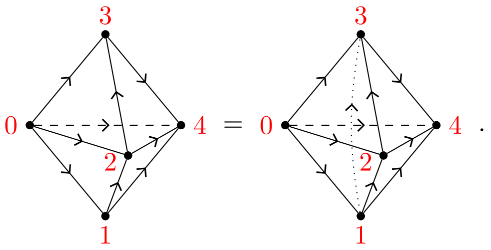
Next, let us look at how to distinguish indices in higher dimensions. A branching structure in n dimensions is define analogous to $1 + 1$ dimensions: A choice of edge directions that is non-cyclic around every triangle. Since every pair of vertices in an n-simplex is connected by an edge, the branching structure fixes all the symmetries of the n-simplex and allows us to order its vertices along the edge directions from 0 to n. This allows us to uniquely label the $n - 1$-dimensional faces by the involved vertices. These faces correspond to the different indices, which in tensor-network diagrams can be distinguished by markings.

In $2 + 1$ dimensions, we choose the following markings for the tensor variable associated to a tetrahedron,



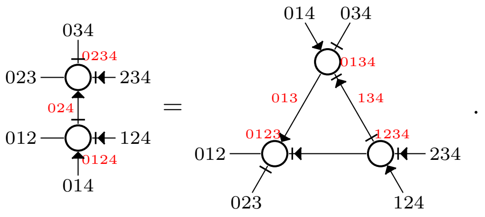
(4.17)

A possible choice of branching structure for the 2-3 Pachner move depicted in Eq. (4.6) is given by



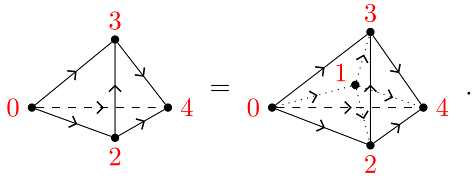
(4.18)

This is by far not the only choice: Pachner moves in 3 dimensions can be obtained by splitting a branching-structure 4-simplex in two halves. There is one such splitting for every choice of edge that plays the role of the 13 edge on the right-hand side above. Since there are 10 edges in the 4-simplex, there are also 10 different choices branching structure for the 2-3 Pachner move. After adding the index markings, the tensor-network equation corresponding to the move in Eq. (4.18) is given by



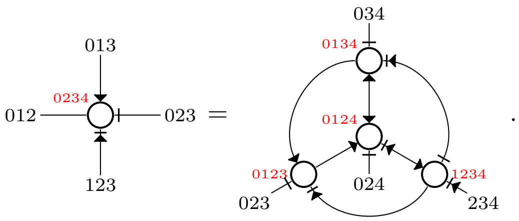
(4.19)

Analogously, we can also equip the 1-4 Pachner moves with edge directions, for example,



(4.20)

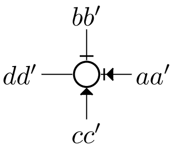
On the left-hand side we have a single tetrahedron. On the right-hand side there are 4 tetrahedra, such that every pair shares a common face, every triple of tetrahedra shares a common edge and all four tetrahedra share the common 1 vertex at the center. The according tensor-network equation is



(4.21)

In total, there are 5 different choices of branching structure for the 1-4 Pachner move, depending on which vertex of the 4-simplex plays the role of the central vertex 1 on the right of Eq. (4.20).

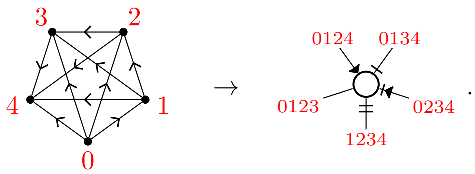
At this point, we have added enough technical detail to give our first real example for a tTS model corresponding to a robust topological phase. Namely, the simplest such phase, the *toric code* can be expressed as a tensor of bond dimension 4, which we split up into two labels each valued in $\mathbb{Z}_2$,



$$= \delta_{dc}\delta_{d',a}\delta_{a',b'}\delta_{d+d',b}\delta_{a+a',c'} \ . \qquad (4.22)$$

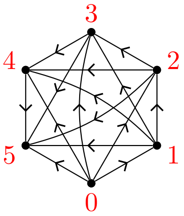
It is not so easy to see that the above defines a tTS model, and that it represents a toric code. We will be able to write down this model, as well as other models, more concisely after we derive the more fine-tuned *block-diagonal* form of the tTS in Section 4.14.

With this, the generalization to $3+1$ dimensions should be straight-forward, but we will spell it out to be explicit. There is one 5-index tensor variable associated to the 4-simplex,



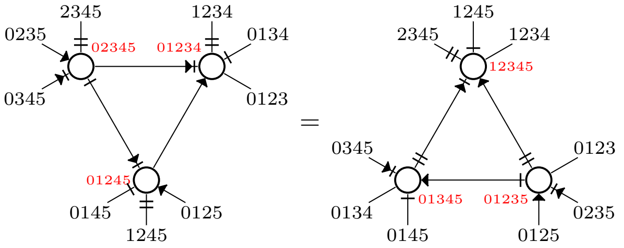
(4.23)

The most restrictive move is the 3-3 Pachner move. One particular choice of branching structure for this move is given by



$$= \text{same 1-skeleton} \; . \tag{4.24}$$

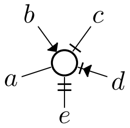
The left-hand side consists of all the 4-simplices adjacent to the 024 face, whereas the right-hand side consists of all 4-simplices adjacent to the 135 face. As indicated, the two sides cannot be distinguished by the drawing that only shows vertices and edges, or in other words, the 1-skeleton of the triangulation. In contrast, the corresponding tensor-network diagrams do fully specify the combinatorics of the move,



$$. \tag{4.25}$$

In total, there are 10 different choices of branching structure for this move: There are 20 triangles of the 5-simplex that could play the role of the 024 face on the left-hand side of the move above, which counts every move twice since swapping the two sides replaces the 024 face with the 135 face.

A simple example for a model of this 4-simplex tTS is given by



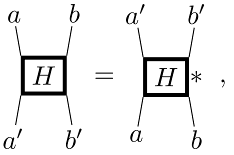
$$= \delta_{a+c,b}\delta_{c+e,d}(-1)^{ae} \; , \tag{4.26}$$

where all indices have bond dimension 2 and the index labels $a, b, c, d, e \in \{0, 1\}$ are interpreted as elements of $\mathbb{Z}_2$. This example is a known lattice TQFT by Kashaev [91].

4.3 Orientation and Hermiticity

In this section we discuss a further feature of topological fixed-point models that is important if we want these models to be quantum mechanical models. Namely, we will show how to impose the fact that quantum mechanics is unitary. Since the spacetimes we are working with have an imaginary time component, *Hermiticity* might be a better name than unitarity.

Objects like Hamiltonians, state vectors or time evolution operators, which occur in the usual pure-state formulation of quantum mechanics, are complex tensors. A "physical" Hamiltonian is Hermitian, which means that interchanging input and output indices of the corresponding complex tensor is equal to complex conjugation. For example, for a Hamiltonian H acting on two degrees of freedom (from the bottom to the top), we have



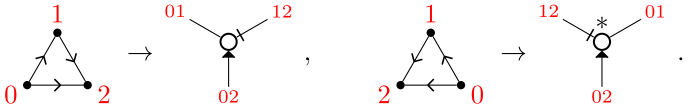
(4.27)

where we denote the entry-wise complex conjugation of a tensor by putting a $*$ next to it.

$e^{-\beta H}$ is Hermitian if H is. Thus the Hermiticity of the Hamiltonian carries over to the tensors of the Trotterized imaginary time evolution tensor network. For this tensor network, Hermiticity implies that inverting the time direction is equivalent to complex conjugation. However, in a Euclidean-signature spacetime there is no distinguished time direction. Also, in a triangulation of a compact topological manifold, there is no notion of globally inverting one direction. Thus, instead, we impose that inverting the *orientation* of the underlying manifold is equal to complex conjugation.

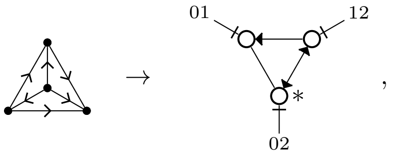
So in order to define tTS whose models have a quantum mechanical interpretation, we need to equip the triangulations with an orientation in the first place. The most lazy way to do this is to just imagine an orientation of the underlying manifold. Using this underlying orientation together with some extra structure like a branching structure, we can label each tensor as being either *positively* or *negatively* oriented. Then, we complex conjugate all negative tensors.

As a first example, consider the 2D tTS in $1+1$ dimensions. Our drawings are embedded into paper and thus equipped with a standard orientation. Then we define a triangle as positive if its 01 edge points clockwise round the triangle, and negative otherwise. Then, negative triangles are represented by a complex conjugated copy of the same tensor variable,



(4.28)

This also applies to the moves of the tTS. For example, the left-hand side of Eq. (4.13) now becomes

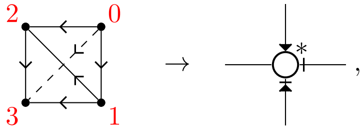


(4.29)

since the bottom triangle is negative while the other two are positive.

In $2+1$ dimensions, we define a tetrahedron as positive if the 01 edge is directed counter-clockwise when looking onto the 012 face from the outside of the tetrahedron.

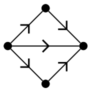
Again, we can determine the orientation of a tetrahedron implicitly from the way it is drawn on paper. For this, it is important to distinguish which edges are in front, and which at the behind in the drawings, which we do by dashing behind edges. Negative tetrahedra are represented by the complex conjugated copy of the tensor,



(4.30)

as opposed to positive tetrahedra as shown in Eq. (4.17). This also applies to the tensors occurring in moves of the tTS. For example, in Eq. (4.20), the 0124 tetrahedron is negatively oriented and the according tensor needs to be complex conjugated in Eq. (4.21).

Next, let us discuss $3+1$ dimensions. Here it is not practical to determine whether a 4-simplex is oriented positively or negatively from a drawing and the implicit orientation of the paper. So we need to deal with the orientation in a proper combinatorial way, which is the more natural and insightful method anyways. In n dimensions, we can simply define an orientation as an assignment of "positively oriented" or "negatively oriented" to every n-simplex. However, we cannot pick any such assignment, but only ones that are locally consistent: From the way in which two n-simplices are adjacent to each other a sub $n-1$-simplex, we can determine whether they have the same or different orientation. For example, it can be easily seen that two branching-structure triangles have opposite orientation if the edge they share is the 02 edge for both of them,



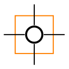
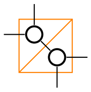
. (4.31)

In general, we can label the $n-1$-simplices of a n-simplex by the number i of the vertex that is not part of it, and use this number to distinguish between *even* and *odd* $n-1$-simplices. Now, two neighboring n-simplices have opposite orientation if the $n-1$-simplex they share is an odd simplex for both of them, or an even simplex for both of them. They have the same orientation if the $n-1$-simplex is odd for one, and even for the other. Now, if we choose a (positive or negative) orientation for one n-simplex, then this fixes the orientation for all other n-simplices in the same connected component. Note than a global consistent choice of positive/negative orientation for each n-simplex on a connected component is only possible if this connected component is *orientable*, and in this case there are exactly two such choices. This global consistent choice for each component is a combinatorial analogue of an orientation. In Section 2.7, we show that the $n-1$-simplices whose adjacent n-simplices have opposite orientation form a $\mathbb{Z}_2$-valued $n-1$-cycle ω_1 representing the first Stiefel-Whitney class. In this formulation, the positive/negative assignment can be interpreted as a $\mathbb{Z}_2$-valued cellular n-chain σ such that $d\sigma = \omega_1$.

Equipped with the general combinatorial definition of an orientation, we can now look at $3+1$ dimensions. We see that the 3-3 Pachner move in Eq. (4.24) is defined precisely in such a way that all involved 4-simplices are positively oriented. Or, all of them are negatively oriented, which is equivalent as it just complex conjugates the whole equation.

4.4 Commuting-projector Hamiltonians

Before we add further details to our path-integral understanding of phases of matter, let us go a step back and think about how they are related to real condensed-matter models. If we choose to describe such models via discrete path integrals, then these will not live on arbitrary triangulations, but on a regular spacetime lattice such as in Eq. (3.2). We can transform our fixed-point model into such a square-lattice path integral by simply restricting to a regular lattice. For example, given a 2D-tTS model, we can divide a square into two triangles, and block the according two tensors into a single one,

 $:=$  . (4.32)

Formally, such an equation between tensors defines a (rather trivial) example of a TS mapping, c.f. Section 2.3.

But what if we prefer a Hamiltonian formulation? As we have seen in Section 3.2, any local Hamiltonian can be Trotterized and approximated by a tensor-network path integral. Unfortunately, this Trotterization does not work in reverse. One can turn any tensor-network or circuit into a time-dependent Hamiltonian evolution, but time-dependent Hamiltonians in imaginary time are not a very common description either. Luckily, there is a standard construction that turns tTS models into Hamiltonians. This works for established fixed-point ansatzes including the ones in this chapter, but not including the ansatzes in Chapter 11 of this thesis. More precisely, the construction is exact and the obtained Hamiltonians are *commuting-projector Hamiltonians*.

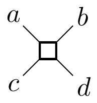
To start, let us review commuting-projector Hamiltonians and realize that they are themselves nothing but models of some TS. A commuting-projector Hamiltonians is specified by a projector P acting on a constant number of qubits on a regular lattice near the origin. It is defined by

$$H = -\sum_i P_i \ , \tag{4.33}$$

where i runs over different site of a regular lattice, and P_i is P shifted from the origin to site i. P has to fulfill

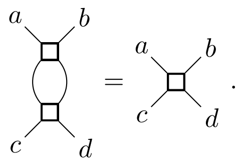
$$P_i P_j = P_j P_i \ , \tag{4.34}$$

for each pair i, j of sites. So P itself is a model of a TS, whose moves correspond to the commutativity and the projector property. As the simplest example, let us consider a nearest-neighbor commuting projector Hamiltonian for qudits on a 1-dimensional spatial lattice. P is a 4-index tensor,

$P =$  , (4.35)

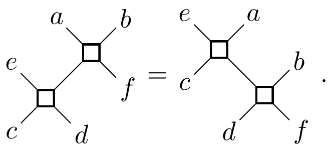
acting as an operator from the bottom to the top two indices. The projector property

$P^2 = P$ becomes a tensor-network equation,



(4.36)

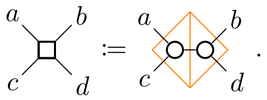
Eq. (4.34) becomes



(4.37)

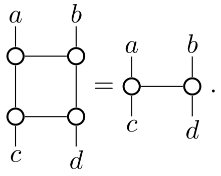
We will refer to the TS given by the above two moves as a *2D commuting-projector TS*.

Now, the construction for obtaining commuting-projector Hamiltonians is nothing but a TS mapping from the commuting-projector TS to the tTS in question. In other words, the projector P defining a commuting-projector Hamiltonian is obtained simply from contracting a tensor network of the tTS. Let us illustrate this with the 2D tTS, where we ignore index markings, branching structure, and Hermiticity for a moment. A 2D commuting-projector-TS model can be obtained from a 2D-tTS model by



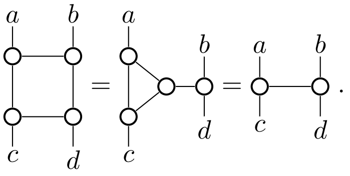
(4.38)

The fact that the resulting tensor will indeed be a commuting projector follows from the fact that above equation defines a TS mapping. To show this, we have to assert that every mapped move of the 2D commuting-projector TS can be derived from the moves of the 2D tTS. For example, if we plug the mapping Eq. (4.38) into the move Eq. (4.36), we get



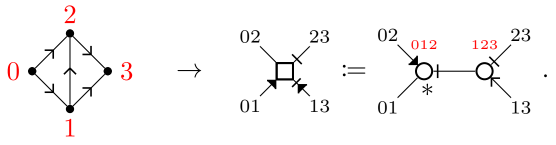
(4.39)

This is equivalent to a sequence of 2D-tTS moves,



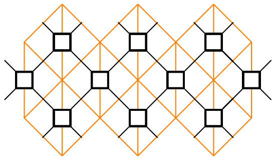
(4.40)

In the first step we applied the 2-2 Pachner move, and in the second step the 1-3 Pachner move. After adding index markings, Eq. (4.38) becomes



(4.41)

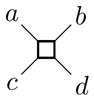
We have successfully built a commuting-projector Hamiltonian from any 2D-tTS model, but we have not yet motivated why the two are just different representations the same physics. This can be done by mapping the Hamiltonian back to a path integral and see if we end up with the original path integral. To this end, we might approximate the Hamiltonian by discrete path integrals via Trotterization as explained in Section 3.2. However, if we apply the Trotterization procedure in Section 3.1 to any Hamiltonian, we will not obtain a fixed-point model. This is because the first excited state always has a finite energy, corresponding to a finite "correlation length in time direction". In particular, Trotterizing the commuting-projector model will not recover the original 2D-tTS model. However, we do not need Trotterization if we have a commuting-projector model. In this case, we can directly take the limit $\beta \rightarrow \infty$ for an individual Hamiltonian term, yielding the projector P defining the commuting-projector Hamiltonian. We then take the product of those projectors P in a some brick-layer like fashion. This product of projectors is a tensor network, and thereby a tensor-network path integral. The natural way of doing this for the 1-dimensional nearest-neighbor model in Eq. (4.35) looks like



(4.42)

If we plug the mapping Eq (4.41) into this tensor network, we arrive back at the 2D-tTS network corresponding to the triangulation drawn in orange.

As a concrete example, let us turn the model in Eq. (4.4) into a commuting-projector Hamiltonian. For bond dimension $d = 2$, plugging Eq. (4.4) into Eq. (4.38) yields



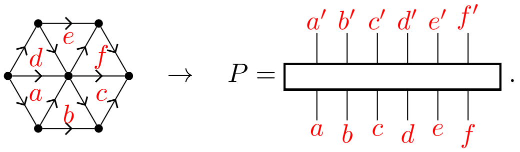
$$= \begin{cases} 1 & \text{if} \quad a = b = c = d \\ 0 & \text{otherwise} \end{cases} \equiv \frac{1}{2}(\mathbb{1} + Z_0 Z_1) \; . \tag{4.43}$$

On the right-hand side we have expressed the tensor as an operator where Z_i denotes the Pauli-Z operator acting on the ith qubit. So we obtain a 1-dimensional Hamiltonian with nearest-neighbor interactions,

$$H = -\sum_i P_i = -\frac{1}{2} \sum_i Z_i Z_{i+1} - \frac{N}{2} \; . \tag{4.44}$$

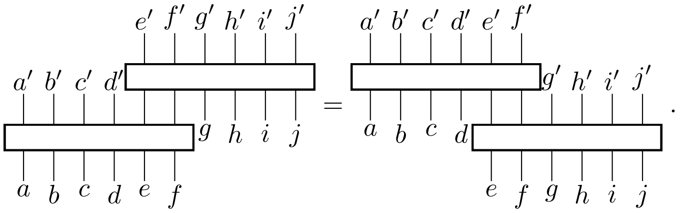
Up to prefactors and offsets, this a well-known model for a symmetry-breaking phase, namely the $1+1$-dimensional *quantum Ising model* without magnetic field.

Let us briefly show how the commuting-projector mapping generalizes to $2+1$ dimensions. A convenient spatial lattice for the commuting-projector TS related to the 3D tTS is a regular triangular grid with one degree of freedom on each triangle. Further we assume that there is one Hamiltonian term on each vertex involving the six degrees of freedom at the surrounding triangles. So, the local ground state projector is a tensor with 12 indices,



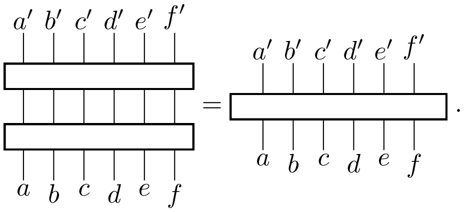
(4.45)

Commutativity of the projectors centered around neighboring vertices yields three different moves, for example,



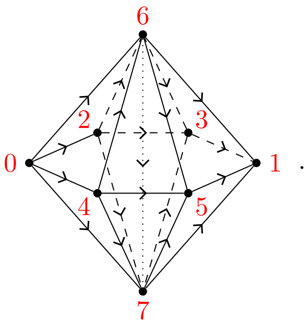
(4.46)

Additionally, there is the projector move,



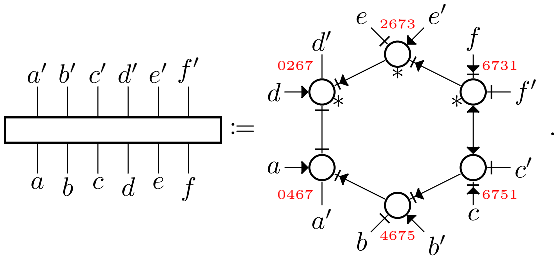
(4.47)

The mapping from this 3D commuting-projector TS to the 3D tTS is as follows. Just like in $1+1$ dimensions, the projector is mapped to a a diamond-shaped patch of spacetime triangulation,



(4.48)

Taking a brick-layer-like product of projectors then corresponds to building a 3-dimensional cellulation/triangulation by stacking these volumes analogous to Eq. (4.42). As depicted above this volume can be triangulated with six tetrahedra, all sharing the (67)-edge, yielding the TS mapping

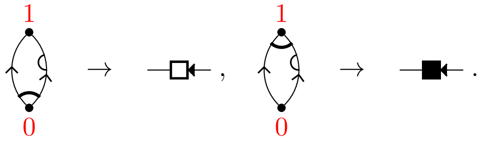


(4.49)

4.5 Weight matrices

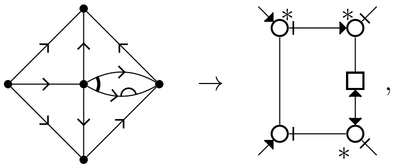
In this section, we will discuss one last, rather technical, modification of our tTS/fixed-point ansatzes. After adding this modification, our tTS models will indeed represent a large class of topological phases, specifically those that admit a topological boundary. The modification consists in inserting additional 2-index tensors, that we refer to as *weight matrices*, at certain places into the tensor network representing a triangulation.

As usual, we start with 2D tTS to illustrate this modification. Even though this example is not directly physically relevant, weight matrices do also show up in 2D-tTS models arising from dimensional reductions of higher-dimensional tTS models. The most important kind of weight matrices in $1+1$ dimensions are *vertex weights*. To this end, we introduce two new tensor variables that we call 0 *vertex weight* and 1 *vertex weight*. They are associated to non-cyclic 2-gon cells where we mark the 0-vertex or the 1-vertex with a little arc,



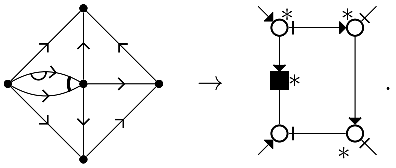
(4.50)

Of course, a 2-gon cannot be embedded non-degenerately into Euclidean space without bending its edges. This is not a problem as we are talking about combinatorial/topological cell complexes and not geometric ones. Note that in order to remove the horizontal reflection symmetry (which the edge directions fail to do in this case), we added a little half-circle marking to the right. This marking coincides with an ingoing arrow at the corresponding index. We consider a 2-gon positively oriented if the marked edge is directed counter-clockwise around the 2-gon, and negatively oriented otherwise. Negatively oriented 2-gons are complex conjugated. Thus, the combinatorial representations of spacetime are not just branching-structure triangulations, but also include 2-gon cells as above. We now demand that in a valid cellulation/triangulation, every vertex must have exactly one adjacent arc. For example, a section of a valid triangulation could look like



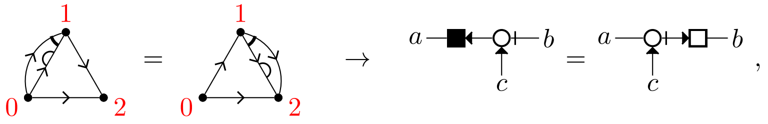
(4.51)

with exactly one arc at the internal vertex, or



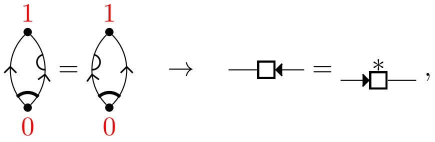
(4.52)

It should not matter where we insert the vertex weight for a given vertex. This can be implemented by so-called *weight commutation moves*, which move the arc at a vertex around this vertex. There is one such move for every corner of the triangle. For example,



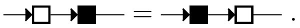
(4.53)

moves the arc past the 1 vertex of the branching-structure triangle. In addition, the weight matrices have to be Hermitian,



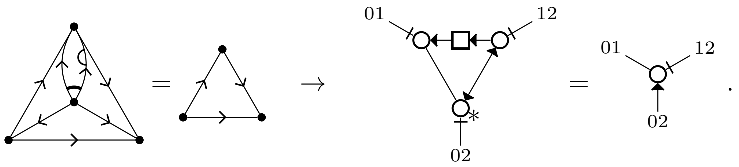
(4.54)

and commute among each other,



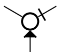
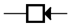
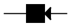
(4.55)

Apart from adding weight commutation moves, we need to incorporate the vertex weight into all moves with internal vertices. For example, the 1-3 Pachner move from Eq. (4.13) becomes



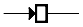
(4.56)

The additional tensor variable leads to a potentially richer set of models. Indeed, consider the following example for a new 2D-tTS model that we have gained, defined for any $\alpha \in \mathbb{R}$: The bond dimension is 1, that is, all tensors are scalars,

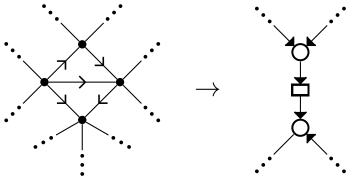
 $= \alpha^{-1/2}$,  $=$  $= \alpha$. (4.57)

One can easily see that the evaluation of this model on a space-time manifold M is $\alpha^{\chi(M)}$, where χ is the euler characteristic. However, such all-scalar models correspond to the trivial phase, and are immediately physically trivial since the Hilbert space is 1-dimensional.

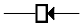
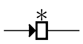
Another type of weight matrices in $1+1$ dimensions are *edge weights*. To this end, we introduce one new bond dimension variable, called edge weight,

 . (4.58)

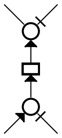
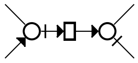
Now, in a tensor network representing a triangulation, we need to insert one edge weight at every bond, for example,

 . (4.59)

The edge weight has to be a Hermitian matrix,

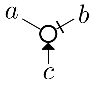
 $=$  . (4.60)

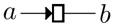
Furthermore, we need to equip every Pachner move with edge weights. For example, the 2-2 Pachner move in Eq. (4.12) becomes,

 $=$  . (4.61)

If we want to impose both edge weights and vertex weights, then there are two possibilities: Either we insert edge weights between all vertex weights and triangle tensors, or we pick the convention that, if a vertex weight is present at a bond, this replaces the edge weight.

Again, edge weights do make the 2D tTS less restricted, which allows for new models. As an example, consider the following model of bond dimension 2, where we interpret the index labels as $\mathbb{Z}_2$ elements,

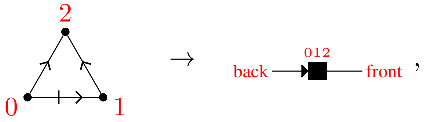


$= 2^{-1/2}(-1)^c \delta_{a+b,c}$, 

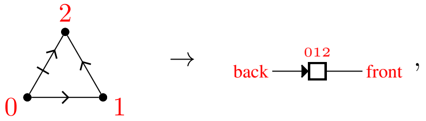
$= (-1)^a \delta_{a,b}$. (4.62)

Strictly speaking, this phase cannot be captured without the edge weights. However, if we restrict ourselves to regular square lattices, then it is in the same phase as the δ-tensor model in Eq. (4.4). So again, edge weights do yield more models and phases mathematically, but not physically. Unless otherwise stated, when we say "2D tTS", we do not include edge weights.

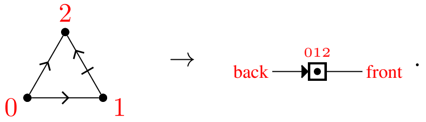
Next, we will look at weight matrices in $2+1$ dimensions. The most important weights are *edge weights*. In contrast to the toy example of vertex and edge weights in the 2D tTS, edge weights are really necessary to get the most general examples of intrinsic topological order in $2+1$ dimensions. Edge weights are implemented by three new tensor variables, which we refer to as the 01*,* 12*, and* 02 *edge weight.* We assign these weights to "pillow" volumes, that is, volumes whose boundary consists of two triangles, and needs to be "inflated" in order to embed it non-degenerately into 3-dimensional Euclidean space. These volumes are marked with a tick on either the 01, 12, or 02 edge,



(4.63)



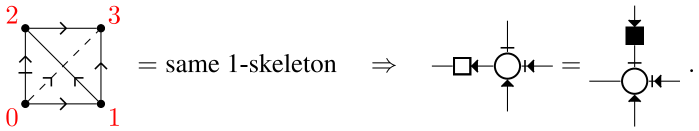
(4.64)



(4.65)

The two indices of the weight matrices correspond to the triangle face at the back, and that in front, as shown. Now, instead of triangulations we consider cellulations with tetrahedra and pillow volumes as above. We then demand that for every (internal) edge there is exactly one adjacent pillow volume with a tick for this edge.

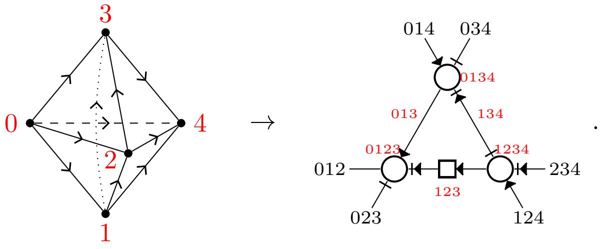
As in $1+1$ dimensions, these edge weights need to be Hermitian and commute among each other similar to Eqs. (4.54) and (4.55). Further, there are moves that allow us to move the weights from one face of the tetrahedron to another, for example,



(4.66)

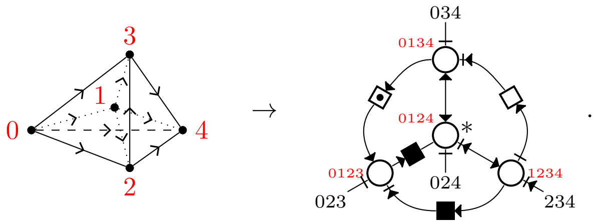
On the left-hand side, a pillow volume with 02 edge weight is attached to the 012 triangle, and a 01 edge weight to the 023 triangle on the right. Specifically, there is one such move for each of the 6 edges of the tetrahedron.

Additionally, we need to add edge weights to all interior edges of the Pachner moves of the 3D tTS. In particular, for the 2-3 Pachner move in Eq. (4.18), the 13 edge on the right-hand side is in the interior but does not yet have a tick yet. To fix this, we need to insert an edge weight pillow-like volume at one of the adjacent triangles. One possibility is to insert a 02 edge weight at the 123 face,

 (4.67)

However, we also could have put a 12 edge weight at the 013 face or a 01 edge weight at the 134 face.

Or, on the right-hand side of Eq. (4.20) there are four internal edges, so we have to include four edge weights,

 (4.68)

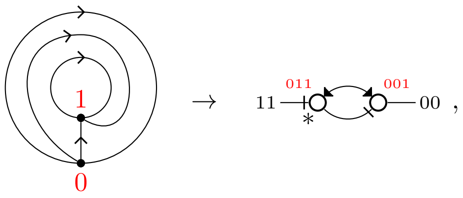
One might also introduce face weights, which are completely analogous to the 2D tTS case. Further, one can introduce vertex weights in the 3D tTS as follows: There are three vertex weight variables, the 0, 1, and 2 vertex weight. If an i vertex weight is inserted into a bond, it is attributed to the ith vertex of the associated triangle. In a valid triangulation, every vertex needs precisely one weight attributed to it. The vertex weight thus shows up in the 1-4 Pachner move, where we need to include a vertex weight for the interior vertex. For example, we would need to include an according vertex weight at any of the bonds in Eq. (4.68). Note, however, that vertex weights turn out to be unnecessary for describing robust phases.

In $3+1$ dimensions, it seems like one needs to include both face weights and edge weights in order to capture the most general phases. Face weights are implemented by four tensor variables, the 012, 013, 023, and 123 face weight, each attributed to one of the four faces of the tetrahedron. For a valid triangulation, we need to insert exactly one suitable face weight for every face, at one of the adjacent tetrahedra. For example, the loop of bonds on the left side of Eq. (4.25) corresponds to the internal 024 face, and the one on the right side to the 135 face. Into each such loop we have to insert exactly one face weight. Edge weights are implemented by 6 tensor variables, the 01, 02, 03, 12, 13, and 23 edge weight, each attributed to one of the 6 edges of the tetrahedron. In a valid triangulation, we need exactly one edge weight attributed to every edge. As usual, is is possible to also introduce volume weights and vertex weights, but neither of these is necessary for describing robust topological phases.

4.6 Robust phases

As we seen in Eq. (4.4), tTS models do not only represent *robust* topological phases, but also non-robust degenerate phases that still feature topological invariance. For a robust tensor-network path integral, an annulus operator like in Eq. (3.30) needs to be rank-1. In this section, we will specialize the robustness condition for tTS models.

For a tTS model, the rank of an annulus operator like in Eq. (3.30) is constant for any annulus-like network. This is because of the topological invariance of the tTS model. Thus, for a tTS model, it suffices to assert robustness for some minimal triangulation of an n-annulus. For example, for the 2D tTS, a minimal annulus is given by



(4.69)

where the 1-gon in the center on the left-hand side is a “hole” in the triangulation. Thus, the 2D tTS model is robust if the operator on the right-hand side is rank-1.

In fact, we can formulate the robustness condition as a *topology-changing* move in addition to the tTS moves. To this end, we note that if we fill in the “hole” in the middle of the n-annulus with an n-ball, this results in another n-ball. Thus, the n-ball is in the support of the annulus operator. If the annulus operator is rank-1, then it is proportional to the exterior product of two vectors corresponding to one n-ball filling its interior space boundary, and one n-ball filling its exterior space boundary. That is, the model is invariant under the topological deformation

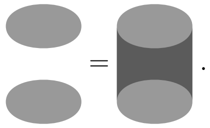
$$S_0 \times B_n = B_1 \times S_{n-1} \;, \tag{4.70}$$

with the two n-balls on the left-hand side and the n-annulus on the right-hand side. To apply this deformation to an n-manifold, we consider an embedding of the left-hand side into the n-manifold, which we then replace by the right-hand side. We can equally apply the deformation in the opposite direction. Note that for this to make sense, the boundary on the left-hand side need to be identified. This is indeed the case since both boundaries are canonically identified with

$$S_0 \times S_{n-1} \;. \tag{4.71}$$

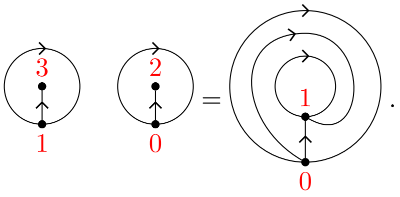
In order to implement this deformation as a tTS move, we need to choose a concrete cellulation for Eq. (4.71), as well as both sides of Eq. (4.70). Note that due to the recellulation tTS moves, all moves for all such cellulations are equivalent, so we aim to choose a minimal one.

For example, in $1+1$ dimensions, the move in Eq. (4.70) equates two disks with an annulus,



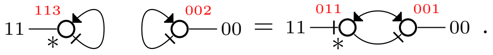
(4.72)

We have already chosen a triangulation for the right-hand side in Eq. (4.69). Each disk on the left can be triangulated with single triangle,



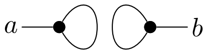
(4.73)

So this yields a move,

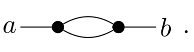


(4.74)

Let us evaluate the robustness for two examples of 2D-tTS models. First, consider the δ-tensor model in Eq. (4.4). Eq. (4.74) does *not* hold for this model,

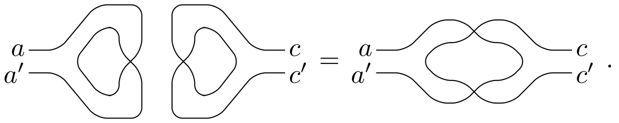


$$= 1\forall a,b \neq \delta_{a,b} =$$



(4.75)

Thus, as we discussed in Section 4.1, this is not a model for robust topological order. The rank of the matrix on the right-hand side is d instead of 1. As a second example, let us consider the model in Eq. (4.15). If we plug this model into Eq. (4.74), we get equivalent diagrams on the left and right-hand side:



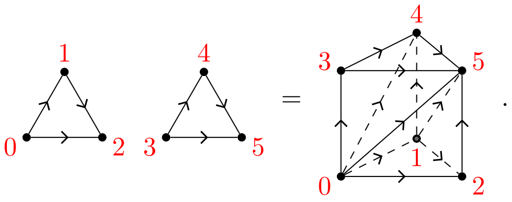
(4.76)

Thus, this model is represents a indeed a robust topological phase. However, as we will see later in Section 5.4, this robust topological phase is the trivial one.

Next, in $2+1$ dimensions, robustness corresponds to a topology-changing deformation,

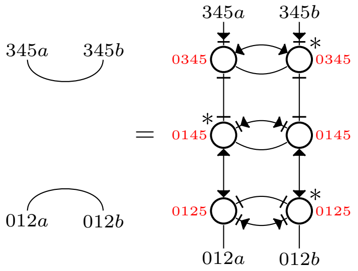
$$S_0 \times B_3 = B_1 \times S_2 \ . \tag{4.77}$$

On the left-hand side there are two 3-balls, and on the right-hand side we have a 3-ball with a hole in the center. We start by triangulating the common boundary, $S_0 \times S_2$. S_2 can be triangulated with two triangles, and $S_0 \times S_2$ by two copies of this. We then pick the following triangulation,



(4.78)

On the left-hand side, we have two pillow-like volumes. On the right-hand side we have two triangles formed by the vertices 012, and two triangles formed by 345, corresponding to $0 \times S_2$ and $1 \times S_2$, respectively, if we identify B_1 with $[0, 1]$. These are separated by two triangle prisms, that can be triangulated by three tetrahedra each. As a tensor-network equation, this looks like:



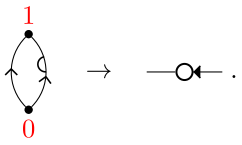
(4.79)

So any tetrahedron tensor fulfilling this equation represents a robust topological phase. For example, the toric-code model in Eq. (4.22) does fulfill this equation up to a prefactor, but it would be too tedious to perform the explicit calculation at this stage. The generalization of the robustness condition to the 4D tTS is straight-forward. Note the robustness moves generally do require global prefactors, whereas one can usually make the recellulation tTS moves work without prefactors.

4.7 Generators and relations in $1 + 1$D

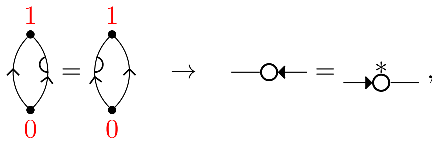
The 2D tTS as it stands has 3 distinct 2-2 Pachner moves as well as 4 distinct 1-3 Pachner moves. It might seem like overkill to impose 7 times essentially the same equation, only differing by edge directions or by how many tensors are on which side. The resulting system of equations is clearly heavily overdetermined. It would be desirable to bring the tTS into a more compact form, that is, to find a simpler presentation of the tTS in terms of *generators and relations*. Luckily, it turns out that an equivalent more compact form of the 2D tTS exists. More precisely, this *simplified* tTS consists of only a single 2-2 Pachner move, along with few much smaller auxiliary tensor variables and moves. The remaining Pachner moves can then be derived from the single Pachner move and the smaller auxiliary moves. The attribute "smaller" here roughly refers to the number of indices of a tensor variable, and the number of contractions and open indices of a move. In this section, we will construct this simplified tTS, and show that it is equivalent to the original, *non-simplified tTS*.

The construction of the simplified tTS has an elegant geometric interpretation. Namely, the diagrams of the tTS correspond to cellulations with other (smaller) 2-cells in addition to triangles, and other (smaller) recellulations in addition to Pachner moves. We start with the triangle tensor and only the 0 vertex weight, and introduce further tensors in the following. There are only two 2-cells that are smaller than a triangle, namely a 2-gon and a 1-gon. The first auxiliary tensor variable that we consider (but drop again in a moment) is associated to a 2-gon with non-cyclic edge directions,



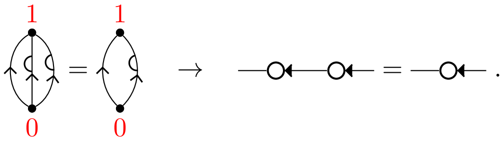
(4.80)

In order to determine whether the 2-gon is positively or negatively oriented, we mark the clockwise edge of a positively oriented one with a little half-circle. If instead the counter-clockwise edge is marked, then by Hermiticity we associate the complex conjugated tensor. By retriangulation invariance, we should impose that this matrix is Hermitian,



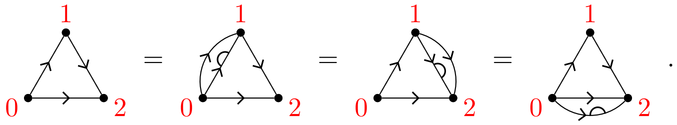
(4.81)

and a projector,



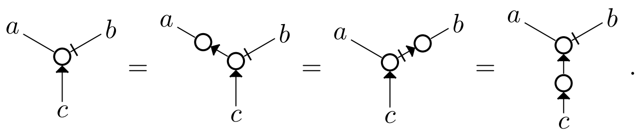
(4.82)

Also, applying it to any index of the triangle tensor is a recellulation,



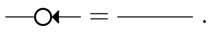
(4.83)

So the support of the triangle tensor on each index is restricted to the support of the 2-gon projector,



(4.84)

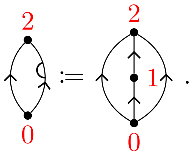
Thus, we can without loss of generality reduce the bond dimension by restricting the vector space of all indices to the support of the 2-gon projector. The 2-gon tensor itself then becomes the identity matrix,



(4.85)

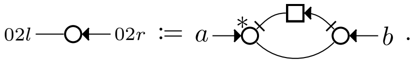
We will refer to this condition as the *full support convention*.

Note that a 2-gon can be triangulated by two triangles,



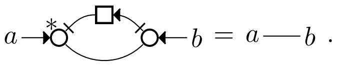
(4.86)

Thus, the 2-gon tensor is determined by the triangle tensor by



(4.87)

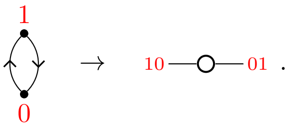
With this identification, Eqs. (4.81), (4.82), and (4.84) can be derived from the Pachner moves. The full support convention can then be imposed without explicitly adding the 2-gon as a new tensor variable, which we will call the *triangle cancellation move*,



(4.88)

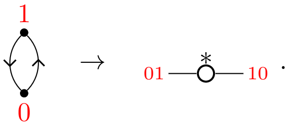
We would like to stress that this move is not a consequence of topological invariance but merely an extra convention to rule out models that are zero restricted to some subspace. In order to perform a fair comparison between the simplified to the non-simplified 2D tTS later, we add this move to both.

The only real additional tensor variable of the simplified tTS is associated to a 2-gon cell with *cyclic* edge orientations,



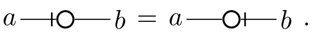
(4.89)

We consider the 2-gon positively oriented if both of its edges are directed clockwise. Due to Hermiticity, a negatively oriented 2-gon is represented by the complex conjugated tensor variable,



(4.90)

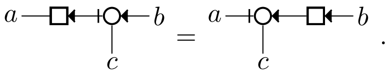
As shown, the 2-gon tensor will be denoted by a circle just like the triangle tensor. It can be distinguished from the triangle tensor due to the different number of indices. Usually, we should distinguish the two indices of the 2-gon tensor by marking one of them with a tick. However, we can omit this since the index permutation invariance,



(4.91)

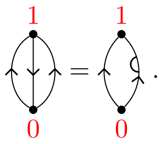
can be derived from the moves below.

The moves of the simplified tTS only contain one single Pachner move, namely the 2-2 Pachner move in Eq. (4.12). We also include the Hermiticity of the 0 vertex weight, and the following weight commutation move,



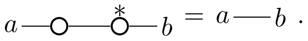
(4.92)

The remaining moves can roughly be divided into two groups, *cancellation moves* and *flip moves*. We have already introduced one cancellation move, namely the triangle cancellation move in Eq. (4.88). There is one other such move, the *2-gon cancellation move*, corresponding to a recellulation,



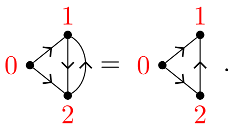
(4.93)

In network notation, we have



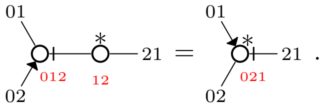
(4.94)

Next, let us state the flip moves, which are related to symmetries of the triangle. Rotating or reflecting the triangle changes the branching structure, so the triangle tensor is not invariant under the corresponding index permutations. However, the changes of the edge directions can be undone by gluing the cyclic 2-gon to the involved edges. This yields, for example, a recellulation that we call the *(12) triangle flip move*



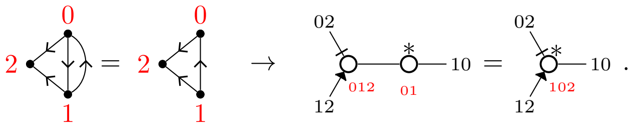
(4.95)

The name refers to effectively interchanging the role of the vertices 1 and 2. In tensor-network notation, this is



(4.96)

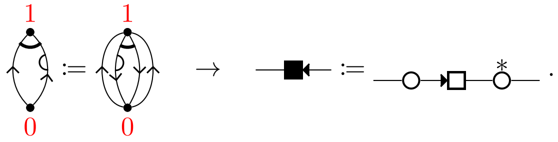
In order to generate the full symmetry group S_3 of the triangle, we only need one further move, the *(01) triangle flip move*,



(4.97)

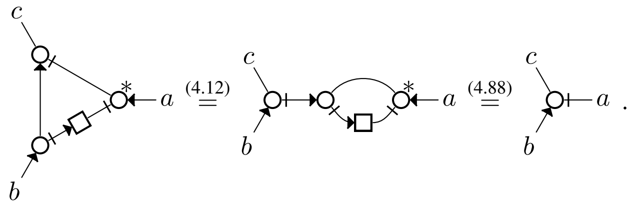
In summary, the tensor variables of the simplified tTS are given by the triangle, cyclic 2-gon, and 0 vertex weight in Eqs. (4.8), (4.89), and (4.50). The moves are given by Eqs. (4.12), (4.88), (4.94), (4.96), (4.97), and (4.92).

Let us now argue that the simplified and non-simplified tTS are equivalent. For this, we should be able to rewrite networks of the simplified tTS as networks of the non-simplified tTS, and vice versa. This can be formalized by two TS mappings $\mathcal{M}_1$ and $\mathcal{M}_2$, going from the non-simplified tTS to the simplified tTS, and back. Let us start with the mapping $\mathcal{M}_1$ from the non-simplified to the simplified tTS. It maps the triangle and 0 vertex weight, which are part of both tTS, onto themselves. The 1 vertex weight is mapped to the following,



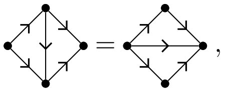
(4.98)

In order to show that this defines indeed a TS mapping, we need to show that the (mapped) non-simplified moves can be derived from the simplified moves. We first notice that one of the 1-3 Pachner moves of the simplified tTS is derived from the 2-2 Pachner move in Eq. (4.12) and the triangle cancellation move in Eq. (4.88),



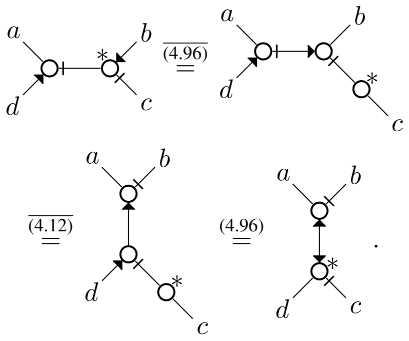
(4.99)

Next we notice that all other versions of branching structure 2-2 Pachner moves are derived from the 2-2 Pachner move in Eq. (4.12), together with the triangle flip and 2-gon cancellation moves. For example, the following 2-2 Pachner move,



(4.100)

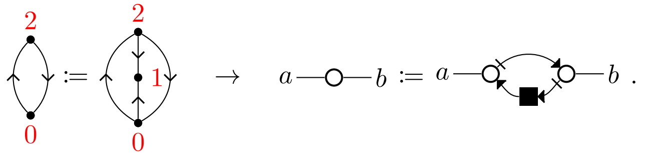
is derived by



(4.101)

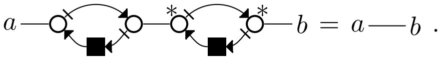
The bar over the referenced equation denotes that the move is applied from right to left. In general, we can flip an internal edge of a triangulation by first applying the reverse 2-gon cancellation move, and then a triangle flip move for each of the two 2-gons and the adjacent triangle. A boundary edge can be flipped on both sides of a move by gluing a 2-gon to this edge on both sides and then performing a triangle flip move with the adjacent triangle. So we see that we can derive all 2-2 Pachner moves from the one in Eq. (4.12), and all other 1-3 Pachner moves are from the one in Eq. (4.99). Also the weight commutation moves involving the 1 vertex weight can be derived, namely from the one in Eq. (4.92) together with the triangle flip moves.

Let us now consider the reverse mapping $\mathcal{M}_2$ from the simplified to the non-simplified tTS. The triangle and 0 vertex weight of the simplified tTS are also part of the non-simplified tTS, and mapped onto themselves. The cyclic 2-gon cell can be triangulated by two triangles, and is mapped accordingly,



(4.102)

In order to show that this is indeed a tTS mapping, we need to find derivations for the mapped simplified moves using the non-simplified moves. For example, if we plug the mapping in Eq. (4.102) into the 2-gon cancellation move in Eq. (4.94), we obtain

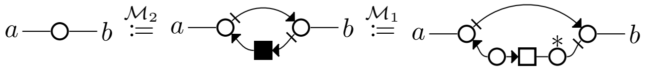


(4.103)

This can be derived by 1) a 2-2 Pachner move, 2) a 1-3 Pachner move, and 3) the triangle cancellation move. We will not explicitly give derivations for each mapped move here. Instead, we would like to remark that the these moves correspond to re-triangulations of a disk. It is known that any two triangulations of the same (piece-wise linear) manifold are related by a sequence of Pachner moves [116]. So, if we rely on this statement about the geometric interpretation, we know that derivations for all mapped moves must exist.

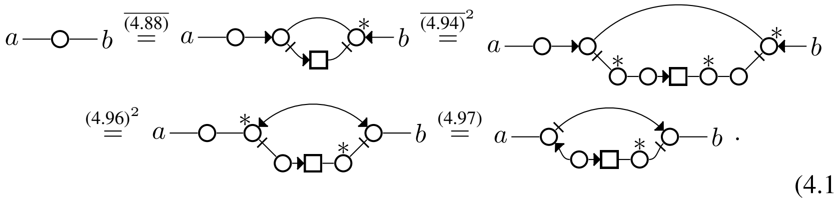
We have found two tTS mappings going from the non-simplified tTS to the simplified tTS and back. Formally, just having to tTS mappings $\mathcal{M}_1$ and $\mathcal{M}_2$ is not enough to show equivalence. For example, for any two tTS, there exists a *trivial mapping* that maps every bond dimension variable to the empty collection of bond dimension variables, and every tensor variable to the empty tensor-network diagram. The reason why $\mathcal{M}_1$ and $\mathcal{M}_2$ show the equivalence of the two tTS is that these mappings are just refinements of the cellulation/triangulation without changing the topology. Formally, this can be formulated as follows. If we apply $\mathcal{M}_2 \circ \mathcal{M}_1$ to any network of the non-simplified tTS, the resulting network should be equivalent to the original one via the moves of the non-simplified tTS. The analogous must hold for $\mathcal{M}_1 \circ \mathcal{M}_2$. If this is the case, we say that $\mathcal{M}_1$ and $\mathcal{M}_2$ are *weak inverses* to each other, and this is indeed the case here. A direct consequence of the existence of two weakly inverse tTS mappings is that applying both mappings to a model yields a model in the same phase. So if two tTS are equivalent via weakly inverse mappings, then their phases are in one-to-one correspondence. By a phase here we mean an equivalence class under invertible domain walls as defined in Section 3.7. We will specialize this notion to the context of tTS models later in Section 5.4.

As an example, let us look at $\mathcal{M}_1 \circ \mathcal{M}_2$. The triangle is mapped again to the triangle. The cyclic 2-gon is mapped to the following diagram,



(4.104)

Indeed, we find that the left and right are related via moves of the simplified tTS,



(4.105)

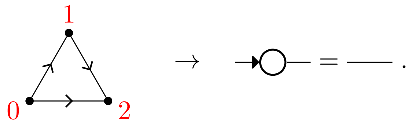
In our specific case, the equivalence between of networks before and after applying $\mathcal{M}_2 \circ \mathcal{M}_1$ or $\mathcal{M}_1 \circ \mathcal{M}_2$ even holds on the level of individual tensors. This implies that the models before and after the double mappings are not only in the same phase, but actually the same model. This will not be the case in general for weakly inverse tTS mappings.

One might think that reducing the number of moves from 8 (including the triangle cancellation move) of the non-simplified tTS to 5 moves of the simplified tTS is not a significant improvement. Let us justify why it actually is. The key task is finding models for our tTS, which means solving the tensor-network equations given by the moves. As a measure of "complexity" of a tTS it thus makes sense to consider the computational cost of evaluating the two networks of each move, and in particular its scaling with the index dimension d. This scaling is always polynomial, but the exponents depend on the move. Very roughly, the exponent will increase proportionally to the "linear size" of a network. Thus, we have a strong preference for moves with small tensor-network diagrams. For evaluating a 2-2 Pachner move we need of the order of d^5 $+$ and $\cdot$ operations. The same holds for a 1-3 Pachner move. All other moves in this section have smaller exponents and thus have a vanishing contribution to the overall complexity when scaling d. So from that perspective we have reduced the complexity from 7 moves to 1 move rather than from 8 to 5 moves.

4.8 Generators and relations in $2+1$D

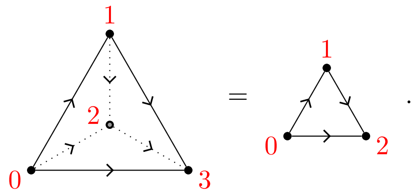
In this section, we will generalize the discussion in Section 4.7 from $1+1$ to $2+1$ dimensions. That is, we will find a simpler equivalent form of the 3D tTS, which we again refer to as the simplified 3D tTS. This simplified tTS will only contain a single Pachner move instead of 10 different 2-3 and 5 different 1-4 moves. Again, there will be new "auxiliary" tensor variables corresponding to different 3-cells other than the tetrahedron. To start with, the simplified tTS contains only the tetrahedron tensor variable, the 01 and 12 edge weights, and one single 2-3 Pachner move. In principle, we could pick any branching structure for this Pachner move, but let us choose the one shown in Eq. (4.19), because it has the neat property that all involved tetrahedra are positively oriented. In the following we introduce the additional tensors and moves of the simplified 3D tTS.

The first new tensor variable, which we will only assign temporarily, corresponds to a pillow-like volume (just like the volume we imagined for the edge weights, but without an edge weight),



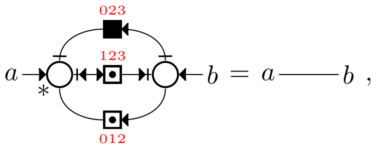
(4.106)

We realize that this tensor variable obeys moves making it a Hermitian projector, which applied to any index of the tetrahedron tensor will leave the latter invariant, analogous to Eq. (4.84). Thus we impose the full support convention that this tensor equals the identity tensor. Realizing that the pillow-like volume can be cellulated with two tetrahedra, we obtain the *tetrahedron cancellation move*,



(4.107)

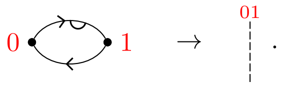
On the left-hand side we have a positively oriented tetrahedron in the front and a negatively oriented one in the back, glued at three of their faces, yielding a pillow volume on the right. To turn this into a tensor network diagram, we have to add edge weights for all three interior edges on the left. One possibility is



(4.108)

where a corresponds to the back and b to the front triangle in Eq. (4.107). For a fair comparison of the simplified and non-simplified tTS, we consider this move as part of both.

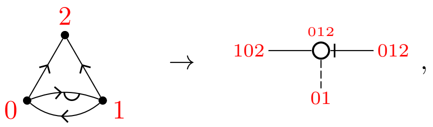
In contrast to the simplified 2D tTS, the simplified 3D tTS does not only contain new tensor variables and moves, but also new bond dimension variables. Since index contractions correspond to faces in the 3-dimensional cellulations, these new bond dimension variables correspond to new types of 2-cells. Specifically, there is one new bond dimension variable corresponding to the cyclic 2-gon, which we represent by dashed lines,



(4.109)

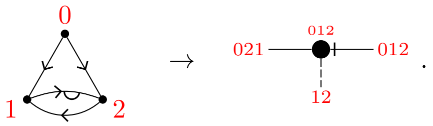
The 2-gon has a rotation symmetry, so there are 2 different ways to identify two glued 2-gons. In order to make the gluing unambiguous, we choose one "favorite edge" marked by the small half circle, and demand that the favorite edges coincide when gluing.

Let us now introduce the new tensor variables. First, there are two tensor variables called *triangle flippers*, corresponding to 3-cells whose boundary consists of two triangles and one cyclic 2-gon. The 01 *triangle flipper*,



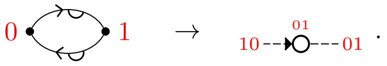
(4.110)

and the 12 *triangle flipper*,



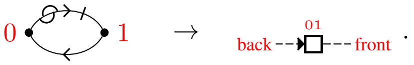
(4.111)

In addition, there is the 2-*gon flipper*,



(4.112)

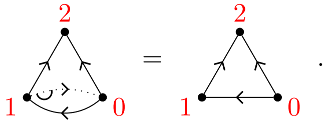
This 3-cell on the left looks like a pillow with two corners, and its boundary consists of two cyclic 2-gons. The favorite edge of the 2-gon in the front is the 01 edge, whereas for the 2-gon at the back it is the 10 edge. At last, there is one new edge weight, namely the 2-*gon edge weight*,



(4.113)

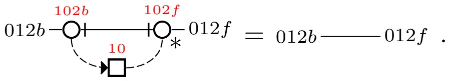
The volume looks the same as for the 2-gon flip, but the favorite edge of both the back and front 2-gon is the 01 edge. According to the name, one of its edges (the 01 edge) carries an edge weight and is therefore marked by a tick. The shape used for the tensor is the same as for one of the triangle edge weights, but it can be distinguished by the line style of its indices.

Next, let us discuss the moves of the simplified 3D tTS. These can be divided into three groups, cancellation moves, flip moves, and weight moves. Let us start with the cancellation moves. We have already introduced the tetrahedron cancellation move in Eq. (4.107). Additionally, we have the 01 *triangle flipper cancellation move*,



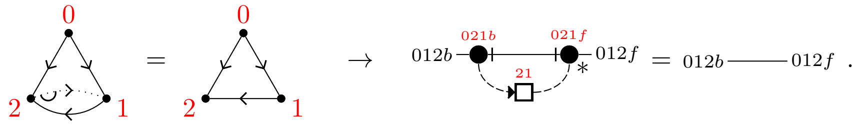
(4.114)

It consists of two 01 triangle flippers on the left-hand side glued by one triangle and the 2-gon, and one pillow-like volume on the right-hand side. Again, the pillow-like volume is interpreted as a free bond in network notation,



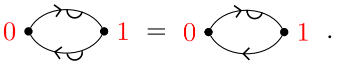
(4.115)

Analogously we have the 12 *triangle flipper cancellation move*,



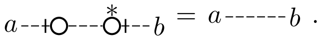
(4.116)

Last, there is the 2-*gon flip cancellation move*,



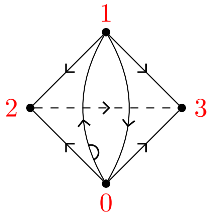
(4.117)

It has two 2-gon flippers on the left-hand side, and a 2-gon pillow on the right where the favorite edge is the same at the back and front. By the full support convention we set this right-hand side to the identity matrix and thus have



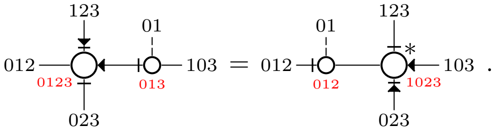
(4.118)

Next, let us introduce the flip moves. The 01 *tetrahedron flip move*,



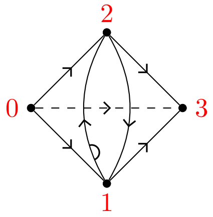
$=$ same 1-skeleton , (4.119)

flips the 01 edge of a tetrahedron. On the left-hand side, we glue a 01 triangle flipper to the 012 triangle of a tetrahedron. On the right-hand side, we glue a 01 triangle flipper to the 013 triangle of a tetrahedron. The two tetrahedra on the left and right-hand side are the same apart from that the direction of the 01 edge is flipped. Both sides result in a volume where the 01 edge of the tetrahedron is doubled to a 2-gon. The geometric drawings for both sides are equal since we only draw vertices and edges but not faces or volumes. The network notation on the other hand completely unambiguous, but does not make the geometric interpretation as apparent,



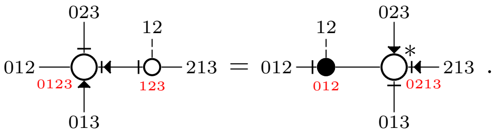
(4.120)

Note that flipping the 01 edge changes the orientation of the tetrahedron from positive to negative, so the tetrahedron tensor on the right-hand side is complex conjugated. Analogously, the 12 *tetrahedron flip move* is given by



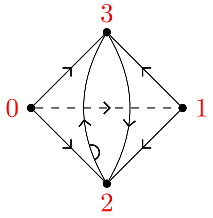
= same 1-skeleton . (4.121)

In tensor-network notation, we have



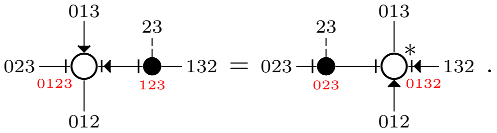
(4.122)

Finally, the 23 *tetrahedron flip move* is given by



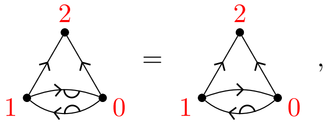
= same 1-skeleton . (4.123)

In tensor-network notation, we have



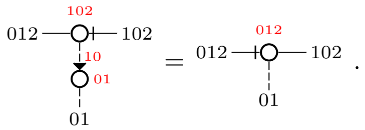
(4.124)

Next, there are moves that flip the favorite edge of the 2-gon of the triangle flippers. Such a flip results in the same triangle flipper which is rotated by π around the axis perpendicular to the 2-gon (if we imagine a geometric realization of the triangle flipper that is symmetric enough). The 01 *triangle flipper flip move*,



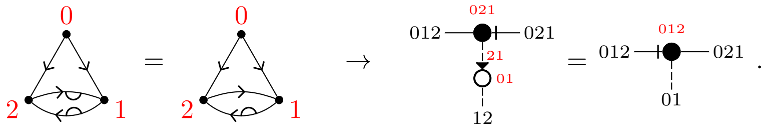
(4.125)

has a 01 triangle flipper on each side, and an additional 2-gon flipper glued to its 2-gon on the left-hand side. In tensor-network notation, this looks like



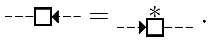
(4.126)

Analogously, there is the 12 *triangle flipper flip move*,



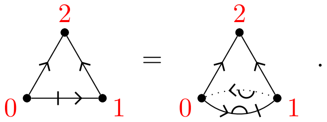
(4.127)

Last, let us consider the weight commutation moves. The first move is the Hermiticity of the 2-gon edge weight,



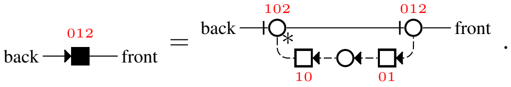
(4.128)

The remaining two moves simply express the 01 and 12 edge weights in terms of the 2-gon edge weight. The 01 edge weight can be obtained by



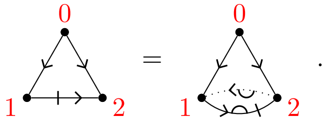
(4.129)

In network notation, this is



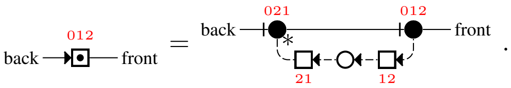
(4.130)

The 12 edge weight is obtained by



(4.131)

In network notation, this is

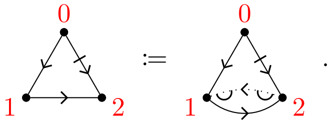


(4.132)

So, all in all, the simplified tTS has two bond dimension variables, corresponding the the triangle and to the cyclic 2-gon in Eq. (4.109). The tensor variables are given in Eqs. (4.17), (4.63), (4.65), (4.110), (4.111), (4.112), and (4.113). The moves are given in Eqs. (4.19), (4.108), (4.115), (4.116), (4.118), (4.120), (4.122), (4.124), (4.126), (4.127), (4.128), (4.130), and (4.132).

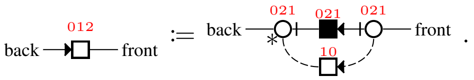
Let us now show that the simplified and non-simplified tTS are equivalent. Like in Section 4.7, we will do this via two weakly inverse TS mappings $\mathcal{M}_1$ from the non-simplified tTS to the simplified tTS and $\mathcal{M}_2$ vice versa.

The more interesting mapping is $\mathcal{M}_1$ from the non-simplified to the simplified tTS, since this shows that the simpler moves can be used to generate all Pachner moves with all choices of edge directions. This mapping is as follows. The tetrahedron, 01, and 12 edge weights are part of both tTS, and mapped to themselves. The 02 edge weight volume can be triangulated as



(4.133)

On the right-hand side, we have a 01 edge weight sandwiched with two 12 triangle flippers. In terms of diagrams, this mapping becomes,



(4.134)

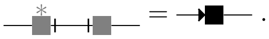
In order to show that this is indeed a TS mapping, we need to show that the mapped moves of the non-simplified tTS are derived by the moves of the simplified tTS. We first notice that the tetrahedron cancellation move in Eq. (4.108) can be used to bring one of the tetrahedra on the left hand side of the 2-3 Pachner move in Eq. (4.19) over to the right hand side, and obtain a 1-4 Pachner move. Next, let us argue that we can reverse the direction of an interior edge e in any triangulation using the tetrahedron flip moves. To this end, we first insert a pair of triangle flippers at a triangle adjacent to e, using one of the triangle flipper cancellation moves in Eqs. (4.115), or (4.116) backwards. Then, using the tetrahedron flip moves in Eqs. (4.120), (4.122), or (4.124), the second of the two triangle flippers can be moved from one triangle to the next triangle adjacent to e. To correctly align the 2-gon favorite edge in these moves, we might need to apply Eqs. (4.126) or (4.127) along the way. We now move the second triangle flipper once around all the triangles adjacent to e, such that it re-appears at the other side of the first triangle flipper. Finally, we apply one of Eqs. (4.115) or (4.116) to remove the two triangle flippers. After this procedure, the direction of the interior edge e is flipped.

Next, we can also flip the direction for edges e that are in the boundary on both sides of a move. To this end, we extend each side by two triangle flippers glued at their 2-gon faces, and glued to the two boundary triangles adjacent to e. Then, on each side, we move the second triangle flipper between the different (interior) triangles adjacent to e, as described in the previous paragraph. We do this until the second triangle flipper is at the same boundary triangle as the first, and then apply Eqs. (4.115) or (4.116) to remove the pair of triangle flippers on both sides. Again, after this, the direction of the boundary edge e will be flipped on both sides.

Equipped with the procedures to flip interior and boundary edge directions described in the two proceeding paragraphs, we can derive all 2-3 Pachner moves and 1-4

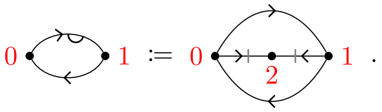
Pachner moves with all different edge directions. Note that also all the weight commutation moves such as in Eq. (4.66) follow from the triangle flip moves and similar, after we plug in the mapping in Eq. (4.134), and the expressions of the 01, and 12 edge weights in terms of the 2-gon edge weight via Eqs. (4.130) or (4.132).

Next, we consider the converse mapping $\mathcal{M}_2$ from the simplified tTS to the non-simplified tTS. The existence of this mapping is perhaps less surprising, since we know that any cellulation can be refined into a triangulation, and any retriangulation can be written as a sequence of Pachner moves. To formally have a purely diagrammatic mapping, we actually need to change our tTS by postulating the existence of a square root S of every weight matrix M, that is, $SS^\dagger = M$. We will denote these square root tensors like the weight tensors but in gray. For example, for the 01 edge weight, we introduce



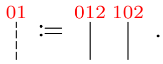
(4.135)

Note that for ordinary complex tensors, these square roots always exist and are unique up to isomorphism. This would not be the case if all tensors were restricted to real tensors and the edge weight had negative eigenvalues, which shows that it is not a property that is guaranteed on a purely diagrammatic level. Equipped with these square root tensors, we can write down the mapping, which involves finding triangulations for all the extra faces and volumes of the simplified tTS. A 2-gon can be triangulated by a pair of triangles, and gluing two 2-gons can be replaced by gluing two triangle pairs instead,



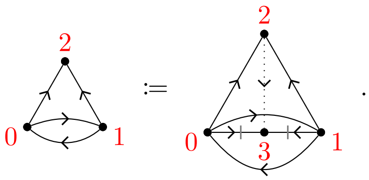
(4.136)

In other words, every time we would glue two 2-gons of the simplified tTS, we now glue two triangle pairs of the non-simplified tTS instead. As indicated by the gray ticks, we include square root weights for the two interior edges. Then, when gluing two 2-gons, the two square root edge weights combine to a proper edge weight for each interior edge. Formally, we make the following identification between bond dimension variables,



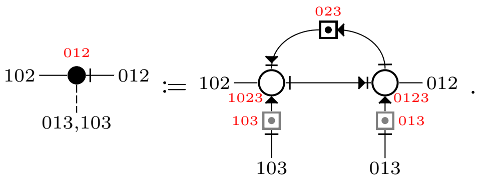
(4.137)

Next, we consider the mapping of the additional tensors. The 01 triangle flipper can be triangulated by two tetrahedra,



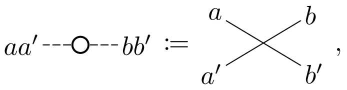
(4.138)

In terms of networks, we have



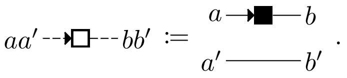
(4.139)

The mapping of the 12 flipper is defined analogously. Moreover, the 2-gon flipper is mapped to two open bonds,



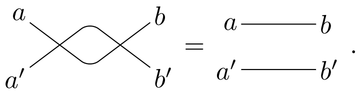
(4.140)

and the 2-gon edge weight can be emulated by an edge weight for one of the two triangles



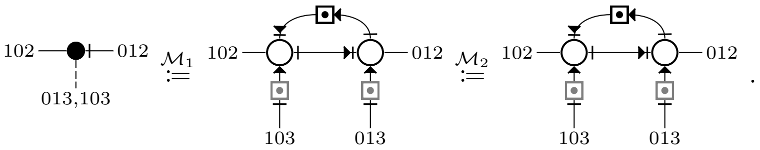
(4.141)

All of the simplified moves correspond to re-triangulations, so they can be derived by the Pachner moves of the non-simplified tTS. A technical exception to this are moves involving the 2-gon flipper cancellation move and triangle flipper flip move, for which it is easy to find derivations. For example, the 2-gon flipper cancellation move in Eq. (4.118) simply becomes



(4.142)

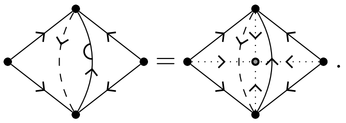
Finally, we have to show that $\mathcal{M}_1$ and $\mathcal{M}_2$ are weak inverses to each other. Intuitively, this is clear since the mappings were defined as refinements of triangulations or cellulations without changing the topology. Formally, applying both mappings to a network should result in a network that is equivalent under locally applying moves. Applying $\mathcal{M}_2 \circ \mathcal{M}_1$ to the 2-gon bond dimension variable of the simplified tTS yields twice the triangle bond dimensions. Thus, applying $\mathcal{M}_2 \circ \mathcal{M}_1$ to, for example, the 01 triangle flipper cannot yield an equivalent network, since resulting network has two open indices where the 01 triangle flipper had one,



(4.143)

So the diagram on the left and on the right cannot be equivalent under moves. To show that the resulting network is still equivalent to the original, we apply the mapping not to an individual triangle flipper, but to a pair of triangle flippers adjacent at their

2-gon faces. Applied to such a configuration, the mapping indeed corresponds to a retriangulation with equal boundary on the left and right-hand side,

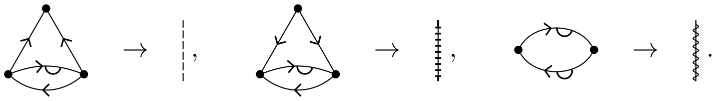


(4.144)

Thus, this equation can be derived from the moves of the simplified tTS. We see that even though applying $\mathcal{M}_2 \circ \mathcal{M}_1$ to a model yields a different model, the two models are in the same phase.

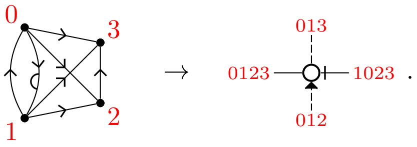
4.9 Generators and relations in $3+1$D

In this section, we briefly sketch how to obtain a simplified tTS in $3+1$ dimensions, without spelling out all the details, and ignoring weight matrices. We start with only the 3-3 Pachner move depicted in Eq. (4.25), and then introduce additional bond dimensions, tensors, and moves, which have geometric interpretations in terms of more general 3-cells, 4-cells, and 4-dimensional recellulations. The 3-cells for the additional bond dimension variables are just the 3-cells of the tensor variables of the $2+1$-dimensional simplified tTS. That is, in addition to the bond dimension variable for the tetrahedron, there is one variable associated to the 01 triangle flipper, 12 triangle flipper, and 2-gon flip,



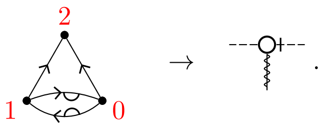
(4.145)

Next, the 4-cells of the new tensor variables can be obtained by taking the recellulations corresponding to the moves of the simplified 3D tTS, and gluing together the left and right-hand side. The 01 tetrahedron flip move yields a new tensor variable that we call the 01 *tetrahedron flipper*,



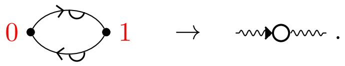
(4.146)

The left-hand side shows a cellulation of the 3-sphere consisting of two tetrahedra and two 01 triangle flipper volumes. Analogously, there is a 12 and 23 tetrahedron flipper. The 01 triangle flipper flip move yields a new tensor variable called the 01 *triangle flipper flipper*,



(4.147)

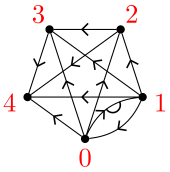
Here, the left-hand side denotes a cellulation of a 3-sphere consisting of two 01 triangle flipper volumes and one 2-gon flipper volume. Analogously, there is a 12 triangle flipper flipper. Last, there is the following *2-gon flipper flipper*,



(4.148)

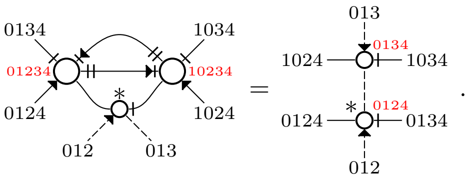
The volume on the left is a cellulation of the 3-sphere consisting of two 2-gon flipper volumes.

As usual, the moves can be divided into cancellation moves, flip moves, and weight moves. In analogy to the triangle cancellation move and tetrahedron cancellation move, we can define a *4-simplex cancellation move* implementing the full support convention. However, it turns out that we also need a more powerful move, namely the *partial 4-simplex cancellation move*,



$= \text{same 1-skeleton}$. (4.149)

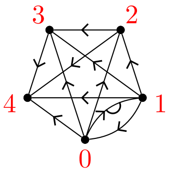
The left-hand side consists of two 4-simplices that are glued not at 4, but at 3 of their tetrahedra. At one of these tetrahedron pairs, we insert a 01 triangle flip flipper volume. On the right-hand side, we have two 01 triangle flip flippers that are glued at one of their triangle flipper volumes. The boundary on both sides is a cellulation of the 3-sphere consisting of four tetrahedra and two triangle flipper volumes. In terms of diagrams, this becomes



(4.150)

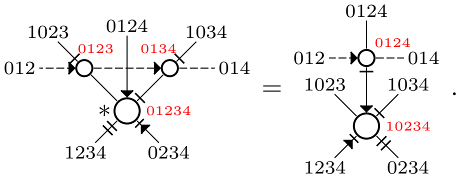
There are analogous cancellation moves for tetrahedron flippers, triangle flipper flippers, and 2-gon flipper flippers, which we do not spell out here.

Next, let us consider one example of a flip move, namely the 01 4-simplex flip move,



$= \text{same 1-skeleton}$. (4.151)

The picture is the same as for the partial 4-simplex cancellation move, but now there is one 4-simplex on each side. In addition, there are two 01 tetrahedron flippers on the left-hand side, and one on the right-hand side. In network notation, we have



(4.152)

Analogously, one can define a 12, 23, and 34 4-simplex flip move. In addition to this, there are also moves related to the symmetries of the tetrahedron flippers, and triangle flipper flippers, which we will not spell out explicitly here.

Note that we do not simply guess the bond dimensions variables, tensor variables, and moves from scratch, but follow some systematics, which we will outline briefly. In general, the bond dimension variables of the nD tTS correspond to different $n-1$-cell representatives, and the tensor variables to different n-cell representatives. The moves are equations between two different n-cellulations of the n-ball, and if we glue both sides of a move together, we get a cellulation of an n-sphere, which can be seen as the boundary of a $n+1$-cell representative. Now, the n-cells corresponding to the tensor variables of the nD tTS are the same as the n-cells corresponding to the moves of the $n-1$D tTS, and the same as the n-cells corresponding to the bond dimension variables of the $n+1$D tTS. For example, the 2-gon 2-cell corresponds to a tensor variable of the 2D tTS in Eq. (4.89) as well as a bond dimension variable of the 3D tTS in Eq. (4.109). Or, the 01 tetrahedron flipper 3-cell corresponds to a bond dimension variable of the 4D tTS in Eq. (4.145), a tensor variable of the 3D tTS in Eq. (4.110), and to the 01 triangle flip move of the 2D tTS in Eq. (4.97). Or, the 4-simplex corresponds to a tensor variable of the 4D tTS in Eq. (4.23) as well as to the 2-3 Pachner move of the 3D tTS in Eq. (4.18).

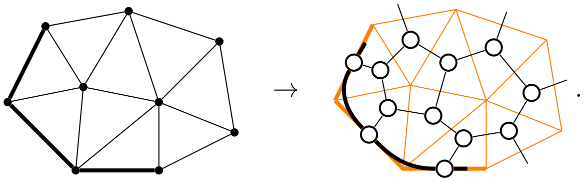
We still need to actually construct the n-cell representatives corresponding to the bond dimension variables, tensor variables, and moves at the same time. The n-cells can be constructed inductively from the $n-1$ cells as follows. First, the n-cells contain a *flipper n-cell*. The boundary of the flipper n-cell consists of two copies of the flipper $n-1$-cell, glued in a rotated way: The flipper 1-cell is a single edge, the flipper 2-cell is the cyclic 2-gon, the flipper 3-cell is the 2-gon flipper, and so on. Second, the n-cells contain the *cone* of every $n-1$-cell. The cone of an $n-1$-cell C is the n-cell whose boundary consists of C, as well as the span of the boundary of C and a *central vertex*. In other words, the n-cell itself is the span of C and the *central vertex*. Note that there are two different choices of branchings structure/edge directions for the cone: Either all new edges point towards the central vertex, or away from it. Both versions of the cone are contained in the n-cells.

For example, n-simplex is the cone of the $n-1$-simplex. Or, the 01 and 12 tetrahedron flipper 3-cells are the two different cones of the cyclic 2-gon 2-cell. Or, the 01 4-simplex flipper 4-cell is the cone of the 01 tetrahedron flipper 3-cell.

Note that the $n+1$-cells do not correspond to a single move, but to many moves depending on how we cut the boundary into two n-cellulations of an n-ball. For example, both the 01 4-simplex flip move as well as the partial 4-simplex cancellation move yield the same 5-cell. We do *not* include all these decompositions into the simplified tTS. For example, different decompositions of the 5-simplex yield all different variants of the 4-dimensional Pachner moves, but we only take a single 3-3 Pachner move.

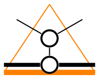
4.10 Boundaries in $1+1$D

In this section and the following, we will anticipate the simple $1+1$-dimensional case of tTS for physical boundaries and domain walls, which is discussed in full detail in Chapter 5. We will do this because it is essential for so-called block-diagonalization of tTS models, which we discuss in Sections 4.13 and 4.14. Let us for a moment neglect technical details such as branching structures, weight matrices, and orientations. Like topological phases in the bulk, topological phases (or superselection sectors) of the boundary will be described via fixed-point models. These fixed-point models are again models of a TS imposing whose moves that correspond combinatorial topological invariance. We will refer to this topological TS as *2D boundary tTS*. The 2D boundary TS assigns tensor networks to 2-dimensional triangulations with boundary, for example,

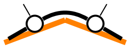

(4.153)

Here, we draw boundary edges of the triangulation with thicker lines. We see that the 2D boundary tTS contains one additional bond dimension variable, of which there are indices contracted between the tensors along the boundary. This new bond dimension variable will also be drawn with thicker lines. Also, there is one additional tensor variable, a copy of which is associated to each boundary edge. We use the same symbol for the bulk and boundary tensor variables, as they can be distinguished by the line width of the indices. Note that the boundaries under consideration are *physical boundaries*. In general, a physical boundary terminates the tensor network with special tensors such that there are no open indices. In contrast to this are *space boundaries* where we merely cut off a tensor network resulting in open indices, corresponding to the non-thick boundary edges in Eq. (4.153).

Next, we need to add moves to the 2D boundary tTS. These moves should be powerful enough such that any two triangulations with boundary can be transformed into each other via a sequence of moves if the underlying manifolds have the same topology. Note that the 2D boundary tTS contains the bulk 2D tTS as a sub-tTS, in the same way as triangulations with boundary contain triangulations without boundary. So not only the triangle tensor, but also the bulk Pachner moves are already part of the tTS. Applying the bulk Pachner moves near the boundary can already be used to “deform” the boundary to a great deal. However, no bulk Pachner move can change the cellulation of the boundary itself, which would involve adding or removing a boundary edge. To this end, we add Pachner moves for the boundary triangulation itself. In our case, there is only one such Pachner move, namely the 1-2 Pachner move acting on the boundary 1-cellulation. Since the boundary is not standalone but attached to the bulk, we need to "pad" one side appropriately,



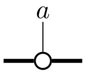
$$=$$



$$. \qquad (4.154)$$

The shown move has one boundary edge on the left and two on the right, and thus represents a 1-2 Pachner move for the boundary cellulation. We need to pad the left-hand side with a bulk triangle, such that on both sides there are two edges that attach to the same bulk. Coincidentally, this move can also be thought of as simply attaching/removing a triangle to/from the boundary.

We can already give an example for a model of the 2D boundary tTS. We take the δ-tensor model of bond dimension d in Eq. (4.4) as 2D-tTS model in the bulk. For every $0 \leq x < d$, there is a boundary with boundary-bond dimension 1 and boundary tensor given by



$$= \delta_{a,x} \ . \qquad (4.155)$$

Since the boundary indices have bond dimension 1, this tensor is just a vector. With this, Eq. (4.154) becomes

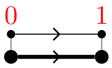
$$\delta_{a,b,c}\delta_{c,x} = \delta_{a,x}\delta_{b,x} \ . \qquad (4.156)$$

If we would impose a symmetry making the δ-tensor model a fixed-point model for a a symmetry-breaking phase, then this is a boundary projecting onto the xth symmetry-broken sector.

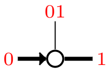
Let us now add in all the technical details. First of all, we need to distinguish the two boundary indices. To this end, we choose a direction for each boundary edge, and mark one index with an arrow aligned with this direction. Second, we distinguish between positively and negatively oriented boundary edges. Positively oriented boundary edges are directed such that they rotate counter-clockwise around the bulk manifold. Third, to make the formalism consistent in particular when we later consider lower-dimensional defects, we realize that triangulations with boundary are equivalent to a type of extended cellulations, c.f. Section 2.4. The type consists of two regions, a 2-region a with empty link (the bulk), and a 1-region b whose link is a point (the boundary),

$$a : (2, \varnothing) \ , \quad b : (1, \bullet) \ . \qquad (4.157)$$

A standard b 1-cell representative is given by a bulk edge with two boundary vertices. Each boundary edge is adjacent to the bulk via this 1-cell representative, and associated a 3-index tensor variable,

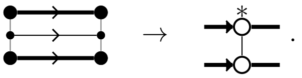


$$\rightarrow$$



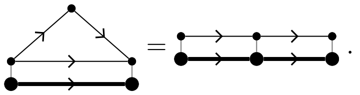
$$. \qquad (4.158)$$

On the left, the boundary edge has been drawn in thick, and the adjacent bulk edge with normal line width. The two thick indices on the right correspond to the boundary vertices, and the third index to the bulk edge. The direction of the boundary edge is aligned with that of the bulk edge. More intuitively, one may imagine the boundary edges as rectangle-shaped 2-cells. Note that this is not only a more complicated way of drawing manifolds with boundary. We can now consider triangulations where two boundary edges are directly adjacent to another via a bulk edge, but with no bulk triangles in between,



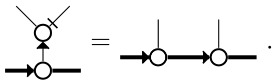
(4.159)

In terms of extended cellulations, the boundary Pachner move of Eq. (4.154) looks like



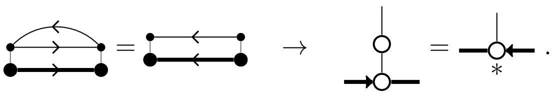
(4.160)

After adding the corresponding markings to the tensor-network diagrams, Eq. (4.154) becomes



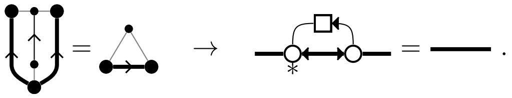
(4.161)

More precisely, we need to impose this move for all choices of edge directions. However, it turns out that it suffices to take the above boundary Pachner move together with the following *boundary edge flip move*,



(4.162)

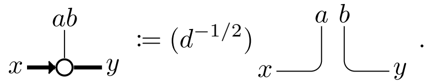
Further, it makes sense to impose the full support convention via the move



(4.163)

The geometric picture on the left shows a new boundary 1-cell representative, consisting of a boundary edge adjacent to a single bulk vertex. The according tensor is the identity matrix due to the full support convention. Since there is an internal bulk vertex on the left, we need to include a vertex weight into the diagram on the right.

After adding the technical details, we are able to give one more example of a boundary model. Namely, consider the triple-identity-matrix model defined in Eq. (4.15). A boundary for it is given by



(4.164)

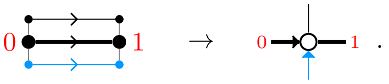
It can be easily seen that plugging this and Eq. (4.15) into Eq. (4.161) yields the same diagram of identity strings on both sides.

4.11 Domain walls in $1+1$D

In this section, we continue anticipating a new type of tTS that belongs to Chapter 5, but which we need for the block-diagonalization in Section 4.14. Namely, we will discuss domain walls between two distinct 2D tTS models that we will label by a and b. Formally, we use extended cellulations of the following extended-manifold type,

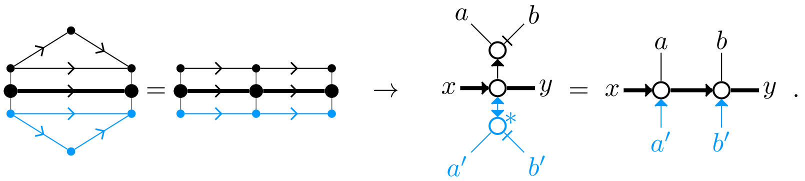
$$a:(2,\varnothing)\,,\quad b:(2,\varnothing)\,,\quad c:(1,\bullet\quad\bullet)\,. \tag{4.165}$$

Extended manifolds of this type (in the collapsed picture) are 2-manifolds separated into a black bulk region (a) and a blue bulk region (b), separated by a domain wall line (c). The standard 1-cell representative for a domain wall edge consists of one a and one b bulk edge, and two domain-wall vertices. The *2D domain-wall tTS* associates a new tensor variable to this domain-wall edge,



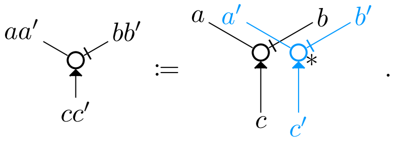
(4.166)

There is one main move, namely the 1-2 Pachner move of the domain-wall 1-cellulation, suitably padded with bulk triangles on one side,



(4.167)

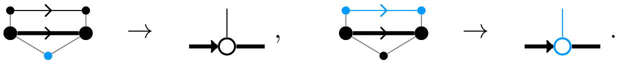
To obtain a simple set of generators and relations, we can add moves analogous to Eq. (4.162) and Eq. (4.163) in the case of boundaries. Now, consider the following tTS mapping from the 2D tTS to two copies (a and b) of the 2D tTS, by simply stacking a and b on top of each other,



(4.168)

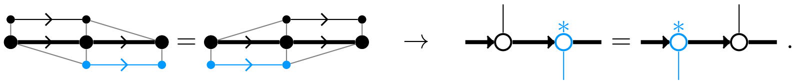
More precisely, the b copy is complex conjugated. It is easy to see that boundaries of this stacked tTS model are identical to domain walls as presented above.

There is an equivalent "simplified" domain wall tTS, which allows us to express models more compactly. In particular, the simplified tTS will make it much easier to define invertibility of a domain wall later in Section 5.4. The simplified domain-wall tTS splits the tensor variable in Eq. (4.166) into two tensor variables. The 1-cell representatives of the simplified tensor variables have a bulk edge on one side, but only a bulk vertex on the other side,



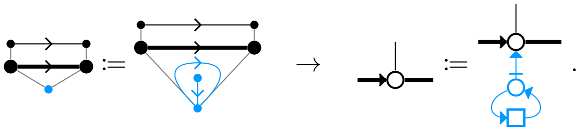
(4.169)

The tensor with the a bulk index has to fulfill all the moves of the 2D boundary tTS with respect to the a 2D tTS. The same holds for the b boundary, just that the b 2D tTS is complex conjugated. In addition, the two boundaries "commute" in the following way:



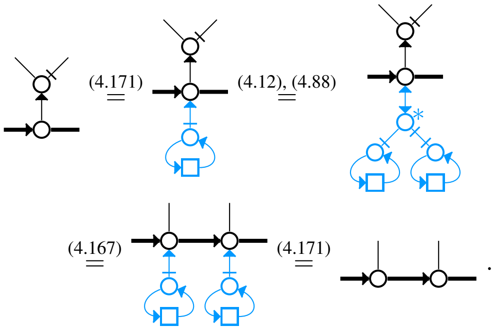
(4.170)

Let us quickly sketch how this simplified 2D domain-wall tTS is equivalent to the original "non-simplified" 2D domain-wall tTS. As usual, we do this by giving a TS mapping from the simplified to the non-simplified tTS, and back, and then show that these mappings are weakly inverse to another. The mapping form simplified to the non-simplified tTS is given by



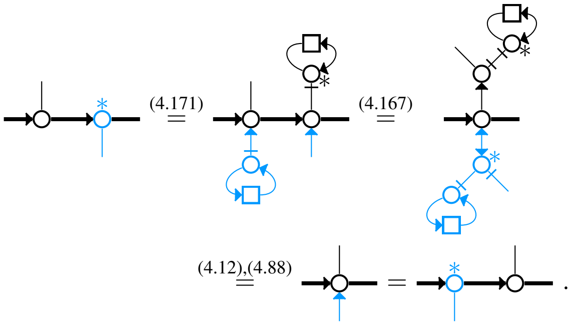
(4.171)

On the left-hand side we showed how after gluing a 1-gon (triangulated by a triangle) to the b bulk edge, only a b bulk vertex remains. The mapping of the second tensor variable in Eq. (4.169) is analogous with a and b exchanged and complex conjugated. Let us quickly sketch how the moves of the simplified tTS can be derived from those of the non-simplified tTS. The boundary Pachner move in Eq. (4.167) is derived by



(4.172)

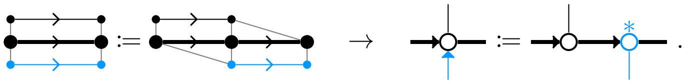
The commutativity in Eq. (4.170) can be derived by



(4.173)

The steps from the second-to-last to the last diagram are the analogous to those from the second-to-last to the first diagram, with a and b exchanged.

The reverse mapping from the non-simplified to the simplified tTS is given by



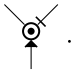
(4.174)

The boundary Pachner move in Eq. (4.167) can be derived from the boundary Pachner moves of the a and b boundary, together with the commutativity move in Eq. (4.170). The two mappings are weakly inverse since they map between different extended cellulations of the same 2-manifolds with domain walls.

4.12 Special Frobenius and *-algebras, and representations

TS models are solutions to tensor-network equations, which are multi-linear equations for tensor-valued variables. Basically all linear-algebraic structures are examples of TS models as well. The tTS arising from the classification of topological phases turn out to have great similarity to well-known algebraic TS. In this section, we will discuss such algebraic TS for the 2D (boundary) tTS. Specifically, we find that the 2D (boundary) tTS is very similar to (*unitary representations* of) both *special Frobenius algebras* and *finite-dimensional *-algebras*. Let us thus start by first introducing algebraic structures as TS models.

An *algebra* is a linear map $\cdot : V \otimes V \to V$, where V is a vector space. A finite-dimensional algebra is represented by its structure coefficients, which form a 3-index tensor,

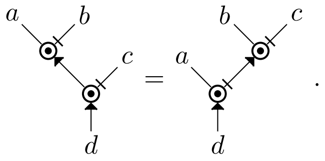


(4.175)

Here, we think of $\cdot$ as a linear map from the top two indices to the bottom index. An algebra is *associative*, if

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) \quad \forall a, b, c \in V\ . \tag{4.176}$$

This can be formulated as an equation between two tensor networks, namely

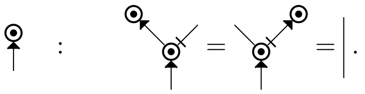


(4.177)

So, associativity defines a move, and associative algebras are models of the TS defined by that move. An algebra is *unital*, if there exists a unit vector 1 such that

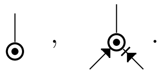
$$1 \cdot a = a \cdot 1 = a \quad \forall a \in V\ , \tag{4.178}$$

or diagrammatically,



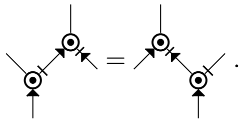
(4.179)

A *coalgebra* is the same as an algebra, just that we reflect all diagrams at the horizontal axis such that inputs are identified with outputs and vice versa. A *Frobenius algebra* consists of an algebra and a coalgebra, both unital. We will use the same symbol for both products and units, and just switch which indices are marked with arrows and which not (while preserving the ticks),



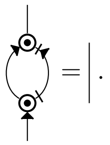
(4.180)

The algebra and coalgebra have to fulfill the following *Frobenius law*,



(4.181)

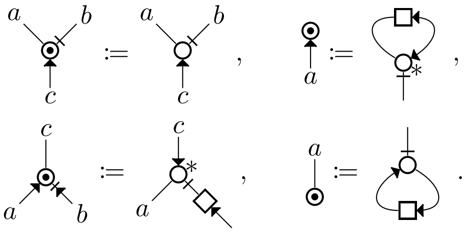
A Frobenius algebra is called *special* if it fulfills the following property,



(4.182)

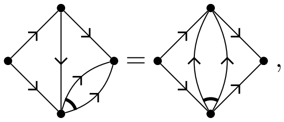
Note that in the literature, this property is sometimes written down up to a scalar prefactor.

Every model of the 2D tTS can be turned into a special Frobenius algebra by a TS mapping from the Frobenius-algebra TS to the 2D tTS:



(4.183)

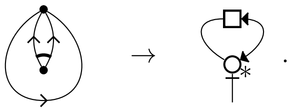
For this to be a mapping, we need to show that all the mapped special-Frobenius moves can be derived from the 2D-tTS moves. Indeed, this is the case. For example, associativity in Eq. (4.177) equals the 2-2 Pachner move in Eq. (4.12). The same holds for the associativity of the co-algebra, just that we have to also use the weight commutation move in Eq. (4.92). The Frobenius law in Eq. (4.181) becomes the recellulation



(4.184)

which can be derived from 1) the inverse 2-gon cancellation move, 2) two triangle flip moves, and 3) the weight commutation move in Eq. (4.92). The “special” property in

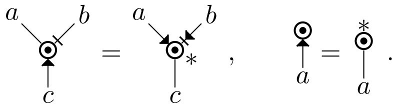
Eq. (4.182) maps precisely to the triangle cancellation move in Eq. (4.88). The diagram assigned to the unit (and co-unit) is the cellulation of a 1-gon,



(4.185)

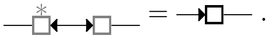
The unitality in Eq. (4.179) is thus a recellulation, and can be derived via Pachner moves, weight commutation moves, and cancellation moves.

In fact, we can equip our algebras with even one more restriction: A *†-Frobenius algebra* is one for which the comultiplication and counit is the complex transpose of the multiplication and unit,



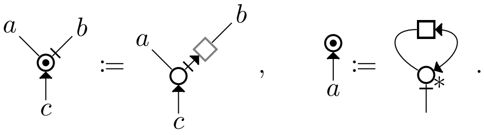
(4.186)

In order to construct a mapping from the 2D tTS to the special †-Frobenius algebra TS, we need to introduce the square root of the 0 vertex weight into our diagrammatic calculus, which we denote by a gray box,



(4.187)

We also demand that the weight commutation move in Eq. (4.92) holds for the square root instead of the vertex weight. Such a square root always exists if we are working with complex tensors, but it cannot be constructed purely diagrammatically. Equipped with this, the mapping becomes

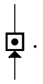


(4.188)

There is an alternative algebraic structure that can be mapped from the 2D tTS: A *finite-dimensional *-algebra* consists of a unital associative algebra, together with an anti-linear involution $*$, such that we have

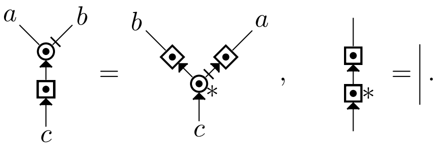
$$(ab)^* = b^* a^* \ , \qquad (a^*)^* = a \quad \forall a, b \in V \ . \tag{4.189}$$

As an anti-linear map, $*$ is given by complex conjugation followed by a linear map, and we will denote this linear map as a tensor



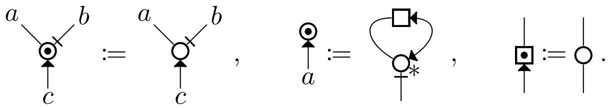
(4.190)

Then the above axioms become



(4.191)

With this, we have the following mapping from the *-algebra TS to the 2D tTS,



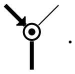
(4.192)

With this, the second axiom in Eq. (4.191) maps to the 2-gon cancellation move in Eq. (4.94), and the first axiom can be derived by three 01 and 12 triangle flip moves of Eq. (4.96) and (4.97).

Next, we will see that also the 2D boundary tTS is very similar to a very well-known algebraic TS. A *representation* of an algebra is a linear map

$$\rho : R \otimes V \to R \,, \tag{4.193}$$

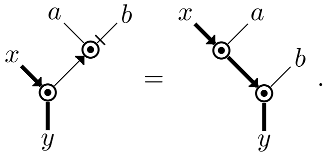
for some separate vector space R. A finite-dimensional representation is determined by its structure coefficients, forming a 3-index tensor,


(4.194)

Here we denote the vector space R by a thick line. A representation has to satisfy

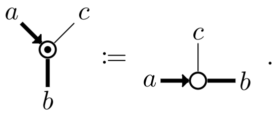
$$\rho(x, a \cdot b) = \rho(\rho(x,a), b) \quad \forall a, b \in V, x \in R \,, \tag{4.195}$$

or in diagrams,



(4.196)

With this, we can extend the mapping from Eq. (4.183) to a mapping from the special-Frobenius-algebras-representation TS to the 2D boundary tTS:



(4.197)

Plugging this mapping into Eq. (4.196) precisely yields Eq. (4.161). However, the move in Eq. (4.162) corresponds to some additional consistency between the representation and the Frobenius structure.

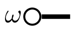
All in all we see that 2D-(boundary)-tTS models yield (representations of) associative algebras with a wide range of extra conditions, such as unital, special Frobenius, †-Frobenius, or $*$-algebras. Vice versa, "nice enough" examples of algebras that fulfill sufficiently many extra conditions do yield 2D-tTS models. In particular, we can interpret previously mentioned tTS models in terms of algebras: The δ-tensor model in Eq. (4.4) corresponds to the algebra of complex functions on a d-element set with pointwise multiplication. The model in Eq. (4.15) corresponds to the algebra of complex $d \times d$ matrices. The boundary models in Eq. (4.155) and (4.164) correspond to the irreducible representations of the two algebras above, respectively.

As we have seen, in order to reformulate 2D-tTS models as some type of algebras, we need to add many extra structures or properties such as "Frobenius", "unital", or "special". This is due to the following key difference between tTS and algebraic TS: The latter have distinct "input" and "output" indices, and diagrams only contract input with output indices in a globally non-cyclic fashion. In other words, the diagrams are restricted by a global "flow of time", which we have picked to run from bottom to top for the diagrams above. This global flow of time is incompatible with the topological nature of the fixed-point ansatzes we want to describe, and the fact that we are dealing with Euclidean-signature spacetimes. Many of the extra structures or properties are secretly attempts to undo the global flow of time. Because of this, we find that tTS models are a more natural way to describe topological fixed-point ansatzes, since they are not constrained by a flow of time from the very beginning on.

4.13 Block-diagonalization in $1+1$D

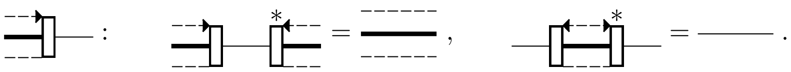
In this section, we state the classification of 2D-tTS models and their boundaries in $1+1$ dimensions. As we have seen in Section 4.12, 2D-tTS models give rise to special Frobenius algebras, in particular special †-Frobenius algebras, and also finite-dimensional *-algebras. All of these types of algebras have enough properties to make them *semi-simple*. Thus, they are isomorphic to a direct sum of $\lambda \times \lambda$ matrix algebras by the *Artin-Wedderburn theorem*. The following proposition applies this classification in the context of tTS models. Since direct sums of $\lambda \times \lambda$ matrices are block-diagonal, we refer to the classification as *block-diagonalization*.

Proposition 1. *For every 2D-tTS model, the following holds. There exists a vector ω,*



(4.198)

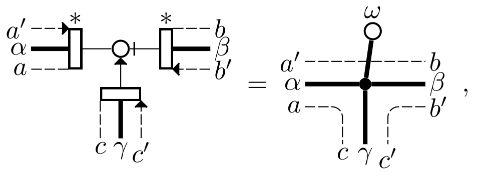
and an isometry between the 2D-tTS vector space and a vector space represented by three indices, denoted as follows,



(4.199)

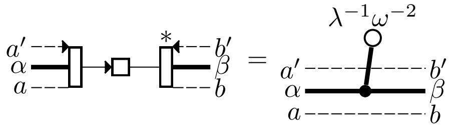
The thicker index of the triple will be called irrep index*, and the dashed indices* internal indices. *The diagrams above represent a generalized tensor-network notation: The bond dimension λ_i of the internal indices depends on the value i of the associated irrep index. As a consequence, we can only contract internal indices if the associated irrep indices are forced to the same value via a δ-tensor.*

After applying this isometry, the triangle tensor is given by



(4.200)

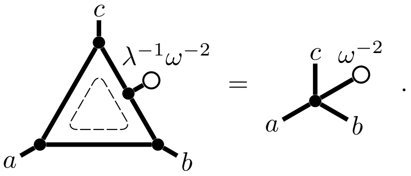
and the vertex weight is given by



(4.201)

The name "irrep index" comes from the fact that it labels isomorphism classes of irreducible representations ("irreps") of the corresponding semi-simple algebra. The internal indices label basis vectors within the λ_i-dimensional irrep i.

Using the above proposition, we can turn each 2D-tTS model into a more restricted state sum: Plugging Eq. (4.200) into a 2D-tTS tensor network yields a network of one δ-tensor at every triangle. At every vertex of the triangulation, we get an internal-index loop, as well as a ring of δ-tensors including one $\lambda_i^{-1}\omega_i^{-2}$ vertex weight. For example, the tensor network around a 3-valent vertex looks like

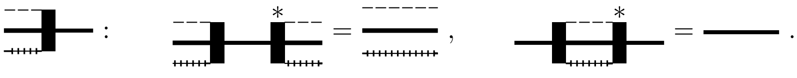


(4.202)

As shown, the internal-index loop is just a trace of a λ_i-dimensional identity matrix, yielding a scalar λ_i, which combined with the vertex weight yields a weight of ω_i^{-2}. The ring of δ-tensors can be shrunk to a single δ-tensor. Taking everything together we get the following state sum: There is one variable at each vertex. At each edge, the variables at its endpoints are forced to the same value, which implies that all vertices on a connected component of the triangulation are forced to the same value. At each vertex, there is a weight ω_i^{-2}. At each triangle, there is a weight ω_i.

At the end of Section 4.12, we have seen that 2D-boundary-tTS models give rise to representations of the corresponding special Frobenius algebras. Irreducible representations of semi-simple algebras are just projections onto individual matrix algebras in the direct sum. The following proposition uses this classification in the context of 2D-boundary-tTS models.

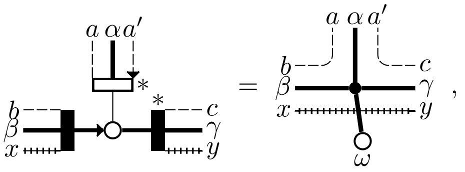
Proposition 2. *For every 2D-boundary-tTS model, the following holds. In addition to the isometry of Proposition 1, there exists an isometry between the boundary bond dimension and a triple of indices,*



(4.203)

The index triple consists of one irrep index, one internal index, and a new type of index called multiplicity index, *drawn as a ticked line. Like the internal indices, the bond dimension m_i of the multiplicity indices depends on the value i of the irrep index.*

Applying this isometry to the boundary edge tensor, together with the isometry of Proposition 1, yields

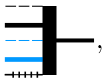


(4.204)

where ω is the vector from Proposition 1.

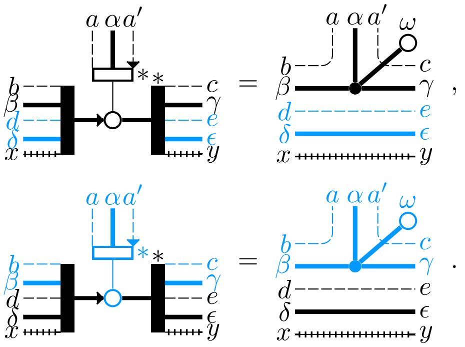
Next, we have seen in Section 4.11 that domain walls between two 2D-tTS models a and b are equivalent to boundaries of the model obtained from stacking a and b. Stacking a and b corresponds to taking the tensor product of the corresponding (special Frobenius) algebras. The irreps of a tensor product of algebras are just pairs of irreps (i, j) of the individual algebras. The dimension for such a tensor-product irrep is the product $\lambda_{(i,j)} = \lambda_i \lambda_j$ of the individual dimensions. Applying the previous proposition in this context yields the following proposition.

Proposition 3. *Consider a 2D-domain-wall-tTS model, or equivalently, two 2D-boundary-tTS models sharing the same boundary bond dimension variable, such that the two boundaries commute as in Eq.* (4.170)*. Then there exists an isometry,*



(4.205)

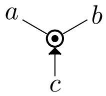
from the boundary/domain-wall bond dimension to a space with one a and one b irrep index, one a and one b internal index, and one common multiplicity index. As shown, we draw b indices and tensors in blue. The dimension of the multiplicity index is $m_{i,j}$ depending on both the a-irrep i and the b-irrep j. Applying this isometry to the boundary tensors yields



(4.206)

The situation is analogous if we have x 2D-boundary-tTS models that share the same boundary bond dimension and commute among each other via Eq. (4.170)*. The common boundary bond dimension decomposes into a tuple of indices. There is one irrep* *index and one internal index for each of the x copies, and one common multiplicity index. The dimension of this multiplicity index is $m_{i_0,\ldots i_{x-1}}$, depending on the values $i_0,\ldots,i_{x-1}$ of the x irrep indices.*

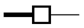
Finally, for Section 4.14, we also need to anticipate a TS from Section 8.3, namely the *2D atTS*. The 2D atTS is like the 2D TS, just that we do not impose any moves that involve interior vertices like the 1-3 Pachner move or the triangle cancellation move. We therefore also do not need a vertex weight. Instead, we add the commutativity move in Eq. (4.10), and therefore denote the generating tensor without a tick marking that distinguishes two of its indices,



. (4.207)

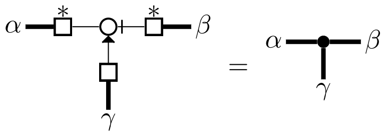
The corresponding algebras do not have the "special" property anymore, but are still Frobenius algebras and *-algebras, and therefore semi-simple. Since they are also commutative, this implies $\lambda\times\lambda$ matrix algebras are trivial 1×1 matrix algebras. Further, we define the *2D atTS with boundary* like the 2D boundary tTS where the bulk is given by the 2D atTS. Note that this is different from the *2D boundary atTS* introduced later in Section 8.5. The 2D-atTS-with-boundary models yield representations of the commutative algebras, and each such representation decomposes into irreducible ones that are projections onto different 1×1 matrix blocks. The following proposition transfers this classification to the context of 2D-atTS models.

Proposition 4. *For every 2D-atTS model, there is an isometry,*



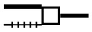
, (4.208)

which maps the model to a δ-tensor model from Eq. (4.4)*,*



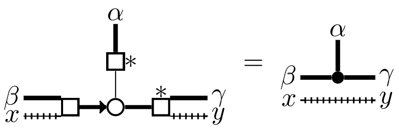
. (4.209)

For each 2D-atTS-with-boundary model, there is an additional isometry,



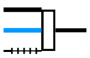
, (4.210)

such that



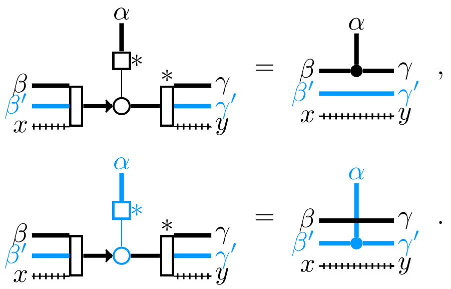
. (4.211)

For every commuting pair of 2D-atTS-with-boundary models, we have an isometry



, (4.212)

such that

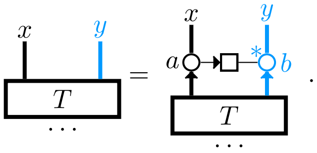


(4.213)

The analogous holds if we have x different commuting 2D-atTS-with-boundary models. The boundary vector space then decomposes into irrep indices $i_0, \ldots, i_{x-1}$, and one multiplicity index of dimension $m_{i_0,\ldots,i_{x-1}}$.

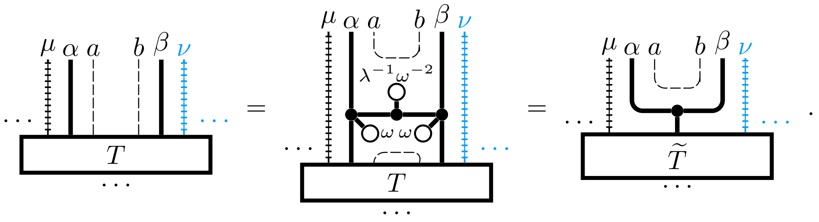
Finally, let us consider two further propositions that are essential when applying the block-diagonalization from this section to higher dimensions.

Proposition 5. *Consider a tensor T and two of its indices, where each is equipped with a set of commuting boundaries of the 2D tTS. Further assume that for the boundary a on the first index, and the boundary b on the second index, the following holds,*



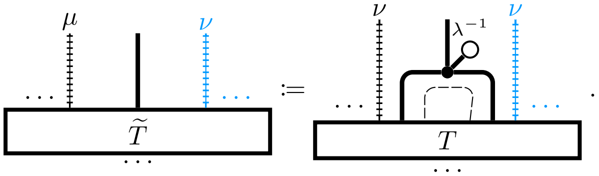
(4.214)

After conjugating with the isometry from Proposition 3 for both sets of commuting boundaries, this equation becomes



(4.215)

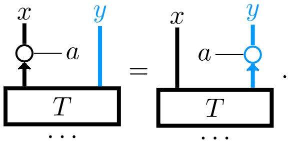
The indices labeled ν and ν are the shared multiplicity indices for the indices x and y in Eq. (4.214)*, respectively. α and β are the irrep indices corresponding to the a boundary of x and the b boundary of y, and a and b are the according internal indices. The three dots on the left and right are symbolic for the commuting boundaries acting on x or y that are not a or b. As shown, we can express T through a simpler tensor $\widetilde{T}$:*



(4.216)

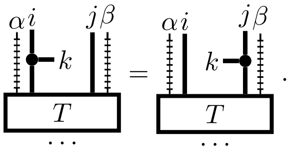
The a and b internal indices of x and y have disappeared, and the according irrep indices are fused into a single one.

Proposition 6. *Consider a tensor T with two indices each of which is equipped with a boundary of the 2D atTS. Assume that the following holds,*



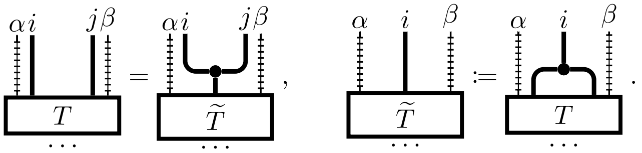
(4.217)

After conjugating with the block-diagonalizing isometry for these boundaries from Proposition 4, this becomes,



(4.218)

We thus have



(4.219)

In other words, the irrep components of the two indices x and y can be fused to a single one.

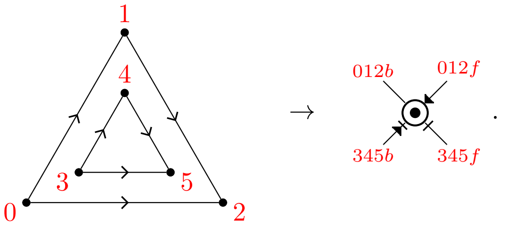
4.14 Block-diagonalization in $2+1$D

In contrast to 2D tTS models in Section 4.13, there is no hope for a general classification of tTS models in $2+1$ or higher dimensions. At least, if by classification, we mean an algorithmically efficient enumeration of the models of bond dimension d. In fact, even writing down concrete examples of higher-dimensional tTS models and asserting the tTS moves is rather inefficient, not in an algorithmic sense, but just in the sense of being very tedious to do with pen and paper. For example, in order to express slightly more involved topological phases in $2+1$ dimensions (such as the double-Ising topological phase) as tTS models, we need medium-large bond dimensions (such as 10). Explicitly spelling out a tensor with, say, four indices (such as the tetrahedron tensor) thus involves writing down 10^4 complex numbers. Checking an equations with, say, 6 open indices on each sides (such as the 2-3 Pachner move) involves checking 10^6 equations of complex numbers. So we see that explicitly working with tTS models can be quite tedious, or even impossible, to do by hand.

In this chapter, we discuss a way to remedy this problem at least to some extent. To do this, we transfer as much as possible from the classification of 2D tTS models in Section 4.13 to 3D tTS models. Accordingly, we will refer to this process *block diagonalization*. This leads to a more fine-tuned fixed-point ansatz, the *block-diagonal form*, which is in fact more complicated, but less tedious for performing computations by hand. The more fine-tuned ansatzes are still given by tensors satisfying equations, but these tensors are *sparse*. That is, the tensor entries are zero unless the index configurations satisfy a set of constraints. Due to this, we need fewer complex numbers to spell out models for non-trivial topological phases.

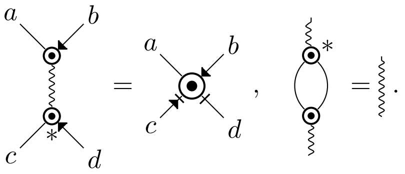
Concretely, we proceed by applying dimension-reduction mappings to distill 2D-(boundary)-tTS models from a 3D-tTS model. Then we block-diagonalize these according to Section 4.13. Finally, we simplify the 3D-tTS models by bringing them in the block-diagonal basis.

The first step is to decompose the 3D-tTS model into robust components. To this end, we construct the following 2D-atTS model from the 3D-tTS model. Note that we have sketched the 2D atTS in Section 4.13, and will properly introduce it later in Section 8.3. We start by considering a pillow-volume inside of which we remove a smaller pillow-volume. The 3D tTS associates to this a tensor with four indices, one for each of the two triangles of each of the two pillow-volumes,



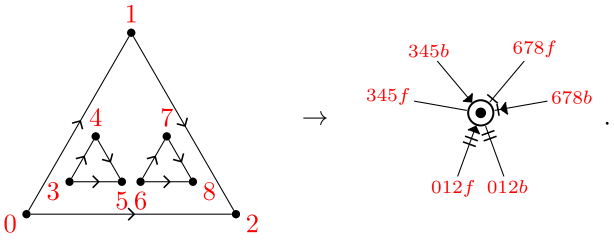
(4.220)

The topology of the volume on the left is $S_2 \times B_1$, and the boundary consists of four triangles. The bulk triangulation does not matter due to the topological invariance of the 3D tTS. Since the volume is reflection symmetric, the tensor is Hermitian when viewed as an operator from the bottom to the top indices. Stacking two of the volumes inside of another yields the same volume again, so the operator is also a projector. Hence, we can find an isometry, such that



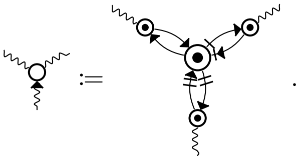
(4.221)

The dimension of the new, squiggly-line bond dimension is equal to the rank of the projector. Now consider the following 6-index tensor corresponding to a pillow volume with two pillow volumes removed on the inside,



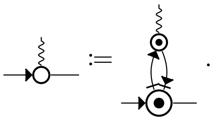
(4.222)

Equivalently, we can imagine this as a 3-sphere with three pillow-volumes removed, just that the 012 pillow has opposite orientation. With this, the defining tensor of the 2D atTS is given by



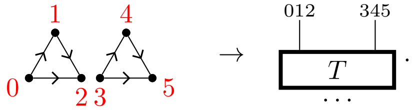
(4.223)

It can easily seen that this is invariant under 2-2 Pachner moves since gluing two of the volumes in Eq. (4.222) yields a 3-sphere with four pillow-like holes. The commutativity move in Eq. (4.10) holds since the volume is invariant under swapping two pillow holes. Next, we can define a boundary of this 2D atTS via the following mapping,



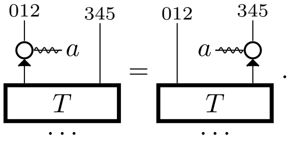
(4.224)

This defines a boundary since gluing two triangles of the 345 pillows of two copies of the volume in Eq. (4.220) yields a 3-sphere with three pillow volumes. Now, consider any two triangles inside the (space) boundary of the same connected component of a triangulation, and the associated tensor T,



(4.225)

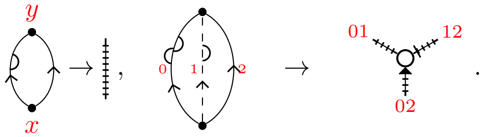
If we glue one triangle of Eq. (4.220) to either triangle, this will introduce a pillow-hole inside the connected component of the interior manifold. Thus, we have



(4.226)

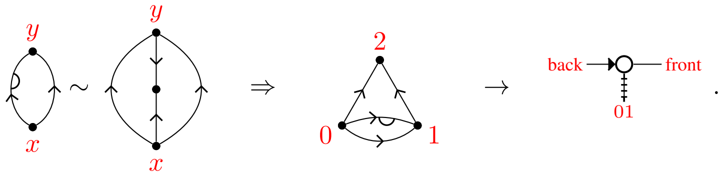
We now apply the block-diagonalizing isometry for this boundary from Proposition 4 to every triangle. Every triangle index then becomes a pair if an irrep label i, and a multiplicity label whose dimension m_i depends on i. We then apply Proposition 6, stating that T is determined by a simpler tensor $\widetilde{T}$ where both irrep labels are forced into a single one. If we apply this Proposition for enough index pairs of the same connected component, we will end up with a tensor $\widetilde{T}_i$ with a single irrep index i for the whole connected component. For every fixed irrep i, the tensors $\tilde{T}_i$ define again a 3D tTS formed by only the multiplicity indices. For each of these models, the tensor associated to Eq. (4.220), with the topology $S_2 \times S_1$ is rank-1. Thus, as discussed in Section 4.6, each of these 3D-tTS models $\widetilde{T}_i$ is indeed robust.

We now continue with each of the robust 3D-tTS models by constructing a 2D-tTS model from it, which we refer to as the *banana 2D-tTS model*. To this end, we start by defining a new bond dimension variable corresponding to a 2-gon with acyclic edge directions, and a new tensor-variable corresponding to a “banana-like” 3-cell,



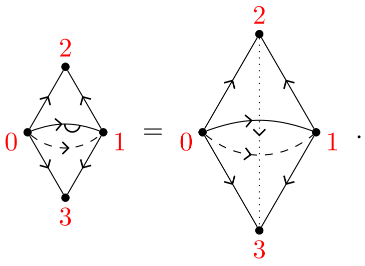
(4.227)

Note that we mark one edge of the 2-gon with a half-circle to break the reflection symmetry and determine whether a 2-gon is positively or negatively oriented. The boundary of the “banana” 3-cell consists of three acyclic 2-gon faces, one in front and two at the back. In order to construct these new tensor and bond dimension variables, we first realize that the acyclic 2-gon can be triangulated with two triangles. Accordingly, we can define a new tensor variable, the *acyclic 2-gon adapter*, whose boundary consists of one acyclic 2-gon face as well as the triangulation of the 2-gon with two triangles,



(4.228)

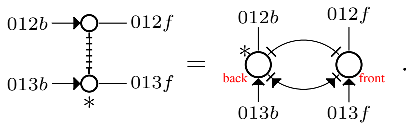
The acyclic 2-gon adapter can be constructed via the following equation,



(4.229)

On the left-hand side, there are two 2-gon adapters glued at their 2-gon faces. On the right-hand side, there are two tetrahedra, glued at two pairs of triangles such that the

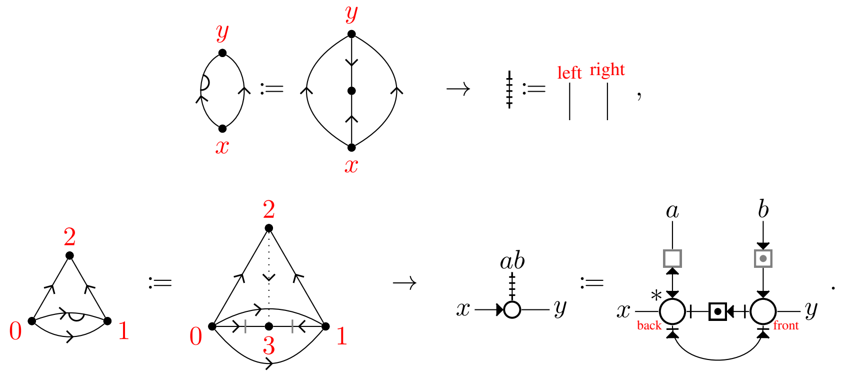
23 edge is in the interior. As a tensor-network equation, this becomes



(4.230)

This equation is of the form $EE^\dagger = A$ where A is the tensor on the right as an operator from bottom to top, and E is the 2-gon adapter as an operator from the 2-gon index to the two triangle indices. We have $A = A^\dagger$ due to Hermiticity and the mirror symmetry Eq. (4.229). Since we are working with complex tensors, this means that E always exists and is defined up to an isometry on the 2-gon index vector space. Further we can make a choice such that $E^\dagger E$ is full rank, corresponding to a full support convention for the 2-gon.

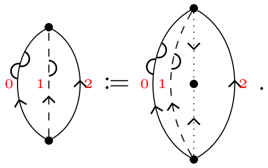
Note that the acyclic 2-gon adapter could also be defined via a TS mapping,



(4.231)

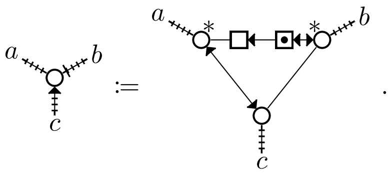
However, this choice does not obey the full support convention for the 2-gon face.

Equipped with the acyclic 2-gon adapter, we can cellulate the banana volume in Eq. (4.227) as



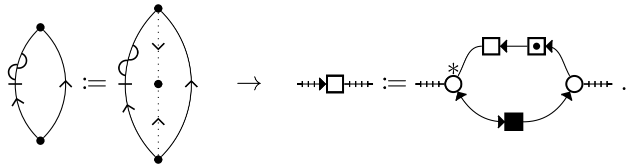
(4.232)

The cellulation on the right-hand side consists of three 2-gon adapters glued in a ring such that every pair shares a triangle, and there are two edges in the interior. In terms of tensor networks, the tensor in Eq. (4.227) now becomes



(4.233)

This new tensor forms a 2D-tTS model, together with the vertex weight,



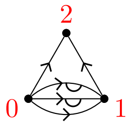
(4.234)

One can explicitly check that this 2D-tTS model fulfills all the moves. For example, the 2-2 Pachner move follows from the fact that gluing two banana-like volumes at two of their 2-gon faces yields a banana-like volume whose boundary consists of four 2-gon faces. In other words, the construction of the 2D-tTS model defines a TS mapping from the 2D tTS to the 3D tTS. Topologically, this mapping maps a 2-manifold M to a 3-manifold via a *suspension*,

$$M \to (M \times [0,1])\big/_{\substack{a\times 0\sim b\times 0\forall a,b\in M,\\ a\times 1\sim b\times 1\forall a,b\in M}} \ . \qquad (4.235)$$

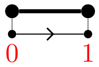
Geometrically, it maps a 2-cellulation to a 3-cellulation as follows. We first introduce two new vertices x and y. Then we replace every vertex of the 2-cellulation by an edge connecting x and y. An edge between two vertices becomes a 2-gon face whose boundary consists of the two according edges connecting x and y. Every face is mapped to a "banana-like" volume whose boundary consists only of 2-gons. This way, a triangle is mapped to the banana volume in Eq. (4.227).

Next, we define a three different 2D-boundary-tTS models for the banana 2D-tTS model, which we refer to as the 01*,* 12*, and* 02 *banana 2D-boundary-tTS* models. The boundary edge tensor for the 01 banana boundary is given by the 2-gon adapter defined in Eq. (4.228). One can explicitly check that this tensor fulfills all the moves. For example, the central move of the boundary 2D tTS in Eq. (4.161) maps to a recellulation,

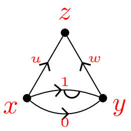


$= \text{same 1-skeleton}$. (4.236)

On the left-hand side, we have two of the triangle-extender volumes in Eq. (4.228) stacked on top of each other and glued at a triangle. On the right-hand side, there is a single such volume together with one banana volume from Eq. (4.227), glued together at a 2-gon. In other words, we have a TS mapping from the 2D-boundary-tTS to the 3D tTS. This TS mapping has again a topological and geometric interpretation: In addition to the vertices x and y introduced for the suspension described above, we add one global vertex z, and two global edges u connecting x and z and w connecting y and z. Every boundary edge gets then mapped to a volume shown in Eq. (4.228),

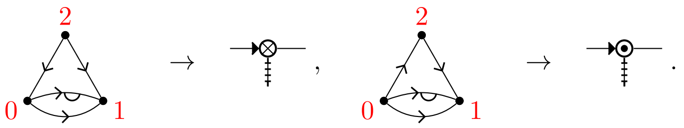


:=



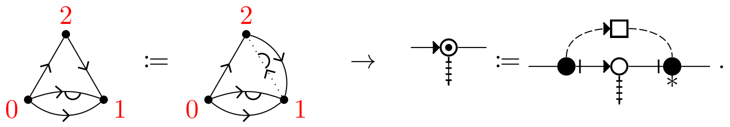
. (4.237)

The 12 and 02 banana boundaries are given by tensors variables assigned to similar volumes,



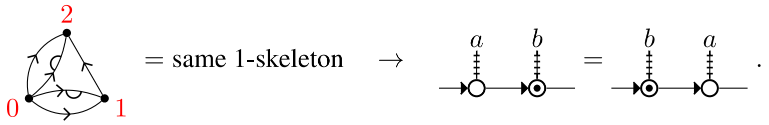
(4.238)

These new tensor variables can be defined from the acyclic 2-gon adapter using the triangle flippers. For example, the 02 triangle extender can be obtained from by wrapping the 2-gon adapter with two 12 triangle flippers,



(4.239)

The 01, 12, and 02 banana boundaries commute among another. For example, the following recellulation corresponds to the commutativity of the 01 and 02 banana boundaries,



(4.240)

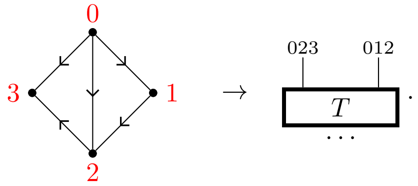
The left-hand side shows a 01 triangle extender which is glued on top of a 02 triangle extender. This yields a volume with the same boundary as if they were glued in the opposite order, but is a different cellulation even though the edges and vertices (which we can draw) are the same. The analogous commutativity moves also hold between the 01 and 12, and between the 02 and 12 triangle extenders. All in all, we find that the banana tensor forms a 2D-tTS model, and the edge extenders form three commuting boundaries for this 2D-tTS model.

We can now apply the isometry from Proposition 1 for the banana 2D-tTS model. At the same time, we apply the isometry from Proposition 3 to the three commuting boundary models given by the 01, 02, and 12 edge extenders. The common bond dimension variable for the commuting boundaries is the one corresponding to the triangle. Thus, each triangle index decomposes into a tuple (i, a, j, b, k, c, μ). Here, i, j, and k are the irrep indices corresponding to the 01, 12, and 02 banana boundaries, respectively. a, b, and c are the according internal indices. μ is the common multiplicity index, whose dimension

$$N_k^{ij} := m_{i,j,k} \; . \tag{4.241}$$

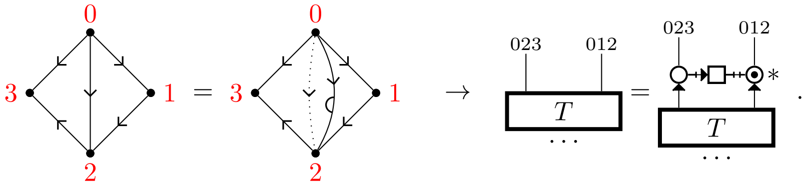
depends on the values i, j, k of the three irrep indices.

Next, we look at our (robust) 3D tTS model in this block-diagonal basis, and see that it can be simplified. Consider the tensor T associated to some 3-cell, or generally some 3-cellulation with space boundary, and the indices associated to two adjacent triangles inside the space boundary, such as



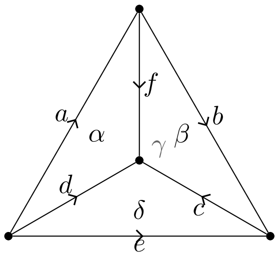
(4.242)

Imagine gluing one banana boundary volume to each of the triangles, such that their 2-gons are adjacent to the edge shared by the triangles, and then glue these two 2-gons. In other words, we take a volume like the one in Eq. (4.229) (with according edge directions) and glue this onto the two adjacent triangles. This gluing changes neither the topology nor the space boundary and thus can be undone by a recellulation, for example,



(4.243)

Note that depending on the edge directions in Eq. (4.242), we have to take different triangle extenders than the 01 and 02 triangle extenders above. Now applying Proposition 5 yields that T is determined by a simpler tensor $\widetilde{T}$. The 01 irrep index of 023 of $\tilde{T}$ on the left and the 02 irrep index of 012 are fused to a single irrep index, which can be associated to the edge separating the two triangles. The according internal index are removed. We now apply this procedure to all triangle pairs on the space boundary of T. This yields a tensor with only one irrep index at every edge, and one multiplicity index at every face. The dimension of the multiplicity index at a triangle is N_k^{ij} depending on the values i, j, k of the irrep indices at its 01, 12, and 02 edge, respectively. In particular, applying this procedure to all edges of the defining tetrahedron tensor yields a tensor $\tilde{T}$ with 10 indices,



$$\rightarrow \quad F_{d\;\;\;\;e\gamma\delta}^{abc\,f\alpha\beta} \, . \tag{4.244}$$

The ordering of sub and superscripts on the right-hand side is essentially arbitrary, but loosely follows some related conventions in the literature. Since the locations of the index labels on the left might not be entirely clear, we explicitly label the indices via the vertex numbers of the the edges and faces they correspond to:

$$F_{(03)}^{(01)(12)(23)}{}^{(13)(013)(123)}_{(02)(012)(023)} \; . \tag{4.245}$$

Now, we plug Eq. (4.215) into the tensor network representing a 3-cellulation/triangulation. For each edge of the cellulation, the identity matrix lines connecting a and b in Eq. (4.215) of the surrounding tensors combine to one internal-index loop. Furthermore, the δ-tensors connected to α and β in Eq. (4.215) form a ring of δ-tensors. For example, for an edge with 3 adjacent volumes, the surrounding tensor-network diagram looks like Eq. (4.202) and can be simplified accordingly. So after this, the tensor network consists of 1) the according $\tilde{T}$ tensor at every 3-cell, 2) an irrep-valued δ-tensor at every edge, 3) a normalization vector ω^{-2} at every edge, contracted with the δ-tensor. We can alternatively express this tensor network as a state sum: There is one irrep variable at every edge, and one multiplicity variable at every face. There is one $\tilde{T}$ weight at every 3-cell, depending on the irrep and multiplicity labels on the edges and faces of the 3-cell. There is one ω_i^{-2} weight at every edge, depending only on the irrep label i of the edge. For better compatibility with other approaches in the literature (see below), we will distribute the normalizations slightly differently. We define

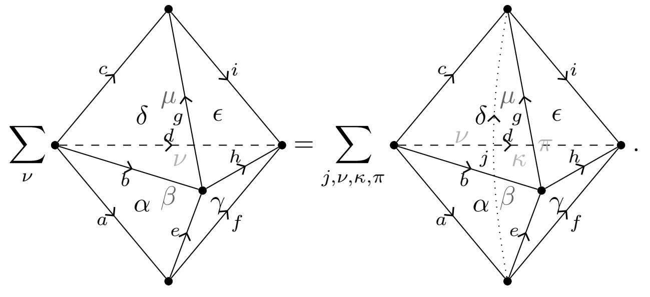
$$D \coloneqq \frac{1}{\sum_i \omega_i^{-4}} \; , \qquad d_i \coloneqq D\omega_i^{-2} \; , \tag{4.246}$$

such that

$$D = \sum_i d_i^2 \; . \tag{4.247}$$

d_i will be called the *quantum dimension* of i, and D the *total quantum dimension*. Then we normalize the $\tilde{T}$ tensor associated to a 3-cell by a factor $D^{1/2}$ for every vertex, and D^{-1} for every edge in the boundary of this 3-cell. After this, the state sum has a normalization of d_i at every edge, and a normalization of D^{-1} at every vertex.

Let us now translate the moves of the tTS to this new block-diagonal form. The 2-3 Pachner move in Eq. (4.18) becomes

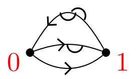


(4.248)

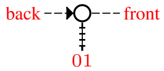
In terms of the tensor F, this is

$$\sum_{\nu} F_d^{aeh}{}^{f\beta\gamma}_{b\alpha\nu} F_d^{bgi}{}^{h\nu\epsilon}_{c\delta\mu} = \sum_{j,\nu,\kappa,\pi} d_j F_c^{aeg}{}^{j\nu\kappa}_{b\alpha\delta} F_f^{egi}{}^{h\gamma\epsilon}_{j\kappa\pi} F_d^{aji}{}^{f\beta\pi}_{c\nu\mu} \; . \tag{4.249}$$

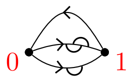
This equation alone is not enough to show topological invariance. For the latter, we can either impose 2-3 and 1-4 Pachner moves with all different choices of edge orientations, or we use the simplified tTS from Section 4.8 in the block-diagonal setting. To this end, we also need to define two more boundaries of the banana 2D-tTS model, whose boundary bond dimension is that of the cyclic 2-gon,



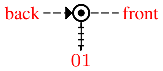
$\rightarrow$



,

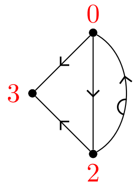


$\rightarrow$

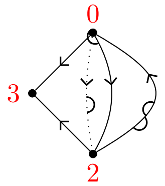


. (4.250)

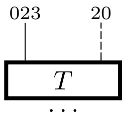
These boundaries again commute with each other. After applying the isometry from Proposition 3, the index at a cyclic 2-gon splits into a tuple (i, a, j, b, μ), where i, j are banana irreps, a, b are the according internal indices, and μ is a common multiplicity index of dimension $N^{ij} := m_{i,j}$. For every edge separating a triangle and a 2-gon, we have the following equation analogous to Eq. (4.243),



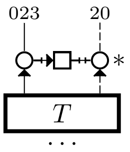
$=$



$\rightarrow$



$=$

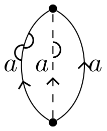


. (4.251)

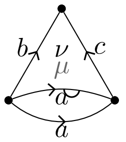
Again, we apply Proposition 5 and find that T is determined by a more fine-tuned tensor $\widetilde{T}$. In addition to edges and triangles, $\widetilde{T}$ has one index at every cyclic 2-gon, whose bond dimension is N^{ij} depending on the values i, j of the irrep indices at its edges. Analogously, every acyclic 2-gon has a multiplicity $N_i^j := m_{i,j}$. However, since the acyclic 2-gon corresponds to the edge tensor of the banana boundary itself, we have

$$N_i^j = \delta_{ij} \ , \tag{4.252}$$

so the two edges of an acyclic 2-gon always have the same irrep label, and the multiplicity label is trivial. Furthermore, the $\widetilde{T}$ tensors associated to the banana 2d-tTS model in Eq. (4.227) its boundaries in Eqs. (4.228) or (4.238) are given by

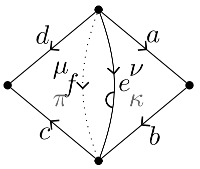


$\rightarrow \quad d_a^{-1/2} \ ,$



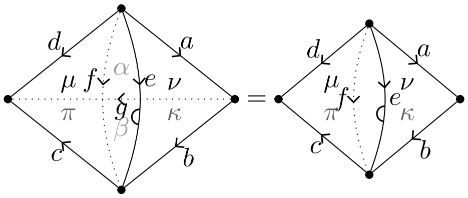
$\rightarrow \quad \delta_{\mu,nu} d_a^{-1/2} \quad \forall b, c \ .$ (4.253)

With this, the block-diagonal $\widetilde{T}$ form of the right-hand side of Eq. (4.243) is given by,



$\rightarrow \quad \delta_{ef}\delta_{\pi\mu}\delta_{\kappa\nu}d_e^{-1} \quad \forall a, b, c, d \ .$ (4.254)

With this, we have the following move,



$$\rightarrow \quad \sum_{g,\alpha,beta} d_g \overline{F^{abc}_{d}{}^{g\alpha\beta}_{e\nu\mu}} F^{abc}_{d}{}^{g\alpha\beta}_{f\kappa\pi} = \delta_{ef}\delta_{\pi\mu}\delta_{\kappa\nu} d_e^{-1} \quad \forall a,b,c,d \; . \tag{4.255}$$

The above equation implies that

$$\sum_e N_e^{ab} N_d^{ec} = \sum_g N_d^{ag} N_g^{bc} \; . \tag{4.256}$$

More generally, analogous considerations for other faces than triangles show that the multiplicity-dimensions N associated to different faces form a 2D tTS model with positive-integer tensor entries. For example, the 2-gon cancellation move in Eq. (4.93) yield an identity

$$\sum_b N^{ab} N^{bc} = N_a^b = \delta_{a,b} \; . \tag{4.257}$$

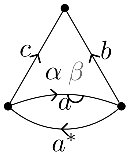
For this to be true, we need to have

$$N^{ab} = \delta_{b,a^*} \tag{4.258}$$

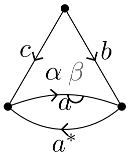
for some function $*$ on the set of irreps with

$$(a^*)^* = a \; . \tag{4.259}$$

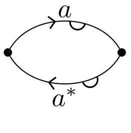
With this, let us now bring the simplified generators-and-relations presentation from Section 4.8 into block-diagonal form. The auxiliary tensors become:



$$\rightarrow \quad G^{ab}_{c}{}_{\alpha\beta} \; ,$$



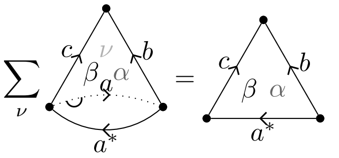
$$\rightarrow \quad H^{ca}_{b}{}_{\alpha\beta} \; , \tag{4.260}$$



$$\rightarrow \quad \kappa_a \; .$$

Next, let us consider the moves. We already stated the 2-3 Pachner move in Eq. (4.249). Also, the move in Eq. (4.255) already implies the tetrahedron cancellation move in

Eq. (4.107). The 01 triangle flipper cancellation move from Eq. (4.114) becomes



$$\rightarrow \quad \sum_\nu d_a G^{ab}_{c\ \nu\alpha} \overline{G^{ab}_{c\ \nu\beta}} = \delta_{\alpha,\beta} \quad \forall a,b,c\ . \tag{4.261}$$

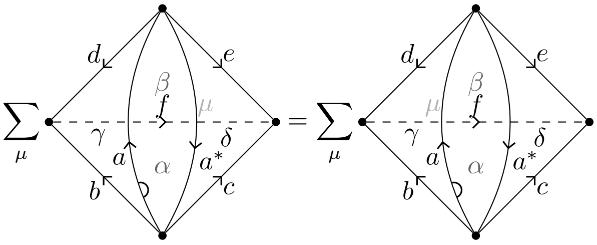
The 12 triangle flipper cancellation move is analogous. The 2-gon flipper cancellation move in Eq. (4.117) becomes

$$\kappa_a \kappa_{a^*} = 1\ . \tag{4.262}$$

For the 2-gon flipper in Eq. (4.260), mirroring at the drawing plane and rotating around the axis perpendicular to the drawing plane exchanges a and a^*, so we have,

$$\kappa_{a^*} = \overline{\kappa_a}\ . \tag{4.263}$$

Now, if $a \neq a^*$, then we can perform a basis change on the 1-dimensional multiplicity vector space corresponding to $N^{aa^*} = 1$, which is just a phase. If we use $\overline{\kappa_a}$ as this basis change, then after the basis change we have $\kappa_a = \kappa_{a^*} = 1$. However, if $a = a^*$, then we have $\kappa_a^2 = 1$, and $\kappa_a = \pm 1$ is independent of the choice of basis. Next, 01 tetrahedron flip move in Eq. (4.119) becomes



(4.264)

$$\sum_\mu F^{gdf\ e\mu\beta}_{c\ \ \ b\gamma\alpha} G^{ae}_{c\ \mu\delta} = \sum_\mu \overline{F^{abf\ c\delta\alpha}_{e\ \ \ d\mu\beta}} G^{ad}_{b\ \mu\gamma}\ .$$

The line above the F-tensor denoted complex conjugation. The other tetrahedron flip moves are analogous. The 01 and 12 triangle flipper flip moves in Eq. (4.125) and following become,

$$G^{ab}_{c\ \alpha\beta} = \kappa_a G^{a^*c}_{b\ \ \beta\alpha}\ , \qquad H^{ca}_{b\ \alpha\beta} = \kappa_a G^{ba^*}_{c\ \ \beta\alpha}\ . \tag{4.265}$$

Let us now see how the block-diagonal form of the 3D tTS that we obtained is related to state-sum constructions in the literature. Indeed, we find that the state-sum that we derived from the general principle of topological invariance is equivalent to the *Turaev-Viro state-sum construction* [130, 14]. This is a state sum that is built from a *unitary fusion category* [64], and was originally constructed in Ref. [130] for specific class of fusion categories, namely quantized enveloping algebras of sl_2. In Ref. [14], the state-sum construction was formulated for arbitrary *spherical* fusion categories. Later, these state sum models have been reformulated as commuting-projector Hamiltonians, and are now well-known in the physics community as *Levin-Wen string-net models* [105].

For concrete computations, it is best to specify a fusion category by its *skeletal data*. This data consists of a set of *simple objects*, a *fusion ring*, a set of *quantum dimensions*, and an *F-symbol* which represents the *associator* of the fusion category. This data is very directly related to our block-diagonal 3D tTS models: The simple objects are the irreps of the banana 2D-tTS model. The structure coefficients of the fusion ring are the multiplicities N, in particular N_k^{ij} defines the multiplication. The quantum dimensions are the numbers d_i. The F-symbol (denoted $\widetilde{F}$ in the following) is related to our tensor F in Eq. (4.244) by a simple normalization,

$$\widetilde{F}_{de\gamma\delta}^{abc\,f\alpha\beta} = F_{de\gamma\delta}^{abc\,f\alpha\beta} d_e^{1/2} d_f^{1/2} \ . \tag{4.266}$$

The F-symbol has to satisfy an equation called the *pentagon equation*, which is equal to Eq. (4.249). Note that due the normalization in Eq. (4.266), the prefactor in Eq. (4.249) is not present in the pentagon equation. For a unitary fusion category, the F-symbol $\widetilde{F}$ is a unitary as an operator from bottom to top indices with the indices a, b, c, and d fixed. Taking into account the different normalizations of F and $\widetilde{F}$, this is precisely Eq. (4.255).

We believe that the tensors G and H in Eq. (4.260) are an explicit skeletal representation of a *spherical pivotal structure* for the fusion category, but have not found anything similar in the math literature. In Appendix E of Ref. [96], Kitaev defines "raising" and "lowering" operators A and B which seem related to our tensors G and H. Note however, that this appendix discusses modular tensor categories and not fusion categories. The 2-gon flip tensor κ in Eq. (4.260) corresponds to what is known as the *Frobenius-Schur indicator* of fusion categories. In fact, it is known that every unitary fusion category admits a unique spherical pivotal structure, see for example Ref. [85].

To be precise, the structure we have defined so far corresponds to more general *multi-fusion categories*. Consider the multiplicity N^a corresponding to a 1-gon. This defines the identity element of the fusion ring. For a fusion category, there is a simple object (or irrep) that we call 1, such that

$$N^a = \delta_{a,1} \ . \tag{4.267}$$

This implies the following,

$$N_b^{a1} = N_b^{1a} = \delta_{a,b} \ , \quad N_1^{ab} = \delta_{a,b^*} \ . \tag{4.268}$$

For general multi-fusion categories, there can be multiple simple objects a such that $N^a \neq 0$. The simplest example for a multi-fusion category that is not a fusion category is one whose fusion ring is a $d \times d$ matrix algebra. The set of simple objects is given by $\{0,\ldots,d-1\} \times \{0,\ldots,d-1\}$, and the multiplication is

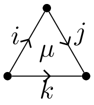
$$N_{(e,f)}^{(a,b)(c,d)} = \delta_{a,e}\delta_{d,f}\delta_{b,c} \ . \tag{4.269}$$

The F-tensor is just 1 for every configuration for which all multiplicities are 1 (and non-existent otherwise). The identity is defined by

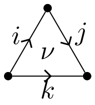
$$N^{(a,b)} = \delta_{a,b} \ , \tag{4.270}$$

and thus consists of d different simple objects. However, this model is in a trivial phase. The corresponding tensor network consists of one disconnected web of δ-tensors around every vertex. One can more generally argue that any robust model with a non-simple unit of the fusion ring is in the same phase as a model with a simple unit. On the level of categories, this means that any indecomposable multi-fusion category is Morita equivalent (c.f. Section 5.4) to a fusion category. Note that Levin-Wen models for multi-fusion categories have been studied in Ref. [49].

The tTS moves are invariant under changes of basis for every bond dimension variable. That is, if we act on all tensors of a tTS model by applying the same unitary to every index of a fixed bond dimension variable, this yields a tTS model again. In the block-diagonal form, we can perform unitaries U_k^{ij} on the multiplicity indices, which can parametrically depend on the associated irrep indices i, j, k. The unitary U then acts on each triangle as follows,



$$\rightarrow \quad \sum_\nu U_{k\ \nu}^{ij\,\mu}$$

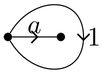


$$. \tag{4.271}$$

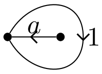
Applying U to the tensor F yields a new tensor F',

$$F'^{abc\,f\alpha\beta}_{d\quad e\gamma\delta} := \sum_{\mu,\nu,\kappa,\pi} U_{d\ \nu}^{af\,\alpha} U_{f\ \nu}^{bd\,\beta} \overline{U_{e\ \gamma}^{ab\,\kappa}}\,\overline{U_{d\ \delta}^{ec\,\pi}} F^{abc\,f\mu\nu}_{d\quad e\kappa\pi} \ . \tag{4.272}$$

Now, for a block-diagonal 3D-tTS model with simple unit (c.f. Eq. (4.267)), consider the following two auxiliary tensor variables,



$$\rightarrow \quad e^{2\pi i\eta_a} d_a^{1/2} \ ,$$



$$\rightarrow \quad e^{2\pi i\omega_a} d_a^{1/2} \ . \tag{4.273}$$

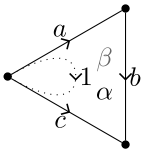
The boundary of these 3-cells consists of one 1-gon (in the back) and one triangle (self-glued, in the front). Now, by applying a basis transformation

$$U_a^{1a} = e^{-2\pi i\eta_a} \ , \quad U_a^{a1} = e^{-2\pi i\omega_a} \ , \tag{4.274}$$

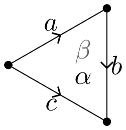
we get

$$I'_a = d_a^{1/2} \ , \quad J'_a = d_a^{1/2} \ . \tag{4.275}$$

Now, consider the recellulation



$$=$$



$$\begin{gathered}\rightarrow \quad I_a I_c F_{c\quad a\varnothing\beta}^{1ab\,c\varnothing\alpha} = \delta_{\alpha,\beta} \\ \Rightarrow \quad F_{c\quad a\varnothing\beta}^{1ab\,c\varnothing\alpha} = d_a^{-1/2} d_c^{-1/2} \delta_{\alpha,\beta} \ .\end{gathered} \tag{4.276}$$

The left-hand side of the recellulation consists of one tetrahedron with one of the 3-cells in Eq. (4.273) glued to each of the 012 and 013 triangle. The label $\varnothing$ indicates that the corresponding multiplicity dimension is 1 and thus the index can be suppressed. Analogously, we have

$$F_c^{a1b}{}^{b\alpha\varnothing}_{a\varnothing\beta} = d_a^{-1/2} d_b^{-1/2} \delta_{\alpha,\beta} \;, \quad F_c^{ab1}{}^{b\alpha\varnothing}_{c\beta\varnothing} = d_b^{-1/2} d_c^{-1/2} \delta_{\alpha,\beta} \;. \tag{4.277}$$

In other words, there exists a basis for the multiplicity spaces such that the tensor F is "trivial" if the irrep label at either the 01, 12, or 23 edge is 1.

Note that in the physics literature, it is common to impose unnecessary restrictions to simplify the presentation for a physics audience ant to avoid having to properly deal with the pivotal structure. In the original Ref. [105], it is assumed that $0 \leq N_k^{ij} \leq 1$ for all i, j, k, such that all multiplicity labels can be dropped. It is also assumed that the F-symbol is invariant under an ad hoc *tetrahedral symmetry* given by 1) flipping the direction of a tetrahedron edge, 2) changing its label from a to a^*. In our construction, changing an edge direction is instead done via the tensors G and H corresponding to the triangle flippers in Eq. (4.264). Even with these restrictions, we still get many interesting models, but they are incompatible with some more general models. Note that the tetrahedral symmetry also depends on the choice of basis of the multiplicity vector space, and will not hold for a generic basis. There are fusion categories for which no basis exists such that the tetrahedral symmetry holds. The prototypical example for this are Turaev-Viro models known as (twisted) *Dijkgraaf-Witten gauge theories* [58], or as *twisted quantum doubles* in a Hamiltonian context [82], which we will study in the next Section 4.16. There has been a series of papers aimed at generalizing string-net models beyond the ad hod symmetry, including Refs. [80, 83, 108, 77]. In this thesis, we reinforce that the most general structure for string-net models are precisely unitary (spherical) fusion categories, which in a way was known since Ref. [14] in 1993.

One critique that applies to all approaches to fixed-point models of topological order in $2 + 1$ dimensions is that they are essentially an educated guess: It is by no means a priori clear that fusion categories have anything to do with topological order, or why we should consider models consisting of string nets in the first place. Even though it is appreciated in the older state-sum literature (and also but less in the more modern physics literature) that these models obey topological invariance in spacetime, we are not aware of any efforts to argue that these models are in fact the most general implementations of topological invariance. Our approach closes this gap: We start with an general and course-grained ansatz formulated in terms of high-level tensor networks, imposing only topological invariance as the only starting condition. Then we derive a set of generators and axioms, and apply block diagonalization to *derive* the structures equivalent to unitary fusion categories, instead of merely postulating them. This program is completed when we prove universality of our fixed-point ansatzes in Chapter 10. This universality proof also explains why the existence of topological boundaries is directly wired into the structure of established fixed-point ansatzes.

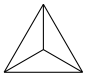
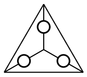
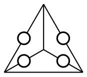
After introducing the block-diagonal form, we are finally ready to give some real examples for a topological fixed-point model. To this end, we only need to look into the mathematics literature on (unitary) fusion categories. We will introduce a large family of examples in the next Section 4.16, which includes some of the simplest examples like the toric code. Let us here give two examples that go beyond this family. As a first example, consider the *double-Fibonacci model* with 2 irreps, labeled 1 and τ. The fusion ring is determined the simple unit 1, $\tau^* = \tau$, and

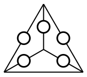
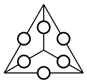
$$N_\tau^{\tau\tau} = 1 \ . \tag{4.278}$$

With Eq. (4.268), this determines all fusion multiplicities. The quantum dimensions are given by

$$d_1 = 1 \ , \quad d_\tau = \phi \ , \quad \phi = \frac{1+\sqrt{5}}{2} \quad \to 1 + \phi = \phi^2 \ , \tag{4.279}$$

where ϕ is the golden ratio. The F-tensor is invariant under flipping the direction of any edge, or in other words, they are invariant under the tetrahedral symmetry discussed above. Thus, we only need to specify the F-tensor for five different configurations. We do this by marking edges with label τ by a dot,

 $= 1$,  $= \phi^{-1/2}$,  $= \phi^{-1}$,

 $= \phi^{-1}$,  $= -\phi^{-2}$. (4.280)

Note that, using the basis where Eqs. (4.276) and (4.277) hold, all except the last F-tensor entry above are predetermined.

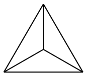
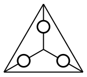
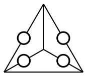
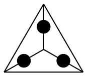
As a second example, consider the *double-Ising model* with 3 irreps, labeled 1, ψ, and σ. The fusion ring is determined by the simple unit 1, and

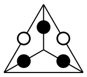
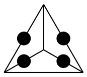
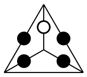
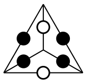
$$\begin{gathered} \psi^* = \psi \ , \quad \sigma^* = \sigma \ , \\ N_\sigma^{\psi\sigma} = N_\psi^{\sigma\psi} = N_\psi^{\sigma\sigma} = 1 \ , \quad N_\psi^{\psi\psi} = N_\sigma^{\sigma\sigma} = N_\psi^{\psi\sigma} = N_\psi^{\sigma\psi} = N_\sigma^{\psi\psi} = 0 \ . \end{gathered} \tag{4.281}$$

The quantum dimensions are

$$d_1 = 1 \ , \quad d_\psi = 1 \ , \quad d_\sigma = 2^{1/2} \ . \tag{4.282}$$

Again, the F tensor is invariant under flipping edge directions/tetrahedral symmetry. Marking ψ edges with an empty dot and σ edges with a full dot, F for the 9 distinct configurations is given by

 $= 1$,  $= 1$,  $= 1$,  $= 2^{-1/4}$,

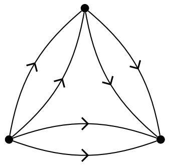
 $= 2^{-1/4}$,  $= 2^{-1/2}$,  $= 2^{-1/2}$,  $= -2^{-1/2}$.

(4.283)

Again, using the basis in Eqs. (4.276) and (4.277), only the last entry is non-trivial. Both examples can be found in Ref. [122], which lists *modular* tensor categories with up to four irreps, yielding fusion categories when simply discarding the braiding. The explicit generating tensors of the Ising modular tensor category can also be found in Ref. [96]. Note that *not* every fusion category can be equipped with a (modular) braiding, and thus conversely, not every fusion category can be obtained by discarding the braiding of a modular tensor category. A full numeric classification of (non-braided) fusion categories with up to 6 irreps can be found in Ref. [33].

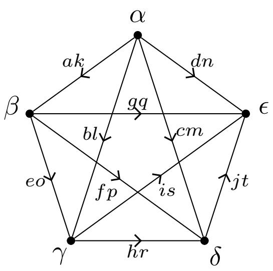
4.15 Block-diagonalization in $3+1$D

In this section, we will sketch block-diagonalization in $3+1$ dimensions. We start by decomposing the 4D-tTS model into robust components by defining a 2D-atTS model analogous to the beginning of Section 4.14, and proceed with each robust component. Analogous to $2+1$ dimensions, we can define a 3D-tTS model via a suspension mapping. The tetrahedron is mapped to a 4-cell representative consisting of four of the banana volumes in Eq. (4.227). We decompose this 3D-tTS model into irreducible components as discussed at the beginning of Section 4.14. We can further define a boundary of a stack of three copies of this 3D-tTS as follows. The boundary triangle is mapped to a 4-cell representative, whose boundary 3-cellulation consists of three copies of the volume

 , (4.284)

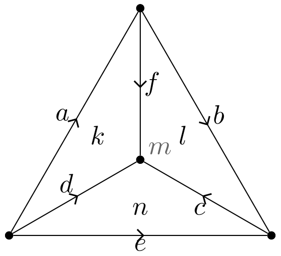
cyclically stacked by gluing along the triangle faces, as well as three banana volumes. Alternatively, we can define three "commuting boundaries" of the 3D tTS, just like a 3D domain wall tTS will be defined in terms of two 3D boundary tTS in Section 5.2. The volume corresponding to a boundary triangle for one of these individual boundaries is obtained by gluing both sides of the recellulation shown in Eq. 4.236. We now decompose this boundary into robust components, for each triple of robust components of the three copies of the 3D-tTS model. Next, we define a 1-dimensional defect of the 3D-tTS and its triple-junctions, whose link is the 1-skeleton of a tetrahedron. The bond dimension running along this defect corresponds to the tetrahedron volume. The 4-cell representative associated to the defining tensor has a boundary 3-cellulation consisting on two tetrahedra and four copies of the volume in Eq. (4.284). This 1-dimensional defect decomposes into components corresponding to the robust components of the surrounding 3D-tTS copies and their junctions. That is, there is one component for every labelling of the tetrahedron with one robust 3D-tTS component at every edge as well as one robust 3D-triple-junction-tTS component at every triangle. For every such configuration, the 4D-tTS index at the tetrahedron can be reduced to a multiplicity index.

In total, the structure of the resulting state sum is as follows. The tensor associated to a 4-simplex then has 3D-tTS-component labels for all of its 10 edges, 3D-triple-junction-tTS-component labels for all of its 10 faces, and multiplicity indices for all of its 5 tetrahedra,



$$\rightarrow \quad F^{abcdefghij}_{klmnopqrst}{}^{\beta\delta}_{\alpha\gamma\epsilon} \,. \tag{4.285}$$

Here, the labels $a, b, c, d, e, f, g, h, i, j$ are associated to the edges as shown. The labels $k, l, m, n, o, p, q, r, s, t$ are associated to the triangles. The label of a triangle is drawn together with the label of the "opposite" edge, that spans the 4-simplex together with this triangle. The labels $\alpha, \beta, \gamma, \delta, \epsilon$ are associated to the tetrahedra. The label of a tetrahedron is drawn at the "opposite" vertex, that spans the 4-simplex together with this tetrahedron. The number of possible label configurations at a triangle is some non-negative integer N_c^{ab} depending on the labels at the three edges of the triangle. The bond dimension of the multiplicity index at a tetrahedron is



$$\rightarrow \quad M_d^{abc}{}^{fkl}_{emn} \,. \tag{4.286}$$

The structure defined above is similar to a *fusion 2-category*. Fusion 2-categories were defined in Ref. [59]. That same reference also constructs a $3+1$-dimensional state-sum model from fusion 2-categories, which is equivalent to the block-diagonal form of the 4D tTS, see also Refs. [134, 84]. Subsets of these models are Dijkgraaf-Witten (twisted) gauge theories [58], 2-group gauge theory [142], the *Crane-Yetter state-sum* based on unitary braided fusion categories [45, 46], and its generalization to G-crossed braided fusion categories [48]. All of the above models were later studied in a physics context as commuting-projector Hamiltonians. Ref. [133] studies the Hamiltonian version of the Crane-Yetter model, now known as *Walker-Wang model*, Ref. [138] studies Hamiltonian models based on G-crossed braided fusion categories, and Ref. [141] studies the models based on general fusion 2-categories. It should be noted that the definition of fusion 2-categories is highly complex, and it is a priori not quite clear which of the many possible definitions is the correct one for describing topological order. In our approach, the obtained algebraic structures are by definition the correct ones for describing the most general $3+1$-dimensional topological fixed-point models. So our approach provides a systematic method for finding the "correct" definition of a fusion 2-category in this context. An explicit set of generating data and axioms can be obtained from the geometric methods in Section 4.9.

4.16 Cohomology models

After discussing block-diagonalization as a method of making general tTS models slightly more tractable in practice, we will now discuss a large family of models that can be found in a computationally efficient way. These models are closely related to cellular cohomology, so we will refer to them as *cohomology models*. An introduction to cellular cohomology can be found in Section 2.5. Cohomology models (or at least some subset of them) are also known as *discrete gauge theories*. Note that in contrast to the standard condensed-matter or high-energy notion of gauge theories, we here consider "pure" gauge theories that do not have matter degrees of freedom but only have gauge degrees of freedom.

Just like the block-diagonal models, cohomology models are best understood as state sum models. Recall that in the block-diagonal form, we sum over configurations of labels where the dimension, such as N_k^{ij}, of some labels depends on the values i, j, k of other labels. If the dimension of one label is $N_k^{ij} = 0$, then this can be understood as a constraint that the i, j, k-configuration is not allowed. Cohomology models have strong constraints that can be expressed as a group multiplication, such as $ij = k$, or a group action, such as $i \triangleleft j = k$. These constraints fully determine some of the labels (such as k above) from others (such as i, j above), such that the many labels of the weights can be reduced to a few generating labels.

In particular, cohomology models have the following property: In general, the moves in a tTS are systems of many linear equations involving the tensor entries. For cohomology models, each linear equation is an equation between two products of tensor entries, with no additions present. Then, assuming all the tensor entries are nonzero, we can take the logarithm of each equation. This yields a set of linear equations over the ring $\mathbb{R}/\mathbb{Z} \times \mathbb{R}$. The model can then be separated into into a $\mathbb{R}/\mathbb{Z}$-valued model and an $\mathbb{R}$-valued model. However, there of only one "phase" of $\mathbb{R}$-valued solutions to the linear equation, since each model can be continuously connected to the all-0 model by just multiplication with a global scalar α. Note that the all-0 model corresponds to a "trivial" all-1 model after exponentiation. So we see that without loss of generality we can restrict ourselves to a $U(1)$-valued models, and these models are obtained by solving a system of linear equations over the ring $\mathbb{R}/\mathbb{Z}$. Solving such linear equations is computationally efficient.

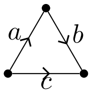
4.16.1 Ordinary cohomology models

In this section, we introduce a specific class of cohomology models, which we will refer to as *ordinary cohomology models*. There, the configurations of state-sum variables form G-valued cellular i-cocycles a. That is, we have one G-label at every i-cell, and one constraint that enforces $da = 0$ at every $i + 1$-cell. There is one weight $\omega[C](a|_C)$ at every n-cell that depends on the n-cell representative C and on the value of a on the i-cells of the n-cell. As we will see, Pachner move/recellulation invariance of these models is equivalent to demanding that ω is a cocycle operation

$$\omega : H^i(G) \to H^n(U(1)) , \qquad (4.287)$$

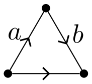
as defined in Section 2.5. Note that for $i = 1$, G can be non-abelian, whereas for $i > 1$, G has to be abelian. Also note that there are only three cases that are physically interesting, in the sense of describing robust topological phases in $n \leq 4$ spacetime dimensions. These cases are $n = 3, i = 1$, $n = 4, i = 1$, and $n = 4, i = 2$. Models for $i = 0$ and $i = n - 1$ correspond to non-robust models, whereas models for $i = n$ are completely trivial.

Let us now consider cohomology models concretely for different i and n. We start with the simplest non-trivial case of $i = 1$ and $n = 2$, even though this does not describe any robust topological phases. We do this both for pedagogical reasons, and because these arise from compactifications when classifying defects in higher-dimensional models, such as in Chapter 6. For simplicity, we will define the cohomology model as a state sum defined on 2-dimensional branching-structure triangulations, even though general cellulations are also possible. To every edge, we assign one G-label, where the group G does not need to be abelian, and the configurations of G-labels form a 1-cocycle. That is, on every triangle, the group labels have to satisfy the constraint



$$\rightarrow \quad ab = c \; . \tag{4.288}$$

At the same time, the triangle carries a weight

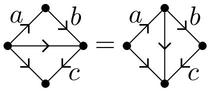


$$\rightarrow \quad \omega(a, b) \; . \tag{4.289}$$

As we can see, c is determined from a and b, so the weight depends only on two instead of three G-elements. Due to Hermiticity, the spacetime triangulation is oriented and we assign the complex conjugate $\overline{\omega(a, b)}$ to a triangle with orientation opposite to the above. We also assign a normalization of $1/|G|$ to every vertex,

$$\bullet \quad \rightarrow \quad \frac{1}{|G|} \; . \tag{4.290}$$

The 2-2 Pachner move in Eq. (4.11) can be equipped with G-labels,



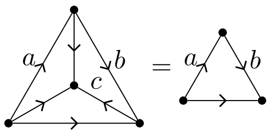
$$. \tag{4.291}$$

This yields the following equation for the weight ω,

$$\omega(a, b)\omega(ab, c) = \omega(a, bc)\omega(b, c) \; . \tag{4.292}$$

We see that this makes ω a group 2-cocycle $\omega \in Z^2(G, U(1))$, since it is equivalent to $d\omega = 1$ using the derivative in Eq. (2.63). In other words, ω is a cocycle operation $Z^1(G) \rightarrow Z^2(U(1))$. This is no surprise, since Pachner moves are precisely obtained by cutting the boundary of a 3-simplex in two halves, so the failure to obey Pachner move invariance equals the value of $d\omega$ on the 3-simplex. Note that the $1/|G|$

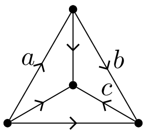
normalization is necessary for the 1-3 Pachner move to hold,



$$\rightarrow \quad \frac{1}{|G|}\sum_c \omega(a,bc)\omega(b,c)\overline{\omega(ab,c)} = \omega(a,b)\ . \tag{4.293}$$

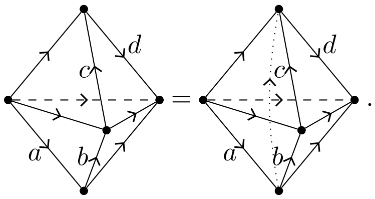
Each individual summand on the left-hand side is equal to the right-hand side, which is an immediate consequence of Eq. (4.292). Since we sum over $|G|$ times this same summand, we have to normalize by $1/|G|$. Also note that Eq. (4.292) implies the 2-2 Pachner move with any choice of branching structure, since we can arbitrarily move ω factors between the left and right side. Concrete formulas and classifications of group cocycles (and more general cohomology operations) can be found in the mathematical literature to a great extent. Also note that, as discussed in general in the beginning of Section 4.16, taking the logarithm makes Eq. (4.292) a system of linear equations over the ring $\mathbb{R}/\mathbb{Z}$, which can be solved computationally efficiently.

Next, let us consider $i = 1$ and $n = 3$. Again we restrict ourselves to branching-structure triangulations. There is one G-element on each edge, and the configurations of these elements are 1-cocycles fulfilling the cocycle constraint in Eq. (4.288) at every triangle. To each tetrahedron, we assign a weight



$$\rightarrow \quad \omega(a,b,c)\ . \tag{4.294}$$

As shown, the G-elements a, b, c on the edges 01, 12, and 23 determine the G-elements on all other edges through the cocycle constraint, so the weight depends not on six but only on three G-elements. We also need to assign an orientation and impose Hermiticity: If the tetrahedron has the opposite orientation to Eq. (4.294), we instead associate the complex conjugate $\overline{\omega}$. Again, we assign a $1/|G|$ normalization to every vertex as shown in Eq. (4.290). The Pachner move in Eq. (4.18) becomes



$$\tag{4.295}$$

Again, all other G-elements in the equation are determined from a, b, c, and d. This yields the equation

$$\omega(a,b,cd)\omega(ab,c,d) = \omega(a,b,c)\omega(b,c,d)\omega(a,bc,d)\ . \tag{4.296}$$

For example, on the left-hand side, the 01, and 12 edge of the bottom tetrahedron have labels a, b, and the 23 edge has label cd, which explains the arguments of the first ω in the equation. This equation is identical to stating that ω is a group 3-cocycle in

$H^3(G, U(1))$, since the equation above is equal to $d\omega = 1$ using the definition of the coboundary in Eq. (2.62) for $j = 3$. In other words, ω defines a cocycle operation $H^1(G) \rightarrow H^3(U(1))$. Again, this is no surprise, since the 2-3 Pachner move arises from decomposing the boundary of the 4-simplex into two pieces, and the boundary $d\omega$ sums the value of ω on the 3-simplices of the 4-simplex. The according models are known as *Dijkgraaf-Witten theory* [58], and have later been rewritten as Hamiltonian models known as *twisted quantum doubles* [82]. Note that these models are a special case of the general block-diagonal 3D tTS, described in Section 4.14, where the set of irreps is G, and the multiplicities are

$$N_k^{ij} = \delta_{ij,k} \;, \quad i^* = i^{-1} \;, \quad 1 = 1 \;. \tag{4.297}$$

The generalization to $i = 1$ and $n = 4$ is straight forward, and the state sum is determined by a group 4-cocycle $\omega(a, b, c, d)$. The models are again equivalent to discrete gauge theory, or twisted quantum doubles in $3 + 1$ dimensions, and a special case of the general block-diagonal 4D tTS where the edge labels are in G, and the face labels are trivial.

Let us briefly discuss the case of $i = 2$ and $n = 4$. There is one G-label at every triangle, for G an abelian group, and the overall G-configuration forms a G-valued cellular 2-cocycle a. That is, at every tetrahedron, there is a constraint enforcing $da = 0$,

$$i_{012} - i_{013} + i_{023} - i_{123} = 0 \;, \tag{4.298}$$

denoting the G-label in the face x by i_x. All G-labels on a tetrahedron are thus determined by the labels at the triangles

$$012, 123, 234, 013, 124, 024 \;. \tag{4.299}$$

Specifically, we have

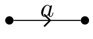
$$\begin{aligned} i_{023} &= i_{013} + i_{123} - i_{012} \;, \\ i_{134} &= -i_{123} + i_{234} + i_{124} \;, \\ i_{014} &= i_{012} + i_{024} - i_{124} \;, \\ i_{034} &= -i_{013} - i_{123} + i_{234} + i_{012} + i_{024} \;. \end{aligned} \tag{4.300}$$

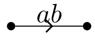
So at every 4-simplex, we have a weight

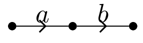
$$\omega(i_{012}, i_{123}, i_{234}, i_{013}, i_{124}, i_{024}) \;. \tag{4.301}$$

For every configurations on a 5-simplex, these weights have to satisfy an according equation. Solutions to these equations have been completely classified and can be expressed in terms of a known cohomology operation, namely the *Pontryagin square*. In the literature, these cohomology models are known as *discrete 2-form gauge theory* [90, 55]. They are a special case of the general block-diagonal 4D tTS where all edge labels are trivial and the face labels are valued in G.

Finally, the trivial case $i = 1$ and $n = 1$ is slightly degenerate. It is determined by



$\rightarrow \quad \omega(a) \;,$


$=$


$\rightarrow \quad \omega(ab) = \omega(a)\omega(b) \;.$ (4.302)

For topological invariance, it would suffice to demand

$$\sum_a \omega(a) = \frac{1}{|G|} \sum_a \omega(a) \sum_b \omega(b) \; . \tag{4.303}$$

So in this rather trivial case, the cocycle condition is stronger than necessary for topological invariance.

Let us consider some concrete examples for $i = 1$ and $n = 3$. First, consider the cyclic group $G = \mathbb{Z}_l$. A set of group 3-cocycles covering all cohomology classes is given by

$$\omega(a, b, c) = e^{\frac{2\pi i}{l^2} p\overline{a}(\overline{b}+\overline{c}-\overline{b+c})} \; , \tag{4.304}$$

for $0 \leq p < l$. The model with $l = 2$ and $p = 0$ is the *toric code*, and $l = 2, p = 1$ is the *double-semion model*.

Second, consider $G = S_3 = \mathbb{Z}_3 \rtimes \mathbb{Z}_2$. We will accordingly denote the group elements by pairs (a, b) with $0 \leq a \leq 2$, and $0 \leq b \leq 1$. The multiplication and inverse are then given by

$$(a, b) \cdot (a', b') = (a + (-1)^b a', b + b'), \qquad (a, b)^{-1} = (-(-1)^b a, b) \; . \tag{4.305}$$

The full set of 3-cocycles is given by

$$\begin{aligned} &\omega((i_0, j_0), (i_1, j_1), (i_2, j_2)) \\ &\quad = e^{\left(2\pi i p\left(\frac{1}{4} j_0(\overline{j_1}+\overline{j_2}-\overline{j_1+j_2})+\frac{1}{9}(-1)^{j_1+j_2} i_0(\overline{(-1)^{j_0} j_1}+\overline{j_2}-\overline{(-1)^{j_0} j_1+j_2})\right)\right)} \; , \end{aligned} \tag{4.306}$$

for $0 \leq p < 6$.

4.16.2 G-set cohomology models

In this section we will consider a variant of cohomology models that we refer to as *G-set models*. In fact, we will show that G-set models are equivalent to ordinary cohomology models. The reason we introduce them here is that they arise from compactifications of higher-dimensional (ordinary) cohomology models when classifying their defects in Chapter 6. As the name suggests, G-set models are defined with respect to a *G-set*, that is, a finite set A equipped with a right action of G,

$$\triangleleft : A \times G \to A \;: \qquad (a \triangleleft g) \triangleleft h = a \triangleleft (gh) \; . \tag{4.307}$$

In $n > 2$ dimensions, these models are robust if A is transitive, that is, it is not a nontrivial disjoint union of smaller G-sets. In this case, A is isomorphic to the set of right cosets $H\backslash G$ with action

$$Hg \triangleleft i = Hgi \; , \tag{4.308}$$

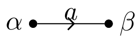
where

$$H = \mathrm{Stab}(X) := \{h \in G : X \triangleleft h = X\} \tag{4.309}$$

is the *stabilizer* of some fixed but arbitrary $X \in A$. Note that since A is transitive, for every $X' \in A$ there is g such that $X' = X \triangleleft g$, and then the stabilizer H' of X'

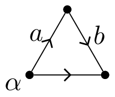
is given by $H' = g^{-1}Hg$. $H\backslash G$ and $(g^{-1}Hg)\backslash G$ define isomorphic G-sets, which is consistent with the arbitrariness of the choice of X.

Just like ordinary cohomology models, G-set cohomology models have one G-element associated to every edge of a triangulation, satisfying the cocycle constraint in Eq. (4.288) at every triangle. Additionally, there is one A-element at every vertex, satisfying the constraint



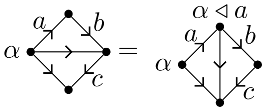
$$\rightarrow \quad \alpha \triangleleft a = \beta \tag{4.310}$$

at every edge. In $1+1$ dimensions, we have one weight



$$\rightarrow \quad \omega^{\alpha}(a,b) \tag{4.311}$$

at every triangle. As we can see, the A-elements at the other two vertices are determined by α through Eq. (4.310), so ω depends on only one and not three A-elements. We still have a $1/|G|$ normalization at every vertex as shown in Eq. (4.290). The 2-2 Pachner move in Eq. (4.11) has labels



$$\tag{4.312}$$

This yields the following equation for the weight ω,

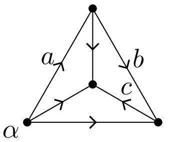
$$\omega^{\alpha}(a,b)\omega^{\alpha}(ab,c) = \omega^{\alpha}(a,bc)\omega^{\alpha\triangleleft a}(b,c)\ . \tag{4.313}$$

We will refer to a weight ω fulfilling this equation as a *G-set 2-cocycle*. Mathematically, these are equivalent to group 2-cocycles $\omega \in H^2(G, U(1)^A)$ where $g \in G$ acts on the left G-module $U(1)^A$ by

$$(g \triangleright \phi)(x) = \phi(x \triangleleft g^{-1})\ . \tag{4.314}$$

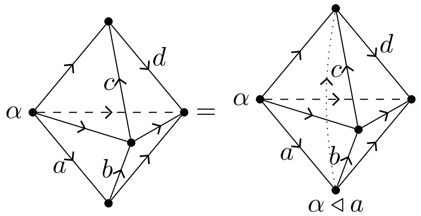
It should be noted that this action is from the left since it is given by precomposing $\phi \in U(1)^A$ with the action on A from the right.

In $2+1$ dimensions, we have a weight



$$= \omega^{\alpha}(a,b,c) \tag{4.315}$$

at every tetrahedron. The 2-3 Pachner move in Eq. (4.18) becomes



(4.316)

$$\rightarrow \quad \omega^{\alpha}(a,b,cd)\omega^{\alpha}(ab,c,d) = \omega^{\alpha}(a,b,c)\omega^{\alpha\triangleleft a}(b,c,d)\omega(a,bc,d) \ .$$

Again, we will call this a G-set 3-cocycle, and it is equivalent to a group 3-cocycle with module $U(1)^A$. The generalization to higher dimensions is straight forward.

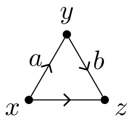
We will now show that for a transitive G-set, such G-set models are equivalent to ordinary cohomology models defined by some $\omega \in Z^n(H,U(1))$, where H is the stabilizer of some fixed $X \in A$. To this end, provide a two mappings from G-set models to ordinary models and vice versa. We first give this equivalence for $n = 2$, and then straight-forwardly generalize it to other dimensions.

We start with the simpler of the two mappings, which constructs an ordinary H model $\tilde{\omega}$ from a G-set model ω. $\tilde{\omega}$ is obtained from ω by restricting the A-label to X and the G-labels to H,

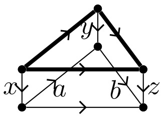
$$\tilde{\omega}(a,b) = \omega^X(a,b) \quad \forall a,b \in H \ . \tag{4.317}$$

Since the α-label in Eq. (4.312) is fixed, $\tilde{\omega}$ fulfills Eq. (4.291) if ω fulfills Eq. (4.312).

Next, let us show how to vice versa construct a G-set model $\hat{\omega}$ with $A = H\backslash G$ from an ordinary model $\omega \in Z^2(H,U(1))$. To this end, we take the cartesian product of the triangulation with a single edge $[0,1]$, and construct a state sum on this "thickened" triangulation as follows: We assign G-elements to all edges, which are constrained to form a 1-cocycle on the thickened triangulation. The G-elements on the "top" edges with coordinate 1 are restricted to H. To every top triangle with coordinate 1, we assign ω as a weight. We then project this thickened state sum back onto the original triangulation,

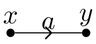


$:=$

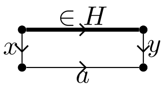


$= \omega(xay^{-1}, ybz^{-1}) \ .$ (4.318)

As one can see, this yields a state sum with an additional G-element on every vertex, corresponding to the vertical edges of the thickened triangle, and weights as shown above. Note that the elements on the top edges, drawn in thick, are determined by the other edge labels via the cocycle constraint. The fact that these elements are restricted to H yields a constraint,



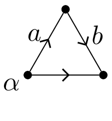
$\rightarrow$



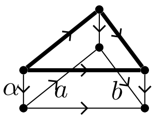
$\rightarrow \quad xay^{-1} \in H \ .$ (4.319)

The new state sum almost looks like a G-set model, but the vertex labels are not valued in $H\backslash G$ but G, and the constraint above is not an action since x and a do not determine y. This can be fixed by realizing that the state sum is invariant under mapping $x \to hx$ where $x \in G$ is the label of a single vertex v, for any $h \in H$. This will multiply h to the H-elements of all the edges adjacent to v in the top layer of the thickened triangulation in Eq. (4.318). Since this is a gauge transformation of the ordinary cohomology model defined by ω, it leaves the state sum invariant. Thus, we can reduce the G-labels at every vertex to $H\backslash G$-labels. The weights for a configuration of $H\backslash G$-labels are obtained by picking a fixed representative $R(\alpha) \in \alpha \subset G$ for each $\alpha \in H\backslash G$. We find

$$\omega^\alpha(a,b) =$$



$$:=$$



$$= \omega(xay^{-1}, ybz^{-1}) \ , \qquad (4.320)$$

where x, y, and z are now shortcuts for

$$x \coloneqq R(\alpha), \quad y \coloneqq R(\alpha \triangleleft a), \quad z \coloneqq R(\alpha \triangleleft ab) \ , \qquad (4.321)$$

Let us now straight-forwardly generalize the equivalence between G-set and ordinary cohomology models to other dimensions. In $0+1$ dimensions, can map a G-set model to an ordinary model using

$$\tilde{\omega}(a) \coloneqq \omega^X(a) \forall a \in H \ , \qquad (4.322)$$

and vice versa using

$$\hat{\omega}^\alpha(a) \coloneqq \omega(xay^{-1}) \ , \qquad (4.323)$$

still referring to the short-hand notation in Eq. (5.75). In $2+1$ dimensions, we have

$$\tilde{\omega}(a,b,c) \coloneqq \omega^X(a,b,c) \forall a,b \in H \ , \qquad (4.324)$$

and vice versa,

$$\hat{\omega}^\alpha(a,b,c) \coloneqq \omega(xay^{-1}, ybz^{-1}, zcw^{-1}) \ , \qquad (4.325)$$

with $w = R(\alpha \triangleleft abc)$ in addition to Eq. (5.75).

In contrast to G-set vertex labels, we can also introduce *free* vertex labels valued in some finite set B that are not restricted by a G-action. Such free vertex labels will become relevant mostly for $0+1$-dimensional state sums arising from compactifications when classifying lower-dimensional defects. Such a $0+1$-dimensional free-label state sum associates to every edge a weight,

$$\mu \bullet\!\xrightarrow{x}\!\bullet \nu \qquad \to \qquad \omega(x)^\nu_\mu \ , \qquad (4.326)$$

without any constraints for $\mu, \nu \in B$. Topological invariance is enforced by

$$\mu \bullet\!\xrightarrow{x}\!\bullet\!\xrightarrow{y}\!\bullet \nu \ = \ \mu \bullet\!\xrightarrow{xy}\!\bullet \nu \qquad \to \qquad \sum_\kappa \omega(x)^\kappa_\mu \omega(y)^\nu_\kappa = \omega(xy)^\nu_\mu \ . \qquad (4.327)$$

Mathematically, this means that ω is a linear representation of the group G. Analogous to the discussion in Section 4.16.1, this condition is not strictly speaking necessary for topological invariance, but state sums arising from compactifications later will obey this condition. Next, we can mix free and G-set labels,

$$\alpha\mu \bullet\!\xrightarrow{x}\!\bullet \beta\nu \quad \rightarrow \quad \omega^{\alpha}(x)^{\nu}_{\mu} \ , \tag{4.328}$$

where $\mu, \nu \in B$ are free, but $\beta \in A$ is determined through $\alpha \in A$ through Eq. (4.310). Topological invariance becomes

$$\begin{aligned} &\alpha\mu \bullet\!\xrightarrow{x}\!\bullet\!\xrightarrow{y}\!\bullet \nu \ = \ \alpha\mu \bullet\!\xrightarrow{xy}\!\bullet \nu \\ \rightarrow \quad &\sum_{\kappa} \omega^{\alpha}(x)^{\kappa}_{\mu} \omega^{\alpha \triangleleft x}(y)^{\nu}_{\kappa} = \omega^{\alpha}(xy)^{\nu}_{\mu} \ . \end{aligned} \tag{4.329}$$

Now, models with free and (transitive) G-set labels are equivalent to models with only free labels,

$$\begin{aligned} \tilde{\omega}(a)^{\nu}_{\mu} &\coloneqq \omega^{X}(a)^{\nu}_{\mu} \quad \forall a \in H \ , \\ \hat{\omega}^{\alpha}(a)^{\nu}_{\mu} &\coloneqq \omega(xay^{-1})^{\nu}_{\mu} \ . \end{aligned} \tag{4.330}$$

Chapter 5

Boundaries, domain walls, and phases

In this chapter, we will equip our topological fixed-point models with a topological boundary. We will also study domain walls between two different models. *Invertible* domain walls will be instrumental in defining *phases* as equivalence classes of tTS models. Finally, we will also discuss invertible and robust topological phases.

5.1 Boundaries in $2+1$D

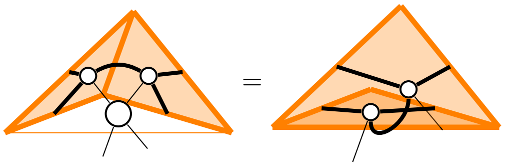
We have already prematurely discussed 1+1-dimensional boundaries and domain walls in Sections 4.10 and 4.11. The generalization to higher dimensions is straight forward. In this section, we will look at boundaries in $2+1$ dimensions.

5.1.1 tTS models

We start by constructing the *3D boundary tTS* that extends the 3D tTS. Let us again begin with a simplified picture without branching structure. The 3D boundary tTS assigns a tensor network to every 3-triangulation with boundary. To every boundary triangle, we associate a copy of a 4-index tensor variable,

. (5.1)

The three thick lines are indices of a new bond dimension variable, and are contracted between adjacent domain wall triangles. The thin line is contracted with the tensor at the adjacent bulk tetrahedron below. The boundary can be deformed through bulk Pachner moves acting near the boundary. In addition, we need Pachner moves of the boundary triangulation itself, which need to be padded with bulk such that both sides can be adjacent to the same tetrahedra. For example, the 2-2 boundary Pachner move is given by



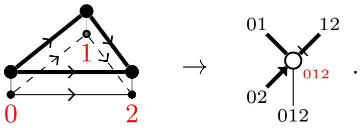
. (5.2)

On both sides there are two boundary triangles. On the left-hand side, these are additionally padded with a bulk tetrahedron, which ensures that the space boundary on both sides is compatible. In other words, this move removes a bulk tetrahedron two of whose faces are at the boundary.

After this simplified picture, let us add the usual technicalities. Rather than triangulations with boundary, we use extended cellulations (c.f. 2.4) of the following type,

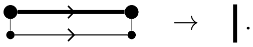
$$a:(3,\varnothing)\ ,\quad b:(2,\bullet)\ . \tag{5.3}$$

As a consequence, we will draw a boundary triangle separately from the bulk triangle it is adjacent to. Apart from drawing things differently, the difference between triangulations with boundaries and extended cellulations is that it is now possible for two boundary triangles to be adjacent to the same bulk triangle, without being separated by bulk tetrahedra. The boundary triangulation itself also carries a branching structure. It suffices to take a single boundary 2-cell representative, namely a triangle attached to a bulk triangle with the same branching structure. To this 2-cell representative, we associate the following tensor variable,



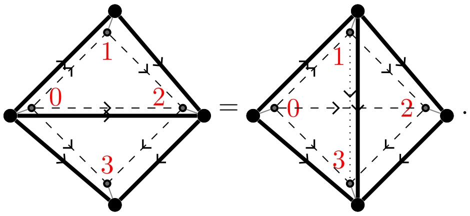
(5.4)

It may be instructive to geometrically think of the left-hand side as a triangle-prism-like volume, as we tried to indicate by dashing edges at the back. As shown, there is one index for each boundary edge, and one for the bulk triangle. The bond dimension variable of these indices corresponds to the same boundary 1-cell representative as the boundary tensor variable in $1+1$ dimensions in Eq. (4.158),



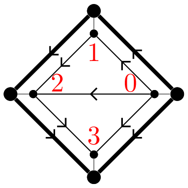
(5.5)

After adding the branching structure, the 2-2 boundary Pachner move in Eq. (5.2) becomes



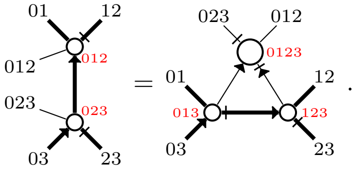
(5.6)

On the left-hand side there are two boundary triangles glued at a "horizontal" edge, depicted together with the adjacent bulk triangles and edges. On the right-hand side, there are two boundary triangles glued at a "vertical" edge. Additionally there is a bulk tetrahedron adjacent to both bulk triangles in the back. The space boundary on both sides is given by the following 2-dimensional extended cellulation,



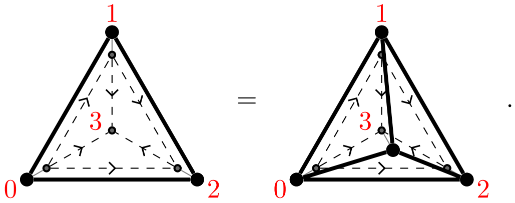
, (5.7)

where we have rotated the drawing such that the back is now in the front. In terms of tensor-network diagrams, we get the following equation:



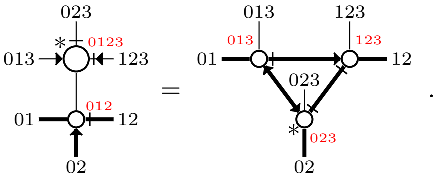
. (5.8)

Note that we are ignoring weight matrices in this section: To be fully precise, we should introduce one weight at every bulk edge and one at every boundary vertex. We also have 1-3 boundary Pachner moves, such as



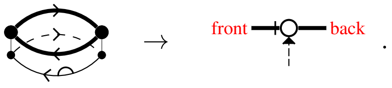
. (5.9)

This move consists of one boundary triangle and one bulk tetrahedron on the left, and three boundary triangles only on the right. In terms of tensor-network diagrams, we have



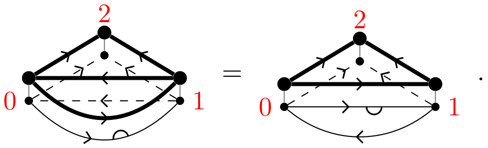
. (5.10)

Instead of imposing all boundary 2-2 and 1-3 Pachner moves with all choices of edge directions, there exists a simplified tTS with a single 2-2 Pachner move along with simpler tensor variables and axioms. The tensor variables and moves of this simplified tTS are given by the same 2-cells and recellulations as for the simplified 2D tTS. There is one additional tensor variable, the boundary 2-gon,



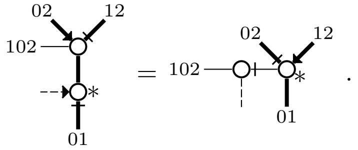
. (5.11)

The boundary 2-gon has one index for each edge attached to other boundary 2-cells, and one index for its bulk 2-gon attached to a bulk 3-cell. The 01 triangle flip move in Eq. (5.12) becomes the following *boundary* 01 *triangle flip move*,



(5.12)

On the left-hand side, we have a boundary triangle and a boundary 2-gon. On the right-hand side, there is a boundary triangle padded with a bulk 01 triangle flipper as in Eq. (4.110). The corresponding tensor-network equation is

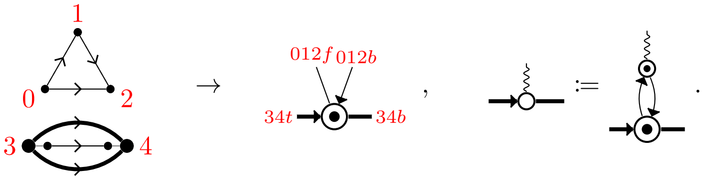


(5.13)

In general, the moves of the simplified 2D tTS correspond to the same 3-cells as the tensor variables of the 3D tTS. The moves of the simplified boundary tTS are obtained by taking the 2D-tTS moves padded with the according bulk 3-cells.

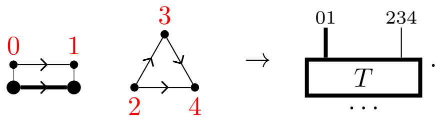
5.1.2 Block-diagonal form

In this section, we will generalize the block diagonalization procedure from Section 4.14 to the boundary. The first step is again to decompose the boundary into robust components. We first notice that the robust components of the bulk also yield different components of the boundary. To this end, we consider the 2D-atTS model defined in Eq. (4.223), and this time we construct a boundary of this model as follows,



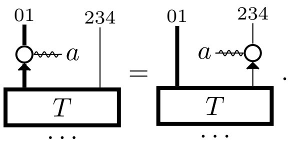
(5.14)

The left-hand side represents a 3-ball with a hole in the middle. Its space boundary cellulation consists of the two 012 triangles in the interior, and the two 34 edges on the exterior part of the boundary. The remaining 2-ball of the exterior boundary is physical boundary. The cellulation of the bulk and physical boundary does not matter due to the topological invariance of the tTS. Now, we apply the isometry from Proposition 4 to every edge index of the tensor T associated to some extended cellulation with space boundary, such that the index decomposes into one irrep and one multiplicity part. Now, consider a boundary edge and a bulk triangle inside the space boundary of the same connected component of an extended cellulation, and the associated tensor T,



(5.15)

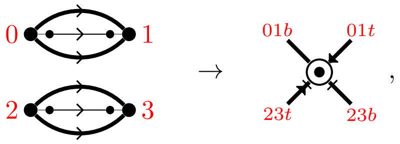
Gluing one boundary edge in Eq. (5.14) to either the boundary edge 01 or 23 above has the same effect of introducing a 2-triangle hole inside the bulk, so we have



(5.16)

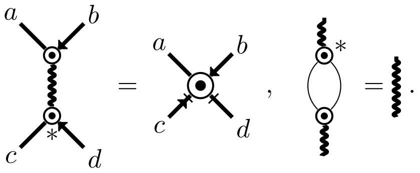
Thus, using Proposition 6, the we can reduce T to a tensor $\widetilde{T}$ where all irrep indices of the bulk triangles and boundary edges inside one connected component are fused to a single one.

In other words, if we decompose the bulk as a direct sum of robust components, then also the boundary gets decomposed as a direct sum. However, the summands are not necessarily robust. In the next step, we decompose each summand of the boundary further into robust components. To this end, for each summand, we define a tensor,



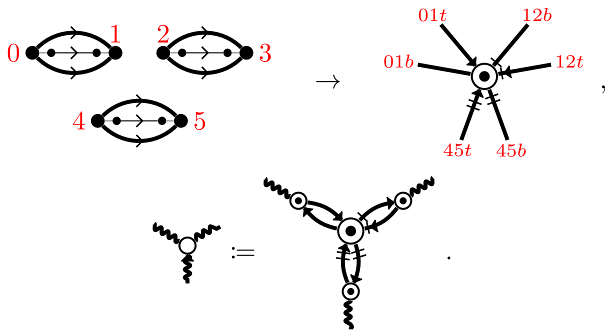
(5.17)

associated to a 3-ball whose boundary consists of the drawn extended cellulation, and an annulus like physical boundary. This tensor defines a Hermitian projector from the bottom to the top indices, and thus, we can define an isometry,



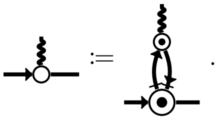
(5.18)

Next, we can construct a 2D-atTS model as follows,



(5.19)

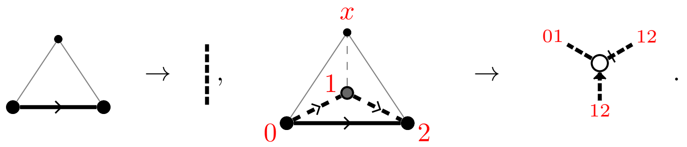
The left-hand side shows a 3-ball with three space-boundary disks embedded inside an otherwise physical boundary. A boundary for this 2D-atTS model is given by



(5.20)

So again, applying the isometry from Proposition 4, each index at a boundary edge in the space boundary decomposes into a pair (i, α), consisting of one Eq. (5.19)-irrep i, and one multiplicity label α of dimension m_i. As usual, consider T associated to an extended space boundary, and two boundary edges inside the same connected physical-boundary component. Gluing one boundary edge of Eq. (5.17) to either of the two edges has the same effect of introducing a space-boundary hole inside the physical boundary. Thus, we can use Proposition 6 to reduce T to a simpler tensor $\widetilde{T}_i$ with only one irrep index i for every connected boundary component. For every fixed choice of i, $\widetilde{T}_i$ defines a robust 3D-boundary-tTS model.

We proceed for each of the obtained robust 3D-boundary-tTS models by constructing the following *cone 2D-tTS model*. The bond dimension and tensor variable determining this 2D-tTS model correspond to the following new boundary 1-cell and 2-cell representatives,



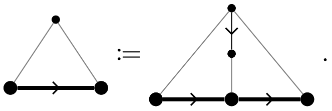
(5.21)

The *boundary triangle* 1-cell representative on the left consists of two boundary vertices only, and the *boundary tetrahedron* 2-cell representative on the right consists of three boundary edges with the representative on the left. Note that this model can be viewed as the result of the following *cone mapping*. Topologically, the mapping maps a 2-manifold to a 3-manifold with boundary (and with a singularity) by

$$M \to (M \times [0,1]) \big/_{a\times 1 \sim b \times 1 \forall a,b \in M} . \tag{5.22}$$

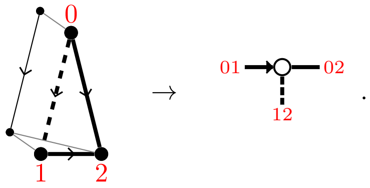
The boundary on the right is given by $M \times 0$. Geometrically, we start with a 2-cellulation/triangulation. Then, we turn every vertex or edge or face into a boundary vertex or edge or face. We add a single bulk vertex x to which the whole boundary is attached.

In order to define the new bond dimension variable, we cellulate the 1-cell representative in Eq. (5.21), for example using two of the standard 1-cell representatives in Eq. (5.5),



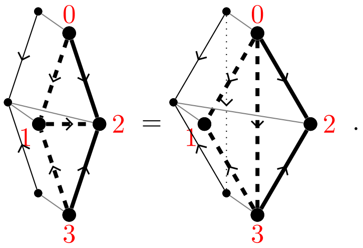
(5.23)

From this cellulation, we define the *boundary triangle adapter* as a 2-cell representative whose boundary consists of both sides of Eq. (5.23) glued together,



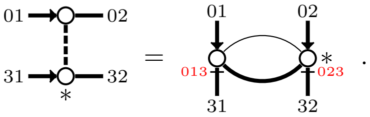
(5.24)

It can be defined via



(5.25)

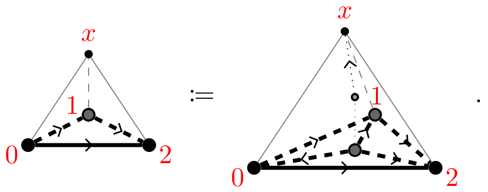
On the left-hand side, there are two boundary triangle adapters glued at their boundary-triangle edges. On the right-hand side, there are two standard boundary triangles, which are glued at one standard boundary edge, but also at their bulk triangles. In terms of tensor networks, this becomes



(5.26)

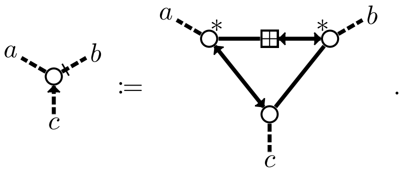
Analogous to Eq. (4.230), interpreting the boundary triangle adapter as an operator E, this equation becomes $EE^\dagger = A$, and we can pick any right-invertible E that solves this equation.

Equipped with the boundary triangle adapter, we can build the tensor on the right-hand side of Eq. (5.21) by the following triangulation,



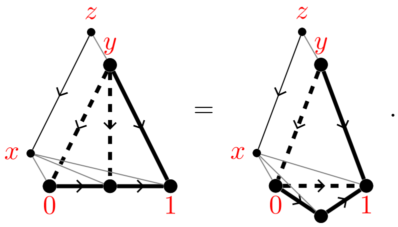
(5.27)

The right-hand side consists of three boundary triangle extenders glued in a ring such that every pair shares one standard boundary edge. In terms of tensor networks, the cone 2D-tTS model can be constructed by,



(5.28)

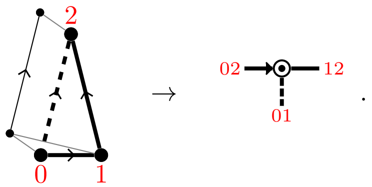
Note that we are using a new weight matrix which for a boundary edge introduces a weight on its adjacent bulk edge. It can be defined from the bulk edge weights. Next, we define two different boundaries for the cone 2D-tTS model, which we refer to as the 0 *and* 1 *cone 2D-boundary-tTS models*. The tensor variable of the 1 cone 2D-boundary-tTS model is simply the boundary triangle adapter from Eq. (5.24). One can explicitly check that this mapping is compatible with the moves. For example, the 2-1 boundary Pachner move in Eq. (4.161) maps to a recellulation,



(5.29)

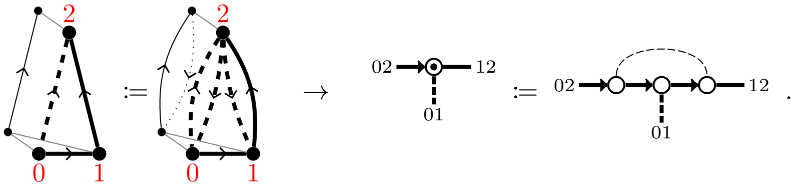
On the left-hand side, we have two of the boundary triangle adapters next to each other glued at a standard boundary edge. On the right-hand side, there is a single boundary triangle adapter together with one boundary tetrahedron from Eq. (5.21), glued together at the boundary triangle edge at the bottom. Note that this 2D-boundary-tTS model can also be viewed as the result of an extension of the cone mapping: We add a single boundary vertex y, a single adjacent bulk vertex z, and a single bulk edge connecting x (from the cone mapping) and z.

The tensor variable determining the 0 cone 2D-boundary-tTS model is associated to the following new boundary 2-cell representative,



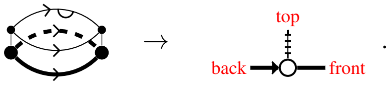
(5.30)

It can be constructed by wrapping two acyclic boundary 2-gons from Eq. (5.11) around the boundary triangle adapter,



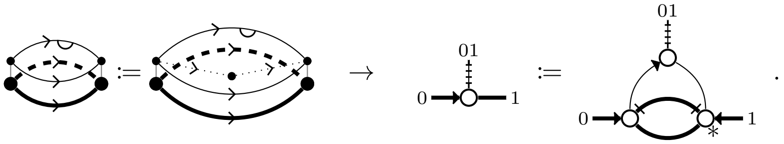
(5.31)

Finally, we can define the following boundary for the banana 2D-tTS model in Eq. (4.227), which we refer to as the *boundary banana 2D-boundary-tTS model*. The defining tensor variable is the following acyclic boundary 2-gon,



(5.32)

It can be constructed via



(5.33)

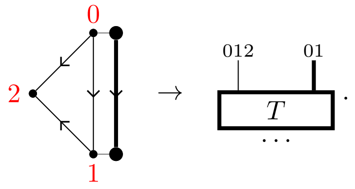
The right-hand side consists of two boundary triangles that are glued together at two of their edges, and one acyclic 2-gon adapter that is glued at both its triangle faces.

The 0 and 1 cone 2D-boundary-tTS models and the boundary banana 2d-boundary-tTS model share the same bond dimension variable, namely the one corresponding to a standard boundary edge. All these boundaries commute for reasons analogous to the discussion around Eq. (4.240). Thus after applying the isometry from Proposition 3, the boundary bond dimension decomposes into a tuple of indices

$$i, a, j, b, k, c, \mu \; . \tag{5.34}$$

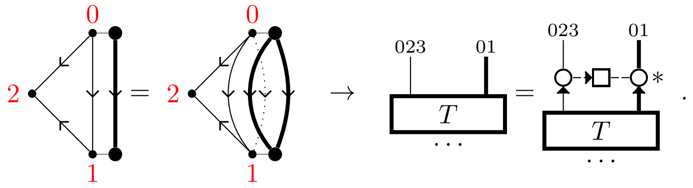
where i, j, and k are irrep labels of the 0 and 1 cone 2D-boundary-tTS model, and the boundary banana 2d-boundary-tTS model, respectively. a, b, and c are the according internal indices, and μ is the common multiplicity index.

Now, consider the tensor T associated to some extended cellulation with space boundary, and an adjacent pair of one bulk triangle and one boundary edge inside the space boundary, like



(5.35)

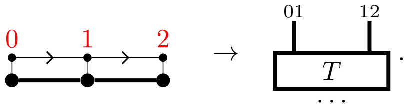
Imagine gluing the volume in Eq. (4.228) to the bulk triangle, and the volume in Eq. (5.32) to the edge, while gluing the two 2-gon faces of these volumes. This gluing does not change the space boundary and thus can be undone by a bulk recellulation, for example,



(5.36)

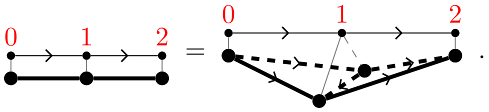
Thus, we can apply Proposition 5 and find that T is determined by a simpler tensor $\widetilde{T}$ where the 01 irrep index of the 012 triangle and the 01 irrep index of the 01 edge are fused together, and the according internal indices have disappeared.

Next, we can look at a tensor T associated to an extended cellulation, and two adjacent boundary edges inside the space boundary, for example



(5.37)

Now, glue one of the volumes in Eqs. (5.24) and (5.30) to each of the two boundary edges, while gluing the triangle boundary edges of these volumes. This gluing changes neither the space-boundary cellulation nor the internal topology,

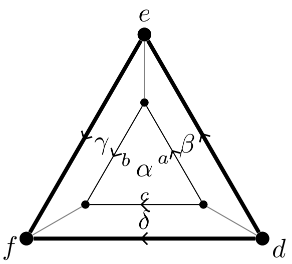


(5.38)

Again, we can apply Proposition 5, stating that T is determined by some simpler tensor $\widetilde{T}$. The 1 cone irrep index of the 01 boundary edge is fused with the 0 cone irrep index of the 12 boundary edge, and the according internal indices have disappeared.

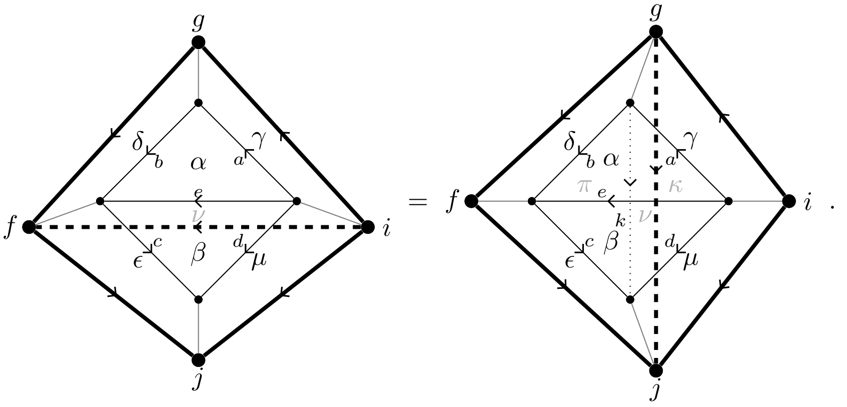
Now we apply the above reasoning to all bulk edges and boundary vertices inside the space boundary of some extended cellulation. We will end up with a tensor $\widetilde{T}$ with one banana irrep index at every bulk edge, one cone irrep index at every boundary vertex, and multiplicity indices at all bulk triangles and boundary edges. The dimension of the boundary edge multiplicity index is M_k^{ij}, depending on its 0 cone irrep i, 1 cone irrep k, and banana irrep j.

If we apply this procedure to the boundary tensor in Eq. (5.4) we get the following 10-index tensor,



$$\rightarrow \quad L_d^{abf}{}^{c\alpha\delta}_{e\beta\gamma}\,. \tag{5.39}$$

The 2-2 boundary Pachner move in Eq. (5.6) becomes



(5.40)

In terms of tensors, this yields

$$\sum_\nu L_i^{abf}{}^{e\alpha\nu}_{g\gamma\delta} L_i^{ecj}{}^{d\beta\mu}_{f\nu\epsilon} = \sum_{\nu,\pi,\kappa,k} d_k L_i^{akj}{}^{d\kappa\mu}_{g\gamma\nu} L_g^{bdj}{}^{k\pi\nu}_{f\delta\epsilon} F_d^{abc}{}^{k\kappa\pi}_{e\alpha\beta}\,. \tag{5.41}$$

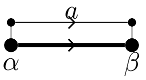
Let us quickly relate the obtained block-diagonal form of the boundary 3D tTS to notions in the literature. The tensor L above is close related to a *module category* of the unitary fusion category related to F. Ref. [97] constructs boundaries of Levin-Wen models from such module categories in a Hamiltonian context.

5.1.3 Cohomology models

After introducing general fixed-point models for boundaries in the previous sections, we will now look at a more restricted family of models that is described through cohomology. Specifically, we will define two different but equivalent types of boundaries for bulk cohomology models introduced in Section 4.16. The first type are *G-set boundaries*, which are more versatile and closer to the block-diagonal form of general tTS models. The second type are *subgroup boundaries*, which are easier to find.

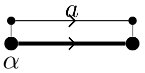
G-set and free boundaries

In this section, we will define G-set boundaries and free boundaries. For pedagogical reasons, we start with the simple case of $1+1$ dimensions. Since non-trivial robust phases $1+1$ dimensions neither exist in the bulk nor at the boundary, these models are not directly physically relevant. However, non-trivial phases exist when we add symmetries or fermions, and also the models are important for defining $0+1$-dimensional defects in higher-dimensional models. For these boundaries, the bulk is given by a cohomology model defined for a group 2-cocycle ω. As the name suggests, *G-set boundaries* are defined with respect to a right G-set A^∂. These boundaries are state sums that associate one A^∂-element to every boundary vertex of an triangulation with boundary, which we view as an extended cellulation. At every boundary edge (c.f. Eq. (5.5)), we impose the following constraint,



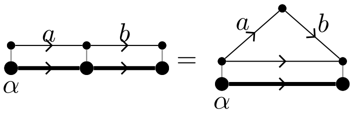
$$\rightarrow \quad \alpha \triangleleft a = \beta \; . \tag{5.42}$$

Additionally, we assign a weight



$$\rightarrow \quad \psi^\alpha(a) \; . \tag{5.43}$$

As usual, there is an orientation, and boundary edges with opposite orientation are assigned the complex conjugate $\overline{\psi^\alpha(a)}$. The boundary Pachner move in Eq. (4.160) equipped with G and A^∂-elements looks like



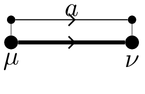
$$. \tag{5.44}$$

Thus, we get a consistency equation

$$\psi^\alpha(a)\psi^{\alpha\triangleleft a}(b) = \psi^\alpha(ab)\omega(a,b) \; . \tag{5.45}$$

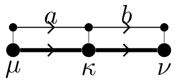
Let us interpret ω as a G-set 2-cocycle that is constant in A, that is, $\omega^\alpha(a,b) := \omega(a,b) \forall \alpha \in A^\partial$. Then ψ is a G-set 1-cochain such that $d\psi = \omega$.

We will not actually use $1+1$-dimensional G-set boundaries, but have merely introduced them for pedagogical purposes. Instead, we will use more general *free boundaries*, where we drop the constraint in Eq. (5.42). The labels at the boundary vertices are now elements of some finite set B without a G-action. At every boundary edge, we have a weight

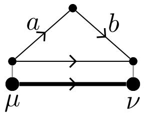


$$\rightarrow \quad \psi(a)^\nu_\mu \; , \tag{5.46}$$

where now $\mu \in B$ and $\nu \in B$ are now independent labels. The boundary Pachner move in Eq. (4.160) has labels



$$= \tag{5.47}$$



Thus, we get a consistency equation

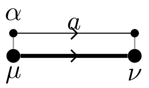
$$\sum_\kappa \psi(a)^\kappa_\mu \psi(b)^\nu_\kappa = \omega(a,b)\psi(ab)^\nu_\mu \ . \tag{5.48}$$

Mathematically, this equation precisely states that ψ is a *projective representation* of G, where the 2-cocycle ω describes the projectivity. Equivalently, ψ is a linear representation of the *ω-twisted G-group algebra*, with basis $\{e_g\}_{g\in G}$ and multiplication

$$e_g \cdot e_h = \omega(g,h)e_{gh} \ . \tag{5.49}$$

Of special interest are the *irreducible* (projective) representations, since they correspond to robust boundaries.

We can also consider free boundaries for a G-set bulk state sum. These associate the following weight to a boundary edge,

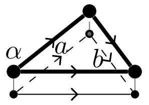


$$\to \quad \psi^\alpha(a)^\nu_\mu \ . \tag{5.50}$$

Pachner move invariance yields

$$\sum_\kappa \psi^\alpha(a)^\kappa_\mu \psi^{\alpha\triangleleft a}(b)^\nu_\kappa = \psi^\alpha(ab)^\nu_\mu \omega(a,b) \ . \tag{5.51}$$

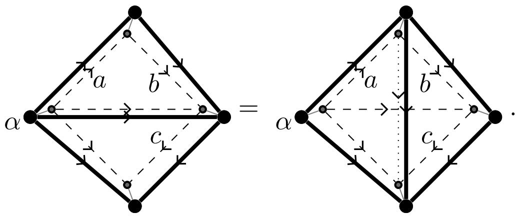
Next, let us look at boundaries of $2+1$-dimensional cohomology models defined by a group G and a group 3-cocycle $\omega \in Z^3(G,U(1))$. For $2+1$ dimensions, G-set boundaries analogous to the ones defined above for $1+1$ dimensions are natural. In fact, it is known that these models capture all boundary phases. Free boundaries like in $1+1$ dimensions do not really make sense in $2+1$ dimensions. For robust boundaries, the G-set A^∂ has to be transitive, and thus isomorphic to $H^\partial\backslash G$ where $H^\partial := \mathrm{Stab}(X^\partial)$ is the stabilizer of an arbitrary but fixed element $X^\partial \in A^\partial$. Like in $1+1$ dimensions, the boundary state sum assigns an A^∂-element to every boundary vertex, and imposes the constraint in Eq. (5.42) at every boundary edge. Furthermore, at every boundary triangle (c.f. Eq. (5.4)), we have a weight



$$\to \quad \psi^\alpha(a,b) \ . \tag{5.52}$$

Note that the G-label at the third bulk edge and the A^∂-labels at the other two boundary vertices are determined from α, a, and b via Eq. (4.288) and Eq. (5.42). Due to

Hermiticity, we are assigning the complex conjugate to boundary triangles with opposite orientation from Eq. (5.4). The boundary Pachner move in Eq. (5.6) carries the following G and A^∂-labels,



(5.53)

This corresponds to the following equation for the weight ψ:

$$\psi^\alpha(a,b)\psi^\alpha(ab,c) = \omega(a,b,c)\psi^\alpha(a,bc)\psi^{\alpha\triangleleft a}(b,c)\ . \qquad (5.54)$$

Again, we can interpret ω as a G-set group 3-cocycle that trivially depends on the A^∂-element, $\omega^\alpha(a,b,c) := \omega(a,b,c)\forall\alpha \in A^\partial$. Then, ψ is a G-set group 2-cochain such that $d\psi = \omega$. We will show how to find solutions to this equations can be reduced to solving $d\psi = \omega$ for ordinary H^∂-valued group cocycles in the following sections. Note that G-set boundaries in $2+1$ dimensions are in the same form as general block-diagonal 3D-tTS models defined in Section 5.1.2.

The generalization of G-set boundaries to $3+1$ dimensions is straight forward. We assign a weight $\psi^\alpha(a,b,c)$ to each boundary tetrahedron. The boundary Pachner move yields an equation

$$\psi^\alpha(a,b,c)\psi^\alpha(b,c,d)\psi^{\alpha\triangleleft a}(a,bc,d) = \psi^\alpha(a,b,cd)\psi^\alpha(ab,c,d)\omega(a,b,c,d)\ , \quad (5.55)$$

that can be compactly stated as $d\psi = \omega$ when we interpret ψ and ω are G-set group cochains. Note however, that in $3+1$ dimensions, these boundaries do not capture all robust boundary phases but only a subset thereof. For example, we can obtain a new boundary by stacking any standalone $2+1$-dimensional robust model on top of the boundary. By this alone, there are infinitely many robust boundary phases.

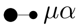
Finally, let us also consider the rather trivial case of $0+1$ dimensions. For a bulk with free vertex labels, a boundary is defined by

$$\bullet\!\!-\!\!\cdot\ \mu \quad \rightarrow \quad \psi^\mu\ , \quad \bullet\!\!-\!\!\cdot\ \mu\ = \bullet\!\!-\!\!\cdot\!\xrightarrow{x}\!\cdot\ \mu \quad \rightarrow \quad \psi^\mu = \sum_\nu \omega(x)^\mu_\nu \psi^\nu \forall x\ . \quad (5.56)$$

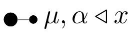
The solutions to this equation form a vector space. Mathematically, this vector space is the subspace corresponding to trivial representation, if we decompose the G-representation ω into irreps. Note that in fact for topological invariance, it would suffice to demand

$$\psi^\mu = \frac{1}{G}\sum_{\nu,x}\psi^\nu\omega(x)^\mu_\nu = \sum_\nu P^\mu_\nu\psi^\nu\ , \quad P^\mu_\nu := \frac{1}{G}\sum_x \omega(x)^\mu_\nu\ . \qquad (5.57)$$

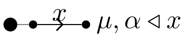
P^μ_ν is the projector onto the subspace corresponding to the trivial representation, so this is equivalent to Eq. (5.56). For a bulk with G-set and free vertex labels, a boundary is defined by



$$\to \quad \psi^{\mu,\alpha}\ ,$$



$$=$$



$$\to \quad \psi^{\mu,\alpha\triangleleft x} = \sum_\nu \omega^\alpha(x)^\mu_\nu \psi^{\nu,\alpha} \forall x\ . \tag{5.58}$$

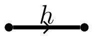
Subgroup boundaries

In the previous section, we have seen that boundaries for cohomology models can be naturally defined via G-sets. In this section, we describe a second way to define boundaries which we will refer to as *subgroup boundaries*. While subgroup boundaries are equivalent to G-set ones, they have the advantage that they are easier to define: They are given by simply restricting the group labels at the boundary to some subgroup $H^\partial \subset G$, and adding boundary weights given by a 2-cochain ψ such that $d\psi = \omega$ holds restricted to H^∂. So, finding subgroup boundaries boils down to finding trivializing cochains for cocycles on H^∂, which is easier than finding solutions to, for example, Eq. (5.54).

We will start by defining subgroup boundaries in $1+1$ dimensions, then in $2+1$ dimensions, and finally comment on their straight forward generalization to arbitrary spacetime dimensions. The bulk is a $1+1$-dimensional bulk cohomology model determined by a group 2-cocycle ω. A subgroup boundary state sum can be defined for any subgroup $H^\partial \subset G$ and a H^∂ group 1-cochain ψ such that $d\psi = \omega|_{H^\partial}$, or more explicitly

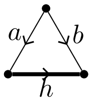
$$\psi(a)\psi(b) = \psi(ab)\omega(a,b) \quad \forall a, b \in H^\partial\ . \tag{5.59}$$

The state sum is defined on conventional triangulations with boundary, which we do *not* implement as extended triangulations. So in drawings, we will not distinguish between a boundary edge and its adjacent bulk edge, and in particular configurations like in Eq. (4.159) are not possible. Every every boundary edge carries a H^∂-element. At the same time, such a boundary edge (which we will draw in thick), has an associated weight



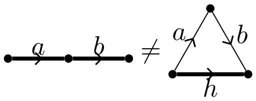
$$\to \quad \psi(h), \quad h \in H^\partial\ . \tag{5.60}$$

As shown, we draw boundary edges as thick lines as opposed to edges in the interior. Note that the cocycle constraints still holds at triangles involving a boundary edge, for example,



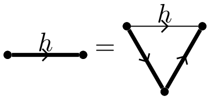
$$\to \quad ah = b\ , \tag{5.61}$$

despite the fact that h is restricted to H^∂. Additionally, there is a normalization of $1/|H^\partial|$ associated to every boundary vertex. The simplicity of the boundary comes at the expense of having a slightly more complicated topological invariance. A move like in Eq. (4.160) for the G-set boundary with just two boundary edges on one side does *not* hold,



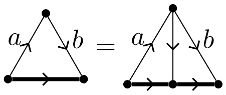
$$\to \quad \delta_{a\in H^\partial}\psi(a)\delta_{b\in H^\partial}\psi(b) \neq \delta_{ab\in H^\partial}\omega(a,b)\psi(ab) \; . \tag{5.62}$$

Note that the above is not to be understood as an inequality that holds, but what we rather mean is that the equality does not follow from Eq. (5.59). The problem is that on the left-hand side, a and b are restricted to H^∂, but on the right-hand side, only their product is. Instead, moves like



$$\to \quad \delta_{h\in H^\partial}\psi(h) = \frac{1}{|H^\partial|}\sum_{g\in G}\delta_{g\in H^\partial}\psi(g)\delta_{g^{-1}h\in H^\partial}\psi(g^{-1}h)\overline{\omega(g,g^{-1}h)}\forall h\in G \; , \tag{5.63}$$

and,



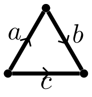
$$\begin{aligned} \to \quad & \delta_{ab\in H^\partial}\psi(ab)\omega(a,b) \\ = & \frac{1}{|H^\partial|}\sum_{c\in G}\delta_{ac\in H^\partial}\psi(ac)\delta_{c^{-1}b\in H^\partial}\psi(c^{-1}b)\omega(a,c)\overline{\omega(c,c^{-1}b)} \; . \end{aligned} \tag{5.64}$$

do follow from Eq. (5.59).

Let us now look at subgroup boundaries in $2+1$ dimensions. Again, a subgroup boundary is defined for $H^\partial \subset G$ and a H^∂ group 2-cochain ψ with $d\psi = \omega|_{H^\partial}$, that is,

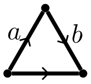
$$\psi(a,b)\psi(ab,c)\omega(a,b,c) = \psi(a,bc)\psi(b,c) \quad \forall a,b,c\in H^\partial \; . \tag{5.65}$$

Again, we restrict the group labels at the boundary edges to H^∂, and impose the cocycle constraint also for triangles involving boundary edges, such as as in Eq. (5.61) and also for the triangles of the boundary,



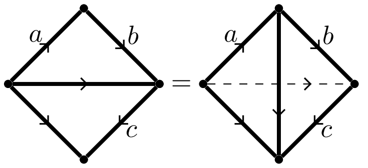
$$\to \quad ab = c \; . \tag{5.66}$$

Furthermore, we assign ψ as a weight to the boundary triangles,



$$= \psi(a,b) \; . \tag{5.67}$$

Again, the topological moves that follow from Eq. (5.65) are more complicated than for G-set boundaries. In fact, the main move in Eq. (5.6) does still hold for the subgroup boundary,



$$. \qquad (5.68)$$

This is because all involved edges labels are constrained to H^∂ and so this is just Eq. (5.65). However, the move in Eq. (5.9), which holds the G-set boundary, does not hold for the subgroup boundary. This is because the three edges adjacent to the central vertex are constrained to H^∂ on the right, but can take values in all of G on the left. This can be fixed by padding the right-hand side with three bulk tetrahedra analogous to Eq. (5.64).

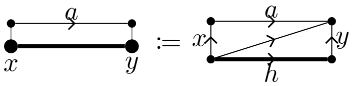
Equivalence of G-set and subgroup boundaries

In this section, we show that G-set and subgroup boundaries are in fact equivalent. More precisely, G-set boundaries with a transitive G-set A^∂ are equivalent to subgroup boundaries for which H^∂ is the stabilizer of some fixed element $X^\partial \in A^\partial$. To show this, we will give a way to construct a subgroup from a G-set boundary and vice versa, and show that these mappings are inverses of another up to phase equivalence. This is analogous to how we showed, for example, equivalence between the non-simplified and simplified 2D tTS in Section 4.7.

As a first example, let us discuss the equivalence of G-set and subgroup boundaries in $1+1$ dimensions. We start with the simpler mapping, constructing a subgroup boundary $\tilde{\psi}$ from a transitive G-set boundary ψ. To this end, we realize that a G-set boundary has the following gauge symmetry around a boundary vertex, for any $\gamma \in G$: First, we change the element a at every edge directed towards the vertex by $a \rightarrow a\gamma$, and by $a \rightarrow \gamma^{-1}a$ for an edge directed away from the vertex. Then, we change the A^∂-label at the vertex itself by $\alpha \rightarrow \alpha \triangleleft \gamma$. Since every such gauge symmetry only affects a single boundary vertex, we can use it to fix all the A^∂-labels to a fixed $X^\partial \in A^\partial$. So $\tilde{\psi}$ is obtained from ψ by simply setting the A^∂-labels to X^∂, which automatically restricts the G-labels at the boundary to its stabilizer H,

$$\tilde{\psi}(a) := \psi^{X^\partial}(a) \ , \quad \forall a \in H^\partial \ . \qquad (5.69)$$

Next, let us discuss the converse mapping, constructing a transitive G-set boundary $\hat{\psi}$ with $A^\partial = H^\partial \backslash G$ from a subgroup boundary ψ. We start by padding each subgroup boundary edge with a bulk rectangle. Then we project all the variables and weights of these rectangles down to a single boundary edge, yielding a state sum with weights

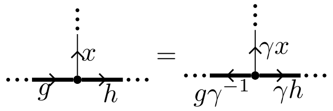


$$= \omega(x,a)\overline{\omega(xay^{-1},y)}\psi(xay^{-1}) \ , \qquad (5.70)$$

and one G-label at every boundary vertex. At each such edge, the labels satisfy a constraint

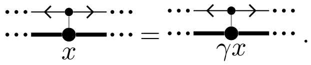
$$xay^{-1} \in H \ . \qquad (5.71)$$

It is easy to see that the moves for the G-set boundary such as Eq. (4.160) follow from the subgroup boundary moves such as in Eq. (5.63) and Eq. (5.64), after we plug in Eq. (5.70). The resulting boundary state sum already looks close to a G-set boundary, but the boundary vertices have labels in G instead of $A^\partial = H^\partial \backslash G$, and the constraint in Eq. (5.71) does not uniquely determine y from x and a. In order to obtain a proper G-set boundary $\hat{\psi}$, we realize that the subgroup boundary has a gauge symmetry acting on the edges adjacent to a boundary vertex,



$$\forall \gamma \in H^\partial \ , \tag{5.72}$$

where the equality is for the weights at the surrounding triangles and boundary edges. On the left of Eq. (5.70), the same gauge symmetry is only acting on a single vertex label,



(5.73)

Thus, we can remove the gauge freedom by replacing the vertex labels such as $x, y \in G$ by right cosets such as $\alpha, \beta \in H^\partial \backslash G$. The constraint $\beta = \alpha \triangleleft a$ of the G-set boundary then directly follows from $xa = hy$ and $h \in H^\partial$ on the right. The weight $\hat{\psi}$ after removing the gauge can be obtained from the weight before by choosing for each $\alpha \in H^\partial \backslash G$ a standard representative

$$R(\alpha) \in \alpha \ , \tag{5.74}$$

and using shorthands

$$x := R(\alpha), \qquad y := R(\beta) = R(\alpha \triangleleft a) \ , \tag{5.75}$$

like in Eq. (4.321). Using these shortcuts in Eq. (5.70), we obtain

$$\hat{\psi}^\alpha(a) := \overline{\omega(x,a)}\omega(xay^{-1}, y)\psi(xay^{-1}) \ . \tag{5.76}$$

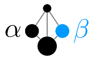
Finally, we need to show that the mappings are inverses of each other up to phase equivalence. In fact, we already made sure that this is the case by the way we constructed the mappings, via pulling back geometric mapping and fixing of a gauge symmetry. However, it is instructive to show this more formally. To this end, we show that applying both mappings consecutively yields an equivalent boundary in the same phase. More precisely, we need to show this for both orderings in which we compose the two mappings. It is easy to see that if we first transform a subgroup boundary into a G-set one, and go back to a subgroup one, we end up with the same subgroup boundary again,

$$\tilde{\hat{\psi}}^\alpha(g) = \psi^\alpha(g) \ . \tag{5.77}$$

However, going from a G-set to a subgroup and then back to a G-set boundary does not yield the same boundary, but one that is in the same phase or G-set cohomology class. In order to see this, we consider an invertible corner (c.f. Section 6.4) between the G-set boundaries ψ, and

$$\hat{\tilde{\psi}}^\alpha(a) := \omega(x,a)\overline{\omega(xay^{-1},y)}\psi^{X^\partial}(xay^{-1}) \;. \tag{5.78}$$

This corner is given by some η,

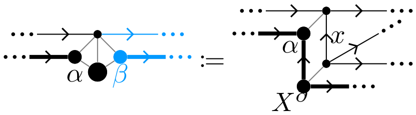


$$\rightarrow \quad \eta^\alpha \delta_{\beta,H^\partial R(\alpha)} \;. \tag{5.79}$$

Here, $\alpha \in A^\partial$, $\beta \in H^\partial\backslash G$, and $R(\alpha) \in G$ is a standard representative for each $\alpha \in A^\partial$ such that

$$X^\partial R(\alpha) = \alpha \;. \tag{5.80}$$

Also, we mark $\hat{\tilde{\psi}}$ in blue. This corner is given by the following mapping



$$\rightarrow \quad \eta^\alpha := \overline{\psi^{X^\partial}(R(\alpha))} \;. \tag{5.81}$$

using the shorthand $x := R(\alpha)$ like in Eq. (5.75). Using this corner η, we find that ψ and $\hat{\tilde{\psi}}$ are related by

$$\hat{\tilde{\psi}}^{H^\partial R(\alpha)}(g) = \psi^\alpha(g)\overline{\eta^\alpha}\eta^{\alpha\triangleleft g} \;. \tag{5.82}$$

If we identify A^∂ and $H^\partial\backslash G$, and interpret η, ψ, and $\hat{\tilde{\psi}}$ as G-set 0-cochains and 1-cochains, this becomes

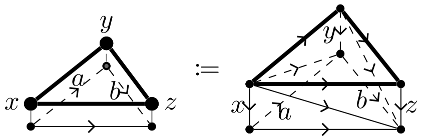
$$\hat{\tilde{\psi}} = \psi + d\eta \;, \tag{5.83}$$

where the sum actually denotes component-wise multiplication. In other words, ψ and $\hat{\tilde{\psi}}$ are in the same cohomology class of G-set 1-cochains relative to a fixed 2-cocycle ω as boundary.

Next, let us generalize the mappings between G-set and subgroup boundaries to $2+1$ dimensions. Again, constructing a subgroup boundary $\tilde{\psi}$ from a G-set boundary ψ is simple, namely by fixing the G gauge symmetry acting on the labels around a boundary vertex. As a result, we restrict all A-elements to a fixed element $X^\partial \in A$, which restricts the boundary G-elements to H,

$$\tilde{\psi}(a,b) := \psi^{X^\partial}(a,b) \quad \forall a,b \in H \;. \tag{5.84}$$

Vice versa, we can construct a transitive G-set boundary $\hat{\psi}$ from a subgroup boundary ψ as follows. Analogous to the $1+1$-dimensional case, we start with a boundary state sum obtained by padding each subgroup boundary triangle with a triangle prism,



$$. \tag{5.85}$$

This state sum has one G-element at every boundary edge and every boundary vertex. The H boundary group labels on the right are determined by the other labels $\alpha, a, b \in G$ so they do not appear on the left. It is easy to see that the resulting state sum obeys the moves for the G-set boundary in Eq. (5.6) and Eq. (5.9). This follows form the subgroup boundary moves such as in Eq. (5.68), after we plug in Eq. (5.85). The state sum is almost a G-set boundary, but the vertex elements are in G and not $H^\partial \backslash G$. Just as in the $1+1$-dimensional case, this can be resolved by fixing a gauge symmetry. This gauge symmetry involves all edges adjacent to a fixed boundary vertex for the subgroup boundary on the right, but only a single vertex G-element for the resulting state sum on the left. So fixing this gauge symmetry means replacing vertex G-elements like x, y, z by vertex elements $\alpha, \beta, \gamma \in H^\partial \backslash G$. Evaluating the right-hand side of Eq. (5.85) then yields

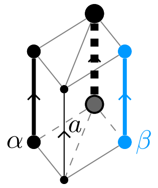
$$\hat{\psi}^\alpha(a,b) := \overline{\omega(x,a,b)}\omega(xay^{-1},y,b)\overline{\omega(xay^{-1},ybz^{-1},z)}\psi(xay^{-1},ybz^{-1}) \ , \tag{5.86}$$

using the shortcuts in Eq. (4.321).

Next, let us study how the subgroup and G-set boundaries actually correspond to the same boundary phase. Analogous to the $1+1$-dimensional case, we have

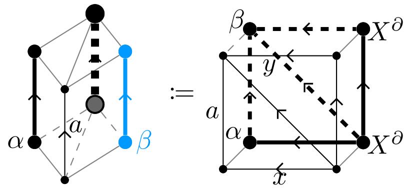
$$\tilde{\hat{\psi}}^\alpha(g,h) = \psi^\alpha(g,h) \ . \tag{5.87}$$

Vice versa $\hat{\tilde{\psi}}$ is not equal to ψ, but the two are separated by a 1-dimensional invertible corner,



$$\rightarrow \quad \eta^\alpha(a)\delta_{\beta,H^\partial R(\alpha)} \ . \tag{5.88}$$

The mapping via which this η can be obtained looks like the cartesian product of Eq. (5.81) with an edge. While in Eq. (5.81) the corner is mapped to a boundary edge, it is now mapped to a boundary rectangle that is triangulated with two boundary triangles,



$$\rightarrow \quad \eta^\alpha(a) = \psi^{X^\partial}(x,a)\overline{\psi^{X^\partial}(xay^{-1},y)} \ , \tag{5.89}$$

using the shorthands in Eq. (4.321). The relation between ψ and $\hat{\tilde{\psi}}$ is now

$$\hat{\tilde{\psi}}^\alpha(g,h) = \psi^\alpha(g,h)\eta^\alpha(g)\eta^{\alpha\triangleleft g}(h)\overline{\eta^\alpha(gh)} \ . \tag{5.90}$$

The generalization of the equivalence between G-set and subgroup boundaries to higher dimensions is straight-forward. To map a subgroup to a G-set boundary of an

n-dimensional spacetime bulk, pad the $n-1$-simplex with the $n-1$-simplex times an edge. The resulting n-cell can be triangulated using n n-simplices. The G-labels at the edges perpendicular to the boundary are set to standard representatives of the $A^\partial = H^\partial \backslash G$-labels of the G-set boundary. Evaluation of the spacetime volume yields an expression with n bulk weights ω and one subgroup-boundary weight ψ.

After relating G-set boundaries to conventional boundaries, we are ready to give some examples for G-set boundaries. As a first example, consider $G = \mathbb{Z}_l$ in the bulk with the cocycle in Eq. (4.304). Now, consider a subgroup boundary using the subgroup

$$H^\partial = \{xk\}_{0 \leq x < l/k} \sim \mathbb{Z}_{l/k} \;, \tag{5.91}$$

for some divisor k of l. $\omega|_{H^\partial}$ is in a trivial cohomology class if $pk = 0 \bmod l$, and in fact in this case, we have $\omega|_{H^\partial} = 1$. Since $H^2(\mathbb{Z}_{k/l}, U(1))$ is trivial there is only one boundary per subgroup, and we can set $\psi = 1$ for this boundary Now, we turn this subgroup boundary into a G-set boundary. We can label the cosets by

$$L_\alpha \coloneqq \{xk + \alpha\}_{0 \leq x < l/k} \;, 0 \leq \alpha < k \;, \quad L_\alpha \triangleleft a = L_{\alpha + a \bmod k} \;. \tag{5.92}$$

As standard representatives, we simply choose $R(\alpha) = \alpha$. With this, we have $x = \alpha$, $y = (\alpha + a) \bmod k$, and $z = (\alpha + a + b) \bmod k$ in Eq. (5.86), which becomes

$$\hat{\psi}^\alpha(a,b) = \overline{\omega(x,a,b)} = e^{-\frac{2\pi i p}{l^2}\overline{\alpha}(\overline{a}+\overline{b}-\overline{a+b})} \;. \tag{5.93}$$

Note that the second and third ω in Eq. (5.86) are equal to 1 since the first argument is in H^∂, and thus a multiple of k.

As a second example, consider $G = S_3$ with $\omega = 1$ as bulk as discussed around Eq. (4.305). There are four conjugacy classes of subgroups H^∂, namely

$$\{(0,0)\} \;, \quad \langle (0,1) \rangle \sim \mathbb{Z}_2 \;, \quad \langle (1,0) \rangle \sim \mathbb{Z}_3 \;, \quad S_3 \;. \tag{5.94}$$

In all four cases, the subgroup has a trivial second cohomology group. So since $\omega = 1$, we can set $\psi = 1$ without loss of generality, and the boundary is fully determined by the G-set A^∂ alone. For the trivial subgroup $H^\partial = \{(0,0)\}$, the cosets are just individual group elements, $A^\partial = \{(0,0)\} \backslash S_3 = S_3$, and the action is the regular action of S_3 on itself. For $H^\partial = \langle (0,1) \rangle$, there are three cosets with action

$$L_\alpha \coloneqq \{(\alpha,0), (-\alpha,1)\} \;, \quad 0 \leq \alpha < 3 \;, \quad L_\alpha \triangleleft (a,b) = L_{(-1)^b(\alpha+a)} \;. \tag{5.95}$$

For $H^\partial = \langle (1,0) \rangle$, there are three cosets with action

$$L_\alpha \coloneqq \{x, \alpha\}_{0 \leq x < 3} \;, \quad \alpha \in \{0,1\} \;, \quad L_\alpha \triangleleft (a,b) = L_{\alpha + b} \;. \tag{5.96}$$

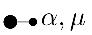
Finally, for $H^\partial = S_3$, there is only one coset, namely all of S_3, and the G-set is trivial.

Boundaries of cohomology models are discussed in various places in the physics literature. For example, Refs. [26] and [43] construct boundaries of quantum double models in $2+1$ dimensions corresponding to our subgroup boundaries are discussed. However, these references focus on Hamiltonian models on rigid spatial lattices via ribbon operators, which is very different from our approach.

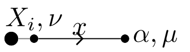
Efficient construction of G-set boundaries of G-set bulk

In this section, we will show how to explicitly construct boundaries for G-set bulk models, a task that will appear prominently when calculating/classifying lower-dimensional defects of cohomology models. Specifically, x-dimensional defects are in one-to-one correspondence with boundaries of $x{+}1$-dimensional state sums arising from compactification, and these state sums are naturally G-set cohomology models. To construct these boundaries, we decompose the G-set A into transitive orbits C_i, determine a representative $X_i \in C_i$, and the stabilizer $H_i = \mathrm{Stab}(X_i) \subset G$ of X_i. Then we look at the boundaries of the equivalent ordinary cohomology model, and construct from them boundaries of the original G-set model.

Let us start with $0+1$ dimensions. Even though this case is rather trivial, it is important for computing point-like defects in spacetime. More precisely, we consider a bulk model with both G-set and free vertex labels defined by $\omega^{\alpha}{}^{\nu}_{\mu}$ as in Eq. (4.328), and construct a boundary defined by $\psi^{\mu,\alpha}$ as in Eq. (5.58). For each transitive orbit C_i, we restrict ω to a bulk model $\tilde{\omega}$ with only free labels, via Eq. (4.330). Then we determine a boundary $\tilde{\psi}^{\mu}$ like in Eq. (5.56) for $\tilde{\omega}$. Finally, we find ψ from $\tilde{\psi}$ by padding $\tilde{\psi}$ with one bulk edge,



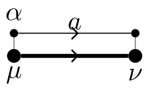
$:=$



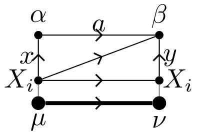
$$\rightarrow \quad \psi^{\alpha,\mu} = \sum_{\nu} \omega^{X_i}(x)^{\mu}_{\nu} \tilde{\psi}^{\nu} \ , \tag{5.97}$$

where $x := R(\alpha)$ is a representative of α like in Eq. (5.80).

Next, consider the case of $1+1$ dimensions, which is important for computing defects that are line-like in spacetime. More precisely, we consider a G-set bulk given by $\omega^{\alpha}(a,b)$ as in Eq. (4.311), and construct a free boundary defined by $\psi^{\alpha}(a)^{\nu}_{\mu}$ as in Eq. (5.50). For every transitive orbit C_i, we restrict the G-set 2-cocycle ω to an ordinary cohomology model $\tilde{\omega}$ via Eq. (4.317). Next, we consider the free boundaries of $\tilde{\omega}$, defined by $\tilde{\psi}(a)^{\nu}_{\mu}$ as in Eq. (5.48). Finally, we construct ψ from $\tilde{\psi}$ by padding it with a rectangle of ω bulk. When doing so, we restrict the bulk A^{∂}-labels at the boundary to X_i,



$:=$



(5.98)

$$\rightarrow \quad \psi^{\alpha}(a)^{\nu}_{\mu} := \omega^{X_i}(x,a)\overline{\omega^{X_i}(xay^{-1},y)}\tilde{\psi}(xay^{-1})^{\nu}_{\mu} \ ,$$

where x and y are chosen representatives of α and β as in Eq. (5.75).

Last, let us consider the case of $2+1$ dimensions. Here, free boundaries are not natural, but instead G-set boundaries represent the most general type of boundary for cohomology models. So we aim to construct G-set boundaries for a G-set bulk given by a G-set 3-cocycle $\omega^{\alpha}(a,b,c)$. Such a boundary is given by a G-set A^{∂} that is distinct from A and a weight $\psi^{\alpha,\beta}(a,b)$, where $\alpha \in A$ and $\beta \in A^{\partial}$. For every transitive orbit C_i, we restrict ω to an ordinary H_i group 3-cocycle $\tilde{\omega}$ via Eq. (4.324). Then we consider the subgroup boundaries of $\tilde{\omega}$, given by a subgroup $H_i^{\partial} \subset H_i$, and a H_i^{∂}

2-cochain $\tilde{\psi}$ such that $d\tilde{\psi} = \tilde{\omega}|_{H^\partial}$. Finally, ψ is obtained from $\tilde{\psi}$ via

$$\begin{aligned}&\psi^{\alpha,\beta}(a,b)\\ &\quad= \omega^{X_i}(x,a,b)\overline{\omega^{X_i}(xay^{-1},y,b)}\omega^{X_i}(xay^{-1},ybz^{-1},z)\tilde{\psi}(xay^{-1},ybz^{-1})\ ,\end{aligned} \tag{5.99}$$

where we used shortcuts,

$$x = R(\beta)\ , \quad y = R(\beta \triangleleft a)\ , \quad z = R(\beta \triangleleft ab)\ . \tag{5.100}$$

The G-set is

$$A^\partial = H_i^\partial \backslash G\ . \tag{5.101}$$

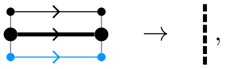
The generalization to higher dimensions is straight-forward, but one should keep in mind that for $n > 3$, not every boundary can be represented as a G-set boundary (or by using cohomology at all).

5.2 Domain walls in $2+1$D

Next, let us look at domain walls in $2+1$ dimensions, analogous to Section 4.11 and 5.1. Domain walls are defined on extended cellulations of the following type,

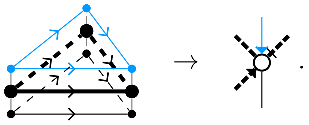
$$a : (3, \varnothing)\ , \quad b : (3, \varnothing)\ , \quad c : (2, ● \quad ●)\ . \tag{5.102}$$

Like in Section 4.11, we could describe domain walls as boundaries of the tTS model obtained via stacking the two tTS models on either side of the domain wall. In this case, we associate one additional bond dimension to a standard domain-wall edge,



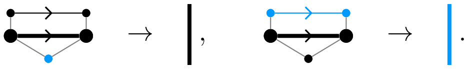
(5.103)

and one additional tensor variable to a standard domain-wall triangle,



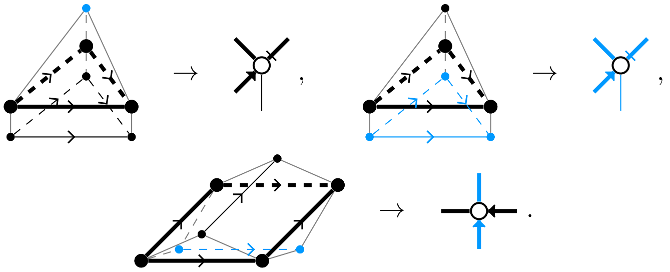
(5.104)

The main move for this tTS is 2-2 Pachner move of domain-wall triangles, similar to Eq. (5.6), just that now the right-hand side is padded with one tetrahedron for each adjacent bulk. However, it is possible to give an equivalent "simplified" form of the domain-wall tTS that is easier to handle in practice. This consists of two new bond dimension variables corresponding to new domain-wall 1-cell representatives,



(5.105)

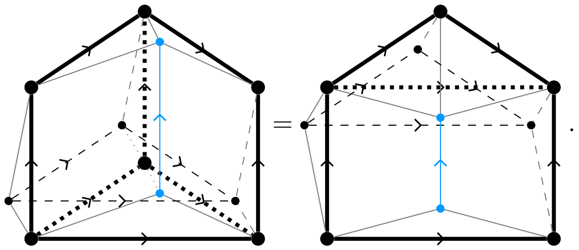
Using these bond dimension variables, we also have three new tensor variables corresponding to new domain-wall 2-cell representatives,



(5.106)

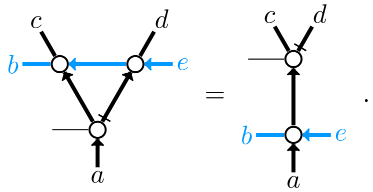
We will refer to the last tensor variable as the *commutor*. Note that the 1-cell representatives in Eq. (5.105) are the same as in Eq. (4.169), and the 2-cell representatives in Eq. (5.106) are the same as for the moves of the 2D domain-wall tTS in Section 4.11. For example, the commutor corresponds to the move in Eq. (4.170).

Next, let us discuss the moves of the simplified 3D domain-wall tTS. The first two tensor variables define boundaries for the a or the b copy of the 3D tTS, respectively. That is, the simplified 3D domain-wall tTS includes the moves of the 3D boundary tTS for each of these tensor variables. In addition, we have the following move involving the commutor above,



(5.107)

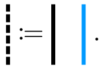
On the left-hand side, we have two commutors and one a boundary triangle. On the right-hand side, there is only one commutor and one a boundary triangle. In terms of tensor-network diagrams, we have



(5.108)

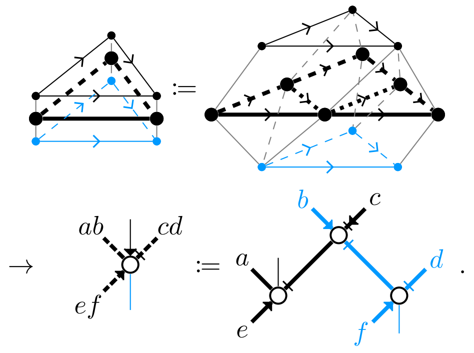
Furthermore, we have the same move with a and b exchanged, that is, with the black boundary tensor replaced by the blue one.

The non-simplified domain-wall liquid can be obtained from the simplified one as follows. First, the standard domain-wall edge in Eq. (5.103) can be triangulated as shown in Eq. (4.174), leading to a mapping of bond dimensions



(5.109)

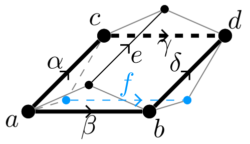
The standard domain-wall triangle in Eq. (5.104) can be obtained from the tensor variables of the simplified tTS as follows,



(5.110)

The right-hand side consists of one commutor, one a boundary triangle, and one b boundary triangle.

The block-diagonal form of the domain wall consists of two block-diagonal boundaries sharing the same set of irreps. In addition, the block-diagonal form of the commutor is



$$\rightarrow \quad C^{aed\;\;b\beta\delta}_{f\;\;\;\;\;c\alpha\gamma} \,. \tag{5.111}$$

5.3 Cone extension boundary

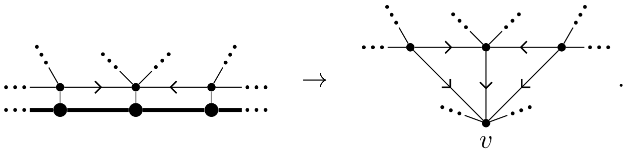
In this section, we will show that any tTS model admits a “standard” topological boundary. The construction works on a purely diagrammatic level, and is simply given by a TS mapping from the nD boundary tTS to the nD tTS. We will call this TS mapping the *cone extension mapping* because its global effect is to extend and close off the boundary with a cone. That is, for a manifold M with boundary $X = \partial M$, we map

$$M \rightarrow M\big/_{a\sim b \forall a,b\in\partial M} \,. \tag{5.112}$$

That is, we identify all points in the boundary with one single point. Note that this only yields a (piece-wise linear) manifold if M is the n-ball. This will not turn out to be a problem, since our tensor-network diagrams still make sense on triangulations that are not manifolds. Actually, the fact that this is not a problem is at the root of why the tTS models discussed in the previous chapter can only represent topological orders that admit a topological boundary.

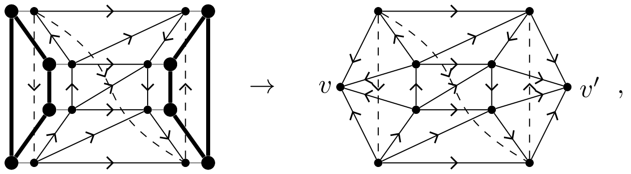
Let us now look at how the mapping is implemented in terms of cellulations. We start by adding a single vertex v to a triangulation with boundary. Then, for every i-simplex inside the boundary of M, we add an $i+1$-simplex spanned by the i-simplex and v. The directions of all the edges spanned by a boundary vertex and v are chosen to point towards v. In n dimensions, the boundary $n-1$-simplex is replaced by a bulk n-simplex, which defines the mapping. Note that we might also take one separate vertex v for every connected component of the boundary: The resulting tensor-network diagrams will not be different.

Let us explicitly consider the cone extension mapping in $1+1$ dimensions. Every boundary edge gets mapped to a bulk triangle, spanned by v and the bulk edge to which the boundary edge was adjacent,



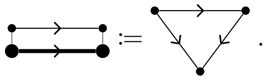
(5.113)

For example, when we apply this mapping to a triangulation of the annulus, we obtain a sphere,



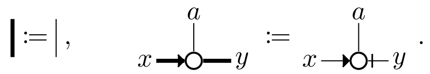
(5.114)

where we chose to draw two separate vertices v and v', one for each boundary circle. Formally, the mapping maps a boundary edge to a triangle,



(5.115)

In terms of diagrams, the mapping becomes

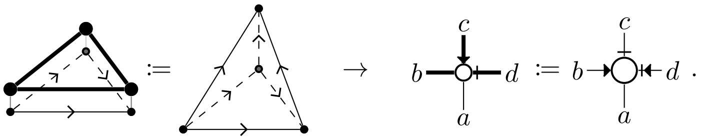


(5.116)

As we can see, the boundary bond dimension is mapped to the bulk bond dimension. For this to define a mapping, we have to derive the mapped boundary Pachner moves from the bulk Pachner moves, which is easy. For example, the mapping of Eq. (4.161) directly yields a 2-2 Pachner move.

As discussed in Section 4.10, boundary 2D tTS models are closely related to representations of the special Frobenius algebras defined by the bulk. In this context, the cone extension mapping in Eq. (5.116) uses the structure constants of the multiplication as those of the representation, which is known as the *regular representation*.

Next, let us consider the case of $2+1$ dimensions. The boundary triangle is mapped to a bulk tetrahedron,

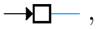


(5.117)

5.4 Phases and invertible domain walls

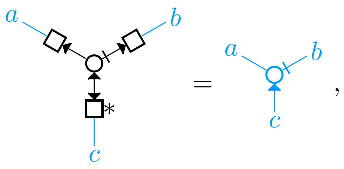
Topological phases of matter are equivalence classes of general models, given for example by a tensor-network path integral. The equivalence relation is the existence of a gapped continuous path, which is complicated and very hard to check for two given models. By restricting to zero-correlation-length path integrals, we have argued that we can resort to a much more tractable notion of equivalence: Namely, invertible domain walls as defined in Section 3.7. Since Chapter 4, we have restricted our models further by demanding combinatorial topological invariance, which implies zero correlation length. In this section, we will discuss how the notion of phases via invertible domain walls specializes to this context.

Just like for the non-topological case in Section 3.7, invertible domain walls are TS models. For a tTS $\mathcal{A}$, these invertible-domain-wall TS are extensions of two copies of $\mathcal{A}$, which we will label by a and b, respectively. Let us give a first simple example for such an invertible-domain-wall TS, namely local changes of basis, or conjugation by on-site unitaries. Since tTS tensor networks are defined on arbitrary triangulations, we need to insert resolutions of the identity at all bonds in the network, and cannot restrict to one particular direction on a square lattice like shown in Eq. (3.43). Let us show this invertible-domain-wall TS explicitly for the 2D tTS. The generalization to higher dimensions is straight forward. The TS is defined by two copies of the 2D tTS, where we will color b in blue, and an additional 2-index tensor variable,

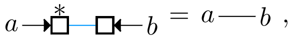


(5.118)

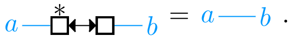
which is the unitary by which we conjugate. The TS also contains the three moves,



(5.119)

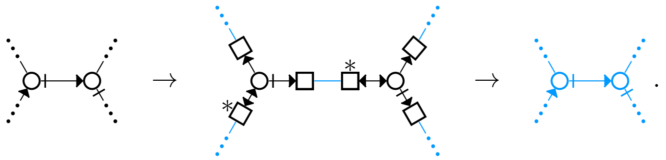


(5.120)



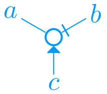
(5.121)

Let us now see how a model of this invertible-domain-wall TS allows us to transform an a 2D tTS tensor network into a b one: First we apply Eq. (5.120) backwards to every bond of the network. After this, every a triangle tensor is surrounded by three 2-index tensors. Then we apply Eq. (5.119) to every triangle tensor together with the surrounding 2-index tensors. After this, we have transformed the a network to a b network:



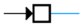
(5.122)

Let us now give a concrete example for two 2D-tTS models that can be shown to be in the same phase via a model of this invertible-domain-wall tTS. As the a model we take the δ-tensor model from Eq. (4.4). As a b model, we take the following parity-even tensor,



$$= \begin{cases} \frac{1}{\sqrt{2}} & \text{if } a+b+c=0 \mod 2 \\ 0 & \text{otherwise} \end{cases} . \tag{5.123}$$

We can easily verify that Eqs. (5.119), (5.120), and (5.121) hold if we make the following choice for the 2-index tensor,



$$= \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} . \tag{5.124}$$

This proves that the two models are in the same phase.

All of the above is really just saying that two models should be considered to be in the same phase if they differ by a change of basis at every index. Even though local changes of basis are a rather restricted ansatz to show phase equivalence, they already help with the classification to a great deal: Taking the quotient with respect to local changes of basis in general appears to reduce from a continuous family of tTS models to a discrete set of equivalence classes. However, two of these discrete equivalence classes can still be in the same phase via a more complicated invertible-domain-wall TS model. So we would like to find a “natural” or “universal” invertible-domain-wall TS that is sufficient to show phase equivalence of any tTS models. To this end, we invoke the second picture for invertible domain walls from Section 3.7: Instead of a sequence of tensor-network equations that map an a to a b tensor network, we imagine a domain wall separating an a and a b area that can be freely moved using the tensor-network equations. Furthermore, it is possible to create or fuse islands of b or a tensor network within a region of a or b tensor network, which we will call the *invertibility condition*. Without the invertibility condition, invertible domain walls thus are just topological domain walls. Thus, it is natural to use the standard nD domain-wall tTS described in Sections 4.11 and 5.2.

Thus, all that remains to construct a universal invertible-domain-wall TS is to implement the invertibility condition for the nD domain-wall tTS. The invertibility condition corresponds to invariance under further, topology-changing, moves, similar to the robustness condition described in Section 4.6. These moves still correspond to recellulations of extended cellulations, however, the topology on the left and right-hand side is different. Since we are already imposing invariance under all topology-preserving recellulations, these topology-changing moves are fully determined by the topology alone. More precisely, such a move is specified by one space-boundary extended manifold, and two different extended manifolds with this space boundary. To implement such a change in topology as a concrete move, we choose any extended cellulation of the common space boundary and any cellulation for each of the two sides. All other realizations of the topology-changing move can be derived from this single move, together with the topology-preserving moves of the (domain-wall) tTS. In practice, we want to implement the moves via the simplest extended cellulations that we can think of.

Concretely, the topology-changing moves implementing the invertibility condition for the nD domain-walls tTS are as follows. There is one such move for every $0 \leq i \leq n$, namely

$$S_{i-1} \times B^{ps,a}_{n-i+1} \cup_p B^{p,b}_i \times B_{n-i} = B_i \times B^{p,a}_{n-i} \cup_p B^{ps,b}_{i+1} \times S_{n-i-1} \ . \tag{5.125}$$

Here, S_x is the x-sphere, B_x is the x-ball, B^p_x denotes an x-ball with physical instead of space boundary, and B^{ps}_x denotes an x-ball whose boundary $x-1$-sphere is split into a $x-1$-ball of space boundary, and a $x-1$-ball of physical boundary. Note also that we adopt the convention that S_{-1} is the empty manifold. The additional superscripts a or b denote whether the bulk is the a or the b model. Each side of the equation is a union of an a bulk and a b bulk followed by identifying their physical boundaries, which is denoted by $\cup_p$. After identifying the physical boundaries of the a and b bulks, they become a domain wall between a and b. This common physical boundary is

$$S_{i-1} \times B_{n-i} \tag{5.126}$$

on the left-hand side, and

$$B_i \times S_{n-i-1} \tag{5.127}$$

on the right-hand side. In order to apply the move above to an n-dimensional extended manifold with a bulk, b bulk, and domain wall, we need an embedding of the left-hand side into the extended manifold. We then remove this left-hand side, and paste the right-hand side. Or, we apply the move in the reverse direction. For this to be a well-defined procedure, we need to identify the space boundaries on both sides. Indeed, the space boundary on both sides is canonically identified with

$$S_{i-1} \times B^{p,a}_{n-i} \cup_p B^{p,b}_i \times S_{n-i-1} \ . \tag{5.128}$$

This space boundary is an $n-1$-dimensional extended manifold with a bulk, b bulk, and domain wall. The examples below will show why these moves indeed intuitively correspond to “creating an fusion *a*/*b* islands within *b*/*a*”. Note that the ith move is related to the $n-i$th move by switching a and b. Also note that the 0th move and the nth move follow from the robustness of b and a, respectively.

A special case of a domain wall is a boundary of a, namely when b is the trivial vacuum. An invertible domain wall to vacuum will be called an *invertible boundary*. The existence of an invertible boundary implies that the a model is in a trivial phase. Removing the b bulk in the invertibility moves of Eq. (5.125) yields

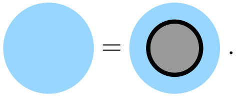
$$S_{i-1} \times B^{ps}_{n-i+1} = B^s_i \times B^p_{n-i} \ . \tag{5.129}$$

The space boundary shared by both sides is

$$S_{i-1} \times B^p_{n-i} \ . \tag{5.130}$$

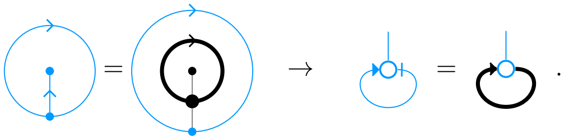
These moves are well-known in topology, namely they are *i-handle attachments*. Note that usually attaching an i-handle would be thought of as directly attaching $B_i \times B_{n-1}$ to the (physical) boundary, but our systematic approach only allows gluing at space boundaries. Similarly, the ith move in Eq. (5.125) corresponds to replacing b with a inside of an i-handle.

Let us now consider the case of $1+1$ dimensions, and show how to impose invertibility for the 2D domain-wall tTS in Section 4.11. To this end, it is convenient to take the simplified version of the 2D domain-wall tTS which consists of two boundaries, interacting via the commutativity move. In $n=2$ dimensions, there are three invertibility moves. For $i=0$, we get



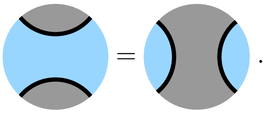
(5.131)

This can be represented by a recellulation of extended cellulations, yielding an equation of tensor-network diagrams,



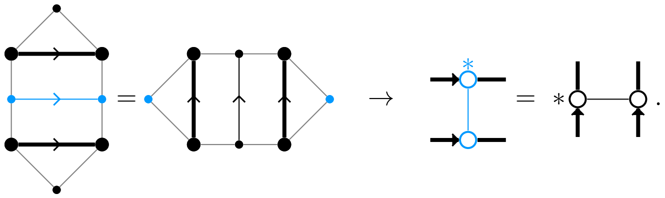
(5.132)

However, as mentioned for the general case, this move follows from the robustness of b. For $i=1$, we find



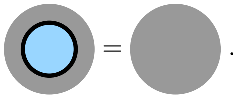
(5.133)

In terms of extended cellulations and diagrams we get



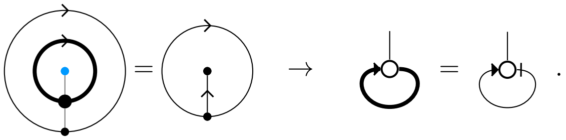
(5.134)

For $i=2$, we get the same as $i=0$ with a and b reversed.



(5.135)

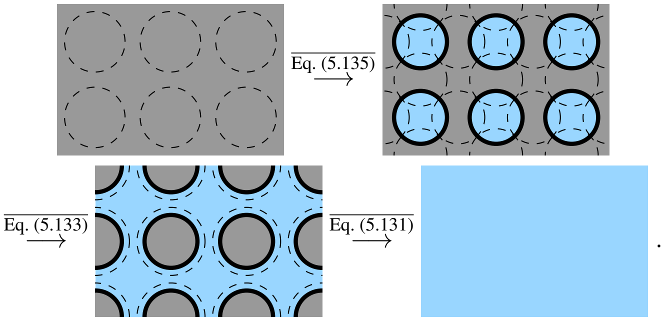
and microscopically,



(5.136)

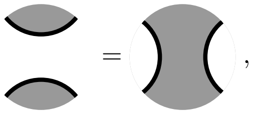
Again, this follows from the robustness of a. All in all, given two 2D tTS models a and b are in the same phase if there exist tensors as in Eq. (4.169), such that Eqs. (5.132), (5.134), and (5.136) hold.

Let us briefly illustrate how a constant-depth circuit of applying these topology-changing moves can transform the a tensor-network path integral into the b path integral,



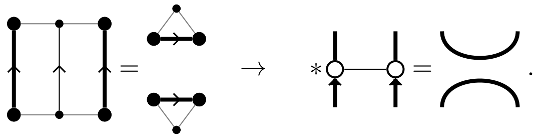
(5.137)

To study whether models are in a trivial phase, let us also look at invertible boundaries, which is a special case of an invertible domain wall where b is replaced by vacuum. For $i = 0$, Eq. (5.129) holds automatically since the vacuum is robust. In fact, the move is a trivial equation of scalars since the common space boundary in Eq. (5.130) is empty. For $i = 1$, Eq. (5.129) yields



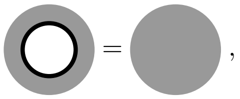
(5.138)

which is just the robustness condition for the boundary itself. It can be implemented by a recellulation and tensor-network equation,



(5.139)

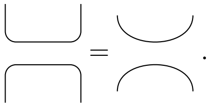
For $i = 2$, we get



(5.140)

and the tensor-network equation looks identical to Eq. (5.136). This follows directly from the robustness of a. We see that any robust 2D tTS model with a robust boundary is automatically in a trivial phase.

Equipped with this, we can now turn to a concrete example. Namely, consider the boundary in Eq. (4.164) of the 2D tTS model in Eq. (4.15). The bulk and boundary are robust, and hence the boundary is invertible. Indeed, plugging Eq. (4.164) into Eq. (5.139) yields the same diagram on both sides,



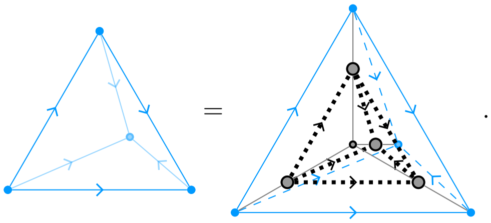
(5.141)

This shows that the model in Eq. (4.15) is in a trivial phase.

Next, let us consider invertibility for a domain wall in $2+1$ dimensions. For $i=0$, we get the move

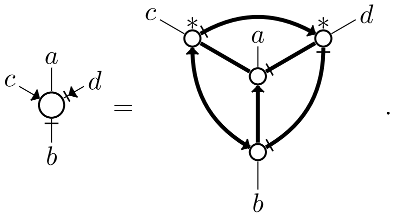
$$B_3^b = B_3^{p,a} \cup_p B_1^{ps,b} \times S_2 \; . \tag{5.142}$$

On the left-hand side, there is a 3-ball of b model. On the right-hand side, there is a 3-ball of b model, with a 3-ball in its center replaced by a-model. A "naive" way to implement this as an extended recellulations is the following,



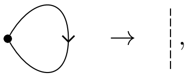
(5.143)

On the left-hand side there is a single b bulk tetrahedron. On the right-hand side, there are four boundary triangles glued together to a tetrahedron, with a single a edge in the center and a b tetrahedron on the outside. As tensor-network diagrams, this equation of extended cellulation becomes



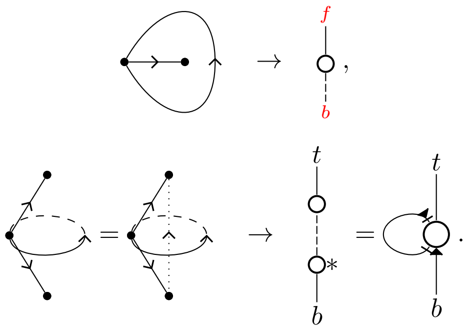
(5.144)

However, one can find even much simpler extended cellulations for both sides. For example, as space-boundary cellulation, we can use a cellulation consisting of two 1-gons. A 1-gon can be cellulated by a single triangle, so the bond dimension variable for 1-gon faces,



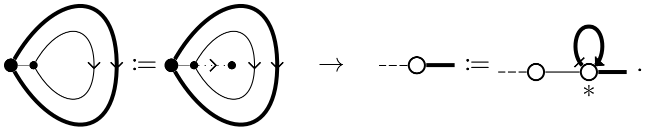
(5.145)

can be defined via the following *1-gon adapter* and the according isometry:



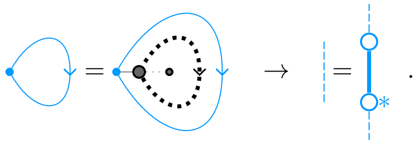
(5.146)

With this, we can define a boundary 1-gon as



(5.147)

Using the boundary 1-gon, the move can be implemented as



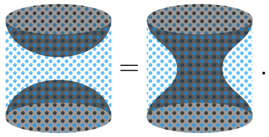
(5.148)

As usual, this move holds automatically if b is robust.

Next, $i = 1$ yields

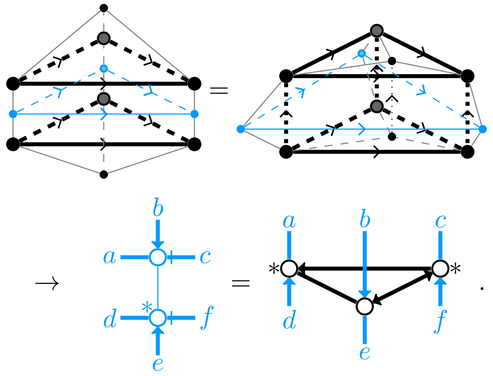
$$S_0 \times B_3^{ps,a} \cup_p B_1^{p,b} \times B_2 = B_1 \times B_2^{p,a} \cup_p B_2^{ps,b} \times S_1 \ . \tag{5.149}$$

We can draw this as follows,



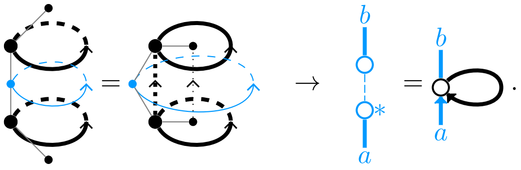
(5.150)

On both sides, we have a cylinder with b space boundary (blue dotted) on the side, and a space boundary (gray dotted) on the bottom and top. On the left-hand side, the b space boundary is connected via a b bulk cylinder, and the a bulk consists of two disconnected 3-balls. The domain wall (shaded gray) itself consists of two disks. On the right-hand side, the two a space-boundary disks are connected via an a bulk cylinder, and the b bulk consists of a solid torus "wrapping around" the a bulk. The domain wall itself is an annulus. A "naive" way to implement this move as an equation between extended cellulations is



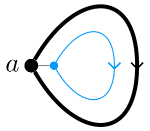
(5.151)

However, we can use the boundary 1-gon cell in Eq. (5.147) to get an even simpler move,



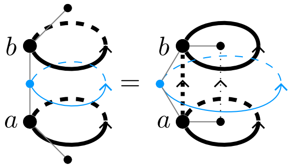
$$\quad (5.152)$$

Let us look at this move in the block-diagonal form. The boundary 1-gon becomes a tensor



$$\to \quad l^a \ . \qquad (5.153)$$

With this, the move becomes



$$\to \quad l^b\overline{l^a} = \sum_y d_y C_1^{ayb}{}^{a\varnothing\varnothing}_{b\varnothing\varnothing} \ . \qquad (5.154)$$

Here, the symbol $\varnothing$ symbolizes that the dimension of the corresponding multiplicity index is always 1.

The moves for $i = 2$ and $i = 3$ are equal to $i = 1$ and $i = 0$ with a and b reversed, so we will omit drawing them explicitly. Also, the $i = 3$ move follows from the robustness of a.

Let us briefly connect our findings with the existing literature. The existence of an invertible domain wall is equivalent to stating that the unitary fusion categories are *Morita equivalent*. An criterion for when a domain wall is invertible and thus defines a Morita equivalence of fusion categories can be found in Ref. [34]. Our criterion is different, since we do not have two but only one C-tensor involved.

5.5 Invertible phases

Similar to the invertibility condition for a domain wall discussed in Sections 5.4, there is also a notion of a model itself being invertible. *Invertible phases* are invariant under a set of topology-changing moves that are well-known in topology as *surgery operations*. The "invertibility" here is with respect to stacking, that is, stacking an invertible model with a time-reversed copy yields a model in the trivial phase. Invariance under surgery operations ensures that this is the case, as we will see in a moment.

In n dimensions and for $-1 \leq i \leq n$, an i*-surgery* operation is the topology-changing move given by

$$S_i \times B_{n+i} = B_{i+1} \times S_{n-i-1} \ . \qquad (5.155)$$

The (space) boundary on both sides is canonically identified with

$$S_i \times S_{n-i-1} \; . \tag{5.156}$$

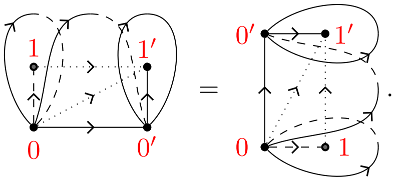
Two manifolds are related by surgery operations for all different i iff they are cobordant, i.e., their disjoint union can be identified with the boundary of a manifold of one dimension higher. So, imposing invariance under all surgery operations can be viewed as going from topological invariance to cobordism invariance. Note that a $n-i-1$-surgery is the same as an i-surgery backwards. Furthermore, 0-surgery, and equivalently $n-1$-surgery equals the move from the robustness condition in Eq. (4.70). Finally, the space boundary is empty for the -1-surgery move and the equivalent n-surgery move. So these moves correspond to equations between scalars, and are thus trivial since we only enforce all equations up to global scalar prefactors. Thus, invertibility is imposed by robustness in Eq. (4.70) together with invariance under x-surgery moves for $1 \leq x \leq \lfloor (n-1)/2 \rfloor$, where $\lfloor a \rfloor$ denotes the largest integer smaller or equal to a.

In $1+1$ dimensions, there are no additional surgery moves, and robust and invertible phases are the same. That is, models of the 2D tTS with the additional move in Eq. (4.74) are models for both robust and invertible phases of matter. Note however that non-trivial robust or invertible phases do not exist in $1+1$ dimensions.

In $2+1$ dimensions, there is a single additional surgery move, namely 1-surgery,

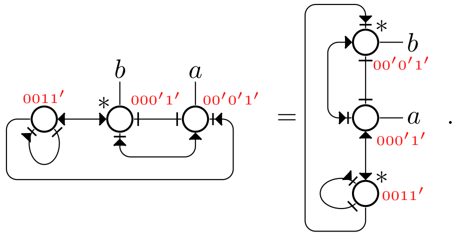
$$S_1 \times B_2 = B_2 \times S_1 \; . \tag{5.157}$$

On both sides, we have a solid torus. However, the boundaries $S_1 \times S_1$ on both sides are identified in a way that gluing both solid tori yields a 3-ball. Let us now find a triangulation for this move. We begin by triangulating the common space boundary, the torus $S_1 \times S_1$, with two triangles. B_2 can represented by a 1-gon which can be triangulated with one triangle as shown in Eq. (4.73), and S_1 is triangulated by a single looping edge. Thus, a triangulation of $B_2 \times S_1$ is obtained from twice the triangulation of $\mathrm{Sx}_2 \times \mathrm{Sx}_1$ with three tetrahedra. So an appropriate retriangulation is given by



(5.158)

On the left-hand side left and right are identified, and on the right-hand side bottom and top are identified. This corresponds to the following tensor-network equation,



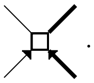
(5.159)

Thus, 3D-tTS models that additionally fulfill the robustness in Eq. (4.79) and the 1-surgery invariance in Eq. (5.159) correspond to to invertible topological phases. For example, the toric-code model in Eq. (4.22) fulfills Eq. (4.79) but not Eq. (5.159), so it corresponds to a robust but non-invertible topological phase. In fact, without adding symmetries or fermions, there are no robust invertible 3D-tTS models in a non-trivial phase. There is one $\mathbb{Z}$ family of non-trivial chiral invertible topological phases, generated by the E_8 *phase*. However, this phase cannot be represented as a 3D-tTS model, but potentially as a model of some other topological TS as discussed in Chapter 11.

In $3+1$ dimensions, we still need to impose only one additional surgery invariance, namely under 1-surgery, which is the same as backwards 2-surgery.

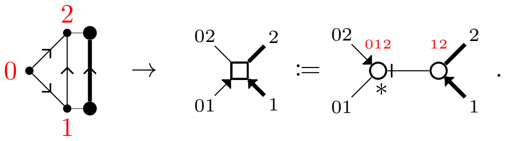
5.6 Commuting-projector Hamitlonians

Like for the bulk in Section 4.4, the nD-boundary-tTS models give rise to commuting-projector Hamiltonians. In general, a Hamiltonian model can have additional degrees of freedom at its boundary. For example, in $1+1$ dimensions, we assume that there is an additional qudit at every boundary point. Then, the boundary of a commuting-projector Hamiltonian is given by an additional projector acting on this boundary qudit and the neighboring bulk qudit. This projector is a tensor acting from the bottom to the top in the following diagram,



(5.160)

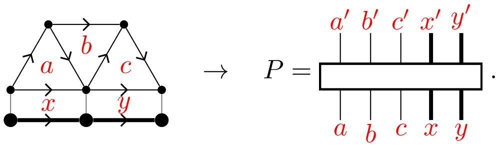
The fact that this tensor is a projector and commutes with the nearby bulk projectors yields two moves analogous to Eqs. (4.36) and (4.37). Together with the bulk projectors and moves, these define the *2D boundary commuting-projector TS*. Now, the commuting-projector model can be obtained from the 2D-boundary-tTS model by the following mapping from the 2D boundary commuting-projector TS to the 2D boundary tTS,



(5.161)

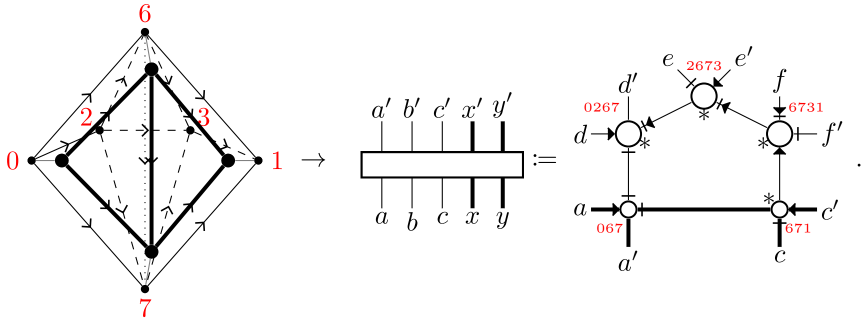
As shown, geometrically, this projector corresponds to a rhombus-like patch like in Eq. (4.41), just that now it is cut in half by a boundary edge on the right.

Let us now show how to extend the $2+1$-dimensional commuting-projector model from Section 4.4 with a boundary. We introduce an additional qudit at every boundary edge, such that at every boundary vertex we get an additional projector acting on the surrounding qudits,



(5.162)

With this, the TS mapping from the 3D boundary commuting-projector TS to the 3D boundary tTS is as follows,



(5.163)

As shown, the boundary projector is mapped onto a diamond-shaped volume like in Eq. (4.48) that has been truncated towards the front with two boundary triangles. The generalization to $3+1$ dimensions and to the block-diagonal form is straight-forward.

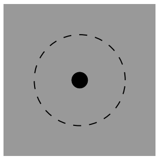
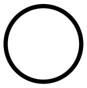
Chapter 6

Defects

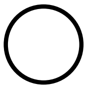
In this section, we will extend the notion of tTS and phases describing topological fixed-point models to topological defects. Roughly speaking, by *defect*, we mean any sort of lower-dimensional submanifold along which the path integral is altered. In general, tTS describing defects assign tensor-network path integrals to extended cellulations of a certain type. Depending on the type of extended manifolds, defects can be things known in physics as anyons, twist defects, boundary anyons, corners, ground states, fusion events, or condensation events. Note that in the physics literature, “defects” often specifically refer to twist defects or symmetry defects, but we use the term much more broadly as in parts of the mathematics TQFT literature. We would even consider boundaries and domain walls types of defects, even though this is certainly not an established terminology in physics.

6.1 Circle ground states in $1+1$D

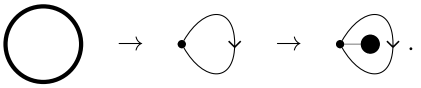
Let us start with a simple toy example for a type of defect in $1+1$ dimensions. We already have looked at two types of 1-dimensional defects, namely boundaries and domain walls. More generally, one could look at lines interfacing three or more copies of the 2D tTS, but this is straight forward and physically not very relevant. Instead, we will look at the simplest type of 0-dimensional defects, namely points embedded into the bulk 2-manifold. The link of such points is a circle,

 $\rightarrow$  , (6.1)

so the overall extended manifold type is

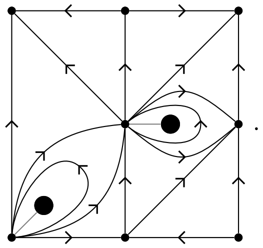
$a:(2,\varnothing)\;,\quad b:(0,$  $)\;.$ (6.2)

As mentioned above, extended manifolds of this type are 2-manifolds (a) with embedded points (b). Physically, such defects correspond to the ground states of the model on (fixed minimally cellulated) a circle. In order to cellulate the extended manifolds, we first cellulate the b link, which can be done via a single looping edge. A defect vertex (in fat below) is then adjacent to the 2-dimensional bulk via this looping edge,



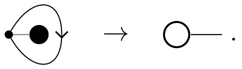
(6.3)

That is, an extended cellulation consists of 1) a collection of defect vertices (that is, a 0-cellulation of M_b), 2) a bulk 2-cellulation of M_a with (space) boundary, and 3) a pairing between the M_a boundary edges and the looping edges adjacent to the different M_b vertices. A patch of such a cellulation could look like:



(6.4)

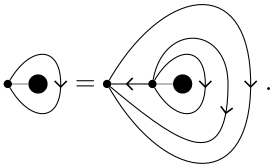
Now, the *2D circle-ground-state tTS* assigns a tensor variable to the defect vertex,



(6.5)

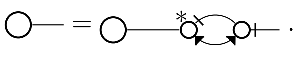
This tensor has one index corresponding to its looping edge, which is contracted with the adjacent bulk triangle tensor.

Next, we need to ensure topological invariance via moves. There is only one such move, namely a trivial 0-0 Pachner move of the defect 0-cellulation, where one side is padded with bulk,



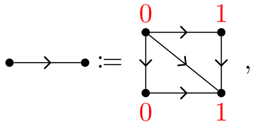
(6.6)

As a tensor-network equation, this is



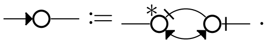
(6.7)

To better understand this axiom, it is insightful to turn the 2-dimensional bulk into a 1-dimensional bulk using a *dimensional reduction*, more specifically a *compactification* mapping from the 1D tTS to the 2D tTS. Topologically, such a mapping takes a 1-manifold and maps it to a 2-manifold by taking the cartesian product with the circle, which is the link of b in Eq. (6.1). On the level of cellulations, we take the cartesian product with the looping edge shown in Eq. (6.3). So an edge of a 1-cellulation is mapped onto an annulus, which can be triangulated with two triangles,



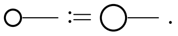
(6.8)

where the bottom and top edge are identified. In terms of tensor-network diagrams, this becomes



(6.9)

This can be extended this to a mapping from the 1D boundary tTS to the 2D circle-ground-state tTS,



(6.10)

Thus, we see that the point defects are identical to boundaries of the compactified 1-dimensional model.

Recall that 2D tTS models give rise to (special Frobenius or C*-)algebras, whereas 1D tTS models are projectors. In this context, the operator defined in Eq. (6.9) is the projector onto the *center* of the algebra, that is, the space of algebra elements that commute with all others.

Similarly to point defects in the bulk, we could look at point defects in the boundary of 2D boundary tTS models. The link of such point defects would be B_1, the interval. However, since these point defects are a bit boring, we will go straight to $2+1$ dimensions.

6.2 Anyons in $2+1$D

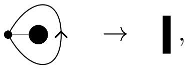
The paradigmatic type of lower-dimensional defects in $2+1$ dimensions are *anyons*.

6.2.1 tTS models

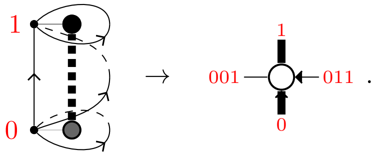
Anyons are point-like defects in a 2-dimensional space. In spacetime, they form worldlines, so the extended manifolds to be used are 3-manifolds (the bulk) with embedded 1-manifolds (the worldlines). The normal space at a point on the anyon worldline is a plane perpendicular to it, and an ϵ-sphere within this plane is a circle. Thus the link of the anyon worldline is a circle, and the the overall extended manifold type is given by

$$a : (3, \varnothing)\ , \quad b : (1, \bigcirc)\ , \tag{6.11}$$

which is the same as in Eq. (6.2), but one dimension higher. In order to cellulate extended manifolds of this type, we need to define standard 0-cell and 1-cell representatives of the anyon worldline. The standard 0-cell representative is given by the looping-edge cellulation of the circle link in Eq. (6.3). The *3D anyon tTS* assigns a new bond dimension variable to every anyon vertex attached to a bulk looping edge,


(6.12)

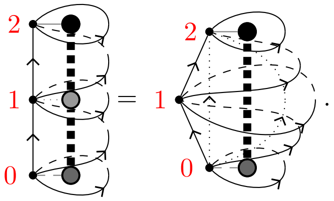
which we draw as (very) thick lines. A standard anyon 1-cell representative can be obtained by taking the cartesian product of the anyon edge with the the looping edge, yielding a rectangle wrapping around to an annulus. This rectangle can be divided into two triangles. The anyon edge is attached to this triangulated annulus. To each anyon edge, we assign a copy of a new tensor variable,



(6.13)

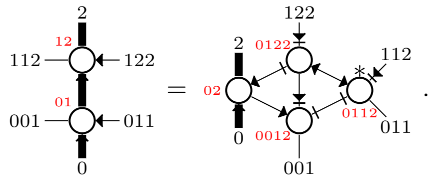
Two of the indices correspond to the top and bottom anyon vertex, and the other two correspond to the bulk triangles to which the anyon edge is attached. It might be helpful to imagine the left-hand side as a cylinder-like 3-cell, even though this picture does not fully generalize to other types of defects. Note that this extended 1-cell is the same as gluing both sides of the move in Eq. (6.6). An extended cellulation consists of 1) an anyon 1-cellulation attached to a 2-cellulation consisting of many annuli and 2) a bulk 3-cellulation whose boundary is identified with this 2-cellulation. Note that in addition to the above characterization, two anyon edges are allowed to be adjacent to the same bulk triangle without any bulk tetrahedra in between. More intuitively, an extended cellulation is a 3-cellulation with embedded sequences of cylinder-like segments.

The topological invariance is imposed by combinatorial moves as usual. These moves are just Pachner moves for the anyon worldline 1-cellulation, suitably padded with bulk tetrahedra. There is one Pachner move in 1 dimension, namely the 2-1 Pachner move. It yields a move with two anyon edges on the left-hand side, one anyon edge on the right-hand side. The right-hand side is padded by bulk tetrahedra such that the space boundary on both sides is the same,



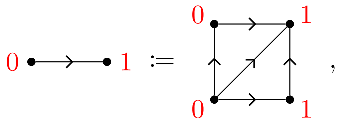
(6.14)

Concretely, the padding on the right forms a solid torus "with triangle profile", whose boundary is divided into three annuli. The solid torus is wrapped around the anyon cylinder, such that its inside annulus is glued to the side of the anyon cylinder. So after padding with the solid torus, there are two annuli on the outside, just like on the left-hand side. Finally, we triangulate each annulus with two triangles, and the solid torus with three tetrahedra. In terms of tensor-network diagrams, this becomes



(6.15)

Every tensor as in Eq. (6.13) fulfilling Eq. (6.15) defines an anyon, though usually only robust anyons are really called anyons. In order to gain more insight into this equation, it is useful to apply a compactification mapping from the 2D tTS to the 3D tTS, analogous to the mapping from the 1D tTS to the 2D tTS in Section 6.1. Topologically, this mapping maps a 2-manifold to a 3-manifold by taking the cartesian product with the circle. On the level of cellulations, we take the cartesian product of a 2-cellulation with the looping edge in Eq. (6.3). Each edge of the 2-cellulation/triangulation is mapped to an annulus of the 3-dimensional triangulation,

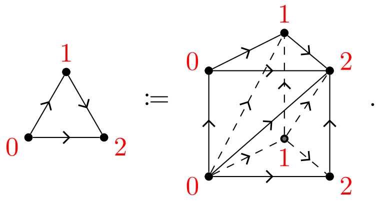


(6.16)

which is triangulated by two triangles, with bottom and top identified in the picture. Thus, the 2D-tTS bond dimension variable corresponding to an edge is mapped to two copies of the 3D tTS bond dimension variable,

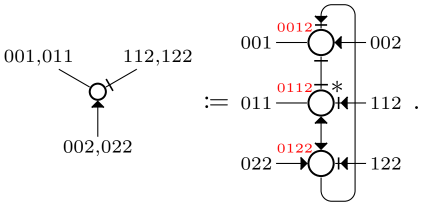
$$\Big| := \Big| \quad \Big| . \tag{6.17}$$

Next, each triangle is mapped to a "triangle prism" with top and bottom identified,



(6.18)

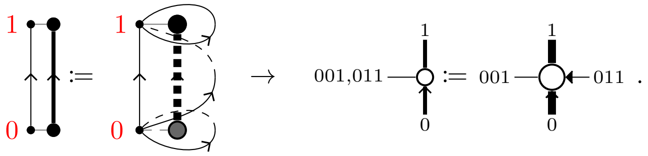
As shown, this triangle prism can be triangulated with three tetrahedra, which is the same triangulation as used in Eq. (6.15). So, the triangle tensor variable is mapped to a diagram with three copies of the tetrahedron tensor variable,



(6.19)

In order to formally show that this defines a TS mapping, we would have to find derivations for all the mapped moves of the 2D tTS. This is not difficult to show, but we will not do this explicitly here. Instead we want to remark that after taking the cartesian product with the circle, the 2-dimensional Pachner moves become 3-dimensional recellulations. Any 3-dimensional recellulation can be obtained by a sequence of 3-dimensional Pachner moves, which shows the claim.

Next, we extend the above to a mapping from the 2D boundary tTS to the 3D anyon tTS. To this end, we simply map the 1-dimensional boundary to a 1-dimensional anyon worldline. Combinatorially, we map a boundary edge in 2 dimensions to an anyon edge in 3 dimensions,

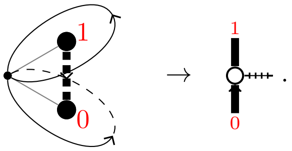


(6.20)

With this, the mapped boundary Pachner move in Eq. (4.160) is equal to the Pachner move for the anyon worldlines in Eq. (6.15). Thus, we see that anyons are in one-to-one correspondence with the boundaries of the compactified 2D-tTS model.

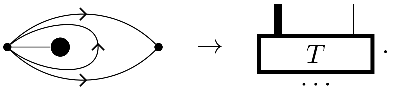
6.2.2 Block-diagonal form

After introducing the tTS describing anyons, let us sketch what happens when we block-diagonalize the surrounding bulk. Recall the banana 2D-tTS model from Eq. (4.227). We can define another boundary for this banana 2D-tTS model, which we call the *anyon banana 2D-boundary-tTS model*. The defining tensor is the one associated to the following anyon 1-cell representative consisting of two anyon vertices and one bulk 2-gon,



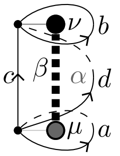
(6.21)

After applying the isometry from Proposition 3, the anyon bond dimension decomposes into a triple (i, a, μ) of one banana irrep i with internal index a, and one multiplicity μ. The dimension of the multiplicity index is $B^i := m_i$, depending on the irrep i. Now, consider the tensor T associated to some extended cell representative, or any extended cellulation with space boundary. Further, consider and edge within this space boundary that is adjacent to an anyon vertex and a bulk triangle,



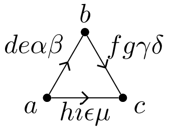
(6.22)

Applying Proposition 5 analogous to Sections 4.14 and 5.1.2 shows that T is determined by a simpler tensor $\widetilde{T}$, where the anyon banana irrep index and the 01 banana irrep index are fused, and the according internal indices removed. Applying this to all edges, we get a tensor $\widetilde{T}$ with one irrep index per edge, one N_k^{ij}-dimensional multiplicity index at each triangle and one B^i-dimensional multiplicity index at each anyon vertex. For example, the defining tensor in Eq. (6.13) becomes

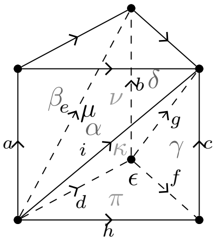


$$\rightarrow \quad \rho_{ab}^{cd\alpha\beta}{}^{\nu}_{\mu} \ . \tag{6.23}$$

The compactification mapping in Eq. (6.18) becomes



$$:= \sum_{\nu,\kappa,\pi}$$



$$\rightarrow \quad \beta^{abc}{}^{(de\alpha\beta)(fg\gamma\delta)}_{(hi\epsilon\mu)} := \sum_{\nu,\kappa,\pi} F_i^{dfc}{}^{g\kappa\gamma}_{h\pi\epsilon} \overline{F_i^{dbf}{}^{g\kappa\delta}_{e\alpha\nu}} F_i^{adf}{}^{h\mu\pi}_{e\beta\nu} \ . \tag{6.24}$$

The resulting 2-dimensional state sum has one irrep label at every vertex of the triangulation, and two irrep labels and two multiplicity labels at each edge. There is one weight d_i/D at every vertex with irrep label i, one weight $d_d d_e$ at every edge with irrep labels d and e, and one weight β as defined above at every triangle. With this, the main move for anyons in Eq. (6.15) becomes

$$\sum_{\pi} \rho_{ab}^{de\alpha\beta}{}^{\pi}_{\mu} \rho_{bc}^{fg\gamma\delta}{}^{\nu}_{\pi} = \sum_{hi\epsilon\mu} d_h d_i \beta^{abc}{}^{(de\alpha\beta)(fg\gamma\delta)}_{(hi\epsilon\mu)} \rho_{ac}^{hi\epsilon\mu}{}^{\nu}_{\mu} \ . \tag{6.25}$$

We can make the 2-dimensional compactified state sum into a standard 2D-tTS model with triangle tensor

$$\widetilde{\beta}^{(abde\alpha\beta)(b'cfg\gamma\delta)}_{(a'c'hi\epsilon\mu)} := \delta_{aa'}\delta_{bb'}\delta_{cc'}\sqrt{d_e d_g d_i d_d d_f d_h}\beta^{abc}{}^{(de\alpha\beta)(fg\gamma\delta)}_{(hi\epsilon\mu)} \ . \tag{6.26}$$

Then the robust anyons are then precisely the irreps when using Propositions 1 and 2. Thus, finding anyons boils down to finding the block-diagonalizing isometry of a (special Frobenius, *-) algebra.

Note that the weight β defining the compactified state sum defines a *linear category*, also known as *algebroid.* The irreps at the vertices are the simple objects of this linear category, and the morphisms form the vector space

$$\mathrm{Hom}(a,b) = \mathbb{C}^{\sum_{d,e} N_e^{ad} N_e^{db}} \ . \tag{6.27}$$

β^{abc} are then the structure coefficients of the composition

$$\mathrm{Hom}(a,b) \times \mathrm{Hom}(b,c) \rightarrow \mathrm{Hom}(a,c) \ . \tag{6.28}$$

For block-diagonalizing β in practice, it may be beneficial to make use of this additional structure, and not directly switch to the ordinary algebra $\tilde{\beta}$. For example, often the set of irreps decomposes into irreducible orbits C_i with $\bigcup_i C_i = \text{irreps}$ and

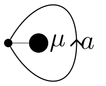
$$a \in C_i, b \in C_j, i \neq j \quad \Rightarrow \quad \sum_{d,e} N_e^{ad} N_e^{db} = 0 \ . \tag{6.29}$$

In this case, we can perform the block-diagonalization on each orbit separately.

In the literature, the algebra defined by Eq. (6.26) is known as *tube algebra*, see for example Refs. [50, 115, 38, 103]. Approaches in the literature are usually focused on obtaining an abstract classification of anyons in a given bulk model, and not on explicitly introducing anyons into a microscopic model as we do here. A work that introduces anyons explicitly into a lattice model is Ref. [81].

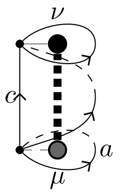
6.2.3 Cohomology models

Let us now discuss anyons when the underlying bulk is described by an ordinary cohomology model. In fact, the form of the anyon state sum will not be restricted, but will be the general block-diagonal form discussed in the previous section, with one free label at every anyon vertex,



. (6.30)

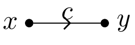
The dimension of this free label, called μ above, is B^a depending on the value of the G-element at the adjacent bulk loop edge, called a above. Due to the cocycle constraint of the G-labels, the weight ρ associated to an anyon edge in Eq. (6.23) reduces to



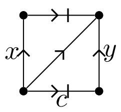
$$\rightarrow \quad \rho^a(c)^\nu_\mu \ . \tag{6.31}$$

Indeed, all other G-labels are determined by a and c via the 1-cocycle constraint.

To classify the anyon worldlines we apply the compactification mapping in Eq. (6.18) or (6.24) to get a 2-dimensional state-sum. Thereby we take all variables and weights of the mapped 3-dimensional triangulation and assign them to the original 2-dimensional triangulation. Every edge of the 2-triangulation becomes a rectangle of the 3-triangulation with top and bottom edge identified,



$\rightarrow$



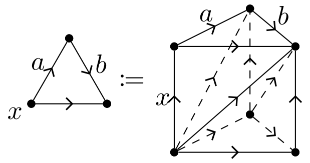
. (6.32)

We can see that there is one G-variable at each vertex and each edge of the 3-triangulation. The 1-cocycle constraint on the 3-dimensional triangulation yields a constraint $y =$

$c^{-1}xc$. Thus, the resulting state sum is a G-set state sum as discussed in Section 4.16.2, with G-set given by

$$A = G\ , \quad x \triangleleft g = g^{-1}xg\ . \tag{6.33}$$

A triangle is mapped to a triangle prism,



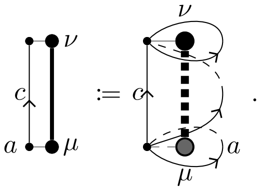
(6.34)

where to top and bottom triangle are identified, and thus carry the same labels a and b. The weights of the three tetrahedra on the right-hand side form the triangle weight β of the 2-dimensional G-set state sum:

$$\beta^x(a,b) := \omega(a,b,(ab)^{-1}xab)\overline{\omega(a,a^{-1}xa,b)}\omega(x,a,b)\ . \tag{6.35}$$

By construction, the topological invariance of the 2-dimensional state sum follows from the topological invariance of the 3-dimensional state sum. In other words, β is a G-set 2-cocycle with respect to the action in Eq. (6.33).

Next, let us consider the extension of the mapping to a boundary in Eq. (6.20). Pulling back the anyon state sum along this mapping yields a 2-dimensional boundary state sum,



(6.36)

The resulting boundary state sum is a free boundary of a G-set bulk as defined in Section 5.1.3, with the G-set bulk given by Eq. (6.34). The mapping above identifies anyon state sums with such boundary state sums. The weights $\rho^a(c)^\nu_\mu$ are thus determined by the equation

$$\beta^a(c,d)\rho^a(cd)^\nu_\lambda = \sum_\mu \rho^a(c)^\mu_\lambda \rho^{c^{-1}ac}(d)^\nu_\mu\ . \tag{6.37}$$

We have shown in Section 5.1.3 how to find and classify such free boundaries. We start by decomposing the G-set defined in Eq. (6.33) into transitive orbits. These orbits are precisely the conjugacy classes C_i of G. For each conjugacy class C_i, we choose a representative $X_i \in C_i$ and consider its stabilizer subgroup $H_i := S(X_i) \subset G$, which is isomorphic to the *normalizer* of the conjugacy class. Then we restrict β to an ordinary H 2-cocycle $\tilde{\beta}_i(c,d) := \beta^{X_i}(c,d)$, for each i. For each i, we block-diagonalize the $\tilde{\beta}$-twisted group algebra of H_i, finding the isometry $\tilde{\rho}_{ij}(c)^\nu_\mu$ from the index $c \in G$ to triples (j,μ,ν). For each fixed i and j, $\tilde{\rho}$ is an irrep of the $\tilde{\beta}_i$-twisted H_i group algebra, or equivalently a $\tilde{\beta}_i$-projective irrep of H_i. Finally, we apply Eq. (5.98) to get $\rho^a_{ij}(c)^\nu_\mu$.

Let us now give some concrete examples for anyon worldlines inside 2+1-dimensional cohomology models. We start with $G = \mathbb{Z}_l$, and ω given in Eq. (4.304). Since the group is abelian, the conjugacy classes are individual group elements,

$$C_i = \{i\} \ , \quad X_i = i \ , \quad H_i = \mathbb{Z}_l \ , \quad 0 \leq i < l \ . \tag{6.38}$$

Evaluating Eq. (6.35) yields

$$\beta^x(a,b) = e^{\frac{2\pi i p}{l^2}\left(\overline{x}(\overline{a}+\overline{b}-\overline{a+b})-\overline{a}(\overline{x}+\overline{b}-\overline{x+b})+\overline{a}(\overline{b}+\overline{x}-\overline{b+x})\right)} = e^{\frac{2\pi i p}{l^2}\overline{x}(\overline{a}+\overline{b}-\overline{a+b})} \ . \tag{6.39}$$

Since $H_i = G$, we have $\tilde{\beta}_i(a,b) = \beta^i(a,b)$. The ordinary $\mathbb{Z}_l$ group 2-cocycle $\tilde{\beta}_i$ is trivial for every i, namely the boundary of the following 1-cochain,

$$\tilde{\rho}_{i,0}(a) = e^{\frac{2\pi i p}{l^2}\overline{i}\overline{a}} \ . \tag{6.40}$$

Thus, the block-diagonalizing isometry of the $\tilde{\beta}_i$-twisted group algebra consists of the ordinary $\mathbb{Z}_l$ Fourier transform multiplied with $\tilde{\rho}_{i,0}$ above,

$$\tilde{\rho}_{i,j}(a) = e^{2\pi i p(\frac{1}{l^2}\overline{i}\overline{a}+\frac{1}{l}ja)} \ . \tag{6.41}$$

Note that we have omitted the indices μ and ν since these indices are 1-dimensional. Since $H_i = G$, we have $x = y = R(i) = 0$ in Eq. (5.98) and we have

$$\rho^i_{i,j}(a) = \beta^i(0,a)\overline{\beta^i(a,0)}\tilde{\rho}_{i,j}(a) = \tilde{\rho}_{i,j}(a) \ . \tag{6.42}$$

As a second example, let us consider $G = S_3$ which we will denote as in Eq. (4.305), and $\omega = 1$. The conjugation action is given by

$$\begin{aligned}(i,j)^{-1}\cdot(a,b)\cdot(i,j) &= (-(-1)^j i, j)\cdot(a+(-1)^b i, b+j)\\ = (-(-1)^j i + (-1)^j(a+(-1)^b i), b) &= ((-1)^j a - (-1)^b i, b) \ .\end{aligned} \tag{6.43}$$

There are three conjugacy classes, with their representatives,

$$\begin{aligned}C_0 = \{(0,0)\} \ , X_0 = (0,0) \ , H_0 = S_3\\ C_1 = \{(x,1)\}_{0\leq x<3} \ , X_1 = (0,1) \ , H_1 = \langle(0,1)\rangle \sim \mathbb{Z}_2\\ C_2 = \{((-1)^x,0)\}_{0\leq x<2} \ , X_2 = (1,0) \ , H_2 = \langle(1,0)\rangle \sim \mathbb{Z}_3 \ .\end{aligned} \tag{6.44}$$

Since $\omega = 1$, we have $\beta = 1$ and $\tilde{\beta}_i = 1$ for all i. Thus, $\tilde{\rho}_{ij}$ are just ordinary irreps of H_i. $H_0 = G = S_3$ has three irreps, the 1-dimensional trivial ($j = 0$) and sign ($j = 1$) irreps, and the 2-dimensional standard ($j = 2$) irrep,

$$\begin{aligned}\tilde{\rho}_{0j}((a,b)) &= (-1)^{jb} \ , 0 \leq j < 2 \ ,\\ \tilde{\rho}_{02}((a,b))^\nu_\mu &= \delta_{b+\mu,\nu}e^{\frac{2\pi i}{3}(-1)^\mu a} \ .\end{aligned} \tag{6.45}$$

Let us briefly verify that the last expression indeed defines a representation,

$$\begin{aligned}&\tilde{\rho}_{02}((a,b)(a',b'))^\nu_\mu = \tilde{\rho}_{02}((a+(-1)^b a', b+b'))^\nu_\mu\\ &= \delta_{b+b'+\mu,\nu}e^{\frac{2\pi i}{3}(-1)^\mu(a+(-1)^b a')}\delta_{b+b'+\mu,\nu}e^{\frac{2\pi i}{3}((-1)^\mu a+(-1)^{\mu+b}a')}\\ &\qquad = \sum_\epsilon \tilde{\rho}_{02}((a,b))^\epsilon_\mu \tilde{\rho}_{02}((a',b'))^\nu_\epsilon \ .\end{aligned} \tag{6.46}$$

Since $H_0 = G$, Eq. (5.98) is trivial and we have $(\rho_{0j})^{(0,0)} = \tilde{\rho}_{0j}$. Next, the irreps of H_1 are given by

$$\tilde{\rho}_{1,j}((0,b)) = (-1)^{jb} \ , \qquad j \in \{0,1\} \ . \tag{6.47}$$

In order to get ρ from $\tilde{\rho}$ via Eq. (5.98), we choose representatives

$$\begin{gathered} X_1 \triangleleft (i,0) = (i,0)^{-1}(0,1)(i,0) = (-i,0)(-i,1) = (i,1) \\ \Rightarrow R((i,1)) = (i,0) \ . \end{gathered} \tag{6.48}$$

With this Eq. (5.98) becomes

$$\begin{aligned} x &:= (i,0) \ , \\ y &:= R((i,1) \triangleleft (a,b)) = R((a,b)^{-1}(i,1)(a,b)) \\ &= R((-(-1)^b a, b)(i-a, b+1)) = R((-(-1)^b a + (-1)^b (i-a), 1)) \\ &= R(((-1)^b (a+i), 1)) = ((-1)^b (a+i), 0) \ , \\ x(a,b)y^{-1} &= (i,0)(a,b)((-1)^b(a+i),0)^{-1} \\ &= (a+i,b)(-(-1)^b(a+i),0) = (0,b) \ , \\ (\rho_{1,j})^{(i,1)}((a,b)) &= \tilde{\rho}_j((0,b)) = (-1)^{jb} \ . \end{aligned} \tag{6.49}$$

Finally, the irreps of $H_2 = \mathbb{Z}_3$ are

$$\tilde{\rho}_{2,j}(a) = e^{\frac{2\pi i}{3} ja} \ , \quad j \in \{0,1,2\} \ . \tag{6.50}$$

For the representatives, we choose

$$\begin{gathered} X_2 \triangleleft (0,i) = (0,i)^{-1}(1,0)(0,i) = ((-1)^i, i)(0,i) = ((-1)^i, 0) \\ \Rightarrow \quad R(((-1)^i, 0)) = (0,i) \ . \end{gathered} \tag{6.51}$$

Then, Eq. (5.98) becomes

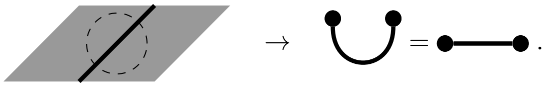
$$\begin{gathered} x := (0,i) \ , \\ y := R(((-1)^i, 0) \triangleleft (a,b)) = R((a,b)^{-1}((-1)^i,0)(a,b)) \\ = R((-(-1)^b a, b)(a + (-1)^i, b)) = R(((-1)^{b+i}, 0)) = (0, b+i) \ , \\ x(a,b)y^{-1} = (0,i)(a,b)(0,b+i)^{-1} = (0,i)(a,i) = ((-1)^i a, 0) \ , \\ \rho_{2,j}^{((-1)^i,0)}((a,b)) = e^{\frac{2\pi i}{3} j (-1)^i a} \ . \end{gathered} \tag{6.52}$$

6.3 Boundary anyons in $2+1$D

As a next example for a 1-dimensional defect, we consider worldlines within the boundary of a model, which we will refer to as *boundary anyons*.

6.3.1 (Block-diagonal) tTS models

As usual, we will describe boundary anyons by a tTS, the *3D boundary-anyon tTS*. The extended manifolds describing this tTS are manifolds with boundary, together with worldlines embedded inside this boundary. The normal space of these worldlines is a half-plane, and an ϵ-sphere within this half-plane is a half-circle or interval,

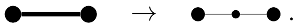


(6.53)

The overall extended manifold type is thus given by

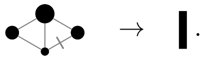
$$a:(3,\varnothing)\,,\quad b:(2,\bullet)\,,\quad c:(1,\bullet\!\!-\!\!\!-\!\!\!-\!\!\bullet)\,. \tag{6.54}$$

The 3D boundary-anyon tTS extends the 3D boundary tTS, and associates tensor networks to extended cellulations of the above type. In order to get standard cell representatives for the 1-region c, we start by finding an extended cellulation for its link. The simplest such cellulation consists of only two boundary vertices adjacent to the same bulk vertex, with no bulk edges in between,



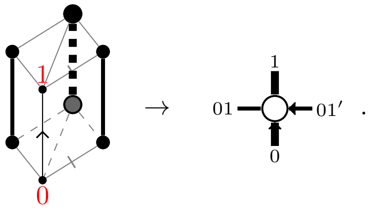
(6.55)

This is the standard 0-cell representative, to which each boundary-anyon vertex is adjacent to,



(6.56)

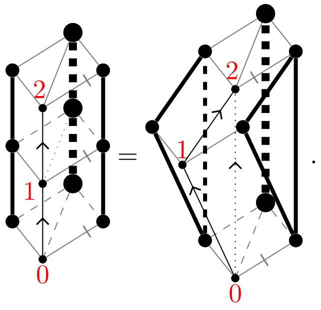
As shown, these boundary-anyon vertices correspond to a new bond dimension variable that we draw in thick. In order to indicate how a the extended 0-cell is identified with the 0-cell representative in Eq. (6.55), we have marked one edge with a tick. This is necessary because of the vertical-axis reflection symmetry of Eq. (6.55). It may be helpful to think of each boundary-anyon vertex to be "thickened" to a little rectangle face. In order to obtain a standard boundary-anyon edge representative, we take the product of the edge with Eq. (6.56),



(6.57)

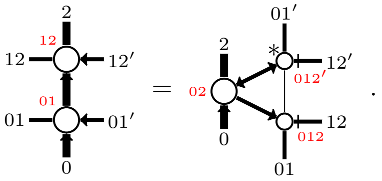
As shown, we associate one copy of a new tensor variable to each such boundary-anyon edge. It may be useful to think of the boundary-edge as "thickened" to a little cubic volume. Two indices of the tensor variable are associated to the each of the two boundary-anyon vertices, and two more to each of the two attached boundary edges. The indices are contracted with those of the adjacent boundary-anyon edges and boundary triangles accordingly.

Next, let us discuss the moves that ensure topological invariance of the tTS model. As usual, there is one main move which is a 2-1 Pachner move for the boundary-anyon 1-cellulation, where the right-hand side is suitably padded with boundary triangles,



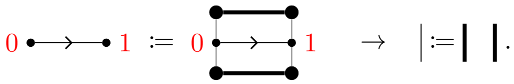
(6.58)

On the left-hand side, there are two boundary-anyon edges, depicted with the adjacent boundary edges and bulk edge. On the right-hand side, there is a single boundary-anyon edge, padded by two boundary triangles that are adjacent to the same common 012 bulk triangle. Expressed in terms of diagrams, this becomes



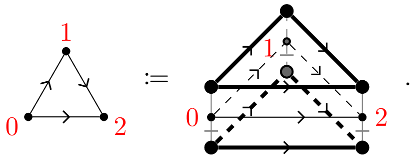
(6.59)

Analogous to bulk anyons, this move can be understood better by considering a compactification mapping from the 2D tTS to the 3D boundary tTS. Topologically, this mapping takes the cartesian product with the c interval link in Eq. (6.53). On the level of extended cellulations, we take the cartesian product of the 2-cellulation with the cellulated link in Eq. (6.55). Every edge of the 2-cellulation is mapped to a pair of boundary edges of the extended 3-cellulation, adjacent to the same bulk edge. Accordingly, the 2D-tTS bond dimension variable is mapped to two copies of the 3D-boundary-tTS bond dimension variable,



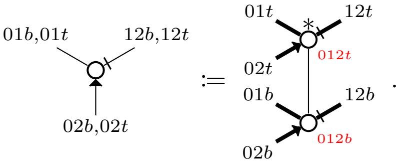
(6.60)

Each triangle is mapped to a "sandwich" of two boundary triangles adjacent to the same bulk triangle,



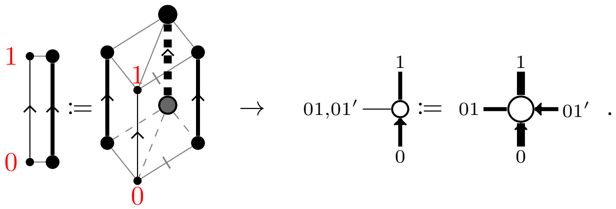
(6.61)

Here we see that our "fattened" way of drawing boundary faces is convenient, since is allows us two draw two boundary triangles with an "infinitely thin" bulk in between. As tensor-network diagrams, the mapping becomes:



(6.62)

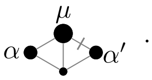
We then extend the above to a mapping from the 2D boundary tTS to the 3D boundary-anyon tTS. Topologically, we simply map the 1-dimensional boundary to a 1-dimensional boundary-anyon worldline. Combinatorially, we map a 2D boundary edge to a 3D boundary-anyon edge,



(6.63)

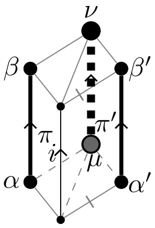
The topological invariance of boundary-anyon worldlines in Eq. (6.59) is equivalent to the 2-dimensional boundary Pachner move in Eq. (4.160) through the mapping given in Eq. (6.62) and Eq. (6.63). Thus, boundary anyons are in one-to-one correspondence with boundaries of the compactified tTS.

Finally, let us consider the block-diagonal form of boundary anyons. We end up with a state sum with one additional multiplicity label at every boundary-anyon vertex,



(6.64)

The dimension if this multiplicity index, called μ above is $B^{\alpha,\alpha'}$ depending on the irrep labels α and α' at the two adjacent boundary vertices. Thus, we associate the following weight to the boundary-anyon edge in Eq. (6.57),

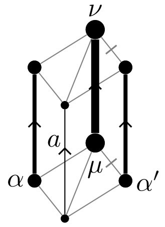


$$\rightarrow \quad \kappa^{i}{}^{\alpha'\beta'\pi'\nu}_{\alpha\beta\pi\mu} \quad . \tag{6.65}$$

6.3.2 Cohomology models

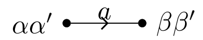
In this section, we will consider boundary anyons for a bulk cohomology model given by $\omega(a,b,c)$ and a G-set boundary given by a G-set A and $\psi^\alpha(a,b)$ as described in

Section 5.1.3. In this case, the multiplicity labels π and π' disappear, and β and β' in Eq. (6.65) are determined by the A^∂-elements α and α' and the G-element i through the G-set action. Thus, we can remove these labels from the weight,

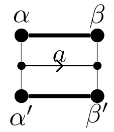


$$\rightarrow \quad \kappa^{\alpha,\alpha'}(a)^{\nu}_{\mu}\,. \tag{6.66}$$

For efficient classification of the boundary anyons we use the compactification mapping in Eq. (6.61). Every edge is mapped to two boundary edges adjacent to the same bulk edge as shown in Eq. (6.60),



$\rightarrow$



$$. \tag{6.67}$$

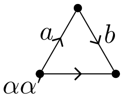
So the resulting state sum has two A^∂-elements at each vertex, one G-label at each edge, and is subject to the constraint

$$(\beta,\beta') = (\alpha \triangleleft a, \alpha' \triangleleft a) \tag{6.68}$$

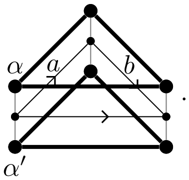
at every edge. Thus, the resulting 2-dimensional state-sum is a G-set state-sum with G-set

$$A = A^\partial \times A^\partial\,, \quad (\alpha,\alpha') \triangleleft g = (\alpha \triangleleft g, \alpha' \triangleleft g)\,. \tag{6.69}$$

The triangle weight β of the 2-dimensional state sum is given by the weights of the two mapped boundary triangles,



$:=$



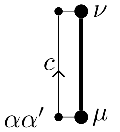
$$. \tag{6.70}$$

The right-hand side consists of two boundary triangles, so we read off the weight as

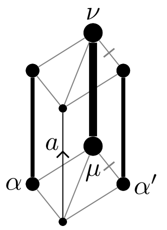
$$\beta^{\alpha,\alpha'}(a,b) := \psi^{\alpha}(a,b)\overline{\psi^{\alpha'}(a,b)}\,. \tag{6.71}$$

Note that one of the boundary triangle is reflected compared to Eq. (5.4) and thus complex conjugated. The topological invariance of the 2-dimensional state sum is inherited from that of the 3-dimensional state sum and its boundary. In other words, β is a G-set 2-cocycle with respect to the action in Eq. (6.69).

Next, the mapping from Eq. (6.63),



$$:=$$



$$, \tag{6.72}$$

identifies boundary anyons with (robust) free boundaries of the 2-dimensional G-set state sum given by β. This allows us to state the defining equation for the weight κ of a boundary anyon as

$$\beta^{\alpha,\alpha'}(c,d)\kappa^{\alpha,\alpha'}(cd)^{\nu}_{\lambda} = \sum_{\mu} \kappa^{\alpha,\alpha'}(c)^{\mu}_{\lambda} \quad \kappa^{\alpha\triangleleft g,\alpha'\triangleleft g}(d)^{\nu}_{\mu} \ . \tag{6.73}$$

We can follow the general recipe in Section 5.1.3 to classify these irreducible κ. First, we decompose the G-set in Eq. (6.69) into irreducible orbits C_i, and choose a representative $X_i \in C_i$ with $H_i := \mathrm{Stab}(X_i) \subset G$. Here we note that if $X_i = (x, y)$, then $\mathrm{Stab}(X_i) = \mathrm{Stab}(x) \cap \mathrm{Stab}(y)$. For every irreducible orbit C_i we reduce β to an ordinary H_i 2-cocycle

$$\tilde{\beta}_i(c,d) := \beta^{X_i}(c,d) \ . \tag{6.74}$$

Then for every i we find the block-diagonalizing isometry $\tilde{\rho}_{ij}(c)^{\nu}_{\mu}$ of the $\tilde{\beta}_i$-twisted H_i group algebra. Finally, we apply Eq. (5.98) to get $\rho^{\alpha,\alpha'}_{ij}(c)^{\nu}_{\mu}$.

In fact, the irreducible orbits are in one-to-one correspondence with double cosets in $i \in H^{\partial}\backslash G/H^{\partial}$, where $H^{\partial} := \mathrm{Stab}(X^{\partial}) \subset G$ for a fixed $X^{\partial} \in A^{\partial}$. To see this, we choose a representative $R(\alpha) \in G$ for all $\alpha \in A^{\partial}$ such that $\alpha = X^{\partial} \triangleleft R(\alpha)$. Then to each $(\alpha, \alpha') \in A^{\partial} \times A^{\partial}$, we assign the double coset

$$H^{\partial}R(\alpha)R(\alpha')^{-1}H^{\partial} \in H^{\partial}\backslash G/H^{\partial} \ . \tag{6.75}$$

Indeed, it is not hard to see that

$$\begin{aligned} H^{\partial}R(\alpha)R(\alpha')^{-1}H^{\partial} &= H^{\partial}R(\beta)R(\beta')^{-1}H^{\partial} \\ \Leftrightarrow \quad \exists g \in G: &\quad (\beta,\beta') = (\alpha,\alpha') \triangleleft g \ . \end{aligned} \tag{6.76}$$

We will not make explicit use of this connection though in the examples below.

Let us now consider some concrete examples for boundary anyons in cohomology models. We start with the $G = \mathbb{Z}_l$ bulk with ω given by Eq. (4.304) and the $H^{\partial} = \mathbb{Z}_{l/k}$ boundary with ψ given by Eq. (5.93). There are l/k irreducible orbits,

$$\begin{aligned} C_i = \{(L_\alpha, L_{\alpha+i})\}_{0\le\alpha<l/k} \ , \quad X_i = (L_0, L_i) \ , \quad H_i = \{kf\}_{0\le f<l/k} \ , \\ 0 \le i < l/k \ . \end{aligned} \tag{6.77}$$

Next, we compute

$$\beta^{X_\alpha,X_{\alpha'}}(a,b) = e^{-\frac{2\pi i p}{l^2}(\overline{\alpha}-\overline{\alpha}')(\overline{a}+\overline{b}-\overline{a+b})} \ , \tag{6.78}$$

which reduces to

$$\tilde{\beta}_i(ka,kb) = \beta^{X_0,X_i}(ka,kb) = e^{\frac{2\pi i p}{l^2}\bar{i}k(\overline{a}+\overline{b}-\overline{a+b})} = 0 \ . \tag{6.79}$$

We can thus set

$$\tilde{\kappa}_{ij}(kf) = e^{\frac{2\pi i}{l}jkf} \ . \tag{6.80}$$

We choose representatives,

$$X_i \triangleleft \alpha = (L_\alpha, L_{\alpha+i}) \quad \Rightarrow \quad R((L_\alpha, L_{\alpha+i})) = \alpha \ . \tag{6.81}$$

With this, Eq. (5.98) becomes

$$\begin{gathered} x \coloneqq \alpha \ , \\ y \coloneqq R((L_\alpha, L_{\alpha+i}) \triangleleft a) = R((L_{\alpha+a \bmod k}, L_{\alpha+i+a \bmod k})) = (\alpha+a) \bmod k \ , \\ xay^{-1} = a + \alpha - ((a+\alpha) \bmod k) \ . \\ \kappa_{ij}^{L_\alpha, L_{\alpha+i}}(a) = e^{\frac{2\pi i}{l}j(a+\alpha-((a+\alpha) \bmod k))} \ . \end{gathered} \tag{6.82}$$

As a second example, consider $G = S_3$ with $\omega = 1$ as bulk, with each of the four boundaries described in Section 5.1.3. Note that we have $\psi = 1$ for each of the four boundaries, and thus $\beta = 1$. First, consider the boundary given by $H^\partial = \{(0,0)\}$, with G-set equal to S_3 with right-regular action. There is one irreducible orbit for every S_3-element,

$$\begin{aligned} C_{(a,b)} = \{((x,y),(a,b)(x,y))\}_{0\le x<3, 0\le y<2} \ , \quad X_{(a,b)} &= ((0,0),(a,b)) \ , \\ H_{a,b} &= \{(0,0)\} \ . \end{aligned} \tag{6.83}$$

So we trivially have

$$\tilde{\kappa}_{(a,b),0}((0,0)) = 1 \ , \quad \kappa_{(a,b),0}^{((x,y),(a,b)(x,y))}(a,b) = 1 \ . \tag{6.84}$$

Next, consider the boundary $H^\partial = \langle (0,1) \rangle$, with G-set spelled out in Eq. (5.95), with $\mathrm{Stab}(L_\alpha) = \langle (\alpha, 1) \rangle$. The G-set A decomposes into two irreducible orbits,

$$\begin{gathered} C_0 = \{(L_\alpha, L_\alpha)\}_{0\le\alpha<3} \ , \quad X_0 = (L_0, L_0) \ , \quad H_0 = \{(0,b)\}_{0\le b<2} \\ C_1 = \{(L_\alpha, L_{\alpha+(-1)^\beta})\}_{0\le\alpha<3, 0\le\beta<2} \ , \quad X_1 = (L_0, L_1) \ , \quad H_1 = \{(0,0)\} \ . \end{gathered} \tag{6.85}$$

For $i = 0$, we set

$$\tilde{\kappa}_{0j}((0,b)) = (-1)^{bj} \ . \tag{6.86}$$

For the representatives, we set

$$X_0 \triangleleft (\alpha, 0) = (L_\alpha, L_\alpha) \quad \Rightarrow \quad R((L_\alpha, L_\alpha)) = (\alpha, 0) \ . \tag{6.87}$$

With this, Eq. (5.98) becomes

$$\begin{gathered} x \coloneqq (\alpha, 0) \ , \\ y \coloneqq R((L_\alpha, L_\alpha) \triangleleft (a,b)) = ((-1)^b(\alpha+a), 0) \ , \\ x(a,b)y^{-1} = (\alpha,0)(a,b)(-(-1)^b(\alpha+a),0) = (0,b) \ , \\ \kappa_{0j}^{(L_\alpha,L_\alpha)}((a,b)) = (-1)^{bj} \ . \end{gathered} \tag{6.88}$$

For $i = 1$, we trivially have

$$\tilde{\kappa}_{10}((0,0)) = 1\ , \quad \kappa_{10}^{(L_\alpha, L_{\alpha+(-1)^\beta})}(a,b) = 1\ . \tag{6.89}$$

Let us continue with the boundary given by $H^\partial = \langle(1,0)\rangle$, with G-set is spelled out in Eq. (5.96). There are two irreducible orbits,

$$C_i = \{(L_\alpha, L_{\alpha+i})\}_{0\leq\alpha<2}\ , X_i = (L_0, L_i)\ , H_i = \langle(1,0)\rangle\ , \quad 0 \leq i < 2\ . \tag{6.90}$$

For any i, we can set

$$\tilde{\kappa}_{ij}((a,0)) = e^{\frac{2\pi i}{3}aj}\ . \tag{6.91}$$

Next, we define representatives,

$$X_i \triangleleft (0,\alpha) = (L_\alpha, L_{\alpha+i}) \quad \Rightarrow \quad R((L_\alpha, L_{\alpha+i})) = (0,\alpha)\ . \tag{6.92}$$

With this, Eq. (5.98) becomes

$$\begin{aligned} x &\coloneqq (0,\alpha)\ , \\ y &\coloneqq R((L_\alpha, L_{\alpha+i}) \triangleleft (a,b)) = (0,\alpha+b)\ , \\ x(a,b)y^{-1} &= (0,\alpha)(a,b)(0,\alpha+b)^{-1} = ((-1)^\alpha a, 0)\ , \\ \kappa^{L_\alpha, L_{\alpha+i}}((a,b)) &= e^{\frac{2\pi i}{3}(-1)^\alpha aj}\ . \end{aligned} \tag{6.93}$$

Finally, consider the boundary $H^\partial = S_3$ with trivial G-set $A^\partial = \{L_0 = \{S_3\}\}$. There is a single trivial orbit

$$C_0 = \{(L_0, L_0)\}\ , \quad (X_0, X_0') = (L_0, L_0)\ , \quad H_0 = S_3\ . \tag{6.94}$$

We thus have

$$\kappa_{0j}^{(L_0,L_0)}((a,b))_\mu^\nu = \tilde{\kappa}_{0j}((a,b))_\mu^\nu = \tilde{\rho}_{0j}((a,b))_\mu^\nu\ , \tag{6.95}$$

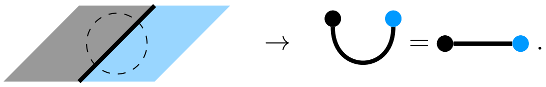
where $\tilde{\rho}$ are the S_3 irreps given in Eq. (6.45), noting that μ and ν are trivial for $j = 0, 1$.

6.4 (Invertible) corners in $2+1$D

In this section, we consider *corners*, which are 1-dimensional defects inside the boundary that separate two different boundary conditions.

6.4.1 tTS models

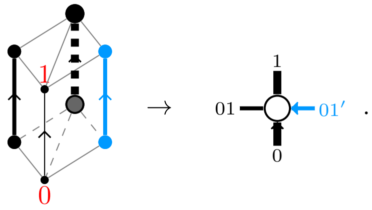
Corners are the same as boundary anyons, just that the boundary on one side is different from the other. We will distinguish the second from the first type of boundary by coloring the tensors, indices, and cell complex edges and vertices in blue. With this coloring, corners are 1-dimensional defects whose link is an interval with differently colored boundary vertices,


(6.96)

The overall extended manifold type is thus given by

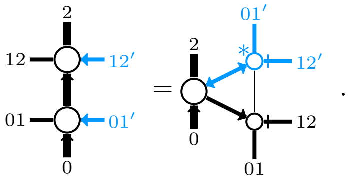
$$a : (3, \varnothing)\ , \quad b : (2, ●)\ , \quad c : (2, ●)\ , \quad d : (1, ●\!\!-\!\!-\!\!-\!\!●)\ . \tag{6.97}$$

A corner edge with the attached bulk and boundary edge, and the according tensor variable looks like follows,



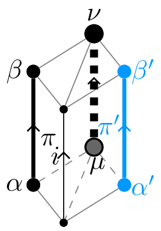
(6.98)

Note that since there are now two different boundaries colored differently, it is no longer necessary to remove the symmetries of the cellulated link with a tick. The moves ensuring topological invariance of corners is the same as for boundary anyons in Eq. (6.59) except that two different types of boundaries are involved,



(6.99)

The block-diagonal form associates the following weight to a corner edge,

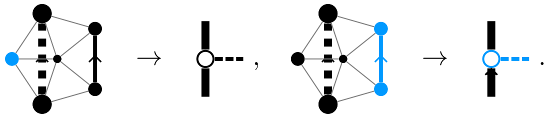


$$\rightarrow \quad \kappa^{i\,\alpha'\beta'\pi'\nu}_{\alpha\beta\pi\mu}\ . \tag{6.100}$$

6.4.2 Invertible corners and boundary phases

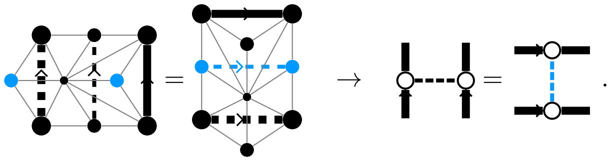
Just like bulk models, different boundaries or defects can be divided into *phases*, which are equivalence classes under local restructuring. The most natural way to define a phase is relative to a fixed model for the bulk or higher-dimensional defects. Then, two defect models can be considered in the same phase if there is an invertible domain wall that acts only in this defect, but not on the surrounding bulk or higher-dimensional defects. For boundaries in $2+1$ dimensions, these invertible domain walls acting only within the boundary are precisely corners between the two different copies of the boundary. All that remains is to impose the invertibility condition for the corners. Topologically, the invertibility condition corresponds to the moves shown in

Eqs. (5.131), (5.133), and (5.135), just that they now hold for the boundaries and corner, with a fixed bulk attached. If the bulk and boundary are robust, then only the move in Eq. (5.133) is non-trivial. To implement this move in terms of concrete cellulations, it is helpful to consider the tensor variables corresponding to the following volumes,



(6.101)

With this, the invertibility can be implemented via the following move,



(6.102)

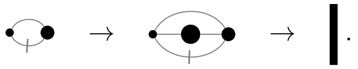
6.5 Twist defects in $2+1$D

Another example for 1-dimensional defect lines in a 3-dimensional spacetime are *twist defects*. Twist defects are lines where a domain wall between a model and itself terminates. The extended manifold type is thus given by

$$a:(3,\varnothing)\,,\quad b:(2,\bullet\quad\bullet)\,,\quad c:(1,\bigcirc\!\!\!\!\bullet)\,. \tag{6.103}$$

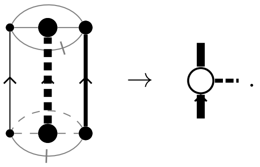
Here, a is the bulk, b is the domain wall (between the model and itself), and c is the twist defect. Note that since the domain wall is between the model and itself, both sides are represented by the same color in contrast to Section 5.2. Instead, we will have to distinguish the two sides of the domain wall by marking one side with ticks.

To construct extended cellulations of this type, we cellulate the link of c. The simplest cellulation is by a single domain wall (b) vertex and one bulk (a) vertex,



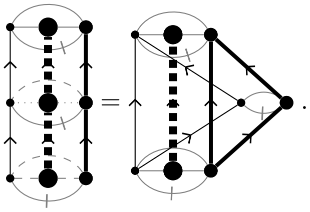
(6.104)

As shown, this link is attached to a c vertex, and we associate to it a bond dimension variable that we denote by a very thick line. Next, we take the cartesian product of this link with a single edge. To this cartesian product, we assign a new tensor variable,

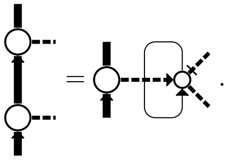


(6.105)

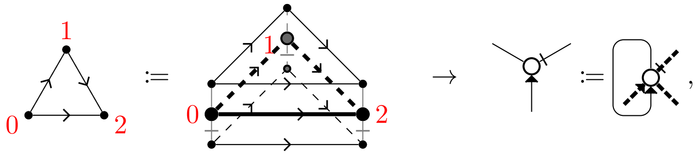
Like in the sections before, the moves are 2-1 Pachner moves of the 1-dimensional defect worldline, suitably padded by the surrounding bulk. Specifically, we impose the following move,

 (6.106)

On the left-hand side there are two of the twist-defect edges from Eq. (6.105) stacked on top of each other. On the right-hand side, there is a single twist-defect edge, padded with the standard domain-wall triangle from Eq. (5.104), whose adjacent bulk triangles on both sides are identified. In terms of tensor networks, this is

 (6.107)

Alternatively, we can consider the following compactification mapping



(6.108)

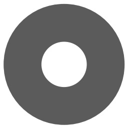
where the bulk triangles on the top and bottom are identified. Twist defects are then in one-to-one correspondence with boundaries of the compactified 2D-tTS model.

6.6 Torus ground states in $2 + 1$D

In the previous section, we have looked at 1-dimensional defects inside a $2 + 1$-dimensional spacetime. We will now go one dimension lower, and consider 0-dimensional defects in spacetime, i.e., embedded points. Physically, such 0-dimensional defects are ground states of the model on a fixed (small) extended cellulation of the link. As a first example, in this section, we consider 0-dimensional defects whose link is a torus, which we consequently refer to as *torus ground states*.

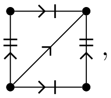
6.6.1 tTS models

The overall extended-manifold type describing torus-ground-state defects is given by

$a:(3,\varnothing)\,,\quad b:(0,$  $)\,.$ (6.109)

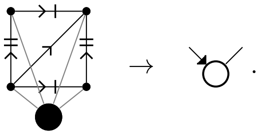
In the collapsed picture, these extended manifolds are 3-manifolds (a), with point-like *singularities* (b) with a torus-like neighborhood. A standard model for such a point is in the center of a cone of the torus, which we connect with a straight line to every point on the torus.

To define extended cellulations of this type, we pick a triangulation of the 2-manifold link. The smallest triangulation of the torus consists of two triangles,



(6.110)

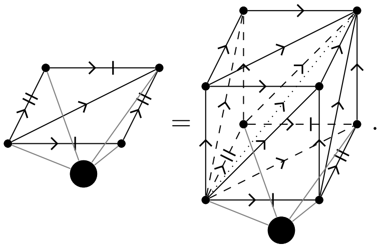
with left/right and bottom/top identified. This is the “standard” torus-ground-state 0-cell representative, to which a torus-ground-state vertex is attached,



(6.111)

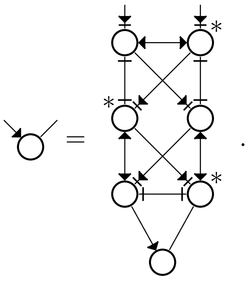
As shown, each ground-state-torus vertex is represented by a copy of a 2-index tensor variable.

Let us now look at the moves that impose topological invariance for the point singularities. As usual, these moves are Pachner moves of the defect manifolds themselves, suitably padded by the surrounding bulk or higher-dimensional defects. In 0 dimensions, there is only the trivial 0-0 Pachner move. This move becomes non-trivial when padding the right-hand side with a layer of bulk tetrahedra,



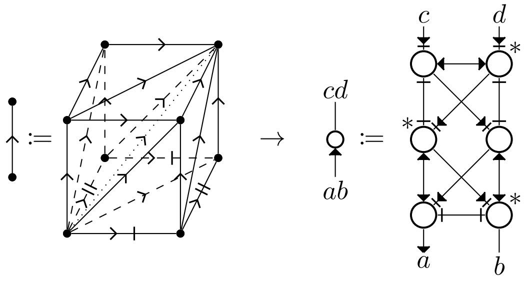
(6.112)

In terms of tensor networks, this equation becomes



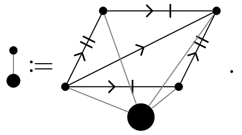
(6.113)

This can be formalized via a TS mapping from the 1D tTS to the 3D tTS taking the cartesian product with the torus,



(6.114)

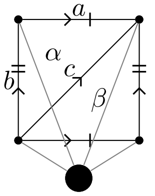
This mapping can be extended to a mapping from the 1D boundary tTS to the 3D torus-ground-state tTS,



(6.115)

With this, torus ground states as defined in Eq. (6.111) are in one-to-one correspondence with boundaries of the compactified 1D-tTS model.

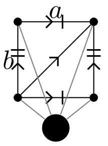
In the block-diagonal form, torus ground states are determined by a tensor,



$$\to \quad S_{abc\alpha\beta}\ . \qquad (6.116)$$

6.6.2 Cohomology models

Let us now classify torus ground states in cohomology models. The tensor in Eq. (6.116) simplifies to



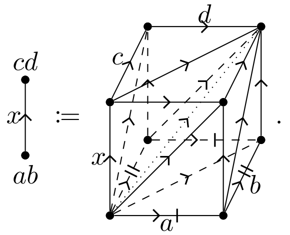
$$\to \quad S^{a,b}\ . \qquad (6.117)$$

Note that a and b are not fully independent but subject to the constraint

$$ab = ba\ , \qquad (6.118)$$

that is, a and b must commute.

In order to classify the torus ground states, we apply the compactification mapping in Eq. (6.114),



(6.119)

At each such edge, the labels satisfy the constraints $d = x^{-1}bx$ and $c = x^{-1}ax$. Thus the resulting 1-dimensional state sum is a G-set state sum with G-set

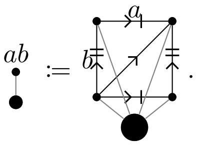
$$A = \{(a,b) \in G \times G : ab = ba\} \ , \quad (a,b) \triangleleft x = (x^{-1}ax, x^{-1}bx) \ . \tag{6.120}$$

The weight associated to the edge is given by

$$\begin{aligned} P^{a,b}(x) := \overline{\omega(a,b,x)}\omega(a,x,x^{-1}bx)\overline{\omega(x,x^{-1}ax,x^{-1}bx)} \\ \omega(b,a,x)\overline{\omega(b,x,x^{-1}ax)}\omega(x,x^{-1}bx,x^{-1}ax) \ . \end{aligned} \tag{6.121}$$

The topological invariance of this state-sum follows by construction from the topological invariance of the original state-sum. In other words, P is a G-set 1-cocycle with respect to the action in Eq. (6.120).

The extension of this mapping in Eq. (6.115) becomes,



(6.122)

So torus ground states are in one-to-one correspondence with boundaries of the 1-dimensional compactified state sum. This allows us to conveniently express the equation that S has to satisfy,

$$P^{a,b}(x)S^{a,b} = S^{x^{-1}ax,x^{-1}bx} \quad \forall x \ . \tag{6.123}$$

Note that as discussed around Eq. (5.57), the following equation would suffice to impose 1D topological invariance,

$$\frac{1}{|G|}\sum_x P^{a,b}(x)S^{x^{-1}ax,x^{-1}bx} = S^{a,b} \ , \tag{6.124}$$

but this is equivalent to Eq. (6.123). If we interpret S as a G-set 0-chain with respect to Eq. (6.120), then Eq. (6.123) becomes

$$dS = P \ . \tag{6.125}$$

This equation is linear in S, thus the ground states form a vector space, which is a feature of all 0-dimensional defects.

In order to efficiently find a basis for the torus ground states, we follow the prescription in Section 5.1.3. We first decompose the G-set in Eq. (6.120) into irreducible orbits, which are commuting double-conjugacy classes C_i. For each i, we choose X_i with $H_i := \mathrm{Stab}(X_i) \subset G$. Then we reduce P to an ordinary H_i 1-cocycle $\tilde{P}$ via Eq. (4.330),

$$\tilde{P}_i(x) = P^{X_i}(x) \; . \tag{6.126}$$

Then we find $\tilde{S}_i$ such that

$$\tilde{S}_i \tilde{P}_i(x) = \tilde{S}_i \; . \tag{6.127}$$

This is the case exactly if $\tilde{P}_i(x) = 1 \forall x$, in which case we can set $\widetilde{S}_i = 1$. In this case, a basis vector of the torus-ground-state space can be obtained via Eq. (5.97),

$$S_i^{a,b} = \begin{cases} 1 & \text{if } (a,b) \in C_i \; , \\ 0 & \text{otherwise} \; . \end{cases} \tag{6.128}$$

If there exists x with $\tilde{P}_i(x) \neq 1$ then i does not contribute a ground state.

Let us now consider some concrete examples. Let us start with $G = S_3$ denoted as in Eq. (4.305) with trivial $\omega = 1$. There are 8 different commuting double-conjugacy classes,

$$\begin{aligned} C_0 &= \{((0,0),(0,0))\} \; , \\ C_1 &= \{((0,0),(1,0)),((0,0),(2,0))\} \; , \\ C_2 &= \{((0,0),(0,1)),((0,0),(1,1)),((0,0),(2,1))\} \; , \\ C_3 &= \{((1,0),(0,0)),((2,0),(0,0))\} \; , \\ C_4 &= \{((0,1),(0,0)),((1,1),(0,0)),((2,1),(0,0))\} \; , \\ C_5 &= \{((1,0),(1,0)),((2,0),(2,0))\} \; , \\ C_6 &= \{((0,1),(0,1)),((1,1),(1,1)),((2,1),(2,1))\} \; , \\ C_7 &= \{((1,0),(2,0)),((2,0),(1,0))\}. \end{aligned} \tag{6.129}$$

Since $\omega = 1$, we have $P = 1$, so all of these are basis states for the space of $S^{a,b}$.

As a next example, consider $G = \mathbb{Z}_2^3$ with

$$\omega((a_0,a_1,a_2),(b_0,b_1,b_2),(c_0,c_1,c_2)) = (-1)^{a_0 b_1 c_2} \; . \tag{6.130}$$

Since G is abelian, every pair of G-elements $((a_0,a_1,a_2),(b_0,b_1,b_2))$ defines a commuting double-conjugacy class. We thus find

$$\tilde{P}_{(a_0,a_1,a_2),(b_0,b_1,b_2)}((x_0,x_1,x_2)) = (-1)^{x_0(a_1b_2+a_2b_1)+x_1(a_0b_2+b_0a_2)+x_2(a_0b_1+b_0a_1)} \; . \tag{6.131}$$

We see that a pair of G-elements defines a basis state if

$$a_1b_2 = a_2b_1 \; , \quad a_0b_2 = b_0a_2 \; , \quad a_0b_1 = b_0a_1 \; . \tag{6.132}$$

This is true iff

$$\begin{gathered} a_0 = a_1 = a_2 = 0 \text{ , or} \quad b_0 = b_1 = b_2 = 0 \; , \\ \text{or} \quad (a_0 = b_0 \text{ and } a_1 = b_1 \text{ and } a_2 = b_2) \; . \end{gathered} \tag{6.133}$$

There are $2^3 = 8$ configurations satisfying the first, second, and third condition, respectively. $((0,0,0),(0,0,0))$ is the only configuration that satisfies more than one of the conditions, namely all three. Thus, we have $3 \cdot 8 - 2 = 22$ different basis states, specifically,

$$\begin{gathered}
((0,0,0),(0,0,0)), \quad ((0,0,1),(0,0,0)), \quad ((0,1,0),(0,0,0)),\\
((0,1,1),(0,0,0)), \quad ((1,0,0),(0,0,0)), \quad ((1,0,1),(0,0,0)),\\
((1,1,0),(0,0,0)), \quad ((1,1,1),(0,0,0)), \quad ((0,0,0),(0,0,1)),\\
((0,0,0),(0,1,0)), \quad ((0,0,0),(0,1,1)), \quad ((0,0,0),(1,0,0)),\\
((0,0,0),(1,0,1)), \quad ((0,0,0),(1,1,0)), \quad ((0,0,0),(1,1,1)),\\
((0,0,1),(0,0,1)), \quad ((0,1,0),(0,1,0)), \quad ((0,1,1),(0,1,1)),\\
((1,0,0),(1,0,0)), \quad ((1,0,1),(1,0,1)), \quad ((1,1,0),(1,1,0)),\\
((1,1,1),(1,1,1)).
\end{gathered} \tag{6.134}$$

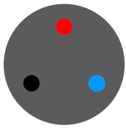
6.7 Anyon fusion events in $2+1$D

As a next example for 0-dimensional defects, we consider *anyon fusion events*, which are points in the 3-dimensional spacetime where three anyon worldlines meet.

6.7.1 tTS models

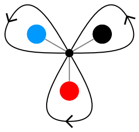
Consider an infinitesimal 2-sphere around an anyon fusion event, and its overlap with the surrounding spacetime. Every adjacent anyon worldline intersects with the 2-sphere at a point, so the link of anyon fusion events is a 2-sphere with three embedded points. So the overall extended manifold type is given by

$$a : (3, \varnothing)\ , \quad b : (1, \bigcirc)\ , \quad {\color{blue} c} : (1, \bigcirc)\ , \quad {\color{red} d} : (1, \bigcirc)\ , \tag{6.135}$$

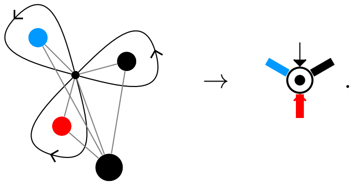
$e : (0,$  $)$.

Physically, anyon fusion events are at the same time ground states on a sphere with three anyons. Note that here we think of anyons as defects, so "ground state with anyons" means ground states of a Hamiltonian that is altered at some points to enforce the existence of anyons.

In order to construct extended cellulations of the above type, we start by cellulating the e link. The simplest triangulation consists of three anyon vertices (of different types) and one bulk triangle,

 . (6.136)

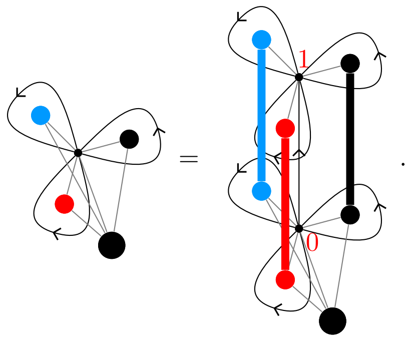
Note that in this drawing, the topology is a sphere and the bulk triangle is "wrapped around the back". Each anyon-fusion vertex is attached to the cellulated link above,



(6.137)

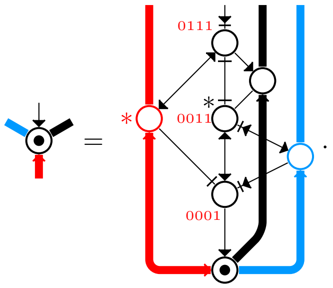
As shown, we associate a copy of a 4-index tensor variable to each anyon-fusion vertex. The three anyon vertices are attached to three anyon edges of the adjacent anyon worldlines. The bulk triangle is attached to a bulk tetrahedron. Intuitively, it might help to think of the left-hand side of Eq. (6.137) as a volume at whose center is the anyon-fusion vertex.

Similar to the previous section, the moves are 0-0 Pachner moves where the right-hand side is padded with bulk and other higher-dimensional regions:



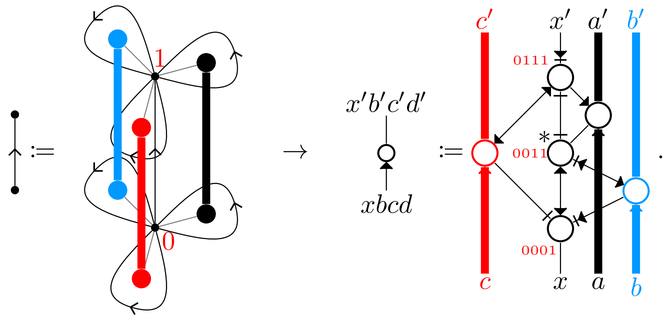
(6.138)

On the left-hand side, there is a single anyon-fusion vertex. On the right-hand side, there is an anyon-fusion vertex that is surrounded by three anyon edges, and one bulk triangle prism. The triangle prism can be triangulated using three tetrahedra. To not clutter the drawing, we have not shown the diagonal edges resulting from this triangulation. The topology of the padding is just the link in Eq. (6.135) times B_1. In terms of tensor-network diagrams, the move becomes



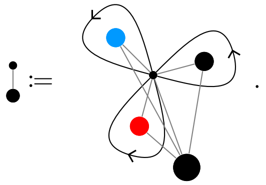
(6.139)

To further understand the structure of this equation, let us consider the following mapping from the 1D tTS to the 3D anyon-fusion tTS. On the level of topology, the mapping consists in taking the cartesian product with the link in Eq. (6.135). On a combinatorial/geometric level, we take the cartesian product of the 1-dimensional cellulation with the cellulated link in Eq. (6.136),



(6.140)

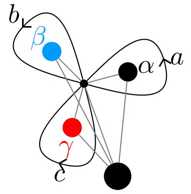
This mapping can be extended to a mapping from the 1D boundary tTS via



(6.141)

With this, torus-singularity point defects as defined in Eq. (6.111) are in one-to-one correspondence with boundaries of the compactified 1-dimensional model.

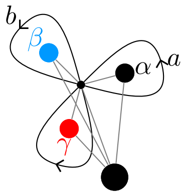
In the block-diagonal form, the tensor in Eq. (6.137) becomes,



$$\rightarrow \quad S^{abc}_{\alpha\beta\gamma} \,. \tag{6.142}$$

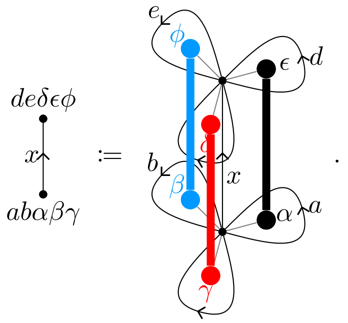
6.7.2 Cohomology models

Let us now look at anyon fusion events for cohomology models. The block-diagonal weight shown in Eq. (6.142) simplifies slightly,



$$\rightarrow \quad S^{a,b,\alpha\beta\gamma} \,. \tag{6.143}$$

The compactification mapping in Eq. (6.140) becomes



(6.144)

The resulting 1-dimensional state sum has one G-label at every edge, and two G-labels and three B-labels at every vertex, subject to the constraints $d = x^{-1}ax$, $e = x^{-1}bx$ at every edge. Thus, it is a G-set state sum with G-set

$$A = G \times G\;, \quad (a,b) \triangleleft x = (x^{-1}ax, x^{-1}bx)\;, \tag{6.145}$$

with additional free vertex labels whose dimension is

$$B_f^{a,b} := B^a B^b B^{ab}\;, \tag{6.146}$$

depending on $(a,b) \in A$, where "f" stands for "fusion", recalling that B^a is the dimension of the free anyon-vertex label. Note that in contrast to the previous Section 6.6.2, we do not restrict to commuting pairs in $A := G \times G$. The weight associated the edge above is obtained by considering the cellulation on the right consisting of three anyon edges and three bulk tetrahedra,

$$\begin{aligned} P^{a,b}(x)_{\alpha\beta\gamma}^{\delta\epsilon\phi} := \omega(a,b,x)\overline{\omega(a,x,x^{-1}bx)}\omega(x,x^{-1}ax,x^{-1}bx)& \\ \rho^a(x)_\alpha^\delta \rho^b(x)_\beta^\epsilon \overline{\rho^{ab}(x)_\gamma^\phi}&\;. \end{aligned} \tag{6.147}$$

The extension of the mapping in Eq. (6.141) identifies anyon fusion events with boundaries of the compactified 1-dimensional state sum. Using this identification, the defining equation for anyon fusion events becomes

$$\sum_{\alpha\beta\gamma} P^{a,b}(x)_{\alpha\beta\gamma}^{\delta\epsilon\phi} S^{a,b,\alpha\beta\gamma} = S^{x^{-1}ax,x^{-1}bx,\delta\epsilon\phi}\;. \tag{6.148}$$

A basis for the vector space of anyon fusion events can thus be efficiently found using the methods in Section 5.1.3. First, we determine the irreducible orbits of the G-set, which are double-conjugacy classes C_i. For each i, we choose a representative X_i with $H_i := \mathrm{Stab}(X_i)$. Then, we reduce the compactified 1-dimensional model to a H_i cohomology model with only free vertex labels, given by

$$\tilde{P}_i(x)_{\alpha\beta\gamma}^{\delta\epsilon\phi} := P^{X_i}(x)_{\alpha\beta\gamma}^{\delta\epsilon\phi}\;, \quad x \in H_i\;, \tag{6.149}$$

which defines an ordinary H_i representation. Then we look for basis vectors $\tilde{S}_i^{\alpha\beta\gamma}$ that are invariant under $\tilde{P}$,

$$\sum_{\alpha\beta\gamma} \tilde{P}_i(x)^{\delta\epsilon\phi}_{\alpha\beta\gamma} \tilde{S}^{\alpha\beta\gamma}_{i,j} = \tilde{S}^{\delta\epsilon\phi}_{i,j} \ . \tag{6.150}$$

Then S can be obtained from Eq. (5.97),

$$S^{a,b,\alpha\beta\gamma}_{i,j} = \sum_{\delta\epsilon\phi} P^{X_i}(x)^{\alpha\beta\gamma}_{\delta\epsilon\phi} \tilde{S}^{\delta\epsilon\phi}_{i,j} \ . \tag{6.151}$$

The dimension of this support vector space is the ground space dimension on the sphere with three anyons, or equivalently the fusion multiplicity N of the three anyons given by ρ, ρ and ρ. It can be calculated by taking the trace of the projector, or another compactification to a 0-dimensional state-sum via the cartesian product with the circle in Eq. (6.12). Plugging Eq. (6.144) into this compactification means summing and identifying a and d, b and e, c and f, α and δ, β and ϵ, γ and ϕ, yielding,

$$\begin{aligned} N = &\sum_{\substack{a,b,x\in Z(a)\cap Z(b),\\ \alpha,\beta,\gamma}} P^{a,b}(x)^{\alpha\beta\gamma}_{\alpha\beta\gamma} \\ = &\sum_{a,b,x\in Z(a)\cap Z(b)} \omega(a,b,x)\overline{\omega(a,x,b)}\omega(x,a,b)\operatorname{Tr}(\rho^a(x))\operatorname{Tr}(\rho^b(x))\operatorname{Tr}(\overline{\rho^{ab}(x)}) \ , \end{aligned} \tag{6.152}$$

where $Z(x)$ denotes the centralizer of x, that is, the set of G-elements that commute with x.

As a concrete example consider S_3 as denoted in Eq. (4.305) with $\omega = 1$, and different triples of anyons from Section 6.2.3. The irreducible orbits of Eq. (6.145) are the commuting double-conjugacy classes in Eq. (6.129), together with the non-commuting ones,

$$\begin{aligned} C_8 &= \{((0,1),(1,1)),((0,1),(2,1)),((1,1),(0,1)),((1,1),(2,1)),\\ &\quad ((2,1),(0,1)),((2,1),(1,1))\}, \quad X_8 = ((0,1),(1,1)), \quad H_8 = 1\\ C_9 &= \{((0,1),(1,0)),((0,1),(2,0)),((1,1),(1,0)),((1,1),(2,0)),\\ &\quad ((2,1),(1,0)),((2,1),(2,0))\}, \quad X_9 = ((0,1),(1,0)), \quad H_9 = 1\\ C_{10} &= \{((1,0),(0,1)),((2,0),(0,1)),((1,0),(1,1)),(2,0),(1,1)),\\ &\quad ((1,0),(2,1)),((2,0),(2,1))\}, \quad X_{10} = ((1,0),(0,1)), \quad H_{10} = 1 \ . \end{aligned} \tag{6.153}$$

Since determining a basis of fusion events for all 8^3 triples of anyons of the model would be too lengthy for this discussion here, we restrict to a few illustrative examples. First, let us consider the fusion of three times the "non-abelian charge" anyon,

$$\rho = \rho_{0,2} \ , \quad \rho = \rho_{0,2} \ , \quad \rho = \rho_{0,2} \ . \tag{6.154}$$

The dimension $B_f^{a,b}$ in Eq. (6.146) is 8 for $a = b = (0,0)$, and 0 otherwise. So the only double-conjugacy class that yields a basis vector is C_0 with $H_0 = S_3$. Since $\omega = 1$, we find

$$\tilde{P}_0((a,b))^{\delta\epsilon\phi}_{\alpha\beta\gamma} = \tilde{\rho}_{0,2}((a,b))^{\delta}_{\alpha}\tilde{\rho}_{0,2}((a,b))^{\epsilon}_{\beta}\overline{\tilde{\rho}_{0,2}((a,b))^{\phi}_{\gamma}} \ , \tag{6.155}$$

where $\tilde{\rho}_{0,2}$ is the standard irrep of S_3 given in Eq. (6.45). Mathematically, $\tilde{P}_0$ is the S_3 representation given by the tensor product of three times the standard irrep. The subspace of invariant vectors S is the subspace of this representation corresponding to the trivial irrep. From the representation theory of S_3, we know that trivial irrep is contained once in the tensor product, so the subspace is one-dimensional. An element of the subspace can be obtained by simply applying $\tilde{P}_0$ to some random vector, as long as the result is non-zero. For example, we can take,

$$\begin{aligned}\tilde{S}_{0,0}^{\delta\epsilon\phi} &= \frac{1}{|H_0|}\sum_{(a,b)\in H_0,\alpha,\beta,\gamma} \tilde{P}_0((a,b))^{\delta\epsilon\phi}_{\alpha\beta\gamma}\delta_{\alpha,0}\delta_{\beta,0}\delta_{\gamma,1}\\ &= \frac{1}{6}\sum_{(a,b)\in S_3} \tilde{P}_0((a,b))^{\delta\epsilon\phi}_{001}\\ &= \frac{1}{6}\sum_{0\leq a<3,0\leq b<2} \tilde{\rho}_{02}((a,b))^{\delta}_{0}\tilde{\rho}_{02}((a,b))^{\epsilon}_{0}\overline{\tilde{\rho}_{02}((a,b))^{\phi}_{1}}\\ &= \frac{1}{6}\sum_{0\leq a<3,0\leq b<2} \delta_{b+0,\delta}e^{\frac{2\pi i}{3}(-1)^0 a}\delta_{b+0,\epsilon}e^{\frac{2\pi i}{3}(-1)^0 a}\delta_{b+1,\phi}e^{-\frac{2\pi i}{3}(-1)^1 a}\\ &= \frac{1}{6}\delta_{\delta=\epsilon=\phi+1}\sum_{0\leq a<3} e^{\frac{2\pi i}{3}a}e^{\frac{2\pi i}{3}a}e^{\frac{2\pi i}{3}a}\\ &= \frac{1}{2}\delta_{\delta=\epsilon=\phi+1}\ .\end{aligned} \tag{6.156}$$

Since $H_0 = G$, Eq. (5.97) is trivial, and we have

$$S_{0,0}^{(0,0),(0,0),\alpha\beta\gamma} = \frac{1}{2}\delta_{\alpha=\beta=\gamma+1}\ . \tag{6.157}$$

As another example, consider fusion events of

$$\rho = \rho_{1,j}\ , \quad \rho = \rho_{1,j'}\ , \quad \rho = \rho_{2,j''}\ , \quad 0\leq j,j' < 2\ , \quad 0 < j'' < 3\ . \tag{6.158}$$

First, we find the double-conjugacy classes with $B_f^{a,b} \neq 0$. For these double-conjugacy classes, the first and second element need to be in the S_3-conjugacy class C_1 in Eq. (6.44), and their product needs to be in C_2. This is only the case for C_8, with $H_8 = 1$, in which case $B_f^{a,b} = 1$. We find

$$P^{((-1)^\beta\alpha,1),((-1)^\beta(\alpha+1),1)}(a,b) = (-1)^{(j+j')b}e^{\frac{2\pi i}{3}j''(-1)^\beta a}\ , \tag{6.159}$$

that trivially reduces to

$$\tilde{P}_8((0,0)) = 1\ . \tag{6.160}$$

Thus, we get one anyon-fusion basis state defined by $\tilde{S}_{8,0} = 1$. In order to obtain S from $\tilde{S}$ via Eq. (5.97), we choose standard representatives,

$$\begin{aligned}X_8 \triangleleft (a,b) &= ((a,b)^{-1}(0,1)(a,b),(a,b)^{-1}(1,1)(a,b))\\ &= ((-(-1)^b a,b)(-a,b+1),((-(-1)^b a,b)(-a+1,b+1))\\ &= (((-1)^b a,1),((-1)^b(a+1),1))\ ,\\ &\Rightarrow R(((-1)^b a,1),((-1)^b(a+1),1)) = (a,b)\ .\end{aligned} \tag{6.161}$$

With this, Eq. (5.97) yields

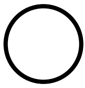
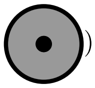
$$\begin{aligned}
x &\coloneqq (a,b)\ ,\\
y &\coloneqq R(((-1)^\beta\alpha,1),((-1)^\beta(\alpha+1),1)\triangleleft(a,b))\\
&= (\alpha,\beta)(a,b) = (\alpha+(-1)^\beta a,\beta+b)\ ,\\
x(a,b)y^{-1} &= (a,b)(\alpha,\beta)^{-1} = (a-(-1)^{b+\beta}\alpha,b+\beta)\ ,\\
S_{8,0}^{((-1)^\beta\alpha,1),((-1)^\beta(\alpha+1),1)} &= P_8^{(0,1),(1,1)}((a-(-1)^{b+\beta}\alpha,b+\beta))\tilde{S}_{8,0}\\
&= (-1)^{(j+j')(b+\beta)}e^{\frac{2\pi i}{3}j''(a-(-1)^{b+\beta}\alpha)}\ .
\end{aligned} \tag{6.162}$$

6.8 Anyon condensation events in $2+1$D

As a next example for a 0-dimensional defect, let us consider points in the boundary where anyon worldlines terminate. In physics terms, such events are referred to as *anyon condensation*.

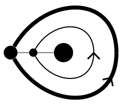
6.8.1 tTS models

Consider an infinitesimal sphere around an anyon condensation vertex. Half of this sphere overlaps with the bulk, and inside this half-sphere, the intersection with the adjacent anyon worldline yields a point. So the link of an anyon condensation event is a disk with an embedded point. Thus, the overall extended manifold type is given by

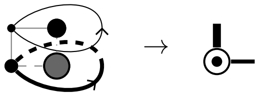
$a : (3, \varnothing)$, $b : (2,$ ● $)$, $c : (1,$  $)$, $d : (0,$  $)$. (6.163)

a is the bulk, b the boundary, c the anyon worldline(s), and d the condensation event(s).

In order to define extended cellulations of the above type, we need a standard anyon-condensation 0-cell representative, which we do by choosing an extended cellulation of the d link. The simplest such cellulation consists of one looping boundary edge and one anyon vertex, both adjacent to the same bulk edge,

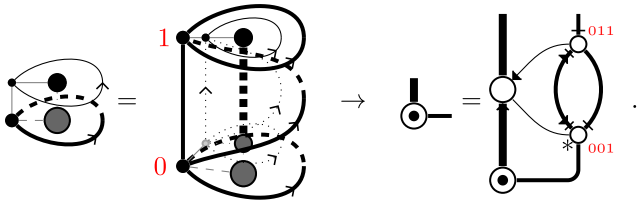
 . (6.164)

Every anyon-condensation vertex is adjacent to the surrounding bulk, boundary, and anyon worldlines via this 0-cell representative,

 . (6.165)

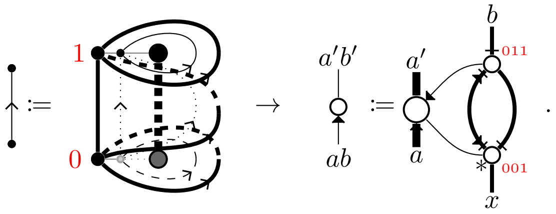
As shown, we associate a copy of a 2-index tensor variable to every such anyon-condensation vertex. Intuitively, it might be best to picture such a condensation vertex as a 3-cell with the shape of a 1-gon prism, with the anyon-condensation vertex in its center.

As usual, topological invariance is imposed by a 0-0 Pachner move, which becomes non-trivial by padding the right-hand side with suitable bulk, boundary, or anyon cells,



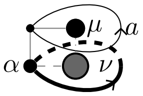
(6.166)

As shown, this padding consists of one anyon edge that has two boundary triangles wrapped around it. Alternatively, we can consider the following compactification mapping from the 1D tTS to the 3D boundary tTS with additional bulk anyons,



(6.167)

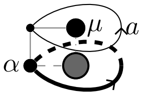
In the block-diagonal form, Eq. (6.165) is replaced by the following tensor,



$$\rightarrow \quad S^{a,\alpha,\mu\nu} \,. \tag{6.168}$$

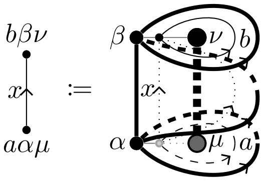
6.8.2 Cohomology models

In this section, we will look at anyon condensation events for bulk cohomology models. In this case, the weight in Eq. (6.168) simplifies to



$$\rightarrow \quad S^{a,\alpha,\mu} \,. \tag{6.169}$$

The compactification mapping in Eq. (6.167) becomes



$$\rightarrow \quad P^{a,\alpha}(x)^{\nu}_{\mu} := \psi^{\alpha}(a,x)\overline{\psi^{\alpha}(x,x^{-1}ax)}\rho^{a}(x)^{\nu}_{\mu} \,. \tag{6.170}$$

The resulting 1-dimensional state sum is a G-set state sum with G-set

$$A \coloneqq \{(a,\alpha)\in G\times A^\partial : \alpha \triangleleft a = \alpha\}\ , \quad (a,\alpha)\triangleleft g = (g^{-1}ag, \alpha\triangleleft g)\ , \tag{6.171}$$

and an additional free vertex label μ of dimension B^a. The defining equation for a condensation event is thus

$$S^{x^{-1}ax,\alpha\triangleleft x,\nu} = \sum_\mu P^{a,\alpha}(x)^\nu_\mu S^{a,\alpha,\mu}\ . \tag{6.172}$$

As usual, a basis for the anyon condensation events can be obtained using the methods from Section 5.1.3.

As a concrete example, let us consider $G = S_3$ with $\omega = 1$ as bulk, with the boundary with $H^\partial = \{(0,i)\}_{0\leq i<2}$ and $\psi = 1$, whose G-action is given explicitly in Eq. (5.95). The G-set in Eq. (6.171) decomposes into two irreducible orbits,

$$\begin{aligned} C_0 &= \{((0,0),L_\alpha)\}_{0\leq\alpha<3}, \quad X_0 = ((0,0),L_0), \quad H_0 = H^\partial \\ C_1 &= \{((\alpha,1),L_\alpha)\}_{0\leq\alpha<3}, \quad X_1 = ((0,1),L_0), \quad H_1 = H^\partial\ . \end{aligned} \tag{6.173}$$

Since $\psi = 1$, $\widetilde{P}$ via Eq. (6.170) reduces to

$$\widetilde{P}_i((0,b)) = \rho^{(0,i)}((0,b))^\nu_\mu\ . \tag{6.174}$$

So we find the following $\tilde{P}$ for the different anyons,

$$\begin{aligned} \rho_{0,j} :& \tilde{P}_0((0,b)) = (-1)^{bj}, & \tilde{P}_1 &= \varnothing\ , \quad 0\leq j<1 \\ \rho_{0,2} :& \tilde{P}_0((0,b))^\nu_\mu = \delta_{b+\mu,\nu}, & \tilde{P}_1 &= \varnothing, \\ \rho_{1,j} :& \tilde{P}_0 = \varnothing, & \tilde{P}_1((0,b)) &= (-1)^{bj}\ , \quad 0\leq j<1 \\ \rho_{2,j} :& \tilde{P}_0 = \varnothing, & \tilde{P}_1 &= \varnothing\ , \quad 0\leq j<2\ . \end{aligned} \tag{6.175}$$

$\varnothing$ indicates that the dimension $B^{(0,b)}$ of μ and ν is 0. Thus, we find one basis vector each for the following anyons

$$\begin{aligned} \rho_{0,0} &: \tilde{S}_{0,0} = 1 \\ \rho_{0,2} &: \tilde{S}^\mu_{0,0} = 1\forall\mu \\ \rho_{1,0} &: \tilde{S}_{1,0} = 1\ , \end{aligned} \tag{6.176}$$

and none for all other anyons. In order to obtain S from $\tilde{S}$ via Eq. (5.97), we choose standard representatives,

$$\begin{aligned} R(((0,0),L_\alpha)) &= (\alpha,0)\ , \\ R(((\alpha,1),L_\alpha)) &= (\alpha,0)\ . \end{aligned} \tag{6.177}$$

For $i = 0$, we then have

$$\begin{aligned}
x &\coloneqq (\alpha, 0)\ , \\
y \coloneqq R(((0,0), L_\alpha) \triangleleft (a,b)) &= R((0,0), L_{(-1)^b(\alpha+a)}) = ((-1)^b(\alpha+a), 0)\ , \\
x(a,b)y^{-1} &= (\alpha,0)(a,b)((-1)^b(\alpha+a),0)^{-1} \\
&= (\alpha+a,b)(-(-1)^b(\alpha+a),0) = (0,b)\ , \\
\rho_{0,0} : S^{(0,0),L_\alpha} &= P^{(0,0),L_\alpha}((0,b)) = \rho_{0,0}^{(0,0)}((0,b)) = 1\ , \\
\rho_{0,2} : S^{(0,0),L_\alpha,\mu} &= \sum_\nu P^{(0,0),L_\alpha}((0,b))^\mu_\nu = \sum_\nu \rho_{0,2}^{(0,0)}((0,b))^\mu_\nu = 1 \forall \mu\ .
\end{aligned} \tag{6.178}$$

For $i = 1$, we then have

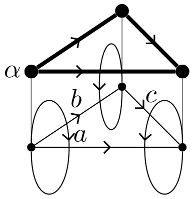
$$\begin{aligned}
x &\coloneqq (\alpha, 0)\ , \\
y &\coloneqq R(((\alpha,1), L_\alpha) \triangleleft (a,b)) \\
&= R((((-1)^b(\alpha+a),1), L_{(-1)^b(\alpha+a)}) = ((-1)^b(\alpha+a), 0)\ , \\
x(a,b)y^{-1} &= (\alpha,0)(a,b)((-1)^b(\alpha+a),0)^{-1} \\
&= (\alpha+a,b)(-(-1)^b(\alpha+a),0) = (0,b)\ , \\
\rho_{1,0} : S^{(\alpha,1),L_\alpha} &= P^{(\alpha,1),L_\alpha}((0,b)) = \rho_{1,0}^{(\alpha,1)}((0,b)) = 1\ .
\end{aligned} \tag{6.179}$$

6.9 Membrane defects in $3 + 1$D

In this section we show how to classify $1 + 1$-dimensional defects inside a $3 + 1$-dimensional bulk model, corresponding to the following extended manifold type,

$$a : (4, \varnothing)\ , \quad b : (2, \bigcirc)\ . \tag{6.180}$$

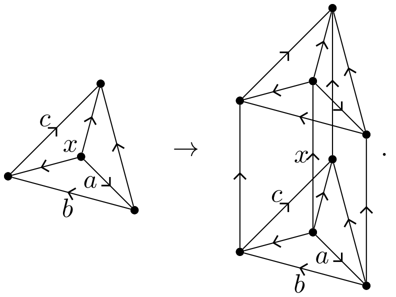
To get straight to computations, we do this directly for cohomology models determined by a group 4-cocycle ω. The standard 2-cell representative for b is a triangle adjacent to a 3-cellulation given by the triangle times a looping edge,



$$\rightarrow \quad \eta^{a,\alpha}(b,c)\ . \tag{6.181}$$

In order to not clutter the above picture, we have not drawn "diagonal" edges resulting from triangulating the volume with three tetrahedra. The cohomology model has one additional A^m-element at every membrane (b) vertex, where A^m is a G-set and the A^m-elements have to fulfill the usual constraint at every edge. There is a weight associated to every b triangle as shown above.

To express the topological moves more concisely, we compactify the 4-dimensional state sum by a cartesian product with the looping-edge, in order to obtain a 3-dimensional state sum,



(6.182)

On the left-hand side, there is a tetrahedron. On the right-hand side, there is the cartesian product of the tetrahedron with a looping edge, yielding a 4-volume of topology $B_3 \times S_1$. Note that in the drawing the top and bottom tetrahedron are identified. The boundary of the 4-volume is a 3-cellulation consisting of four triangle prisms. The resulting 3-dimensional state sum is a G-set state sum (c.f. Section 4.16.2), with G-set given by G with conjugation as in Eq. (6.33). The weight of the state sum associated to a 3-simplex can be read off from Eq. (6.182) by triangulating the right-hand side with four 4-simplices,

$$\begin{aligned}\beta^x(a,b,c) = \omega(x,a,b,c)\overline{\omega(a,a^{-1}xa,b,c)}\\ \omega(a,b,(ab)^{-1}xab,c)\overline{\omega(a,b,c,(abc)^{-1}xabc)}\ .\end{aligned} \tag{6.183}$$

Membrane defects given by η as above are then in one-to-one correspondence with G-set boundaries of this G-set bulk state sum. Note that since it is known that all boundary phases of $2+1$-dimensional cohomology models can be represented by G-set boundaries, we also know that all membrane-defect phases of $3+1$-dimensional cohomology models can be represented by a model of the form in Eq. (6.181).

In order to classify the different G-set boundaries, we follow the prescription at the end of Section 5.1.3. We first decompose $A = G$ into irreducible orbits C_i, which are just conjugacy classes, choose a representative $X_i \in C_i$ and consider $H_i = \operatorname{Stab}(X_i) \subset G$. For each conjugacy class C_i, we then obtain an ordinary H_i group 3-cocycle $\tilde{\beta}$ via Eq. (4.324),

$$\tilde{\beta}(a,b,c) = \beta^{X_i}(a,b,c) \quad \forall a,b,c \in H_i\ . \tag{6.184}$$

Then, we choose a subgroup $H_i^\partial \subset H_i$ and a H_i^∂ 2-cochain $\tilde{\eta}$ such that $d\tilde{\eta} = \tilde{\beta}|_{H_i^\partial}$. Finally, we use apply Eq. (5.99) to obtain η from $\tilde{\eta}$. Thereby, A_i^m is given by $H_i^\partial \backslash H_i$. All in all, we find that membrane defects in $3+1$-dimensional cohomology models are classified by 1) a conjugacy class C_i, 2) a subgroup $H_i^\partial \subset H_i$ such that $\tilde{\beta}|_{H_i^\partial}$ is a H_i^∂ group coboundary, and 3) an element in $H^2(H_i^\partial, U(1))$ relative to an arbitrary standard 2-cochain α such with $d\alpha = \tilde{\beta}|_{H_i^\partial}$.

As a first example, let us consider $G = \mathbb{Z}_l \times \mathbb{Z}_l$ with group 4-cocycle

$$\omega((a,a'),(b,b'),(c,c'),(d,d')) = e^{\frac{2\pi i p}{l^3}(\overline{a}+\overline{b}-\overline{a+b})(\overline{c}'+\overline{d}'-\overline{c'+d'})}\ . \tag{6.185}$$

Since G is abelian, the conjugacy classes are just single elements,

$$C_{(i,j)} = \{(i,j)\} \; , \quad X_{(i,j)} = (i,j) \; , \quad H_{(i,j)} = G = \mathbb{Z}_l \times \mathbb{Z}_l \; . \tag{6.186}$$

Furthermore, the conjugations in Eq. (6.183) disappear. Since further the 4-cocycle is symmetric under exchanging the first two, or the last two arguments, we have

$$\beta^{(i,j)}((a,a'),(b,b'),(c,c')) = 1 \; . \tag{6.187}$$

Hence, membrane defects are labeled by (i,j), as well as an arbitrary subgroup $\mathbb{Z}_k \times \mathbb{Z}_m \sim H^\partial \subset \mathbb{Z}_l \times \mathbb{Z}_l$, and a group 2-cocycle on $\mathbb{Z}_k \times \mathbb{Z}_m$. There are $\gcd(k,m)$ such inequivalent 2-cocycles.

As a next example, let us consider $G = S_3$ with the trivial 4-cocycle $\omega = 1$. The conjugacy classes, representatives, and stabilizers can be found in Eq. (6.44). Since $\omega = 1$, we have $\beta = 1$. For C_0, $H_0 = S_3$ has four subgroups, listed in Eq. (5.94). Since $H^2(S_3, U(1))$ is trivial, there is exactly one membrane defect for each subgroup, given by $\tilde{\eta} = 1$ For C_1, $H_1 = \mathbb{Z}_2$ has two subgroups, $\{0\}$, and $\mathbb{Z}_2$. Again, $H^2(\mathbb{Z}_2, U(1))$ is trivial, so this yields two membrane defects. Finally, for C_2, $H_2 = \mathbb{Z}_3$ has two subgroups yielding two membrane defects again. So in total we find that there are 8 different robust membrane defects.

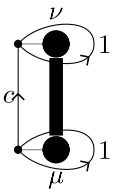
6.10 Anyons in $3+1$D

Next we consider the $0+1$-dimensional defects, that is, "anyon" worldlines in a $3+1$-dimensional bulk. That is, we consider the tTS corresponding to the following extended manifold type,

$$a : (4, \varnothing) \; , \quad b : (1, \bullet) \; . \tag{6.188}$$

The link of b is a 2-sphere. Extended manifolds of this type are (oriented) 4-manifolds with embedded b worldline loops. The worldlines are framed, which means that the ϵ-sphere inside their normal space is smoothly identified with a standard sphere. For every closed worldline loop there are exactly two such choices of framing.

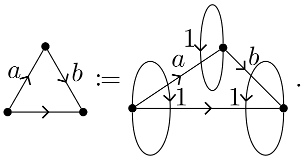
As in the previous section, we skip the general tTS models for these defects, and go right to cohomology models. In these state sums, every anyon vertex is equipped with a free label, just as for anyons in $2+1$ dimensions. Each such vertex is attached to a cellulation of the 2-sphere link. The simplest such cellulation consists of two 1-gon faces, glued together at their edges. A standard anyon edge representative is then attached to the cartesian product of this pillow triangulation with an edge,



$$\rightarrow \quad \rho(c)^\nu_\mu \; . \tag{6.189}$$

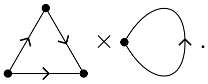
This picture looks similar $2+1$-dimensional anyons in Eq. (6.31), but the two looping edges here are now cellulated 2-spheres consisting of two 1-gon edges. As a consequence, the G-label on these looping edges must be trivial. The 3-volume to which this edge is attached can be cellulated with two 1-gon prism 3-cells.

In order to handle these line defects more conveniently, we consider the mapping given by the cartesian product with the (cellulated) 2-sphere,



(6.190)

The right-hand side shows the 1-skeleton of a 4-cellulation consisting of two copies of the same 4-cell (one orientation reversed). This 4-cell is the cartesian product of the triangle and one 1-gon face in the cellulation of the 2-sphere link,



(6.191)

Since the G-labels at the looping edges are constrained to 1, the compactified 2-dimensional state-sum is just an ordinary cohomology model, or a G-set model with trivial G-set. The deeper cause for this is that the sphere link of b has trivial 1-cohomology, in contrast to the circle link of anyons in $2+1$ dimensions. Normally, in order to evaluate the triangle weight of the compactified state sum, we would have to refine the cellulation on the right-hand side of Eq. (6.190) into a triangulation and take the product of the ω weights for all the 4-simplices. However, the cellulation consists of a 4-cell and its orientation-reversed copy, so we see that no matter how we would triangulate the 4-cell, the two weights always cancel. So the weight of the 2-dimensional ordinary cohomology model is trivial,

$$\beta(a,b)=1\;. \tag{6.192}$$

Anyons are in one-to-one correspondence with free boundaries of this compactified untwisted 2-dimensional cohomology model, which are just given by the (irreducible) representations of G. So the weight $\rho(c)_\mu^\nu$ defining an irreducible anyon is given by an irreducible representation of G, irrespective of the 4-cocycle ω defining the bulk model.

For example, for $G=\mathbb{Z}_l$, there are l different irreducible anyons given by

$$\rho_j(c)=e^{\frac{2\pi i}{l}jc}\;,\quad 0\leq j<l\;, \tag{6.193}$$

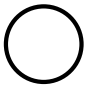
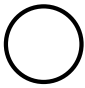
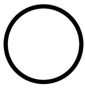
with trivial free labels. For $G=S_3$, there are 3 different anyons, given by

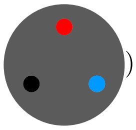
$$\begin{aligned}\rho_j(a,b)&=(-1)^{jb}\;,\quad 0\leq j<2\;,\\ \rho_2(a,b)_\mu^\nu&=\delta_{b+\mu,\nu}e^{\frac{2\pi i}{3}(-1)^\mu a}\;.\end{aligned} \tag{6.194}$$

like in Eq. (6.45).

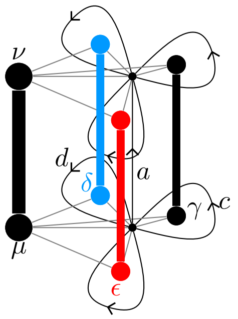
6.11 Membrane fusion worldlines in $3+1$D

Next, let us consider the fusion of three membrane defects inside a $3+1$-dimensional bulk. This fusion happens at $0+1$-dimensional defects, which we will accordingly call *membrane fusion worldlines*. The extended manifold type is the same as for anyon fusion events in $2+1$ dimensions in Eq. (6.135), just that every region is one dimension higher,

$a:(4,\varnothing)\;,\quad b:(2,$  $)\;,\quad c:(2,$  $)\;,\quad d:(2,$  $)\;,$

$e:(1,$  $)\;.$ (6.195)

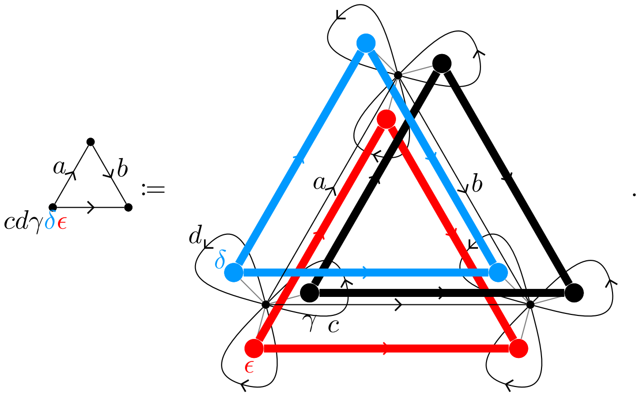
As in the previous sections, we go straight to cohomology models. As in Section 6.7, we can cellulate the link of e with three anyon vertices and one triangle. A standard membrane-fusion edge representative is given by the cartesian product of this link cellulation with an edge,



$$\rightarrow\quad \rho^{c,d,\gamma,\delta,\epsilon}(a)^{\nu}_{\mu}\;. \tag{6.196}$$

As shown, we have a weight associated to every membrane-fusion edge, and an additional free label associated to every membrane-fusion vertex. The dimension of these free labels is $B_{\mathrm{fw}}^{c,d,\gamma,\delta,\epsilon}$ depending on the values of the corresponding G and A^m-labels.

To classify membrane fusion worldlines, we consider the mapping from 2-dimensional triangulations by taking the cartesian product with the cellulated link of e.



. (6.197)

Assume the three membrane defects are robust and correspond to conjugacy classes C^m, C^m, and C^m, respectively. The resulting 2-dimensional state sum is a G-set state sum with G-set given by

$$\begin{gathered} A = \{c, d \in C^m \times C^m : cd \in C^m\} \times A^{m_b} \times A^m \times A^m \ , \\ (c, d, \gamma, \delta, \epsilon) \triangleleft g = (g^{-1}cg, g^{-1}dg, \gamma \triangleleft g, \delta \triangleleft g, \epsilon \triangleleft g) \ . \end{gathered} \tag{6.198}$$

The triangle weight can be read off Eq. (6.197) by triangulating the corresponding volume. The cartesian product of the triangle with the b, c, or d membrane-defect vertex yields a copy of the b, c, or d standard membrane-defect triangle in Eq. (6.181). The cartesian product of the triangle with the triangle can be triangulated with 6 4-simplices. Thus we get:

$$\begin{aligned} &\beta^{c,d,\gamma,\delta,\epsilon}(a, b) := \\ &\omega(a, b, (ab)^{-1}cab, (ab)^{-1}dab)\omega(a, a^{-1}ca, c^{-1}bc, (ab)^{-1}dab) \\ &\omega(a, a^{-1}ca, a^{-1}da, (cd)^{-1}bcd)\omega(c, c^{-1}ac, c^{-1}bc, (ab)^{-1}dab) \\ &\omega(c, c^{-1}ac, a^{-1}da, (cd)^{-1}bcd)\omega(c, d, (cd)^{-1}a(cd), (cd)^{-1}bcd) \\ &\qquad\qquad\qquad \eta^{c,\gamma}(a, b)\eta^{d,\delta}(a, b)\overline{\eta^{cd,\epsilon}(a, b)} \ . \end{aligned} \tag{6.199}$$

Membrane fusion worldlines are now in one-to-one correspondence with free boundaries of the 2-dimensional G-set cohomology model above. These boundaries can be constructed and classified following Section 5.1.3 as usual.

As a concrete example, let us consider $G = S_3$ with $\omega = 1$. Calculating fusion worldlines for all 8^3 triples of irreducible membranes would be too lengthy, so we consider one specific triple of membrane defects:

$$\begin{gathered} C^m = C_1 \ , H^m = \{(0, \alpha)\}_{0 \leq \alpha < 2} = H_1 \ , \eta = 1 \\ C^m = C_1 \ , H^m = \{(0, 0)\} \subset H_1 \ , \eta = 1 \\ C^m = C_2 \ , H^m = \{(\alpha, 0)\}_{0 \leq \alpha < 3} = H_2 \ , \eta = 1 \ . \end{gathered} \tag{6.200}$$

Let us now decompose the G-set in Eq. (6.198) into irreducible orbits. The action restricted to the first two components is transitive with trivial stabilizer, so every configuration of the remaining components generates one orbit with trivial stabilizer,

$$\begin{gathered} X_{\alpha,\beta,\gamma,\delta} = ((0,1),(1,1),L_\alpha,L_{(\beta,\gamma)},L_\delta) \;, \\ C_{\alpha,\beta,\gamma,\delta} = \{((0,1),(1,1),L_\alpha,L_{(\beta,\gamma)},L_\delta) \triangleleft (i,j)\}_{0\leq i<3, 0\leq j<2} \;, \\ H_{\alpha,\beta,\gamma,\delta} = \{(0,0)\} \;. \end{gathered} \tag{6.201}$$

Since $\omega = 1$ and $\eta = \eta = \eta = 1$, we have $\tilde{\beta} = \beta = 1$. Thus, every irreducible orbit defines precisely one membrane fusion worldline with $\rho = 1$. In total, there are 36 such fusion worldlines.

Let us briefly review approaches in the literature to classifying defects in $3+1$-dimensional models of topological order. Ref. [12] provides many examples and constructions of membrane and line defects, but it does not seem to provide a general classification of all such defects. The methods there are certainly very different from our approach. Ref. [38] discusses defects in $3+1$ dimensions via compactification and a higher-dimensional tube algebra. However, it seems like what is called “loop-like excitations” (in space) in that work corresponds to point singularities (line singularities in spacetime) in our systematic classification, whose link is a torus. In Ref. [54], the “loop-like excitations” from Ref. [38] are called “torus excitations”, which better fits our point of view. Ref. [141] also uses a higher-dimensional tube algebra to calculate “excitations” of general microscopic models of topological order in $3+1$ dimensions. Again, according to our formalism, the central irreducible idempotents of this tube algebra classify line-like singularities and not line or membrane-like defects in spacetime.

Chapter 7

Symmetries and fermions

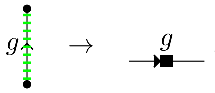
In all previous chapters, we have been describing topological phases of qudit systems. In this chapter, we discuss other “types of matter”. First, we will enrich the phases by imposing invariance under a global symmetry, which might be time-reversing. Second, we include fermionic degrees of freedom into the classification.

7.1 Symmetries

In this section, we add ordinary global symmetries to our classification. By a *symmetry* we mean a unitary representation of a finite group G acting on the Hilbert space of each degree of freedom, such that the global representation is the tensor product of the individual representations. In principle, fixed-point ansatzes with symmetry are the same as without. We just use *symmetric tensors* instead of ordinary tensors. Instead of associating only a bond dimension to each index, symmetric tensors are defined with respect to a unitary representation of G at each index. A symmetric tensor is then a tensor that is invariant under the overall unitary representation, that is, the tensor product of the representations at the individual indices. The graphical calculus for symmetric tensors is the same as for ordinary tensors, apart from one small difference: We need to distinguish between input and output indices, and the input indices are associated with the complex conjugated representation. Using symmetric tensors instead of ordinary tensors reproduces all known classifications of SPT, SET, and symmetry-breaking phases for unitary finite-group symmetries. Note that when defining phases via invertible domain walls, these need to be formed by symmetric tensors as well. This is why the classification is not only constrained, but also refined by adding symmetries.

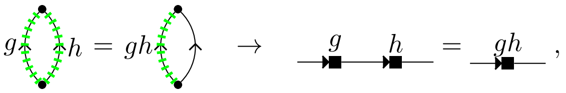
A perhaps more elegant way to incorporate symmetries is to stay with ordinary tensors but instead enrich the triangulations with a topological extra structure. This topological extra structure will also be necessary in the following sections where we add fermions or time-reversal symmetries. Furthermore, it makes the notion of *gauging* a symmetry particularly simple in our formalism. In the continuum, the topological extra structure is a 1st cohomology class in $H^1(M,G)$ with coefficients in the symmetry group G. Combinatorially, it is a $n-1$-cycle $\eta\in Z^1(M,G)$ on the triangulation or cellulation, c.f. 2.5. Physically, this $n-1$-cycle describes *symmetry defects* in the model.

Let us now describe how to build a tTS with symmetry defects. To be concrete, let us start with $1+1$ dimensions, and later sketch the straight-forward generalization to higher dimensions. We will define a variant of the 2D tTS with a homology class, the *2D G-cycle tTS*. Like the 2D tTS in Section 4, the 2D G-cycle tTS associates one tensor to every triangle of a triangulation, or 2-cell of a cellulation. In addition, we also associate a *homology tensor* to each edge e, whose variable depends on the $\eta(e)$,



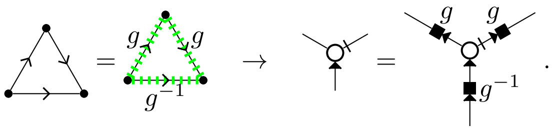
(7.1)

For edges $\eta(e) = g = 1$, we do not assign a tensor, or equivalently, the symmetry tensor is the identity matrix. The moves of the 2D G-cycle tTS contain the Pachner moves of the 2D tTS with trivial η. In addition, there are *homology moves* that locally change η without changing its homology class. By definition, such moves are generated by adding the boundary of a single 2-cell to η. It suffices to add two families of moves, namely



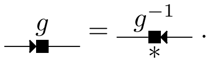
(7.2)

and



(7.3)

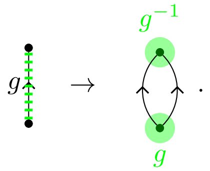
Hermiticity corresponds to a further move,



(7.4)

That is, when inverting the orientation, we also need to invert the G-elements. The first move implies that the collection of edge tensors for different g forms a linear representation of G. The second move implies that the triangle tensor is invariant under this representation. The third move implies that the representation is unitary. All in all we see that a 2D-G-cycle-tTS model is nothing but a 2D-tTS model together with an on-site unitary representation, under which it is invariant.

An alternative formulation of the 2D G-cycle tTS is to equip every 2-cell representative X with a 0-cycle $\omega[X]$, such that the cyclic product over the group elements on all vertices is the identity 1. Then we restrict ourselves to cellulations such that at each vertex x, the cyclic product of all the group elements that every adjacent 2-cell associates to x is 1. With this, we can stick to our design paradigm that tensor variables correspond to different 2-cell representatives. The tensor variable at an edge is now associated with a 2-gon with the following 0-cycle,



(7.5)

Let us give some examples of 2D-G-cycle-tTS models. The first family of examples is defined for any subgroup $H \subset G$. It consists of the δ-tensor 2D-tTS model in Eq. (4.4) with label set $H \backslash G$, together with the homology tensor

$$Hx \overset{g}{\rightarrow\!\!\blacksquare\!\!-\!\!-} Hy = \delta_{Hxg,Hy} \ . \tag{7.6}$$

These models are fixed-point models for *symmetry-breaking phases*, where the symmetry group G is spontaneously broken down to a subgroup H.

The next family of examples is defined for a *projective representation* ρ of G, that is, a matrix $\rho(g)$ for every $g \in G$ such that

$$\sum_b \rho(g)_{ab}\rho(h)_{bc} = \omega(g,h)\rho(gh)_{ac} \ , \tag{7.7}$$

for some $\omega(g,h) \in U(1)$. It consists of the matrix-algebra model in Eq. (4.15), together with

$$aa' \overset{g}{\rightarrow\!\!\blacksquare\!\!-\!\!-} bb' = \rho(g)_{ab}\overline{\rho(g)_{a'b'}} \ . \tag{7.8}$$

The simplest non-trivial concrete example is for $G = \mathbb{Z}_2 \times \mathbb{Z}_2$, where a projective representation is given by

$$\rho((g,g'))_{ab} = \delta_{a+g',b}(-1)^{ag} \ , \tag{7.9}$$

with

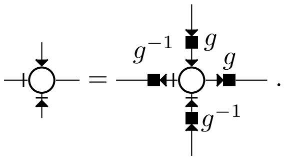
$$\omega((g,g'),(h,h')) = (-1)^{g'h} \ . \tag{7.10}$$

The homology tensor is thus given by

$$aa' \overset{(g,g')}{\rightarrow\!\!\blacksquare\!\!-\!\!-} bb' = \delta_{a+g',b}(-1)^{ag}\delta_{a'+g',b'}(-1)^{a'g} \ . \tag{7.11}$$

These models are fixed-point models for *symmetry-protected topological* (SPT) phases in $1+1$ dimensions. They are a spacetime version of the isometric MPS in Ref. [127] and of the "dimer crystal" models in Ref. [41].

The generalization to other dimensions, boundaries, or other defects is straightforward. In $2+1$ dimensions, η is a 2-cycle, that is, it associates a G-element to every triangle. There is a 2-index homology tensor associated to every triangle t whose tensor variable depends on $\eta(t)$ and is the identity if $\eta(t) = 1$. The homology tensors have to fulfill the same moves as in Eqs. (7.2), (7.3), and (7.4). So they form a unitary representation of G, under which the tetrahedron tensor is invariant,

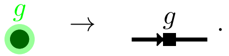


(7.12)

For boundaries in n spacetime dimensions, we need to decorate the extended cellulations with a $n-2$-cycle η_∂ inside the boundary $n-1$-cellulation, in addition to the $n-1$-cycle η in the bulk cellulation, such that

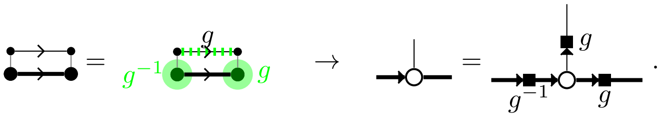
$$\delta\eta = \eta_\partial \ , \tag{7.13}$$

where the right-hand side means the $n-2$-chain consisting of the bulk $n-2$-cells to which the boundary $n-2$-cell is attached. In other words, η needs to fulfill the constraint that for every bulk $n-2$-cell, the product of η on the attached bulk $n-1$-cells and η_∂ on the attached boundary $n-2$-cells is the identity. Let us again consider the case $n=2$. Then, for a bulk vertex adjacent to a boundary vertex, η_∂ on this boundary vertex also contributes to the cycle constraint. Additionally to the bulk edges, we also associate a 2-index homology tensor to each boundary vertex v, whose variable depends on $\eta_\partial(v)$,



(7.14)

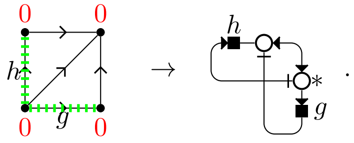
Again, this tensor fulfills the moves as in Eqs. (7.2), (7.3), and (7.4). The boundary edge tensor is invariant under adding homology tensors at all indices,



(7.15)

In general, for any tTS with defects, we introduce one unitary G-representation for every $d_r - 1$-cell representative of every region r, and demand that all tensors are invariant under the respective representations.

The fact that the tensor-network path integral is defined on manifolds enriched with a 1-homology class also leads to more powerful invariants. For example, for $n=2$, for every pair $g, h \in G$ with $gh = hg$, we can evaluate the path integral on a torus with a non-contractible g loop in one direction and a non-contractible h loop in the other direction,



(7.16)

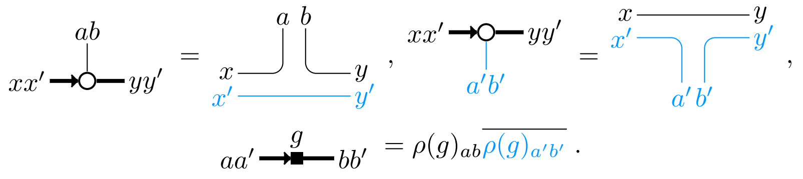
If we evaluate this path integral for the model in Eq. (7.8), we obtain

$$\omega(g,h)\overline{\omega(h,g)} \ . \tag{7.17}$$

For example, for the concrete $\mathbb{Z}_2 \times \mathbb{Z}_2$ projective representation in Eq. (7.9), we have

$$\omega((g,g'),(h,h'))\overline{\omega((h,h'),(g,g'))} = (-1)^{g'h+gh'} \;, \tag{7.18}$$

which is non-trivial. This suggests that this 2D-G-cycle-tTS model is in a non-trivial phase. Indeed, the invertible boundary for the matrix-algebra model in Eq. (4.164) can *not* be equipped with a unitary representation on its boundary bond dimension to yield a 2D G-cycle-boundary tTS model for the 2D-G-cycle-tTS model in Eq. (7.8). This is because the boundary bond dimension in Eq. (4.164) consists of a single index that is invariant under the projective representation, but not a linear representation. Vice versa, two models as in Eq. (7.9) are in the same phase if the projective representations have the same ω. An according invertible domain wall is defined by



(7.19)

Note that the invertibility property of the model itself in Section 5.5 is unaffected by the existence of a symmetry. Thus, the model in Eq. (7.8) still represents an invertible phase, even though it is non-trivial.

7.2 Time-reversal symmetries

Next, let us consider models that have a time-reversal symmetry. As we have argued in Section 4.3, in a topological Euclidean-signature spacetime, reversing time is as good as reversing any direction. So reversing time is the same as changing the orientation. In order to discuss time-reversal symmetries, we thus need to properly define an orientation: An orientation is a $\mathbb{Z}_2$-valued n-chain σ such that $\delta\sigma = \omega_1$, where ω_1 is the first Stiefel-Whitney class as defined in Section 2.7. The n-cells with $\sigma = 0$ are positively oriented, and n-cells with $\sigma = 1$ are negatively oriented and correspond to a complex conjugated copy of the corresponding tensor variable.

In general, time-reversal is part of a larger symmetry group G. There is a homomorphism

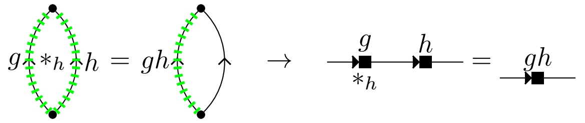
$$s : G \rightarrow \mathbb{Z}_2 \;, \tag{7.20}$$

which assigns to every G-element whether it is time reversing or not. Just like the G-cycle tTS in Section 7.1, a tTS with a time-reversal symmetry associates tensor networks to triangulations/cellulation equipped with a G-valued $n-1$-cycle η. The only difference is that now, instead of an orientation, we define an n-chain σ such that

$$\delta\sigma = \omega_1 + s(\eta) \;. \tag{7.21}$$

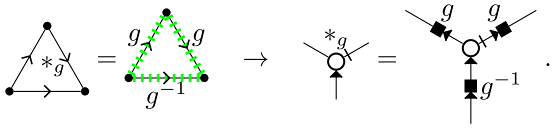
We still associate a complex conjugate copy of the corresponding tensor variable to all n-cells where $\sigma = 1$.

Let us again start with the simple example of $1+1$ dimensions, where we define the following *2D G, s-cycle tTS*. Just like the 2D G-cycle tTS, the 2D G, s-cycle tTS associates a *homology tensor* to every edge e, depending on $\eta(e)$ as shown in Eq. (7.1). Further, we need to add moves that add the boundary of a 2-cell to η, valued with some group element $g \in G$. However, in order to maintain the constraint in Eq. (7.21), we have to at the same time switch the value of σ on the 2-cell itself if g is time reversing. The moves in Eqs. (7.2) and (7.3) thus become



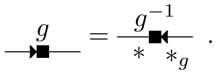
(7.22)

and



(7.23)

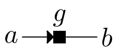
Here, the marking $*_g$ inside a 2-cell means that σ for this 2-cell is different from the orientation of this 2-cell that is implicit by the drawing. $*_g$ next to a tensor denotes complex conjugation if $s(g)=1$. The Hermiticity condition in Eq. (7.4) becomes



(7.24)

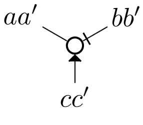
Eq. (7.22) is equivalent to imposing that $K^{s(g)}$ followed by the homology tensor defines a (mixed linear/anti-linear) group representation, where K denotes complex conjugation. Eq. (7.23) states that the triangle tensor is invariant under this representation. Eq. (7.24) states that all homology tensors are unitaries (and consequently anti-unitaries when composed with K if $s(g)=1$).

Let us consider two simple examples of 2D-G, s-cycle-tTS models with $G=\mathbb{Z}_2$ and $s=\mathrm{id}_{\mathbb{Z}_2}$. That is, we have a symmetry under a single time-reversing operation that squares to the identity. The first example is given by the δ-tensor model in Eq. (4.4) with homology tensor

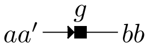


$$= \delta_{a+g,b} \ . \tag{7.25}$$

In other words, the homology tensor for the time-reversing element $g=1$ is given by a Pauli-X matrix. Since Eq. (4.4) is invariant under conjugating with Pauli-X matrices as an ordinary symmetry, and all involved tensors are real, this defines a valid model. This model can be considered a time-reversing variant of the Ising symmetry-breaking phase. The second example is given by a model with label set $\{0,1\} \times \{0,1\}$,

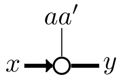


$$= \frac{1}{2}\delta_{a+b,c}\delta_{a'+b',c'}(-1)^{ab+a'b'+a'b} \ ,$$



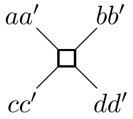
$$= \delta_{a,b}\delta_{a',b'} \ . \tag{7.26}$$

That is, the symmetry is given by complex conjugation alone. Note that the algebra defined by this model according to Section 4.12 is the *quaternion algebra*. As a complex algebra, the quaternion algebra is isomorphic to the 2×2 matrix algebra. Accordingly, without the symmetry, the model has an invertible boundary given by



$$= \delta_{x+a',y}(-1)^{ax} i^{a+a'} \ . \qquad (7.27)$$

However, the symmetry restricts us to real-valued models, and the quaternion algebra as a real algebra is not isomorphic to the 2×2 matrix algebra. This suggests that this model is indeed in a non-trivial phase. Let us briefly consider the commuting-projector Hamiltonian for this model, via Eq. (4.41),



$$\begin{aligned} &= \frac{1}{4}\delta_{a+b,c+d}\delta_{a'+b',c'+c'}(-1)^{(c+c'+b)(a+c)+(c'+b'+b)(a'+c')} \\ &= \frac{1}{4}(\mathbb{1} - XZ \otimes Z \otimes XZ \otimes \mathbb{1} - \mathbb{1} \otimes XZ \otimes Z \otimes XZ - XZ \otimes X \otimes X \otimes XZ) \\ &= \frac{1}{4}(\mathbb{1} - XZ \otimes Z \otimes XZ \otimes \mathbb{1})(\mathbb{1} - \mathbb{1} \otimes XZ \otimes Z \otimes XZ) \ . \end{aligned} \qquad (7.28)$$

In the second and third line, we denote the tensor as an operator acting on four qubits whose configuration corresponds to the value of (c, c', d, d'). Each summand in the second line corresponds to a fixed value of $a + c$ and $a' + c'$. The expression above shows that one reinterpret the model as a chain of qubits with range-3 Hamiltonian terms, instead of 4-dimensional qudits with range-2 terms. The 3-qubit projector is

$$P = \frac{1}{2}(\mathbb{1} - XZ \otimes Z \otimes XZ) \ , \qquad (7.29)$$

and the Hamiltonian is

$$H = \sum_i (XZ)_{i-1} Z_i (XZ)_{i+1} \ , \qquad (7.30)$$

which is invariant under a time-reversal symmetry that is just complex conjugation K. After conjugating with the Clifford matrix S, the Hamiltonian becomes

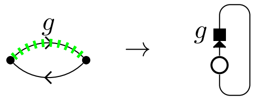
$$H = \sum_i -X_{i-1} Z_i X_{i+1} \ , \qquad (7.31)$$

and the anti-unitary time-reversal symmetry is given by

$$T = K \bigotimes_i Z \ . \qquad (7.32)$$

This is a known model for a time-reversal SPT phase that is referred to as *cluster Hamiltonian* in Ref. [132].

Note that due to the interaction of η with the orientation, we can put the path integral also on *non-orientable* manifolds. For example, in $n=2$, for any $g\in G$ with $s(g)=1$ can put it on the real projective plane,



, (7.33)

where on the left-hand side the bottom and top edge are identified. If we compute the above invariant for the model in Eq. (7.25), we get 0. For the model in Eq. (7.26), we get -2. This again suggests that these two models are indeed in non-trivial phases.

7.3 Fermions

In this section we will consider models with fermionic degrees of freedom. Similar to the two previous sections, we do this by equipping the topological manifolds with some homological extra structure. However, this alone is not enough. We also need to account for the non-trivial exchange statistics of fermions. These exchange statistics are of a global nature, and they require the use of another *tensor type* [22], that is, another consistent interpretation of the tensor-network diagrammatic calculus. A mathematical definition of such *fermionic tensors* suitable for our needs can be found in the preliminaries in Section 2.2.

Let us start by motivating the use of fermionic tensors from the perspective of a condensed matter theorist. We consider a Hilbert space consisting of n fermionic modes labeled $0,\ldots,n-1$, with creation and annihilation operators c_i and $c_i^\dagger$ for $0\leq i<n$ fulfilling the familiar commutation relations,

$$c_ic_j=-c_jc_i\ ,\quad c_i^\dagger c_j^\dagger=-c_j^\dagger c_i^\dagger\ ,\quad c_ic_j^\dagger=-c_j^\dagger c_i+\delta_{i,j}\mathbb{1}\ . \tag{7.34}$$

The spacetime tensors in our formalism correspond to operators acting on the Hilbert space, in particular to the imaginary time evolution $e^{-\beta H}$. Any operator A acting on this Hilbert space can be expanded in terms of creation and annihilation operators as

$$\begin{aligned}&\sum_{\substack{s_0,\ldots s_{n-1}\\ s'_0\ldots,s'_{n-1}}} A_{s'_0,\ldots,s'_{n-1},s_{n-1},\ldots,s_0} \left|s_0\right\rangle_0\cdots\left|x_{n-1}\right\rangle_{n-1}\left\langle s_{n-1}\right|_{n-1}\cdots\left\langle s_0\right|_0\\ &\qquad=\sum_{\substack{s_0,\ldots s_{n-1}\\ s'_0\ldots,s'_{n-1}}} A_{s'_0,\ldots,s'_{n-1},s_{n-1},\ldots,s_0}\\ &\qquad(c_0^\dagger)^{s_0}\left|0\right\rangle_0\cdots(c_{n-1}^\dagger)^{s_{n-1}}\left|0\right\rangle_{n-1}\left\langle 0\right|_{n-1}(c_{n-1})^{s'_{n-1}}\cdots\left\langle 0\right|_0(c_0)^{s'_0}\ ,\end{aligned} \tag{7.35}$$

where the s_i and s'_i are either 0 or 1 depending on whether the corresponding mode is occupied or not. Any physical operator must preserve fermion parity, so we have

$$\sum_i s_i+\sum_i s'_i\neq 0\mod 2\quad\Rightarrow\quad A_{s_0,\ldots,s_{n-1},s'_{n-1},\ldots,s'_0}=0\ . \tag{7.36}$$

Further, we have ordered the creation and annihilation operators in the same way as the sub and superscripts of A. If we want to exchange two sub or superscripts, for example s_0 and s_1 above, we need to multiply by a sign, for example

$$\tau^{01}(A')_{s_1,s_0,s_2,\ldots} = (-1)^{s_0 s_1} A_{s_0,s_1,s_2,\ldots} \; . \tag{7.37}$$

This is due to the anti-commutation relations in Eq. (7.34). So we see that A is a fermionic tensor with input indices s'_i and output indices s_i, where each index has bond dimensions $d^+ = d^- = 1$ so that $|s_i| = s_i$.

If the fermionic tensor A represents a fermionic Hamiltonian, then the operator in Eq. (7.35) needs to be Hermitian. This is the case if complex conjugating A is the same as swapping input and output indices of A as an ordinary tensor,

$$A_{s_0,\ldots,s_{n-1},s'_{n-1},\ldots,s'_0} = \overline{A_{s'_0,\ldots,s'_{n-1},s_{n-1},\ldots,s_0}} \; . \tag{7.38}$$

However, if we interpret A as a fermionic tensor, the above equation involves applying fermionic index transposition to arrive at the same index ordering on both sides. We notice that the index ordering on the left and right-hand side is exactly reversed. Combined with the complex conjugation, this is the same as a fermionic conjugation, as introduced in Section 2.2. So if we interpret A as a fermionic tensor, then Eq. (7.38) becomes

$$A_{s_0,\ldots,s_{n-1},s'_{n-1},\ldots,s'_0} = A^*_{s'_0,\ldots,s'_{n-1},s_{n-1},\ldots,s_0} \; . \tag{7.39}$$

In order to go from fermionic operators to fermionic tensors, we generalize two aspects of the latter. First, we allow tensors with different numbers of input and output indices. Second, we need to go from the condensed-matter notion of a fermionic mode to the quantum-information notion of a fermionic degree of freedom: Instead of merely having modes that can either be occupied or unoccupied by a fermion, we allow for additional degeneracies of the occupied and unoccupied state. That is, a fermionic degree of freedom can have d^+ different unoccupied configurations, and d^- occupied configurations.

Note that we are not demanding fermion number conservation, so we only care about the parity of the fermionic charge. When combining two fermionic degrees of freedom into a single one, configurations of this degree of freedom are pairs of original configurations. The parity of these pairs is just the mod 2 sum of the individual parities.

Having motivated the use of fermionic tensors, let us now discuss the homological extra structure that we need to equip our cellulations with. Fermionic tensors are constrained by a fermion-parity symmetry, which is similar to a $\mathbb{Z}_2$ symmetry, so we expect the extra structure to be some $\mathbb{Z}_2$-valued $n-1$-chain η. However, in contrast to an ordinary $\mathbb{Z}_2$ symmetry, η is not a cycle, but its boundary is given by

$$\delta\eta = \omega_2 \; , \tag{7.40}$$

where ω_2 is the second Stiefel-Whitney class, which we have introduced in Section 2.7. η is then a cellular representative of a *spin structure*. The fact that fermionic fixed-point models secretly carry a combinatorial representation of a spin structure was first pointed out in Ref. [70]. Equivalence classes of choices of η up to homology are in one-to-one correspondence with equivalence classes of spin structures on the manifold. If the manifold is *spinnable*, then there are as many equivalence classes as there are $n-1$st $\mathbb{Z}_2$ homology classes of the manifold, otherwise, there are none. In the former case, there is an action of the $n-1$st homology group on the set of spin structures, but no canonical identification between the two.

Like the ordinary nD tTS, the *nD spin tTS* associates a tensor to every n-cell. The tensor-network diagrams will be interpreted as fermionic tensor networks, but this does not matter for the purely diagrammatic definition of the tTS. The only difference is that we have to choose for each index whether it is an input or output index. This can be done using an orientation: Positively oriented $n-1$-cells of a n-cell correspond to input indices, and negatively oriented ones to output indices. Due to Hermiticity, negatively oriented n-cells are associated a fermionically conjugate copy of the corresponding tensor variable. In addition to the n-cells, there is also a 2-index tensor at every $n-1$-cell with $\eta = 1$. However, this 2-index tensor is not arbitrary, but is fixed to the fermion parity tensor. This can be regarded as a lattice analogue of the *spin-statistics relation*: η represents a *spin* structure, whereas the fermion parity of the indices determines the reordering sign that implements the exchange *statistics* of the fermions.

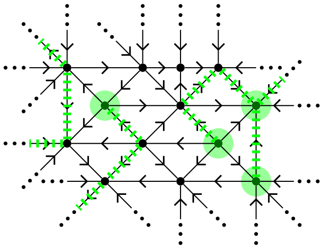
The spin structure affects the moves of the nD tTS in two different ways: First, like in Section 7.1 and 7.2, we need to add *homology moves* that add the boundary of an n-cell to η. However, due to the spin-statistics relation, these moves hold automatically due to the fermion parity constraint in Eq. (2.12). One might vice versa try to argue analogous to Section 7.1 that the definition of the fermion parity and the fermion parity constraint follow automatically from the spin tTS. However, we need the fermion parity to define fermionic tensors in the first place. Second, in contrast to symmetries in Section 7.1 and 7.2, η affects the Pachner or recellulation moves of the tTS nontrivially. This is because ω_2 might be different for the cellulation on the right and left, and we have to include an $n-1$-chain η whose boundary is the difference in ω_2.

7.4 Fermions in $1+1$D

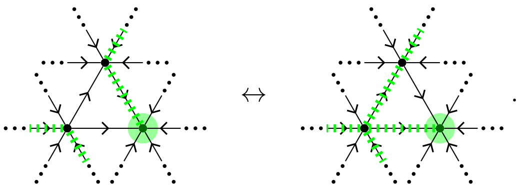
In this section, we will discuss the 2D spin tTS more concretely. Following Section 2.7, in $1+1$ dimensions, the value of ω_2 on a vertex v in a branching-structure triangulation is given by

$$\omega_2(v) = 1 + \sum_{e\in S_1[v]} \delta_{v,e_0} + \sum_{t\in S_2[v]} \delta_{v,t_0} \pmod 2 , \tag{7.41}$$

where e_0 and t_0 denote the 0 vertex of the edge or triangle e or t. As an example, consider the following patch of triangulation with ω_2 (which is fixed), and a choice of η,

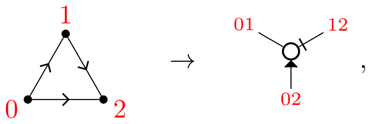
 . (7.42)

Here, if an edge or vertex is marked, this means $\eta = 1$ or $\omega_2 = 1$, otherwise, we have $\eta = 0$ or $\omega_2 = 0$. Two spin structures are equivalent if they differ by local homology moves such as



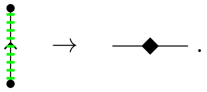
(7.43)

Like in the previous sections, the 2D spin tTS assigns one tensor to every triangle or other 2-cell. One difference is that, since the diagrams are interpreted via fermionic tensors, we need to choose for every index whether it is an input or output index. We choose every index corresponding to a clockwise edge to be an output index, and the others to be input indices, so in



(7.44)

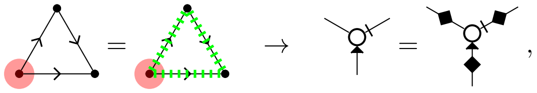
01 and 12 are output indices, whereas 02 is an input index. Note that the input indices are also the ones carrying an arrow marking. In addition, we associate a 2-index tensor to every $\eta = 1$ edge. This 2-index tensor is fixed to the fermion parity tensor by the spin-statistics relation,



(7.45)

Here, the output index is on the left, and the input index on the right.

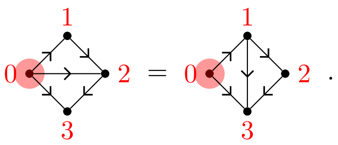
Let us now discuss the moves of the tTS. The homology move for the triangle,



(7.46)

is the same as the fermion parity constraint for the triangle tensor. The red marked points on the boundary represent the value of $\omega_2^I + d\eta$ on the boundary, where ω_2^I denotes the contributions to ω_2 from edges and faces in the interior. For a valid move, these marked points have to agree on the left and right-hand side, which is trivially the case here since the cellulation does not change.

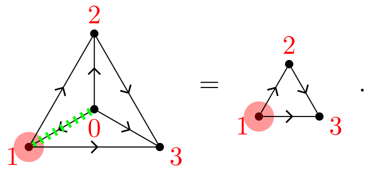
Next, we need to add the spin structure to the Pachner moves. For the standard 2-2 Pachner move in Eq. (4.11), we obtain



(7.47)

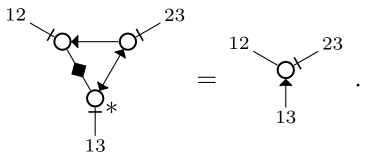
The value of ω_2^I is the same for all vertices on the left and right-hand side. Therefore, we do not need to introduce any $\eta = 1$ edges, and diagrammatically, the move looks identical to the ordinary 2-2 Pachner move in Eq. (4.12).

This is *not* the case for all Pachner moves. For example, consider the 3-1 Pachner move in Eq. (4.13),



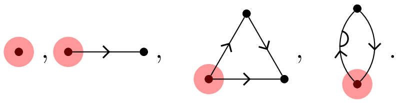
(7.48)

Here, we have $\omega_2^I(1) = 0$ on the left-hand side but $\omega_2^I(1) = 1$ on the right-hand side. In addition, we have $\omega_2^I(0) = 1$ on the left-hand side. To make $d\eta + \omega_2^I$ trivial on the interior vertex 0 on the left-hand side, and agree on the boundary on both sides, we introduce η such that $d\eta = 0 + 1$. The simplest such η is given by $\eta = 01$ as shown. In the corresponding tensor-network diagram, we have to insert a fermion parity operator for this 01 edge,



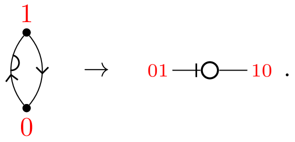
(7.49)

Let us now discuss how to the spin structure enters in all the moves of the simplified 2D tTS presented in Section 4.7. To this end, we need to define spin structures on arbitrary 2-cellulations. This can be done by choosing a "favorite" vertex of every 0-cell, 1-cell, and 2-cell. We have already chosen the favorite vertex of the vertex, edge and triangle to be the 0 vertex. In addition, we have to choose a favorite vertex for the cyclic 2-gon of the simplified 2D tTS in Eq. (4.89),



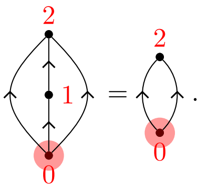
(7.50)

As we see, in order to determine the favorite vertex of the cyclic 2-gon, we need to break its rotation symmetry by marking the edge whose 0 vertex is the favorite vertex. Therefore, we need to distinguish the indices of the associated tensor variable by marking one with a tick,



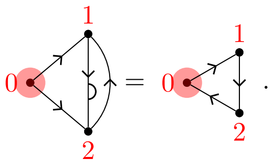
(7.51)

With this, we are ready to look at the moves of the simplified 2D tTS. The triangle cancellation move in Eq. (4.86) becomes,



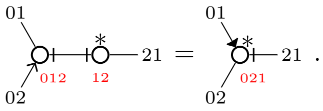
(7.52)

The tensor on the right-hand side is the fermionic identity matrix by the full support convention, associated to the acyclic 2-gon whose favorite vertex is the 0 vertex. As we can see, ω_2^I is the same on both sides, so diagrammatically the move looks identical to Eq. (4.88). The only difference is the interpretation of the diagram in terms of fermionic tensors, and the $*$ as a fermionic conjugation. Next, the 12 triangle flip move in Eq. (4.95) becomes



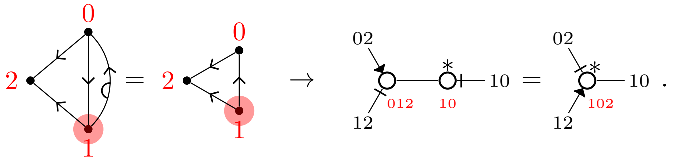
(7.53)

Again, $d\eta + \omega_2^I$ is already the same on both sides. Note that this would not be the case if we had chosen the favorite edge of the cyclic 2-gon differently. Diagrammatically, the only difference to Eq. (4.95) is that the two indices of the 2-gon are distinguished,



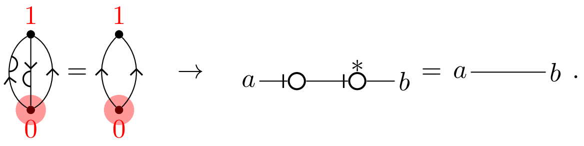
(7.54)

The 01 triangle flip move in Eq. (5.12) becomes



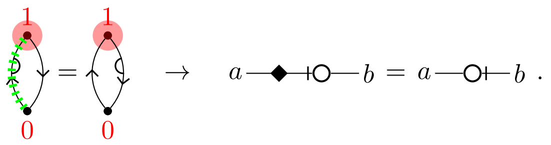
(7.55)

The 2-gon cancellation move in Eq. (4.93) becomes



(7.56)

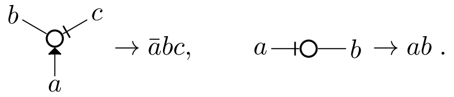
In addition to equipping all moves of the simplified 2D tTS with a spin structure, we also need one additional move that is specific to the 2D spin tTS,



(7.57)

Here we have $\omega_2^I = 0$ on the left but $\omega_2^I = 1$ on the right, so we need to set $\eta = 01$ such that $\omega_2^I + d\eta$ agrees on both sides.

For finding models and classification, it is instructive to express all the moves of the 2D spin tTS as ordinary tensor-network equations, together with sign factors coming from reordering of the fermionic degrees of freedom. To this end, we choose a fixed index ordering for each tensor, which is not necessary in the non-fermionic case. A choice of orderings that turns out to be particularly convenient is



(7.58)

We now determine the reordering signs for each tensor network equation using the methods from Section 2.2. Thereby, we manipulate expressions of the form $S_l|S_r|P$, where S_l is the index ordering on the left-hand side of the equation, S_r is the index ordering on the right-hand side, and P is the collected reordering sign from both sides together. We also allow removing an index label from the ordering that is at the beginning or at the end of both S_l and S_r. For the 2-2 Pachner move in Eq. (7.47) we get the following,

$$\begin{gathered}\bar{x}ab\bar{d}xc|\bar{d}ay\bar{y}bc|0 \to x\bar{x}ab\bar{d}|\bar{d}ab|dx \to ab\bar{d}|ab\bar{d}|dx + d(a+b)\\ \to ||d(x+b+b) \to ||0\ .\end{gathered} \tag{7.59}$$

That is, we find that all the reordering signs on the left and right cancel. Next, for Eq. (4.88), we get

$$\bar{b}xy\overline{\bar{a}xy}|\bar{b}a|0 \to \bar{b}xy\bar{y}\bar{x}a|\bar{b}a|0 \to \bar{b}x\bar{x}|\bar{b}|0 \to ||0\ . \tag{7.60}$$

Eq. (7.54) yields

$$\bar{a}bx\overline{xc}|\overline{\bar{b}ac}|0 \to \bar{a}bc\bar{x}\bar{c}|\bar{c}\bar{a}b|c \to \bar{a}b\bar{c}|\bar{c}\bar{a}b|c \to \bar{c}\bar{a}b|\bar{c}\bar{a}b|0 \to ||0\ . \tag{7.61}$$

Eq. (7.55) yields

$$\bar{a}xb\overline{cx}|\overline{\bar{b}ca}|0 \to \bar{a}xb\bar{x}\bar{c}|\bar{a}\bar{c}b|0 \to x\bar{x}b\bar{c}|\bar{c}b|c \to b\bar{c}|\bar{c}b|c \to ||0\ . \tag{7.62}$$

Eq. (7.56) yields

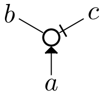
$$ax\overline{xb}|\bar{b}a|0 \to ax\bar{b}\bar{x}|a\bar{b}|a \to x\bar{x}\bar{b}|\bar{b}|0 \to ||0\ . \tag{7.63}$$

Finally, Eq. (7.57) yields

$$ab|ba|a \to ||0\ . \tag{7.64}$$

Thus, we observe that all reordering signs of the 2D spin tTS cancel. Note that this would not have happened if we had not introduced a spin structure. Also note that this is not a general feature of spin tTS: First of all, we would have obtained non-trivial reordering signs, if we had chosen a different index ordering in Eq. (7.58). Second, we will find that in higher dimensions an index ordering for which all reordering signs vanish does not exist.

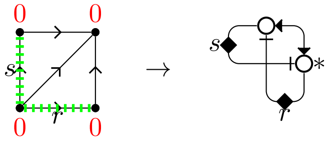
The vanishing of the reordering signs implies that 2D-spin-tTS models are in one-to-one correspondence 2D-tTS models that obey a $\mathbb{Z}_2$-grading. That is, they are one-to-one with 2D-$\mathbb{Z}_2$-cycle-tTS models, where the symmetry representation of $1 \in \mathbb{Z}_2$ is given by the fermion parity matrix. With this, it is easy to give an example: We can take the 2D-tTS model corresponding to the $\mathbb{Z}_2$ group algebra with the obvious parities $|0| = 0$ and $|1| = 1$. With respect to the index ordering in Eq. 7.58, the model is given by



$$= \frac{1}{\sqrt{2}} \cdot \delta_{a+b,c} \ , \tag{7.65}$$

where a, b and c are understood as elements of $\mathbb{Z}_2$.

The fact that the 2D spin tTS depends on a combinatorial spin structure means that there is one invariant corresponding to every spin manifold. Note that in $1+1$ dimensions, every orientable manifold admits a spin structure, and there are as many equivalence classes of spin structures as elements of the 1st $\mathbb{Z}_2$ homology group. For example, we can consider the following spin torus,



$$, \tag{7.66}$$

where $r, s \in \{0, 1\}$ denote the value of η at an edge on the left-hand side, and whether the fermion parity matrix is present or not on the right-hand side. Physically, the path integral on a torus corresponds to the ground-state degeneracy of the model on a circle. However, for a fermionic model, there are four such ground state degeneracies: First of all, the spin structure on the circle can be bounding ($s = 1$) or non-bounding ($s = 0$). Second, each ground state degeneracy has an even part d^+ and and odd part d^-. The $r = 1$ invariant yields $d^+ + d^-$, and the $r = 0$ invariant yields $d^+ - d^-$. Due to symmetry reasons, we can also swap the interpretation of s and r. The reordering signs occurring in this fermionic tensor network is

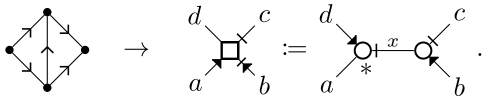
$$\begin{aligned}&\bar{z}yx\overline{\bar{z}xy}|rx + sy \to \bar{z}yx\bar{y}\bar{x}z|rx + sy \to \bar{z}y\bar{y}x\bar{x}z|rx + sy + xy\\&\quad \to \bar{z}z|rx + sy + xy \to |rx + sy + xy + z \to |(r+1)x + (s+1)y + xy \ .\end{aligned} \tag{7.67}$$

So we see that even though all reordering signs cancel for recellulations or Pachner moves, they are non-trivial for the computation of invariants. If we evaluate this invariant for the model given in Eq. (7.65), we thus get

$$\frac{1}{2}\sum_{x,y}(-1)^{(r+1)x+(s+1)y+xy} = (-1)^{(r+1)(s+1)} \ . \tag{7.68}$$

So we see that the circle has a unique ground state, which is even if the spin structure is bounding ($s = 1$), and odd if it is non-bounding ($s = 0$).

Next, let us look at the commuting-projector mapping from Section 4.4,



(7.69)

For the reordering sign, we find

$$\begin{aligned}&\bar a\bar b c d|\overline{\bar d a x}\bar b x c|0 \to \bar a d\bar b c|\bar x\bar a d\bar b x c|d(b+c)\\&\quad\to \bar a d\bar b c|x\bar x\bar a d\bar b c|d(b+c)+bx \to ||cd+b(d+x) \to ||ab+cd\ .\end{aligned} \tag{7.70}$$

Thus, evaluating this for the model in Eq. (7.65), we get a fermionic tensor

$$A_{abcd} = \frac{1}{2}\delta_{a+b,c+d}(-1)^{ab+cd}\ . \tag{7.71}$$

Via Eq. (7.35), this yields an operator

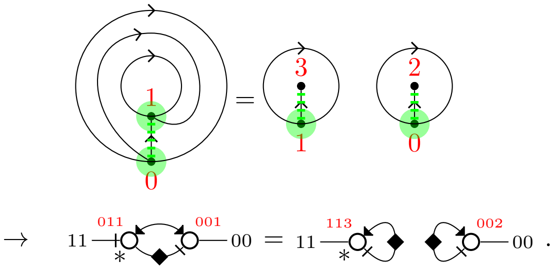
$$\begin{aligned}P = \frac{1}{2}\sum_{a,b,c,d}&\delta_{a+b,c+d}(-1)^{ab+cd}(c_0^\dagger)^d(c_1^\dagger)^c\,|00\rangle\,\langle 00|\,c_1^b c_0^a\\&= \frac{1}{2}(\mathbb{1} - c_1c_0 - c_0^\dagger c_1^\dagger + c_1^\dagger c_0 + c_0^\dagger c_1)\\&= \frac{1}{2}(\mathbb{1} + c_0c_1 - c_0^\dagger c_1^\dagger - c_0 c_1^\dagger + c_0^\dagger c_1)\\&\quad= \frac{1}{2}(\mathbb{1} + (c_0+c_0^\dagger)(c_1-c_1^\dagger))\ .\end{aligned} \tag{7.72}$$

Up to rescaling and shift, this yields the Hamiltonian,

$$H = -\sum_i (c_i + c_i^\dagger)(c_{i+1} - c_{i+1}^\dagger)\ . \tag{7.73}$$

This Hamiltonian is the fixed-point of the *Kitaev chain* [94], a model for a non-trivial invertible fermionic phase in $1+1$ dimensions.

Next, let us look at the fermionic version of the invertibility property of models from Section 5.5. Equipped with spin structure, Eq. (4.73) becomes



(7.74)

The reordering sign for this move is

$$\begin{aligned}\overline{\bar{x}ya}\bar{x}by|\overline{\bar{w}aw}\bar{z}bz|y+w+z &\rightarrow \bar{a}\bar{y}x\bar{x}by|\bar{w}\bar{a}w\bar{z}bz|y+w+z\\ &\rightarrow \bar{a}\bar{y}by|w\bar{w}\bar{a}z\bar{z}b|y \rightarrow \bar{a}y\bar{y}b|\bar{a}b|yb \rightarrow ||yb\;.\end{aligned} \tag{7.75}$$

So again the invertibility move is affected by a reordering sign, even though Pachner moves or recellulations are not.

Let us test invertibility for the model in Eq. (7.65). We get

$$\frac{1}{2}\sum_y \delta_{a,b}(-1)^{yb} = \frac{1}{2}\delta_{a,0}\delta_{b,0}\;. \tag{7.76}$$

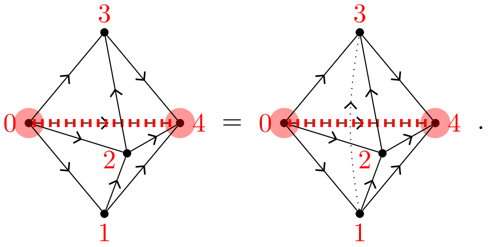
We see that this equation holds up to a prefactor, and thus the model is invertible Note that this is in contrast to the same model interpreted as a 2D-$\mathbb{Z}_2$-cycle-tTS model.

7.5 Fermions in $2+1$D

In this section, we will briefly discuss the 3D spin tTS. The value of ω_2 on an edge e in a branching-structure triangulation is given by

$$\omega_2(e) = 1 + \sum_{t\in S_2(e)} \delta_{e,t_{02}} + \sum_{t\in S_3(e)} \delta_{e,t_{03}}\;. \tag{7.77}$$

The standard 2-3 Pachner move in Eq. (4.18) becomes,



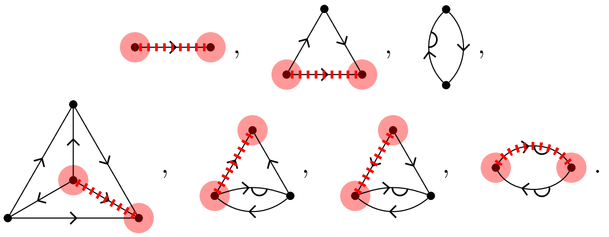
(7.78)

The red edges mark $d\eta+\omega_2^I$ inside the boundary, where again ω_2^I denotes the contributions to ω_2 from the interior faces and volumes only. Its boundary, the red vertices, are equal to ω_2 of the boundary 2-cellulation itself. We see that ω_2^I is already the same on the left and right-hand side, so we do not need to add any η to the move. So diagrammatically the move is the same as Eq. (4.19).

To define the spin analogue of the simplified 3D tTS from Section 4.8, we need to choose a definition of a spin structure for more general cellulations. To this end, we need to choose a 1-chain $\Omega[R]$ inside the boundary of every x-cell representatives R This choice needs to be consistent in the following way,

$$d\Omega[R] = \omega_{x-1}(R)\;. \tag{7.79}$$

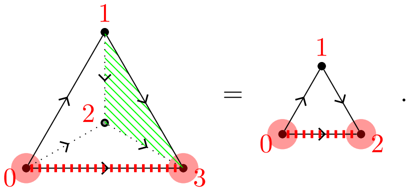
One possible choice is



(7.80)

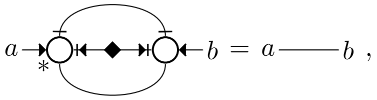
The red points for the two 2-cells above are given by ω_1 inside their boundary, and the red lines represent an orientation. The red points for the four 3-cells are ω_2 of their boundary 2-cellulation, as defined in Section 7.4. The tensor variables associated to the 3-cell above are the same as in Section 4.8.

Let us now consider some examples for how the spin structure enters into the moves of the simplified 3D tTS. First, the tetrahedron cancellation move in Eq. (4.107) becomes



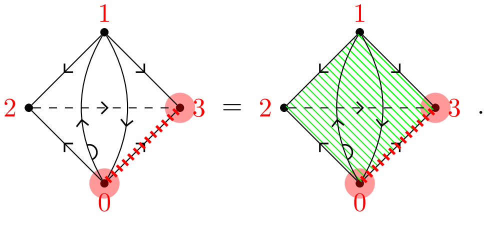
(7.81)

We have $\omega_2^I = 03 + 12 + 23 + 13$ on the left-hand side, but $\omega_2^I = 01$ on the right-hand side. So if we set $\eta = 013$, then $d\eta + \omega_2^I$ is equal inside the boundary on both sides. Diagrammatically, Eq. (4.108) thus becomes



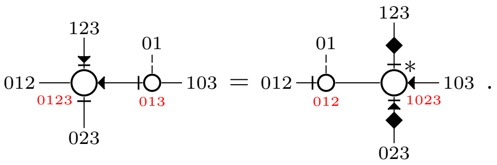
(7.82)

As a second example, the move in Eq. (4.119) becomes



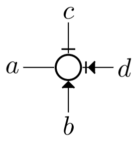
(7.83)

We have $\omega_2^I = 03$ on the left=hand side, and $\omega_2^I = 02 + 12 + 13$ on the right-hand side. So $d\eta + \omega_2^I$ agrees on both sides if we set $\eta = 023 + 123$. Diagrammatically, the move in Eq. (4.120) is changed to,



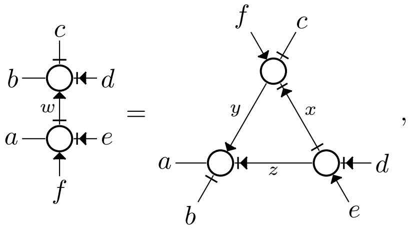
(7.84)

After having discussed the spin structure in the fermionic moves, let us study the fermionic reordering sign occurring in these moves. For this, we first choose an index ordering for the tetrahedron tensor,



$$\rightarrow \quad \bar{d}\bar{b}ca \; . \tag{7.85}$$

With this the reordering sign in Eq. (4.19),



(7.86)

is given by

$$\begin{aligned}\bar{e}\bar{f}wa\bar{d}\bar{w}cb|\bar{d}\bar{e}xz\bar{x}\bar{f}cy\bar{z}\bar{y}ba|0 &\rightarrow \bar{d}\bar{e}\bar{f}w\bar{w}cba|\bar{d}\bar{e}xz\bar{z}\bar{x}\bar{f}cy\bar{y}ba|ad\\ &\rightarrow \bar{d}\bar{e}\bar{f}cba|\bar{d}\bar{e}\bar{f}cba|ad \rightarrow ||ad \; .\end{aligned} \tag{7.87}$$

We can see that the reordering sign does *not* cancel like in the $1+1$-dimensional case. It does also not cancel for any other index ordering, and in fact we have chosen the index ordering to get a minimal expression for the reordering sign. So models of the 3D spin tTS are *not* in any one-to-one correspondence with models of the 3D $\mathbb{Z}_2$-cycle tTS.

Fermionic fixed-point models that are equivalent to our construction were discussed in Ref. [1], though in a rather different language. A fermionic analogue of the Turaev-Viro state sum is also sketched in Ref. [70].

7.6 Fermions with symmetry

In this section, we sketch how to represent phases with both fermionic degrees of freedom and symmetries. Due to the parity-even constraint of all fermionic tensors, $(-1)^{P_f}$ is automatically a symmetry, where P_f the fermion parity. This symmetry forms a $\mathbb{Z}_2$

subgroup of the full symmetry group G. Every symmetry representation has to preserve fermion parity, that is, commute with the corresponding $\mathbb{Z}_2$ subgroup. Thus, the $\mathbb{Z}_2$ subgroup must be normal, in other words, G is a group extension of some "bosonic" symmetry group G_b over $\mathbb{Z}_2$ corresponding to a short exact sequence,

$$\mathbb{Z}_2 \xrightarrow{p} G \longrightarrow G_b \; . \tag{7.88}$$

If the symmetry has time-reversing elements, then these give rise to a homomorphism s from G to $\mathbb{Z}_2$ like in Eq. (7.20). Since the fermion parity itself is not time-reversing, s is given via a homomorphism on G_b, such that we have a sequence of homomorphisms

$$\mathbb{Z}_2 \xrightarrow{p} G \longrightarrow G_b \longrightarrow \mathbb{Z}_2 \; . \tag{7.89}$$

We have to equip our triangulations/cellulations with the following homological extra structure. Namely, we introduce a G-valued $n-1$-chain η and a $\mathbb{Z}_2$-valued n-chain σ such that

$$\begin{aligned} d\eta &= p(\omega_2) \; , \\ d\sigma &= \omega_1 + s(\eta) \; . \end{aligned} \tag{7.90}$$

The resulting *2D G, s, p-chain tTS* is a combination of Sections 7.2 and 7.3. We have a mixed unitary/anti-unitary representation of G under which all the tensors are invariant. All tensors are fermionic tensors, the representation of $p(1) \in G$ is fixed to the fermion parity, and invariance of the tensors under $p(1)$ is equivalent to the parity constraint of fermionic tensors. In a recellulation, we have to choose η and σ such that $p(\omega_2^I) + d\eta$ and $d\sigma^I + s(\eta)$ are trivial in the interior and equal at the boundary on both sides.

Phases of fermions with time-reversing symmetries have been extensively studied in the setting of non-interacting fermionic systems. There, 10 different types of symmetry are known as the *10-fold way* [4, 123]. The 10-fold way classes have been translated to interacting many-body symmetries, see for example table III in Ref. [11]. For example, the 10-fold way class D corresponds to fermions without symmetry, $G = \mathbb{Z}_2$, $p = \mathrm{id}$, $s = 0$. The $\mathbb{Z}_2$ classification for this class D in spatial dimension $d = 1$ in Table 3 of Ref. [123] is precisely the Kitaev-chain model from Section 7.3. Or, the class DIII physically corresponds to a *time-reversal symmetry for spinful fermions*, that is, a time-reversal symmetry that squares to fermion parity. In our setting, this corresponds to $G = \mathbb{Z}_4$, $s = \bmod 2$, and $p = \cdot 2$.

Chapter 8

Axiomatic and extended TQFT

So far, we have been describing topological phases via fixed-point path integrals. In this chapter, we will generalize this perspective to incorporate descriptions in terms of higher-level invariants of a hypothetical fixed-point model. Like for fixed-point models, these higher-level descriptions will be formalized as models of TS, which we will refer to as *axiomatic topological TS (atTS)*. atTS have strong similarities to the notion of a *axiomatic TQFT* [6], more precisely *extended TQFT* [8]. However, the language we use is very different, and we do not attempt to make a direct connection here.

8.1 General framework

In this section, we define the general notion of atTS. Just like tTS, the tensor variables of atTS correspond to combinatorial representations of spacetime topological manifolds. There are however two major differences: First, while a tTS assigns tensor variables to n-cell representatives, an atTS assigns tensor variables to all extended cellulations of a fixed type, as defined in Section 2.4. Second, while the moves of tTS ensure recellulation invariance, the moves of atTS ensure compatibility with *gluing operations*. As such, atTS have infinitely many tensor variables and moves to begin with, however, for reasonable atTS we can find equivalent simplified versions with a finite number of generating tensor variables and axioms.

Before we get to defining atTS, let us first introduce gluing operations. We will first introduce gluing operations for extended manifolds, and then for extended cellulations. For every region r of an extended manifold M, every $0 \leq x < d_r$, and every embedding of $S_x \times B_{d_r-x}$ inside M_r, a gluing operation is the following operation yielding an extended manifold M'. First, we perform an x-surgery operation to M_r, applying

$$S_x \times B_{d_r-x} \to B_{x+1} \times S_{d_r-x-1} \tag{8.1}$$

to the corresponding embedding. Then, for every region r', we attach

$$B_{x+1} \times B_{d_r-x} \times (L_r)_{r'} \tag{8.2}$$

to $M_{r'}$ along $\psi_{r,r'}(S_x \times B_{d_r-x} \times (L_r)_{r'})$, where S_x is identified with ∂B_{x+1}. For example, if the extended manifold type consists of n-manifolds with boundary, then a gluing operation for the boundary region is the same as attaching an $x+1$-handle to the boundary. In the collapsed picture, gluings with $x = 0$ are determined by two points inside M_r, and can be pictured as 1) cutting out an ϵ-ball around each point, and 2) identifying the so-created boundaries.

For an extended cellulation M, a gluing operation can be defined for every pair of d_r-cells s_0, s_1 with the same d_r-cell representative S, and yields an extended cellulation $M|_{s_0\sim s_1}$ as follows. Consecutively, for every $t \in S_{d_r-1}[S]$, we change the pairing ψ to ψ' by setting $\psi'(\psi(t, s_0)) = \psi(t, s_1)$ and removing (t, s_0), and (t, s_1). Then, we remove s_0 and s_1 from M. For later usage, we will let $\mathrm{Int}(M, s_0, s_1) \subset S_{d_r-1}(S)$ denote the set of d_r-1-cells t of S that "disappear inside the adjacent higher-dimensional regions" during the gluing. Combinatorially, this happens when $\psi(t, s_0) = (t, s_1)$ in the process of setting ψ' above. Note that such $d_r - 1$-cells can only exist if s_0 and s_1 share a common d_r-cell.

Let us now consider the effect of gluing two d_r-cells on the corresponding extended manifolds. First, the "regular" case where $\mathrm{Int}(M, s_0, s_1) = \{\}$ corresponds to an $x = 0$ gluing of the extended manifold. If the topology of the d_r-1-cells in $\mathrm{Int}(M, s_0, s_1)$ is equal to $S_{x-1} \times B_{d_r-x-2}$, then this corresponds to a gluing operation on the extended manifolds with that same x. If the topology of $\mathrm{Int}(M, s_0, s_1)$ is different from any of the above, then this does not correspond to a single gluing operation of extended manifolds, but a sequence of gluing operations with different x.

Definition 26. *An* axiomatic topological tensor scheme (atTS) *is determined by*

- *an extended manifold type, and*
- *a labeling of each region as either an* internal *or a* space *region.*

It is defined by the following:

- *It contains one bond-dimension variable for every d_r-cell representative of every space region r.*
- *It contains a tensor variable $T(M)$ for every oriented extended cellulation M of the given type. This tensor variable has one index associated to every d_r-cell of M_r for every space region r, of the corresponding bond-dimension variable.*
- *It contains one* weight matrix *tensor variable $D[r, S, t]$ for every $d_r - 1$-cell t of every d_r-cell representative S for every space region r.*
- *Its moves contain the following* gluing axioms*: For every gluing operation on an extended cellulation M, gluing two d_r-cells s_0, s_1 of a space region r, where s_0 has positive and s_1 negative orientation, we have*

$$T(M|_{s_0\sim s_1})_{\dots} = \sum_{i_{s_0}, i_{s_1}} (T(M))_{\dots,i_{s_0},i_{s_1}} \Big(\prod_{t\in \mathrm{Int}(M,s_0,s_1)} D[r, S, t]^{\dagger_{\sigma(t)}} \Big)_{i_{s_0},i_{s_1}} . \tag{8.3}$$

  The product on the right is a product of matrices, whose ordering does not matter due to Eq. (8.6). *$\dagger_{\sigma(t)}$ denotes Hermitian conjugation if $\sigma[S](t) = 1$.*

- *Its moves contain the* internal invariance axiom*: For R a topology-preserving recellulation that only involves cells of internal regions, the associated tensors are unchanged,*
  $$T(R(M)) = T(M) \ . \tag{8.4}$$

- *Its moves contain the* disjoint union axiom*: For any two extended cellulations M_0 and M_1, we have*
  $$T(M_0 \sqcup M_1) = T(M_0) \otimes T(M_1) \ . \tag{8.5}$$

- *Its moves contain the* weight matrix commutation axiom*,*
  $$D[r, S, t_0]D[r, S, t_1] = D[r, S, t_1]D[r, S, t_0] \ . \tag{8.6}$$

- *Its moves contain the* Hermiticity axiom*,*
  $$T(\overline{M}) = \overline{T(M)} \ . \tag{8.7}$$

  Here, $\overline{M}$ denotes M with reversed orientation, and $\overline{T(M)}$ denotes the complex conjugate of $T(M)$.

Note that for "regular" gluings with $\mathrm{Int}(M, s_0, s_1) = \{\}$, the gluing axiom simplifies to

$$T(M|_{s_0=s_1}) = \sum_i (T(M))_{\ldots,i,i} \ . \tag{8.8}$$

Note that we can at the same time get rid of the internal invariance axiom and disjoint union axiom, and reduce the number of tensor variables. To get rid of the internal invariance axiom, we associate tensor variables to *extended semi-cellulations*, which are extended manifolds where only the space regions are cellulated. To get rid of the disjoint union axiom, we associate tensor variables only to connected extended semi-cellulations. When doing this, then we have to introduce another *disjoint-union-gluing* axiom: Taking the disjoint union of two connected extended semi-cellulations, and then gluing two space-region cells, one on each component, is compatible with taking the tensor product and then contracting the two indices. While Definition 26 is purely combinatorial and more systematic, the definition in terms of semi-cellulations is what we will actually use in practice. We should keep in mind though that in order to precisely define extended semi-cellulations, we would do so as equivalence classes of extended cellulations under internal-region recellulations.

As there is a large variety of extended manifold types, there is also a large variety of atTS. However, in n dimensions, there is a "standard" atTS, which we will simply refer to as the *nD atTS*. Its extended manifold type is given by a single internal n-region, and an infinite number of space d-regions with $d \leq n-2$. Namely, for *every* extended $n-d-1$-manifold L, there is a d-region whose link is L. This seems to be equivalent to what is known as $n-(n-1)-\ldots-2-1$-*extended TQFT*, *TQFT extended down to the circle*, or *almost-fully extended TQFT* in the mathematics literature. One nice feature of these atTS is that there exists simplified version of these atTS with a finite number of generating tensor variables and moves. In particular, the links of all d-regions are generated from a finite set of links via 1) disjoint union, 2) gluing operations on the space regions, and 3) cartesian products with ordinary manifolds. For example, the links of $n-2$-regions are 1-manifolds, and every 1-manifold is a disjoint union of circles. Thus, it suffices to only take a single generating $n-2$-region whose link is a circle. We will discuss 2D atTS in detail in Section 8.3, and 3D atTS in Section 8.4. Physically, these atTS are higher-level descriptions of hypothetical microscopic topological models, and roughly describe the behavior of the defects of dimension $\leq n-2$ in such models.

It should be noted that the nD tTS can also be understood as atTS, which we show in detail in Section 8.2. In particular, they can be understood as atTS with a single internal n-region, and an infinite number of d-regions with $d \leq n-1$ and arbitrary extended $n-d-1$-manifolds as links. This seems to be equivalent to *TQFT extended down to the point*, or *fully extended TQFT*. However, all extended-manifold links can be generated from a single one, namely the point. Thus, it suffices to only take a single space $n-1$-region with point link, which is also where we start from in Section 8.2. This seems to be equivalent to the *cobordism hypothesis* [8, 109] in extended TQFT, which states that a fully extended TQFT is determined by the thing it assigns to a point. Note that while the cobordism hypothesis in extended TQFT is a complex statement that it took decades to prove, it essentially comes for free in our framework. This is also related to the idea that fully extended TQFTs are equivalent to microscopic lattice models, which is folklore in extended TQFT, and essentially by construction in our framework.

More generally, for $0 \leq i \leq n$, one might consider atTS with a single internal n-region and space d-regions for $d \leq n-i-1$ with arbitrary $n-d-1$-manifolds as links. This seems to be equivalent to $n-(n-1)-\ldots-(i+1)-i$*-extended TQFT*, or *TQFT extended down to* i*-manifolds*. In particular, $i=n-1$ corresponds to *ordinary axiomatic TQFT* [6]. Whereas the cases $i=0,1$ were discussed in the previous paragraphs, the usefulness of the cases $i \geq 2$ for the classification of topological phases is very limited, for the following two reasons. First, these atTS do not have a finite set of generating tensor variables and moves. This is true alone by the fact that the links of $n-3$-regions are 2-manifolds of the internal region, and there is an infinite amount of connected 2-manifolds. So the $n-3$-region links are not generated from a finite set of 2-manifolds via disjoint union. Not having a finite generators-and-relations presentation makes dealing with atTS models very cumbersome. Second, the axioms in this case might simply be too lax to describe topological phases of actual microscopic models. As an extreme case of this, take $i=n$, in which case an atTS assigns an arbitrary number to every connected n-manifold. Obviously, not every such assignment represents the invariants of an actual microscopic topological phase. Vice versa, there is no good reason why we should believe that such an atTS model should describe a unique topological phase. In other words, two distinct topological phases might not be distinguishable by the invariants given by the atTS model. For these two reasons, we will not consider atTS with $i \geq 2$.

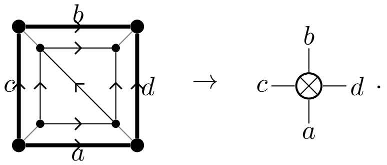
atTS can also be enriched with boundaries and lower-dimensional defects. To this end, we take more than one internal region. Specifically, consider a compound of defect types, and the extended manifold type $\mathcal{A}$ used to describe these defects as tTS models as in Chapter 6. If we want to describe the same defect types as atTS models, whose internal regions are formed by $\mathcal{A}$. For every internal x-region, there are infinitely many space d-regions with $d \leq x - i - 1$ with arbitrary extended $n - d - 1$-manifolds as link, where i can either be 0 or 1 depending on the internal region.

8.2 tTS as atTS

In this section, we will see that tTS are included in the general notion of atTS. Namely, nD tTS are atTS for the extended manifold type

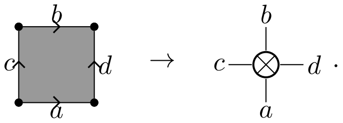
$$\underline{a} : (n, \varnothing) \ , \quad b : (n-1, \bullet) \ , \tag{8.9}$$

that is, manifolds (a) with boundary (b). b is a space region, and a is an internal region, which we indicate by underlining it. To be concrete, we will illustrate this for $n = 2$, and the generalization to higher dimensions is straight forward. A atTS according to Definition 26 associates to each 2-triangulation (or cellulation) with boundary a tensor, for example



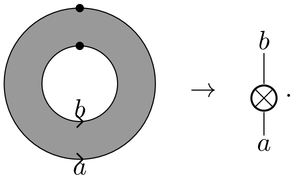
(8.10)

However, we will work with the simplified version of the atTS, where we associate tensors to extended semi-cellulations. In our case, these are 2-manifolds with cellulated boundary. We will indicate the topology of the internal 2-region by shading. So Eq. (8.10) becomes



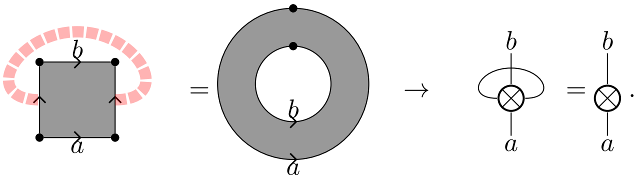
(8.11)

Let us now consider some examples for gluing axioms. We start with an example for a regular gluing. Consider the following tensor associated to an annulus whose boundary consists of two looping edges,



(8.12)

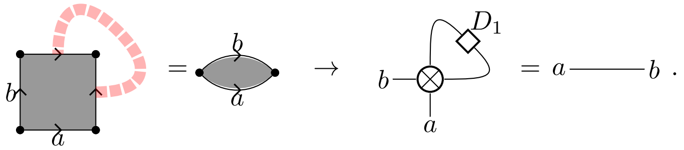
Then a gluing axiom is given by



(8.13)

On the left-hand side, we glue the edges on the left and right of the 4-gon disk in Eq. (8.11). Here and in the following, gluings are indicated by connecting the glued cells with a semi-transparent thick red dashed line. The gluing results in the annulus in Eq. (8.12). Hence, the 2-index tensor in Eq. (8.12) equals the tensor in Eq. (8.11), where we contract the according index pair. Since the two glued edges do not share any vertices, the gluing is regular, $\mathrm{Int}(M, s_0, s_1) = \{\}$, so we do not include any weight matrices in the contraction.

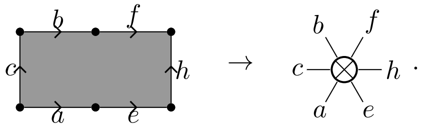
As an example for a non-regular gluing, consider



(8.14)

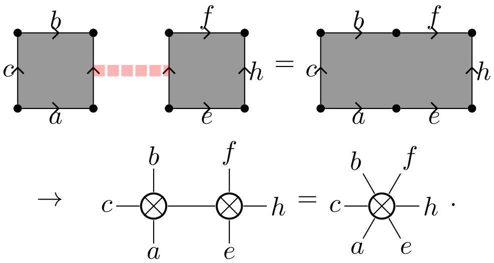
On the left-hand side of the first equation, we have the disk in Eq. (8.11), where we glue a pair of edges which have the same 1 vertex. As shown, this results in a 2-gon disk with two boundary edges on the right-hand side of the first equation. One might imagine "folding over" the two glued edges until they touch. Thereby, 1 vertex separating the glued edges disappears inside the interior region. In other words, we have $\mathrm{Int}(M, s_0, s_1) = \{1\}$ for this gluing. Therefore, we have to insert the weight matrix $D_1 := D[b, \mathrm{Sx}_1, 1]$ into the contraction on the left-hand side of the second equation. On the right-hand side, we have implemented the full support convention: Since gluing one edge of the 2-gon to any edge of any extended semi-cellulation, we can without loss of generality set the associated tensor to the identity.

Last, consider an example for a disjoint-union-gluing. To this end, we introduce a new tensor associated to a disk with 6 boundary edges,



(8.15)

With this, we have a disjoint-union-gluing axiom,

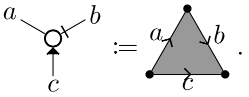


(8.16)

On top, we have a disjoint union of two copies of the 4-gon disk from Eq. (8.11). We glue one pair of their edges, yielding a 6-gon disk from Eq. (8.15). Hence, the tensor in Eq. (8.15) equals the tensor product of two copies of the tensor in Eq. (8.11), where we contract the two according indices.

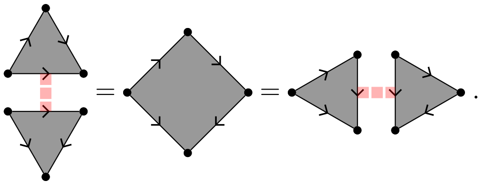
Let us now argue that the 2D tTS is just a simplified, finite generators-and-relations form of the above atTS. To this end, we show that the two are equivalent as TS by giving TS mappings between them that are inverses of each other. In one direction, there is a TS mapping from the above atTS to the 2D tTS as follows. The tensor variable for a connected extended semi-cellulation is mapped to the 2D-tTS tensor network associated to an arbitrary but fixed triangulation of the internal 2-region. We need to show that the mapped gluing and disjoint-union-gluing axioms follow from the 2D-tTS moves. The tensor networks on both sides of a mapped (disjoint-union-) gluing axiom are different internal triangulations of the same extended semi-cellulation. Thus, by the fact that all triangulations with the same topology are related by Pachner moves, this mapped move can be derived from the 2D-tTS moves.

Vice versa, the mapping from the 2D tTS to the above atTS simply maps the triangle tensor to a disk with three boundary edges,



(8.17)

The moves of the 2D tTS then follow from the (disjoint-union-) gluing axioms of the atTS. For example, the mapped 2-2 Pachner move in Eq. (4.11) is derived from the following two disjoint-union-gluing axioms,



(8.18)

The generalization to higher dimensions is straight forward: For example, in $2+1$ dimensions, extended semi-cellulations are 3-manifolds with triangulated boundary. To each such extended semi-manifold, we associate a tensor with one index for each boundary triangle. The 3D tTS is a simplified version of this atTS: An extended semi-cellulation can be mapped to the 3D-tTS tensor network associated to a fixed internal triangulation. Vice versa, the 3D-tTS tetrahedron tensor is mapped to the tetrahedron interpreted as a 3-manifold with cellulated boundary.

Also tTS with boundaries or other defects, described by some extended manifold type, correspond the finite generators-and-relations form of some atTS. Each region r of the tTS external manifold type yields an internal region $\underline{r}$ of the atTS extended manifold type whose link $\underline{L_{\underline{r}}}$ equals L_r. In addition, each tTS region r also yields a space atTS d_r-1-region $\overline{r}$, whose link $\underline{L_r}$ is determined by

$$\begin{gathered}(\underline{L_r})_{\underline{r'}} = (L_r)_{r'}\times[0,1]\ , \quad (\underline{L_r})_{r'} = (L_r)_{r'}\ , \quad (\underline{L_r})_{\underline{r}} = B_0\ ,\\ \psi^{\underline{L_r}}_{\underline{r},\underline{r'}}(0,x) = (x,0)\ , \psi^{\underline{L_r}}_{r,\underline{r'}}(x,l) = (x,l)\ ,\\ \psi^{\underline{L_r}}_{r',r''} = \psi^{L_r}_{r',r''}\ , \quad \psi^{\underline{L_r}}_{\underline{r'},\underline{r''}} = \psi^{L_r}_{r',r''}\times \mathrm{id}_{[0,1]}\ .\end{gathered} \tag{8.19}$$

In the collapsed picture, $\underline{L_r}$ is the *cone* of L_r, obtained by adding a single point corresponding to $\underline{r}$ and connecting it with every point in L_r by a line. For example, for manifolds with boundary, we get

$$\underline{a} : (n, \varnothing)\ , \quad \underline{b} : (n-1, ●)\ , \quad {\color{blue}c} : (n-1, ●)\ , \quad d : (n-2, ●\!\!-\!\!-\!\!{\color{blue}●})\ . \tag{8.20}$$

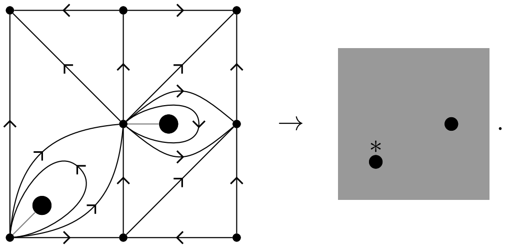
Extended manifolds of this type are n-manifolds ($\underline{a}$) with both physical ($\underline{b}$) and space (${\color{blue}c}$) boundary, separated along a $n-2$-manifold (d) inside the boundary. d is the space boundary of the physical boundary. Extended semi-cellulations are extended manifolds where ${\color{blue}c}$ as well as its boundary d are cellulated. The atTS associates indices to the $n-1$-cells of ${\color{blue}c}$ as well as the $n-2$-cells of d.

8.3 Axiomatic TQFT in $1+1$D

In this section, we will consider 2D atTS, that is, the "standard" atTS as discussed in Section 8.1. The extended manifold type consists of one internal 2-region $\underline{a}$, as well as space 0-regions with arbitrary 1-manifolds as links. Since all 1-manifolds are disjoint unions of the circle, it suffices to only take one space 0-region with circle link. So the extended manifold type is the same as in Section 6.1,

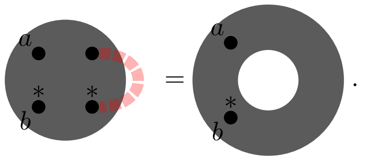
$$\underline{a} : (2, \varnothing)\ , \quad b : (0, ○)\ . \tag{8.21}$$

Like in the previous section, we underline internal regions. As we have seen in Section 6.1, the circle link of the 0-region can be cellulated with a single looping edge, such that an extended cellulation is given by a cellulated 2-manifold with b vertices adjacent to 1-gons. When going to extended semi-cellulations, we neglect the $\underline{a}$ cellulation. So we only draw the b vertices, and indicate the $\underline{a}$ topology by shading the corresponding 2-manifold, for example,



(8.22)

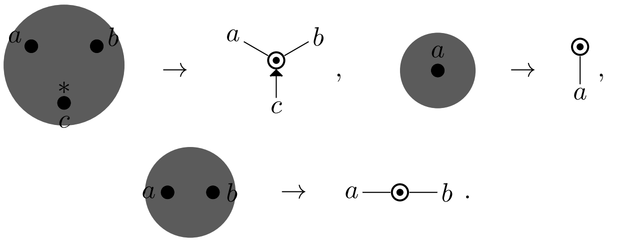
As shown, we mark negatively oriented b vertices with a $*$. So drawings of extended semi-cellulations then look the same as drawings for extended manifolds, apart from the orientation of the vertices. To every extended semi-cellulation, we associate a tensor with one index for each b vertex. In order to glue a positively and a negatively oriented b vertex, we cut out two small disks around them and identify the so-created boundary circles. As an example for a gluing axiom, consider the following,



(8.23)

We glue two vertices of a sphere with 4 b vertices, yielding a torus with 2 remaining b vertices.

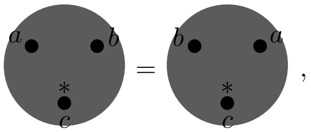
Let us now give a simplified form of the 2D atTS, consisting of a finite set of generating tensors and axioms. There are three generators, namely spheres with one, two, and three vertices,



(8.24)

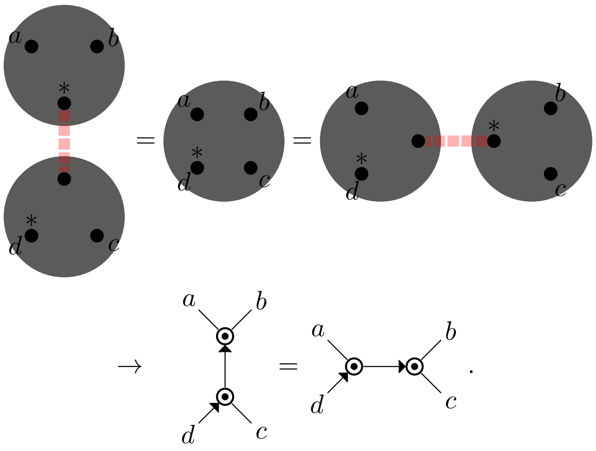
In fact, the 2-vertex sphere at the bottom (and therefore all extended semi-cellulations) can be glued from the first two generating tensors. We do however keep it as an auxiliary generator, since this makes the generating axioms simpler. To see that all extended semi-cellulations can be glued from copies of the above generators, we note that connected oriented 2-manifolds are classified by their genus. To obtain the extended manifold with genus n and x vertices, we first disjoint-union-glue $x+2n-2$ 3-vertex spheres at $x+2n-3$ vertex pairs. Then, we glue n vertex pairs of the resulting $x+2n$-vertex sphere. In this process, the orientation of any vertex can be flipped by gluing with the 2-vertex sphere above.

Next, the generating axioms are as follows. The first axiom is invariance under exchanging the two positively oriented vertices,



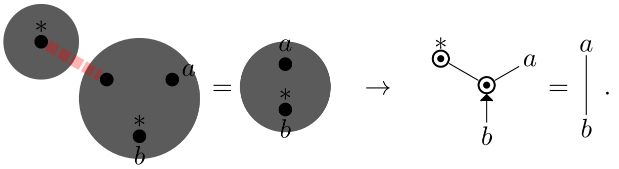
(8.25)

and is already implemented in the diagrammatic notation by the fact that we do not distinguish the corresponding indices by markings. Next, we have



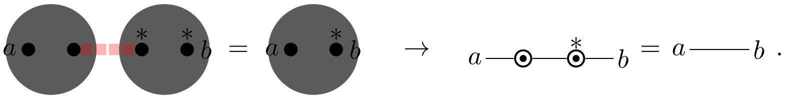
(8.26)

Another axiom is given by



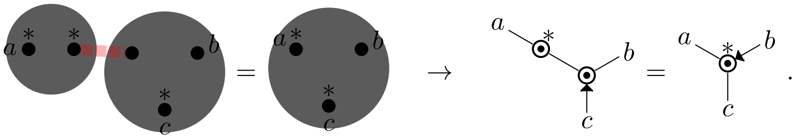
(8.27)

On the right-hand side, we have used the full support convention. Finally, we need two axioms involving the auxiliary generator. The first axiom is,



(8.28)

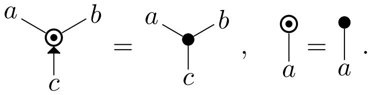
The second axiom is,



(8.29)

To see that the axioms above generate all gluing axioms, we notice that any process of gluing generators corresponds a tensor-network diagram. Using Eqs. (8.28) and (8.29), we can arbitrarily change the index markings of these diagram, such that it can be represented as a graph with 3-valent and 1-valent nodes. Using Eq. (8.27), we can remove all 1-valent nodes, such that only 3-valent nodes remain. The two resulting extended semi-cellulations are equal if the two graphs have the same number of independent loops. Two graphs with the same number of loops can be deformed into each other via the move in Eq. (8.26).

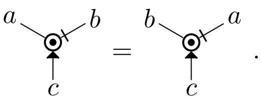
Let us now give an example for a model of the 2D atTS. For any bond dimension d, the δ-tensor provides such a model,



(8.30)

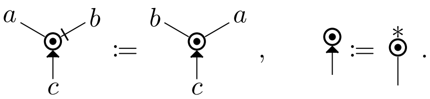
In fact, these are the only models up to a change of basis. Again, this model does *not* describe topological order, but rather a non-robust degenerate GHZ-like superposition.

The 2D tTS is equivalent to an algebraic TS, namely *commutative* †*-Frobenius algebras*. Recall the definition of †-Frobenius algebras in Section 4.12. A (†-Frobenius) algebra is commutative if



(8.31)

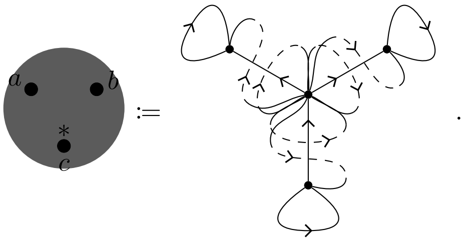
There is a very simple mapping from the commutative †-Frobenius-algebra TS to the 2D atTS, namely



(8.32)

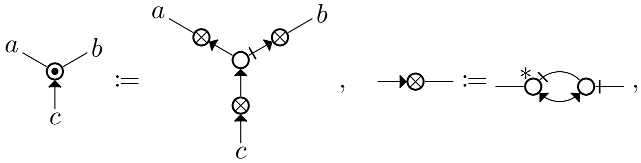
Under this mapping, commutativity becomes Eq. (8.25), associativity becomes Eq. (8.26), unitality becomes Eq. (8.27). Also the Frobenius law in Eq. (4.181) can be derived from the moves of the 2D atTS, and equals a gluing axiom like in Eq. (8.26) with different vertex orientations. There is a well-known alternative formulation of Frobenius algebras using a non-degenerate bilinear form, which can be represented as a 2-index tensor. This 2-index tensor is precisely the 2-vertex sphere from Eq. (8.24). In fact, there is also a weakly inverse mapping back from the 2d atTS to the commutative †-Frobenius-algebra TS, so the two TS are equivalent. In other words, 2D atTS models are precisely commutative †-Frobenius algebras.

As we mentioned in the beginning of this section, 2D-atTS models should be thought of as descriptions of a hypothetical microscopic model in terms more high-level invariants. It should be possible to determine these higher-level invariants from a microscopic model. This is what we will do now: We will construct a TS mapping from the 2D atTS to the 2D tTS, which constructs a 2D-atTS model from any 2D-tTS model. Physically, the 2D-atTS model describes the n-point functions in the microscopic 2D-tTS model. In a tensor-network path integral, an n-point function is computed by evaluating it on a cellulation whose space boundary consists of n small circles. Mathematically, we get the 2D-atTS tensor associated to an extended cellulation by simply evaluating the 2D-tTS model on the internal 2-region $\underline{a}$. Thereby, we triangulate of the circle b link by a single looping edge. Then, for any extended semi-cellulation, we replace every b vertex by a looping edge space boundary, and triangulate the internal 2-region. For example, the three-vertex sphere is mapped to



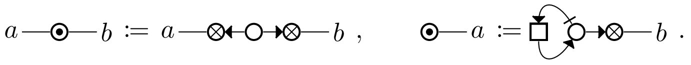
(8.33)

The right-hand side shows a triangulation whose boundary consists of three looping edges. It consist of one triangle in the middle, and three annuli around, each of which is triangulated with two triangles. Each annulus shares one loop edge with an edge of the triangle in the middle, such that the three corner vertices of the middle triangle coincide. Diagrammatically, we have



(8.34)

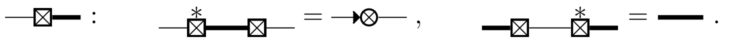
where we have used the 1D tTS tensor arising from the compactification mapping in Eq. (6.9). Similarly, we find



(8.35)

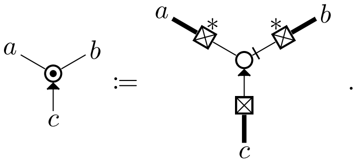
It is not hard to see that the moves of the 2D atTS can be derived from those of the 2D tTS. Algebraically, the compactified 1D tTS tensor is the projector onto the subspace of elements that commute with all others, which is known as the *center*.

Note that the obtained 2D atTS model does not obey the full support convention. This can be fixed by defining an isometry



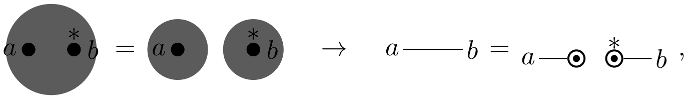
(8.36)

This new tensor has one index of a new bond dimension, which can be smaller than the original one. This is the bond dimension variable of the constructed 2D atTS. The tensors are defined by using the above isometry instead of the compactified 1D tTS tensor, for example,



(8.37)

Note that if we restrict to robust topological phases by imposing invariance of the internal region $\underline{a}$ under surgery moves, as described in Section 4.6, then the 2D atTS becomes immediately trivial. This is because one application of (backwards) 0-surgery yields



(8.38)

so every model obeying the full support convention has bond dimension 1.

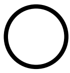
Let us describe how the 2D atTS is related to definitions of TQFT in the literature. 2D atTS models are very directly equivalent to *ordinary axiomatic TQFTs* [6] in $1 + 1$ dimensions [98]: Extended manifolds are equivalent to 2-dimensional *cobordisms*. There are only two small differences between our approach and the literature: First, we do not distinguish between the input and output boundary circles of the cobordisms, which makes our generators and relations a bit more compact compared to the literature. Second, we use the collapsed picture in our drawings, which we believe is closer to the physical meaning of these drawings, namely as n-point functions. Gluing of vertices in our picture is equivalent to composing cobordisms in conventional TQFT, and the tensor we associate to an extended semi-cellulation is equivalent to the linear map associated to a cobordism. The equivalence of ordinary axiomatic TQFT in $1 + 1$ dimensions with commutative-Frobenius algebras is well-known in the literature.

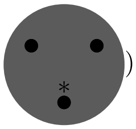
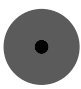
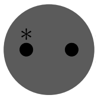
8.4 Anyon theories in $2 + 1$D

In this section, we will look at the 3D atTS. Physically, 3D atTS describe hypothetical topological fixed-point models in $2 + 1$ dimensions via their *anyon content*.

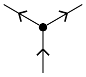
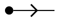
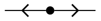
8.4.1 Definition of the bulk

Following the general notion of the nD atTS in Section 8.2, the corresponding extended manifold type that contains a single internal 3-region $\underline{a}$. In addition, there are infinitely many space regions r with $d_r \leq 1$, namely one for with every extended manifold as link. The links of the 1-regions are $\underline{a}$ 1-manifolds, and all such links can be generated from the circle by disjoint union. Thus it suffices to take a single 1-region b, whose link is the circle, like for the space 0-region of the 2D atTS. The links of the 0-regions are precisely extended manifolds of the type in Eq. (8.21), that is, 2-manifolds with embedded points. However, all such extended manifolds are generated from a finite set of links by gluing space regions. These generating 0-region links are precisely the generating extended manifolds for the 2D atTS in Eq. (8.24). So all in all, 3D atTS is generated by the following extended manifold type,

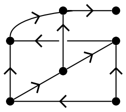
$\underline{a} : (3, \varnothing)$, $b : (1,$  $)$,

$c : (0,$  $)$, $d : (0,$  $)$, $e : (0,$  $)$. (8.39)

In the collapsed picture, extended manifolds are 3-manifolds with an embedded "string network", that is, a network of loops and string segments that braid around each other, and meet at three types of "fusion" vertices,

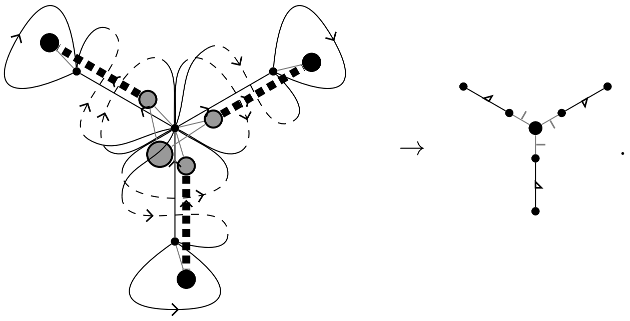
 ,  ,  . (8.40)

The strings are actually "ribbons" which carry a normal framing, and also the ϵ-sphere around every fusion vertex is identified with the corresponding link. When drawing an extended manifold, we will only draw the space regions b, c, d, and e, and specify the orientation of b by equipping each segment with an arrow. The framing is assumed to be "pointing to the right" when looking along the arrow direction, a convention that is also known as "blackboard framing". When drawing the extended manifolds on plain paper, a b segment may cross over or under another one, which we draw as shown in the following example,

 . (8.41)

Next, we need to consider extended semi-cellulations of this type. Extended cellulations of this type are very similar to the ones discussed in Section 6.7. A standard 1-cell representative for the space 1-region b is the one in Eq. (6.13), and a standard 0-cell representative for c looks like Eq. (6.136). The only difference is that all the 1-region vertices correspond to the same 1-region b and are *not* colored differently. Each of the above d_r-cell representatives of each space region r corresponds to a bond dimension variable. The 3D atTS assigns to every extended semi-cellulation a tensor, with one index for every d_r-cell of every space region r.

In order to draw an extended semi-cellulation, we will only draw the extended cellulation formed by the 1-region and 0-regions. For example, the following patch of an extended cellulation simplifies as

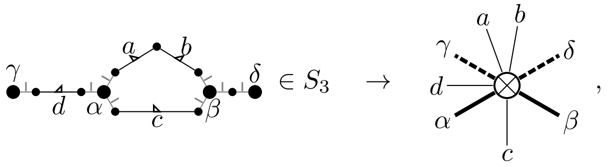


(8.42)

The left-hand side shows a patch of an extended manifold consisting of a vertex of region c and three edges of region b. On the right-hand side, we draw only the vertices and edges of c and b without the surrounding region $\underline{a}$. We also draw the edges and vertices less thick. The directions of the looping edges attached to the b vertices on the left are indicated by a "semi-tick" on the right that either point left or right on the line connecting the c vertex with each adjacent b vertex. Also, we make the arrows indicating the b directions into "flags" that also encode the dual direction. For a sequence of b edges and the potential c vertices at which they end, all dual directions have to agree. It is important that the planar embedding of the c vertices, that is, the cyclic order of the attached b vertices, remains the same in the drawing. With this, the three attached b vertices can be distinguished since the since the tick directions are acyclic. The dual tick and flag directions also allow us to define whether c vertices or b edges are positively or negatively oriented: A b edge is positively oriented if the its dual direction points to the right when looking along the primal direction. A c vertex is positively oriented if the two semi-ticks whose dual directions are aligned point clockwise.

For d and e vertices, the situation is analogous to the c vertices above, just that for the e vertices the two adjacent b vertices cannot be distinguished by the dual directions and the planar embedding. To fix this, we will mark one of the adjacent b vertices with an extra tick. If a segment of b edges is crossing over or under another, then we will denote this as in Eq. (8.41). Other than that, the topology of the 3-region $\underline{a}$ will be described in words, but for most drawings is just a 3-sphere.

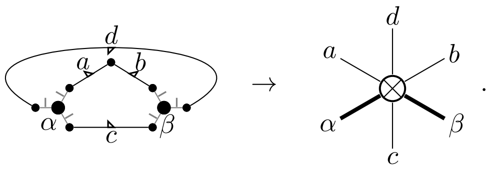
Once again, the 3D atTS assigns a tensor variable to each extended semi-cellulation, with one index at every b edge, and every c, d, or e vertex, for example



(8.43)

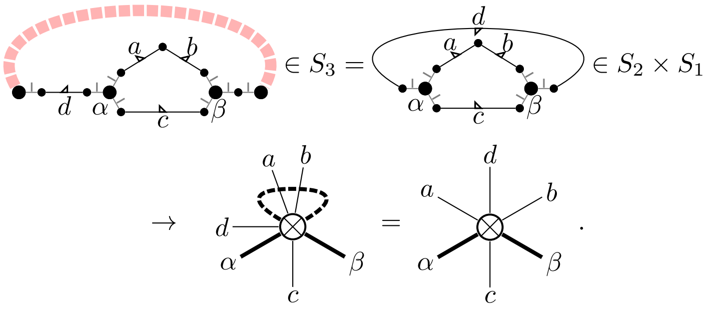
where the indices of the region d are drawn as dashed lines, and "$\in S_3$" denotes that the $\underline{a}$-topology is a 3-sphere.

Let us now look at the gluing axioms of the 3D atTS. There are gluing operations for every space region b, c, d, or e. As a first example, let us consider gluing of two d vertices. In addition to Eq. (8.43), we use the following tensor shape,



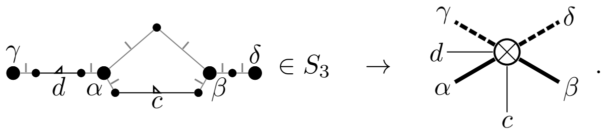
(8.44)

Using this, we get a gluing axiom,



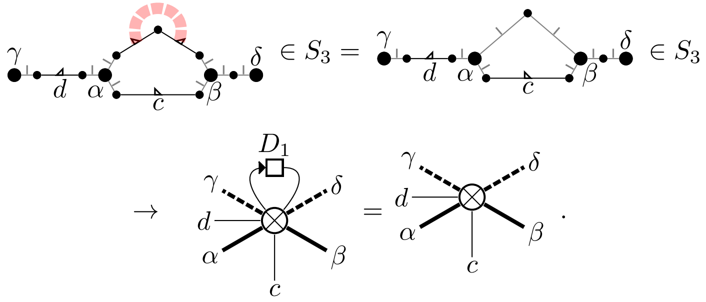
(8.45)

As another example, consider gluing of two b two edges. Using another tensor shape,



(8.46)

We get a gluing axiom,

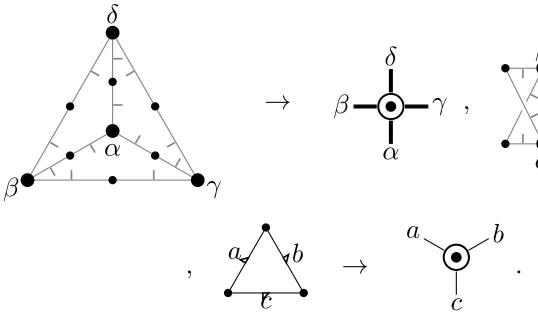


(8.47)

We see that since the two glued edges s_0, s_1 share their 1 vertex, so we have $\mathrm{Int}(M, s_0, s_1) = \{1\}$ and need to insert the weight matrix $D_1 \coloneqq D[b, \mathrm{Sx}_1, 1]$ when contracting over the two corresponding indices.

Note that in order to ensure that we are describing non-degenerate robust phases, we impose invariance of the internal region $\underline{a}$ under 0-surgery moves, in addition to topology-preserving moves, analogous to Section 4.6.

It is possible to find a set of generating tensors and axioms for the 3D atTS. Each of the generating extended semi-cellulations has the overall topology of the 3-sphere S_3. Some of the generators look like

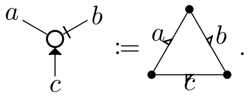


(8.48)

However, before we go to a full description of the generating cellulations and axioms, we will bring the 3D atTS into a block-diagonal form.

8.4.2 Block-diagonal form

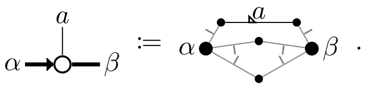
In this section, we will derive a block-diagonal form of the 3D atTS using methods similar to those in Section 4.14 for the 3D tTS. In particular, we again define a 2D-tTS model, which we will call the *anyon 2D-tTS model*,



(8.49)

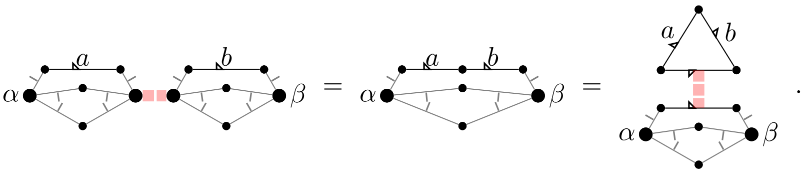
That is, the triangle tensor is mapped to to a loop of three b edges inside S_3. This model is the result of a mapping from the 2D tTS to the 3D atTS. If we view the 2D tTS as an atTS with internal region $\underline{a}$ and space region b as in Section 8.2, then we can apply this mapping to any extended semi-cellulation of type in Eq. (8.9). Then, the mapping maps the internal 2-region $\underline{a}$ in Eq. (8.9) to the internal 3-region $\underline{a}$ in Eq. (8.39) by taking the cartesian product with the circle. The space 1-region b in Eq. (8.9) and its cellulation is directly mapped to b in in Eq. (8.39).

We can also define a boundary for the anyon 2D-tTS model, which we will call the 0 c *anyon 2D-boundary-tTS model*,



(8.50)

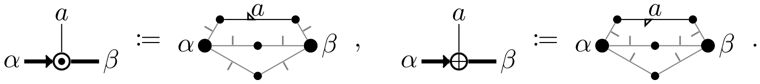
For example, the boundary Pachner move in Eq. (4.160) maps to the following gluing axiom,



(8.51)

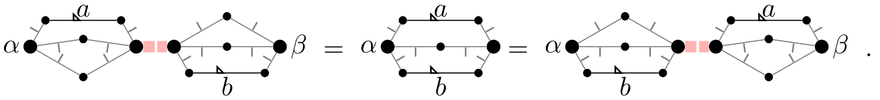
Again, this can be viewed as the result of a dimension-reduction TS mapping that extends the previous TS mapping.

We get two variants of this boundary of the boundary in Eq. (8.50), which we call the 1 *c and the* 2 *c anyon 2D-boundary-tTS model*,



(8.52)

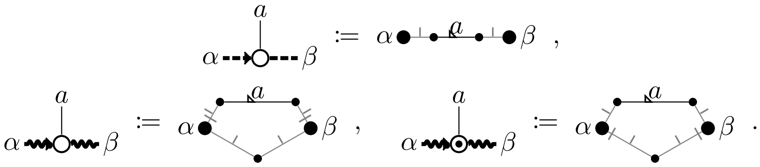
To be precise, the 2 anyon 2D-boundary-tTS model is a boundary of the complex conjugated anyon 2D-tTS model. These three boundaries commute mutually, that is, they satisfy commutation relations such as in Eq. (4.170). For example, the commutation relation for the 0 and 1 boundaries corresponds to the following gluing axiom,



(8.53)

So all in all, by applying the above mappings, we get a 2D-tTS model and three commuting boundaries for it. We can now apply Propositions 1 and 3 from Section 4.13. The 2D tTS models and boundaries are then determined by a set of irreps, their dimensions λ_i, their weights ω_i, and their multiplicities $N_k^{ij} := m_{(i,j,k)}$.

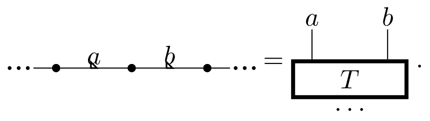
Analogously, there is a *d anyon 2D-boundary-tTS model* as well as 0 *e and* 1 *e anyon 2D-boundary-tTS models*, given by



(8.54)

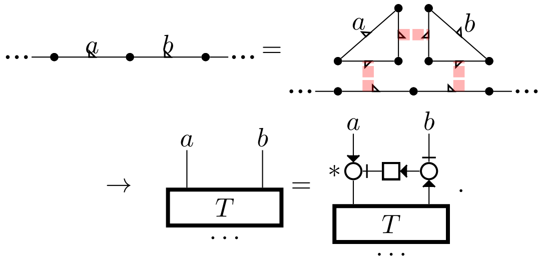
These boundaries are described by additional multiplicities $N^{ij} := m_{(i,j)}$, and $N^i := m_i$. Now apply the block-diagonalizing isometries in Eqs. (4.200) and (4.204) to all the b edge indices and c, d, or e vertex indices of the 3D atTS. Each b edge index becomes a triple (i, a, b), each c vertex index a 7-tuple (i, a, j, b, k, c, μ), each d vertex index a triple (i, a, μ), and each e vertex index a 5-tuple (i, a, j, b, μ). Here, i, j, k are irrep labels, a, b, c are internal indices, and μ are multiplicity indices of dimension N_k^{ij}, N^i, or N^{ij}, respectively.

Now, consider two adjacent b edges inside an arbitrary extended semi-cellulation, and the associated tensor,



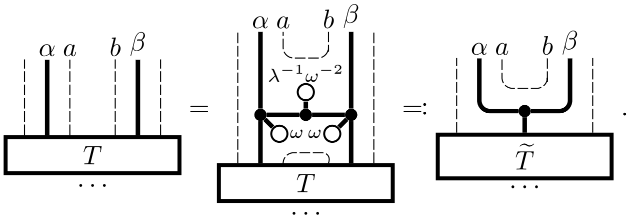
(8.55)

It is invariant under the following gluing,



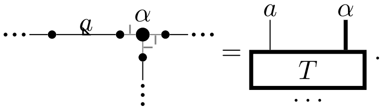
(8.56)

In the block-diagonal basis, this becomes



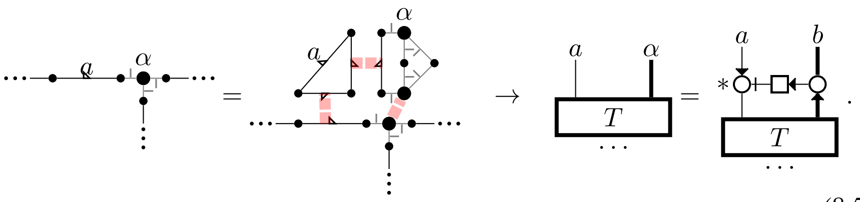
(8.57)

This equation is analogous to Eq. (4.214) taking into account that the 2D tTS tensor defines two commuting boundaries of itself where the 2D-boundary-tTS model is the cone extension boundary from Section 5.3 of the anyon 2D-tTS model. So the above is a variant of Proposition 4.214. $\widetilde{T}$ has one single irrep index for both edges, and the two according internal indices have disappeared. Next, consider a c vertex with a b edge adjacent to its non-marked b vertex, as part of an arbitrary extended semi-cellulation,



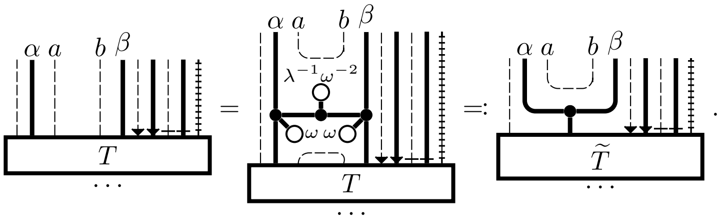
(8.58)

It is invariant under the following gluing:



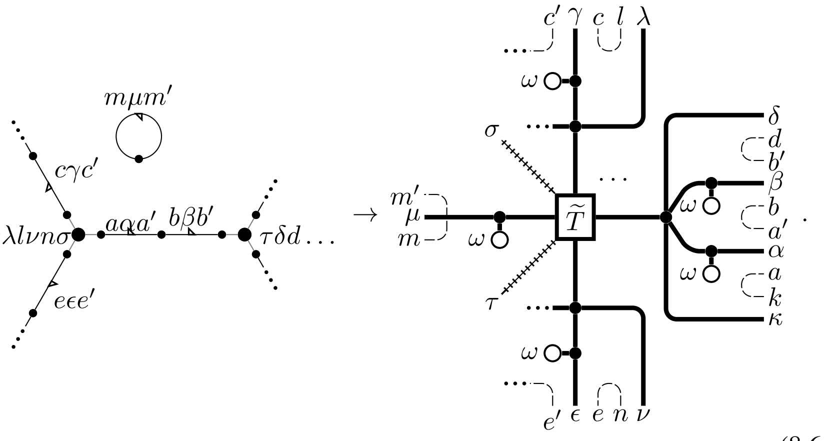
(8.59)

After applying the block-diagonalizing isomorphism, this becomes:



(8.60)

Here we marked the 1 c irrep and internal indices with a tick and the 2 c irrep and internal indices with an arrow. Again, this is a variant of Proposition 4.214. The tensor is determined by a simpler tensor $\widetilde{T}$ with only one common irrep index, and where the according internal indices have been removed. This argument applies to all constellations of a b edge and an adjacent c, d, or e vertex, as well as to adjacent pairs of vertices. Thus, if we apply the above to all such adjacent pairs of an extended cellulation, we end up with a tensor with only one irrep index for each b *segment*. A b segment is either a loop of b edges, a sequence of b edges connecting two c, d, or e vertices, or a single b vertex separating two c, d, or e vertices. Additionally, there is one multiplicity index at every c, d, or e vertex, whose dimension is N_k^{ij}, N^i, or N^{ij} respectively depending the irreps of the adjacent b segments. As an example, consider the following example for how a 3D-atTS tensor is determined by the simpler tensor $\widetilde{T}$:



(8.61)

As shown, for $\widetilde{T}$ to become independent on the number of b edges within a segment, we do include one weight ω_i^{-1} for every edge. After block diagonalization, the 3D atTS associates a tensor to every extended manifold of the type in Eq. (8.39). The tensor has one irrep index for every connected component (segment) of the b region, and one multiplicity index for each c, d, or e vertex. Similar to Section 4.14, we will express the block-diagonal form of the 3D atTS through the following weights,

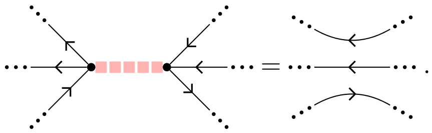
$$D := \frac{1}{\mathcal{T}(S_3)} , \qquad d_i := D\omega_i^{-2} \quad \Rightarrow D = (\sum_i d_i^2)^{1/2} , \tag{8.62}$$

where $\mathcal{T}(S_3)$ is the scalar that the 3D atTS associates to the 3-sphere.

Furthermore, using arguments analogous to around Eq. (4.255), we can show that the numbers N associated to different types of fusion vertices form an non-negative-integer-valued 2D atTS model. This implies that $N^{ab} = \delta_{a,b^*}$, analogous to Eq. (4.258). Due to the commutativity, there must be an irrep 1 such that $N^a = \delta_{a,1}$ as in Eq. (4.267).

Let us now see how the gluing axioms translate to the block-diagonal form. We start with (disjoint-union-)gluing of two c vertices of an extended semi-cellulation. On the level of extended manifolds, this corresponds to gluing two points of the region c

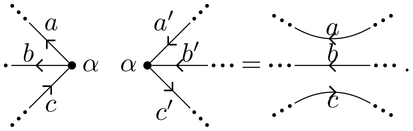
as described in Section 8.1 around Eq. (8.1). In the collapsed picture, we remove the two vertices and connect the adjacent b segments pairwisely,



(8.63)

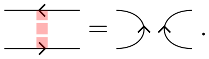
The effect of this gluing on the overall topology is a 0-surgery, see Section 5.5. Contracting the two indices of the 3D-atTS tensors will contract their multiplicity part, and force their irrep components to take the same value. We therefore have

$$\sum_{\alpha} \delta_{aa'}\delta_{bb'}\delta_{cc'}$$



(8.64)

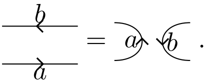
The gluing of d and e vertices is completely analogous. Next, consider (disjoint-union-)gluing two b edges in an extended semi-cellulation, that do not share their 0 vertex or their 1 vertex. The effect of such a gluing on the extended manifolds is to cut the two b segments and pair them of in a different way,



(8.65)

The effect on the overall topology is again a 0-surgery. Since each edge comes with a factor of ω_i, we have to include a normalization of $\omega_i^2 = D/d_i$,

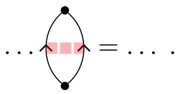
$$(\frac{d_a}{D})^{-1}\delta_{a,b}$$



(8.66)

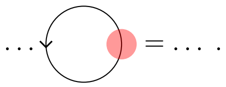
Note that this implies that if there are only two b segments "running along the same 1-handle in opposite directions", then their labels must be the same, like $a = b$ on the right-hand side above. Next, consider the gluing of two b edges that share either their 0 vertices or their 1 vertices. The effect of such a gluing on the extended manifold is trivial, since it just shortens the corresponding b segment. In the contraction, we have to include the corresponding weight matrix, which in the block-diagonal basis is $\lambda_i^{-1}\omega_i^{-2}$. The two ω weights at each edge cancel the ω_i^{-2} part. We also get an internal index loop canceling the λ_i^{-1} part. So (as expected) the effect of contraction on the tensor $\widetilde{T}$ is trivial as well.

Finally, consider the gluing of two b edges that share both their 0 vertex and their 1 vertex,



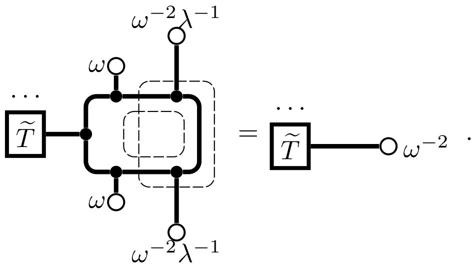
(8.67)

For the corresponding ribbon manifold, this makes a b loop segment disappear,



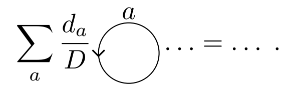
(8.68)

The effect on the overall topology is to apply a 1-surgery to the neighborhood of the b loop. We will denote this gluing operation of a loop edge by putting a red shaded dot somewhere on the loop. In the corresponding contraction, we have one ω weight for every edge, we have to include two weight matrices $\lambda^{-1}\omega^{-2}$, and we get two internal-index loops,



(8.69)

So overall, we need to normalize with $\omega_i^{-2} = d_i/D$,

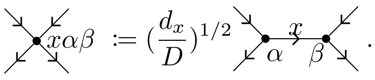


(8.70)

All in all, the block-diagonal form of the 3D atTS associates to each extended manifold a tensor such that the (disjoint-union-)gluing axioms in Eqs. (8.64), (8.66), and (8.70) hold. The translation of the robustness axiom to the block-diagonal setting is trivial. The Hermiticity axiom still states that orientation reversal equals complex conjugation. Since we implicitly determine the orientation by the embedding into the drawing paper, reversing the orientation is given by reflecting a diagram and flipping all segment directions.

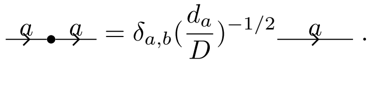
8.4.3 Generators and relations

After going to the block-diagonal form, we are now ready to propose a set of generating tensors and axioms. Before we get to this, let us argue more explicitly why we can restrict to a finite set of space 0-regions and 1-regions. At the beginning of Section 8.4.1, we have postulated that the generating 1-region links are just the generating 0-region links in Section 8.3, and the generating 0-region links are the generating extended manifolds. All other types of fusion vertices can be constructed from the three in Eq. (8.40). For example, a 4-valent fusion vertex can be emulated via



(8.71)

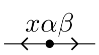
Gluing two of these 4-valent vertices is equivalent to first gluing both vertex pairs labeled α and β via Eq. (8.64), and then applying 1-surgery to the resulting x loop via Eq. (8.70). As another example, a fusion vertex where the orientation does not change can be emulated just by a piece of line,



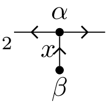
(8.72)

Gluing two of the vertices is equivalent to gluing the two segments via Eq. (8.66).

Just like in Section 8.3, the sphere-with-2-points link is auxiliary and can be emulated by

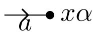


$$:= (\frac{d_x}{D})^{1/2} \qquad (8.73)$$

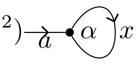


.

Note that if we impose robustness, that is, 0-surgery invariance of the internal region $\underline{a}$, then also the 1-valent vertex can be emulated via the 3-valent one,



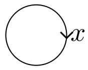
$$:= (d_x^{1/2}) \qquad (8.74)$$



.

Gluing a pair of 1-valent vertices is equivalent to first gluing the pair of 3-valent vertices, then applying the 1-surgery to the resulting x loop via Eq. (8.70), and then applying a backwards 0-surgery on $\underline{a}$. Note that we could even define "singular" fusion vertices, whose link does not have the overall topology of a 2-sphere, but some other 2-manifold, potentially with b vertices. For example, fusion vertices whose link is a torus without any b vertices can be emulated via

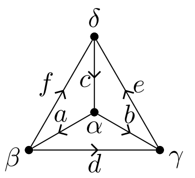
$$\underset{(S_1 \times S_1)}{\overset{x}{\bullet}} := \qquad (8.75)$$



.

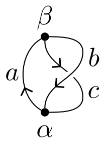
Instead of gluing a pair of singular vertices, we first glue the two according loops via Eq. (8.66), which fuses the two loops to a single loop. Then we perform 1-surgery to the resulting single x loop via Eq. (8.70). The prefactors for these two gluing operations cancel.

Next, we will define the generating tensors of the block-diagonal 3D atTS. These are related to the generating gluing axioms of the 2D atTS in Section 8.3 as follows: Every vertex of the 3D generating tensor corresponds to one generating tensor of the 2D generating axiom one side. The link of the 3D vertex and the extended manifold of the 2D generating tensor are the same. Every b segment connecting two vertices corresponds to either a gluing of between two vertices in the 2D generating axioms, or a pair of matching indices on the left and right-hand side of the 2D generating axiom. The associativity axiom in Eq. (8.26) thus yields the following *tetrahedron* generator,



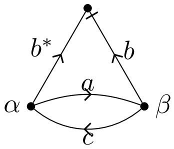
$$\rightarrow \quad F_{ce\gamma\delta}^{fdb\,a\alpha\beta} \; . \qquad (8.76)$$

The overall topology is a 3-sphere. Next, the commutativity axiom in Eq. (8.25) yields the following *braiding* generator,

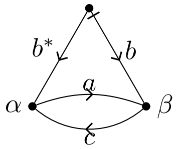


$$\rightarrow \quad R_{a\alpha}^{bc\,\beta} \; . \qquad (8.77)$$

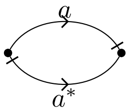
It consists of two 3-valent fusion vertices, one with negative orientation, whose adjacent ribbons are swapped. The overall topology is again inside the 3-sphere. In addition, we introduce the following three auxiliary tensors,



$$\to \quad G_b^{ac\alpha\beta}\,,$$



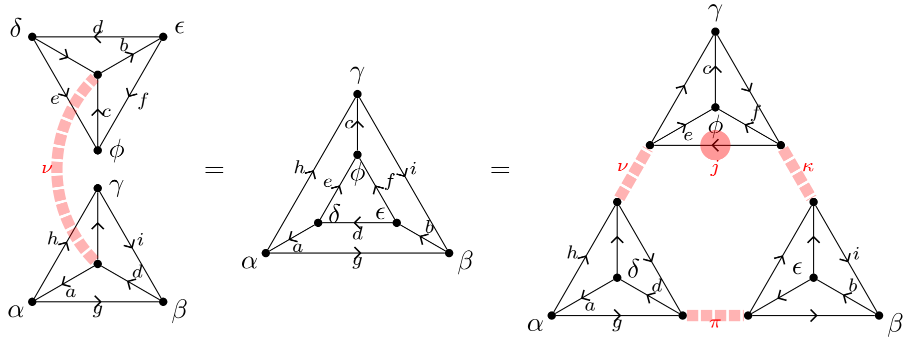
$$\to \quad H_b^{ac}{}_{\alpha\beta}\,, \tag{8.78}$$



$$\to \quad \kappa_a\,.$$

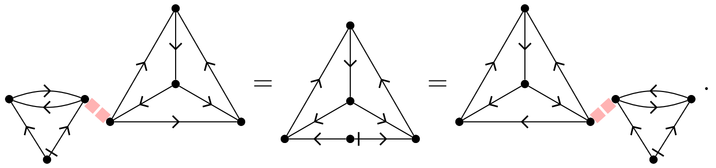
These auxiliary tensors correspond to the 1D "auxiliary" axioms in Eqs. (8.29) and (8.28). Note the great similarities to the diagrams for the triangle and 2-gon flippers in Eqs. (4.111), (4.110), and (4.112). However, the diagrams above are Poincaré dual to these, and moreover are embedded into 3-dimensional space.

Next, we will propose a finite set of generating gluing axioms. We will name these generating axioms in the same way as for the very similar categorical structures. First, we notice that there is the following *mirror-copy densification* mapping from the (block-diagonal) 3D tTS to the present 3D atTS. Recall that the 3D tTS can be thought of as an atTS whose extended semi-cellulations are 3-manifolds with cellulated boundary. This can be mapped to an extended semi-cellulation of the 3D atTS by 1) adding an orientation-reversed copy of the bulk on the other side of the boundary and 2) taking a string network Poincaré dual to what used to be the boundary cellulation. Through this mapping, the 2-3 Pachner move of the 3D tTS yields the following *pentagon equation*:



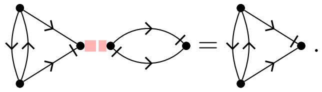
$$\to \quad \sum_{\nu} F_b^{fea}{}^{d\epsilon\delta}_{c\phi\nu} F_d^{chg}{}^{a\nu\alpha}_{i\gamma\beta} = \sum_{j,\nu,\kappa,\pi} \frac{d_j}{D} F_d^{ehg}{}^{a\delta\alpha}_{j\nu\pi} F_i^{feh}{}^{j\kappa\nu}_{c\phi\gamma} F_b^{fjg}{}^{d\epsilon\pi}_{i\kappa\beta}\,. \tag{8.79}$$

Further, there are the triangle flip moves in Section 4.8 yield axioms such as



(8.80)

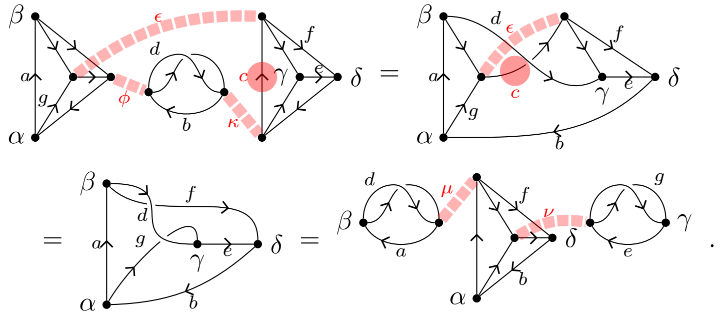
Or, the 01 triangle flipper flip move yields an axiom,



(8.81)

In addition, we get moves from the tetrahedron cancellation move, a triangle flipper cancellation move, and a 2-gon flipper cancellation move.

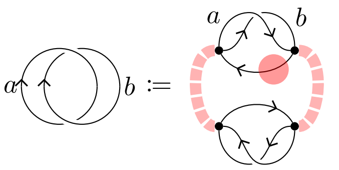
Next, we have moves that include the 3-dimensional structure by including the braiding tensor. The most important such move is the *hexagon equation*,



(8.82)

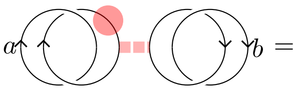
$$\rightarrow \quad \sum_{c,\epsilon,\phi,\kappa} \frac{d_c}{D} F^{gfd}_{b}{}^{a\alpha\beta}_{c\epsilon\phi} R^{dc}_{b}{}^{\phi}_{\kappa} F^{dgf}_{b}{}^{c\kappa\epsilon}_{e\gamma\delta} = \sum_{\mu,\nu} R^{df}_{a}{}^{\beta}_{\mu} F^{gdf}_{b}{}^{a\alpha\mu}_{e\nu\delta} R^{dg}_{e}{}^{\delta}_{\gamma} \,.$$

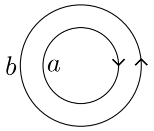
All of the above axioms were such that the topology on both sides is a 3-sphere. There are two more generating axioms that are simpler in terms of the tensors, but topologically more involved. The first one is the *modularity condition*, which we express via another auxiliary tensor,

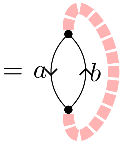


$$\rightarrow \quad S_{ab} \coloneqq \sum_x \frac{d_x}{D} \overline{R^{ab}_{x}{}^{\beta}_{\alpha}}\, \overline{R^{ba}_{x}{}^{\alpha}_{\beta}} \,. \tag{8.83}$$

The diagram on the left shows two linked loops inside a 3-sphere. With this, the axiom is given by



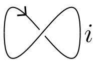




$$\in S_2 \times S_1 , \quad a \sim x \times S_1 , b \sim y \times S_1 \quad \to \quad \sum_b S_{ax}\overline{S_{bx}} = \delta_{a,b} \ , \quad \text{or} \quad SS^\dagger = \mathbb{1} \ . \tag{8.84}$$

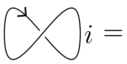
x and y are any two points of S_2.

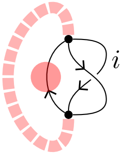
Last, there is the *anomaly-freeness condition*, which is again defined via an auxiliary tensor, namely the *twist*,



$$\to \quad \frac{d_i}{D}\theta_i \ . \tag{8.85}$$

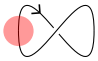
It can be defined from one copy of the braiding,





$$\to \quad \frac{d_i}{D}\theta_i = \sum_{x,\alpha} \frac{d_x}{D} R^{ii\alpha}_{x\ \alpha} \ . \tag{8.86}$$

With this, the axiom is given by

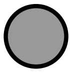


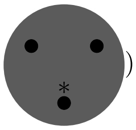
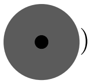
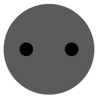
$$= S_3 \quad \to \quad \sum_a (\frac{d_a}{D})^2 \theta_a = \frac{1}{D} \ , \text{or} \quad \sum_a \frac{d_a^2 \theta_a}{D} = 1 \ . \tag{8.87}$$

Let us remark that this is just an equation between scalars, so if we interpret all of the equations up to scalars, then this is a trivial axiom.

8.4.4 Chiral anomaly

In fact, it turns out that we need to relax the anomaly-freeness condition in Eq. (8.87) in order to describe all intrinsic topological phases in $2+1$ dimensions. This is no problem because physically it is fine if all equations only hold up to a global scalar prefactor, and in this sense, the anomaly-freeness condition is a trivial axiom. However, the scalar prefactor has an interesting physical and mathematical interpretation, which we will study in this section. To this end, we will fix the anomaly-freeness by adding a 4-dimensional internal region $\underline{f}$ whose boundary is the internal 3-region a. That is, the extended manifold type is now given by

$\underline{f} : (4, \varnothing)$, $\underline{a} : (3, \bullet)$, $b : (1,$  $)$,

$c : (0,$  $)$, $d : (0,$  $)$, $e : (0,$  $)$, (8.88) where the links of c, d, and e are now 3-balls with b points inside the boundary. The gluing axioms are the same as before, just that 0-surgery operations become 1-handle attachments to $\underline{a}$, and 2-surgery operations become 2-handle attachments. In addition, we add invariance of $\underline{f}$ under arbitrary surgery operations. That is, the hypothetical 4-dimensional model that $\underline{f}$ is describing is invertible. An immediate consequence is that the atTS tensors depend only on the cobordism class of $\underline{f}$, instead of the full topology. The generating tensors are still the same as before, just that we add a 4-ball for the 4-region $\underline{f}$ filling the $\underline{a}$ 3-sphere. As an additional generating tensor, we add the scalar associated to a standalone $\mathbb{C}P(2)$ $\underline{f}$ region,

$$\mathbb{C}P(2) \quad \rightarrow \quad e^{2\pi i \frac{c}{8}} \; , \tag{8.89}$$

where c will be called the *chiral central charge*, and is defined mod 8. $\mathbb{C}P(2)$ is the generator of the $\mathbb{Z}$ oriented cobordism group in 4 dimensions, thus we can arbitrarily change the topology of the $\underline{f}$ region by surgery operations and disjoint union with $\mathbb{C}P(2)$. Note that as we will see later, $e^{2\pi i c/8}$ or $c \mod 8$ is fully determined by the other generators, but $\mathbb{C}P(2)$ cannot be obtained diagrammatically by gluing them.

Let us now look at how the presence of the 4-region $\underline{f}$ affects the generating axioms. The axioms in Eqs. (8.79), (8.82), (8.80), or (8.81) have the topology of a 4-ball on both sides, and are thus unaffected by the presence of $\underline{f}$. Gluing the left-hand side of the modularity condition in Eq. (8.84) yields $S_2 \times B_2$ as $\underline{f}$ region, whereas on the right-hand side, the $\underline{f}$ region that yields the δ-tensor is $B_3 \times S_1$. However, the two $\underline{f}$ regions are precisely related by a 2-surgery (or backwards 1-surgery) operation and are thus in the same cobordism class. So also this gluing axiom holds for after adding the 4-region $\underline{f}$.

Next, let us look at the anomaly-freeness condition in Eq. (8.87). When we attach a 2-handle to the once-twisted loop inside the boundary 3-sphere of a 4-ball, we obtain the complex projective plane with the neighborhood of a point removed, c.f. page 47 of Ref. [111],

$$\mathbb{C}P(2) - B_4 \; . \tag{8.90}$$

In contrast, the right-hand side, to which we associate $\frac{1}{D}$, is a 4-ball B_4. $\mathbb{C}P(2) - B_4$ and B_4 are *not* related by any sequence of surgery operations, since they are in different cobordism classes relative to their fixed boundary S_3. More precisely, they differ by a connected sum (that is, a disjoint union followed by 0-surgery with one point on each summand) with $\mathbb{C}P(2)$. Thus, the axiom becomes

$$\sum_a \frac{d_a^2 \theta_a}{D^2} = \frac{1}{D} e^{2\pi i \frac{c}{8}} \; . \tag{8.91}$$

This can be seen as a definition of $c \bmod 8$ from d and θ.

Another interesting variant is to consider the 3D atTS with the $\underline{f}$ 4-region, but without surgery invariance. As we have noted above, the axioms in Eqs. (8.79), (8.82), (8.80), or (8.81) yield a 4-ball on each side and thus still hold after adding $\underline{f}$. However, the modularity condition in Eq. (8.84) does not hold anymore, and the same is true for the anomaly-freeness condition in Eq. (8.87).

Let us now describe the relation to the existing literature. The block-diagonal form of the 3D atTS with surgery-invariant 4-region $\underline{f}$ turns out to be equivalent to *unitary modular tensor categories (UMTCs)* [131], and we have already named our generating tensors and axioms accordingly. A unitary fusion category together with a *braiding* or *R-matrix* satisfying a *hexagon equation* defines a *unitary braided fusion category*. A unitary braided fusion category is a UMTC if its *S-matrix* is unitary. The associator and R-matrix of the UMTC (denoted as $\tilde{F}$ and $\tilde{R}$ in the following), are equal to the generating tensors F and R up to normalization,

$$\tilde{F}^{fdb\;a\alpha\beta}_{c\;\;\;e\delta\gamma} = (\frac{d_a}{D})^{-1/2}(\frac{d_e}{D})^{-1/2} F^{fdb\;a\alpha\beta}_{c\;\;\;e\delta\gamma} \;, \quad \tilde{R}^{ab\;\beta}_{c\;\;\alpha} = R^{ab\;\beta}_{c\;\;\alpha} \;. \tag{8.92}$$

Due to these normalizations, the quantum dimensions do not appear in the conventional pentagon and hexagon equations, but they do in the generating axioms in Eqs. (8.79) and (8.82). Without the surgery invariance of $\underline{f}$, 3D atTS correspond to unitary braided fusion categories instead of UMTCs. Like for the non-braided unitary fusion categories discussed in Section 4.14, the auxiliary generators G and H seem to be related to an explicit pivotal structure, and κ is the Frobenius-Schur indicator.

In alignment to our approach is the so-called *Reshetikhin-Turaev TQFT* [121, 131], which constructs 3-manifold invariants from a UMTC. These invariants the same as the atTS tensors associated to $\underline{a}$ 3-manifolds with empty space regions/string network. By folklore, Reshetikhin-Turaev TQFT is known to be a 3-2-1-extended TQFT, which fits our proposition that nD atTS models capture the essence of nD TQFT extended down to the circle. An attempt to make the interpretation of Reshetikhin-Turaev TQFT as a 3-2-1-extended TQFT precise, and a derivation of generators and relations thereof can be found in Ref. [16]. A classification of UMTCs with 4 or less isomorphism classes of simple objects, and their explicit F and R-symbols, can be found in Ref. [122]. In the physics literature, it was first proposed in Appendix E of Ref. [96] that UMTCs correspond to *anyon theories*, which describe topological phases in $2+1$ dimensions to a great extent.

8.5 atTS for boundaries and defects

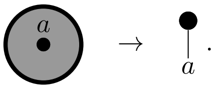
In this section, we will show how to introduce boundaries and lower-dimensional defects into the atTS description of topological phases.

8.5.1 Boundaries

As a warmup, we will consider the slightly trivial case of equipping the 2D atTS with a boundary, which we will refer to as the *2D boundary atTS*. Following the general prescription from Section 8.1, the extended manifold type for 2D tTS in Eq. (4.157) yields the internal regions for the 2D boundary atTS. We consider the case where both the bulk and the boundary are not fully extended, such that there are no space regions for the boundary. So the only space region is that of the 2D atTS in Eq. (8.21),

$$\underline{a} : (2, \varnothing) \;, \quad \underline{b} : (1, \bullet) \;, \quad c : (0, \bigcirc) \;. \tag{8.93}$$

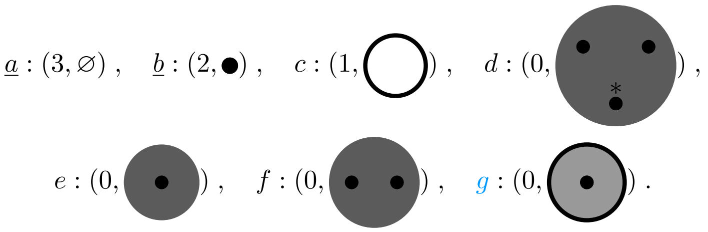
There is only one additional generating tensor,



(8.94)

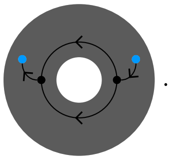
By gluing with this manifold, we can introduce boundary circles, and generate an arbitrary 2-manifold with boundary. There are *no* additional generating gluing axioms. In other words, for a 2D-boundary-atTS model, the tensor in Eq. (8.94) can be completely arbitrary. Note that if we add the robustness axiom to the 2D boundary atTS, it becomes immediately trivial.

Let us now discuss how to equip the 3D atTS with boundaries, yielding a tensor scheme that we will refer to as *3D boundary atTS*. As usual, the internal regions consist of a 3-dimensional bulk and its boundary, and are the same as for the 3D boundary tTS in Eq. (5.3). We again consider the case where the boundary is not fully extended, and thus we have additional boundary space 0-regions. The links of these space 0-regions are arbitrary extended 2-manifolds. A set of generating links is given by the generating tensors of the 2D boundary atTS in Eqs. (8.24) and (8.94). The full extended manifold type is thus given by



(8.95)

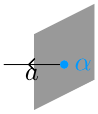
In the collapsed picture, extended manifolds are thus 3-manifolds (a) with boundary (b), with a network of strings (c) in the interior. These strings can terminate on the boundary at a set of points, which form the region g. To make drawings more readable, we will draw the space boundary region g in blue. An example for an extended manifold drawing is then,



(8.96)

This shows a solid 3-torus (where only the 2-dimensional boundary is shaded), with two d fusion vertices inside the bulk, and two g vertices where strings terminate on the boundary.

Let us here directly describe the block-diagonal form of the 3D boundary atTS. The block-diagonal form assigns a tensor to every extended manifold of the above type. As for the bulk, these tensors have one label for every c segment, and one index for every point of the 0-regions d, e, f, and g. The bond dimension of the g boundary-vertex indices is some positive integer m^a depending on the label a of the attached bulk string,

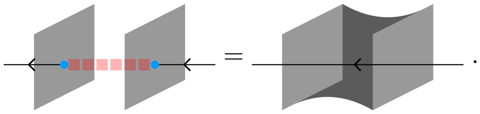


$$: \quad 0 \leq \alpha < m^a \; . \tag{8.97}$$

m^a needs to be an *algebra element* of the fusion ring given by N,

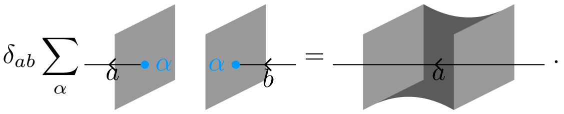
$$\sum_c N_c^{ab} m^c = m^a m^b \; . \tag{8.98}$$

Since we have an additional space 0-region g, there is an additional gluing operation,



(8.99)

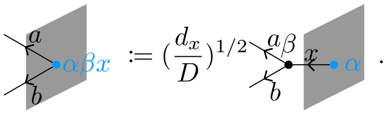
The effect of this gluing on the overall topology is a 1-handle attachment. The gluing axiom for this type of gluing is given by



(8.100)

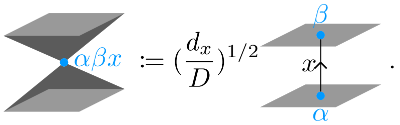
In addition, we should also add a robustness axiom for the boundary. This is implemented by demanding that the internal boundary region $\underline{b}$ is invariant under 1-handle attachments.

Let us now look at the generating tensors and axioms for the 3D boundary atTS. We start by demonstrating how the links in Eq. (8.95) generate all other links. As an example, boundary points with two attached strings can be emulated via a 3-valent bulk vertex,



(8.101)

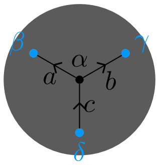
Instead of gluing a pair of 2-valent boundary vertices, we first glue the corresponding pair of boundary vertices, then the pair of bulk vertices, and then apply 1-surgery to the resulting x loop. As a second example, singular boundary fusion vertices with annulus link (but no attached bulk strings) can be emulated by two boundary fusion vertices connected via a bulk string,



(8.102)

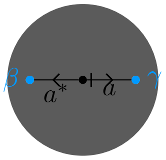
Gluing two of the singular vertices shown on the left can be emulated by gluing the corresponding two pairs of ordinary g boundary vertices, and then performing 1-surgery at the resulting x loop.

With this, we are now ready to look at the generating tensors. In fact, there is only one additional generating tensor, namely a 3-ball with three boundary vertices, each connected to one common 3-valent bulk vertex,



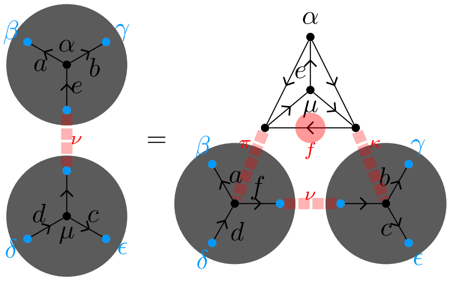
$$\rightarrow \quad M^{ab}{}^{\alpha\delta c}_{\beta\gamma}\;. \tag{8.103}$$

Note that this is just the sphere-with-three-points generator in Eq. (8.24), where we filled the sphere with a 3-ball, and connected every boundary vertex to a bulk vertex in the center. However, to make the generating axioms more concise, we also add the following auxiliary generator,



$$\rightarrow \quad N^a{}_{\gamma\beta}\;. \tag{8.104}$$

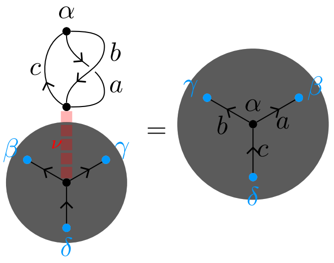
Next, let us look at the generating axioms. The new generating axioms look like the ones of the 2D atTS in Section 8.3, if we restrict only to the boundary regions $\underline{b}$ and g. To obtain a 3D-boundary-atTS axiom from a 2D-atTS axiom, we 1) make each sphere into a ball with a bulk vertex in the center, and 2) glue one side with the 3D generating tensor corresponding to the 2D generating axiom. For example, the associativity axiom in Eq. (8.26) yields



(8.105)

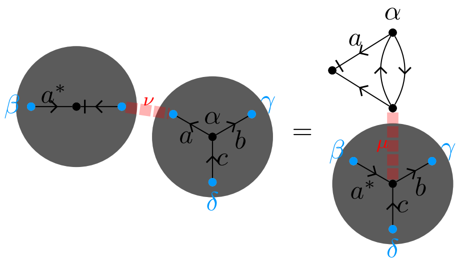
$$\rightarrow \quad \sum_{\nu} M^{ab}{}^{e\nu\alpha}_{\beta\gamma} M^{ec}{}^{d\delta\mu}_{\nu\epsilon} = \sum_{f,\nu,\pi,\kappa} \frac{d_f}{D} M^{af}{}^{d\delta\pi}_{\beta\nu} M^{bc}{}^{f\nu\kappa}_{\gamma\epsilon} F^{cba}_{d}{}^{e\mu\alpha}_{f\kappa\pi}\;.$$

The commutativity axiom in Eq. (8.25) yields



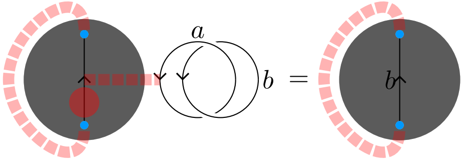
$$\rightarrow \quad \sum_{\nu} R^{ba\;\alpha}_{c\;\;\nu} M^{ab\;c\delta\nu}_{\;\;\;\beta\gamma} = M^{ba\;c\delta\alpha}_{\;\;\;\gamma\beta} \; . \tag{8.106}$$

The axiom in Eq. (8.29) yields



$$\rightarrow \quad \sum_{\nu} M^{ab\;c\delta\alpha}_{\;\;\;\nu\gamma} \overline{N^{a}{}_{\nu\beta}} = \sum_{\mu} \overline{M^{a^*c\;b\gamma\mu}_{\;\;\;\;\beta\delta}} G^{cb\mu\alpha}_{a} \; . \tag{8.107}$$

Finally, there is also one additional axiom that does not correspond to any of the 2D axioms. It is given by



$$\rightarrow \quad \sum_{a} m_a S_{ab} = m_b \; . \tag{8.108}$$

On the right-hand side, we have a solid torus with a non-contractible loop inside. On the left-hand side, we have the same solid torus additionally glue the loop with one loop of the S-matrix diagram. This results in another solid torus with a non-contractible loop.

Let us briefly review how our approach to boundaries via atTS models relates to the literature. Topological boundaries of phases, described by a UMTC via their anyon content, are known to be described by *Lagrangian algebra objects* internal to the UMTC. In terms of category theory, an algebra object is given by an object m, together with a morphism

$$M \in \mathrm{hom}(m \otimes m, m) \; . \tag{8.109}$$

The isomorphism class of m is described by assigning a non-negative integer multiplicity m^i to each simple object i, and the morphism M is described by an M-symbol. The numbers m^i are precisely the multiplicities of our g vertices, and the M symbol is our generating tensor M in Eq. (8.103). The morphism M has to fulfill the associativity axiom inside the UMTC, which is precisely our generating axiom in Eq. (8.105). In fact, we need a *Frobenius algebra object*, which corresponds to the existence of our generating tensor N in Eq. (8.104) fulfilling the axiom in Eq. (8.107). The M-symbol and associativity condition have been described explicitly in Ref. [43]. A Frobenius algebra object is *Lagrangian* if it is maximal in the sense that there is no simple object i such that

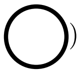
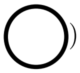
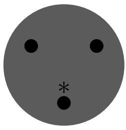
$$m_i = 0 \;, \quad \theta_i = 0 \;, \quad S_{ij} = 0 \quad \forall j : m_j > 0 \;. \tag{8.110}$$

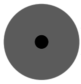
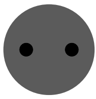
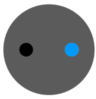
The move in Eq. (8.108) ensures this elegantly.

In the physics literature, anyons i with $m_i > 0$ are said to *condense* at the boundary. There, the atTS description of gapped boundaries is most established in the case where the fusion rules of the UMTC are an abelian group. In this case, UMTCs are in one-to-one correspondence with *metric groups*, and determined by the topological twist of every anyon. Lagrangian algebra objects are then in one-to-one correspondence with *Lagrangian subgroups*, that is, subgroups H of the fusion groups for which all the topological twists are 1. m^i is then determined from H by $m^i = \delta_{i\in H}$, and there is a unique M-symbol, such that the discussion of gapped boundaries greatly simplifies. In this context, Ref. [106] provides a physics argument, why the subgroup should indeed be Lagrangian, which is also known as the *principle of remote detectability*. In contrast, in our formalism, the Lagrangian property is purely a consequence of topological invariance.

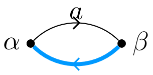
8.5.2 Defects

We can also enrich our 3D atTS with lower-dimensional defects. Let us start by considering the paradigmatic kind of such defects, namely anyons. That is, we will give a higher-level atTS description of hypothetical bulk models with one specific type of anyon worldlines. The funny thing is that the labeled 3D-atTS extended semicellulations already correspond to pattern of anyon worldlines. Hence, all we need to do is to set some of the segment labels to the desired anyon. However, this approach does not generalize to higher dimensions or other types of defects, so we will discuss a more formal equivalent way of including anyons. Namely, we consider the *3D anyon atTS*, which is of the following type,

$\underline{a} : (3, \varnothing)\;,\quad \underline{b} : (1,$  $)\;,\quad c : (1,$  $)\;,\quad d : (0,$  $)\;,$

$e : (0,$  $)\;,\quad f : (0,$  $)\;,\quad g : (0,$  $)\;.$ (8.111)

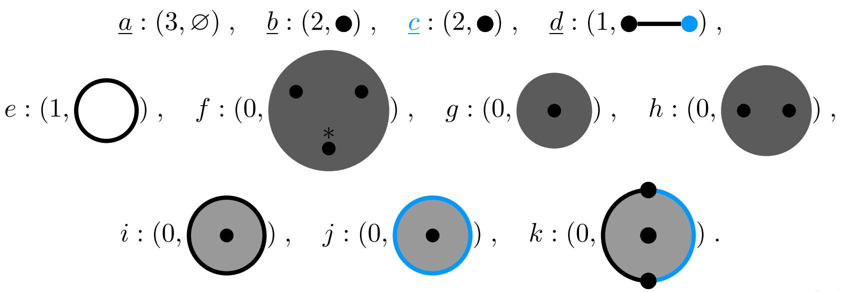
The extended manifolds in the collapsed picture are 3-manifolds with embedded string-nets, and separate $\underline{b}$ segments that do not carry any label. The b segments join c segments at g vertices. There is one index at every g vertex whose dimension is A^a depending on the label a of the adjacent c segment. There are no additional generating tensors or axioms. By the full support convention, we have



$$\to \quad \delta_{\alpha,\beta} \; . \tag{8.112}$$

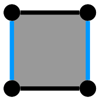
We can use the tensor above to insert $\underline{b}$-segments anywhere. Physically, the anyons described in this way are non-robust composite anyons, and A^a is the multiplicity with which the robust anyon a occurs as part of the composite anyon. If we additionally add robustness of the anyon as a move, then this forces $A^a = \delta_{a,i}$ for some i, that is, the composite anyon is the robust anyon i. In this case, extended semi-cellulations of the type above are the same as 3D-atTS extended semi-cellulations where all $\underline{b}$-segments are c-segments with label i.

Next, let us consider corners in models with two different boundaries. Here, the *3D corner atTS* is of the following type,



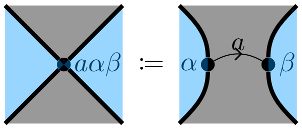
(8.113)

Extended manifolds in the collapsed picture are 3-manifolds with two sorts of boundary ($\underline{b}$ and $\underline{c}$), and corners ($\underline{d}$) separating the two boundaries. In the bulk, we have a network of strings that can terminate on either boundary. In addition, bulk strings can terminate directly on the corners at points of the space region k. The k vertices carry an index of dimension C^a depending on the label a of the adjacent e segment. Physically, C^a is the multiplicity with which the anyon a condenses on the corner. Using the k vertices, we can emulate any other vertex link. For example, consider the following link consisting only of internal regions,



$$, \tag{8.114}$$

corresponding to points in the boundary where four corners meet. These vertices can be emulated as follows,



$$. \tag{8.115}$$

On the right-hand side, the e string with label a connects the two k vertices through the bulk behind. Gluing a pair of the vertices on the left is emulated by first gluing the two pairs of k vertices, then applying 1-surgery to the resulting e loop (labeled a), and then applying an inverse 1-handle attachment corresponding to the robustness of the $\underline{b}$ boundary. We will not explicitly work out the generating tensors and axioms here.

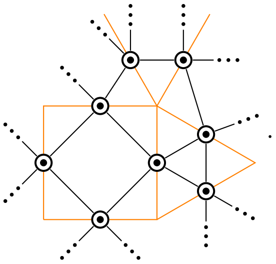
Chapter 9

Non-simplicial fixed-point ansatzes

So far, the tTS that we have used to describe topological phases via fixed-point models had a fixed "standard" geometric structure. Namely, we associate one tensor variable to each n-cell/n-simplex of an n-dimensional (extended) cellulation/triangulation. Topological invariance was then implemented by recellulations, in particular Pachner moves for triangulations. In this section, we generalize this notion of tTS, by allowing the tensor-network diagrams to be any sort of local combinatorial representation of a topological manifold. Equivalently, we still assign tensor networks to cellulations, but the graph of the tensor network does no longer need to look like the Poincaré dual cellulation. We illustrate this by giving two examples for such "non-standard" tTS.

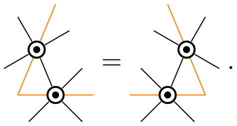
9.1 2D edge tTS

As a first toy example in $1+1$ dimensions, we introduce a tTS that is a "non-standard" alternative to the 2D tTS. We will refer to this tTS as *2D edge tTS*, since not the triangles but the edges are represented by tensors. We start by defining the 2D edge tTS, and then show its equivalence to the 2D (standard) tTS via weakly invertible TS mappings. The 2D edge tTS associates a copy of a 4-index tensor variable to every edge of a 2-dimensional cellulation. Two edge tensors share a common bond if they are adjacent to both a common vertex and a common face. This prescription also works for cellulations with arbitrary n-gons as faces instead of just triangles, for example,



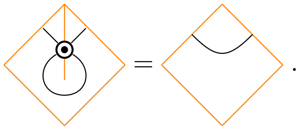
(9.1)

As usual, the orange lines represent the cellulation in the background, whereas the black diagram is the tensor network. For simplicity, we refrain from distinguishing indices by markings for this toy example. There are 3 different moves. The first move consists in taking the endpoint of one edge and moving it along another edge,



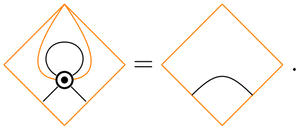
(9.2)

As shown, this corresponds to a tensor-network equation with two tensors on each side. The second move consists in removing a "dangling" edge with a vertex that is only adjacent to this edge,



(9.3)

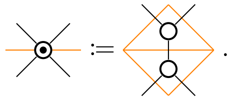
The corresponding tensor network on the left-hand side consists of one tensor two of whose indices have been contracted. On the right-hand side, the tensor-network only consists of a "free bond" that corresponds to the identity matrix with its two open indices. Note that we depicted this move inside a 4-gon cell, but the same move can be applied to an n-gon since the tensors at the outer edges do not participate in the tensor-network equation anyways. The third move is Poincaré dual to the second and removes a looping edge whose endpoints coincide,



(9.4)

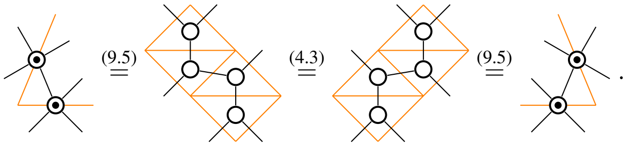
In terms of tensor-network diagrams it looks the same as the second move, which is only the case because we are not distinguishing between indices.

After having defined the 2D edge tTS, we will now argue that it is equivalent to the 2D tTS. We will define the equivalence via two TS mappings. Namely, we construct one mapping from the 2D edge tTS to the 2D tTS, and one mapping from the 2D tTS to the 2D edge tTS, such that the two mappings are weak inverses of another, c.f. Section 4.7. Let us start with the mapping from the 2D edge tTS to the 2D tTS. We can map a cellulation to a triangulation by subdividing each face into triangles in a “pizza-like” manner, with a new vertex in the middle. In doing so, every edge gives rise to two triangles,



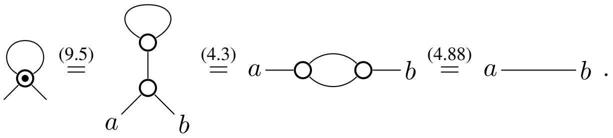
(9.5)

As depicted, this mapping associates to the 2D edge tTS variable a tensor network consisting of two 2D tTS tensors. For this to define a TS mapping, we need to derive the 2D-edge-tTS moves from the 2D-tTS moves. For example, the move in Eq. (9.2) can be derived by



(9.6)

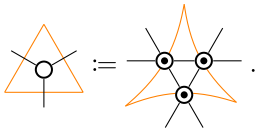
Or, Eq. 9.4 can be derived by



(9.7)

Thus, under the mapping defined in Eq. (9.5) a model of the 2D tTS is mapped to a model of the 2D edge tTS.

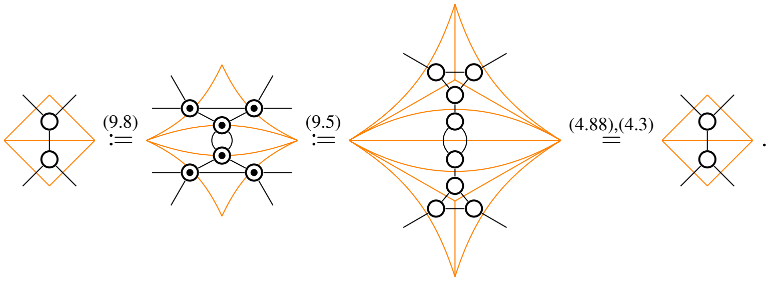
Let us now give the opposite mapping from the 2D tTS to the edge tTS. A triangulation can be turned into a cellulation by making a second copy of every edge such that two copies enclose a new 2-gon face. Every triangle then yields three edges,



(9.8)

The Pachner moves in Eq. (4.3) can be derived using the move in Eq. (9.2).

Last, we want to show that the two TS mappings are weak inverses of another. Intuitively, this is evident from the fact that the two mappings were defined by refinements of cellulations or triangulations. Formally, let us apply both mapping to some patch of a 2D-tTS tensor network,

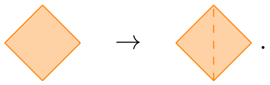


(9.9)

In the last step, to deform the twice-mapped diagram back to the original diagram, we apply one triangle cancellation move at each edge of the original triangulation, and one 1-3 Pachner move at each triangle of the original triangulation. Applying both mappings starting from an 2D-edge-tTS diagram also yields a refined cellulation that can be mapped back to the original cellulation using the moves of the 2D edge tTS.

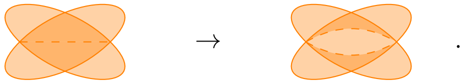
9.2 3D face-edge tTS

As a next example, we will discuss a tTS in $2+1$ dimensions that is an alternative to the 3D (standard) tTS. We will refer to this tTS as the *3D face-edge tTS*, since it associates tensors not to the tetrahedra of a triangulation but to the faces and edges of a 3-dimensional cellulation. It is convenient to restrict ourselves to 3-cellulations where every face is either a triangle or a 2-gon, and that every edge is 3-valent or 2-valent (i.e., it is adjacent to three or two faces). Every cell complex can be brought into this form by simple deformations, for example, a 4-gon can be split into two triangles with a 2-valent edge in between as



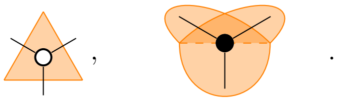
(9.10)

Or, dually, a 4-valent edge can be split into two 3-valent edges with a 2-gon face in between,



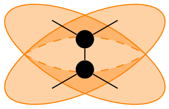
(9.11)

Note that when drawing a standalone edge, we include a little 2-gon "stub" face for every face adjacent to this edge. There is one copy of a 3-index tensor variable associated to every triangle and one copy of a different 3-index tensor variable associated to every 3-valent edge. Note that this means that a model of the 3D face-edge tTS is given by two potentially different 3-index tensors. At every pair of adjacent triangle and 3-valent edge, there is a bond between the two corresponding tensor indices. Since the face and edge tensors are different, we use two different shapes to represent them, namely an empty and a full circle,



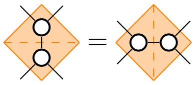
(9.12)

Here, we indicate the background cellulation in orange. On the right-hand side, we indicate that the edge (which itself is dashed) has three adjacent faces by including three 2-gon stub faces as discussed earlier. We will add index markings and talk about orientation later. The 2-gons (including the 2-gon stub faces) and 2-valent edges are not explicitly represented by tensors. Instead, the two edges adjacent to a 2-gon, and likewise the two faces adjacent to a 2-valent edge are directly connected by a bond. For example, two 3-valent edges separated by a 2-gon are represented as



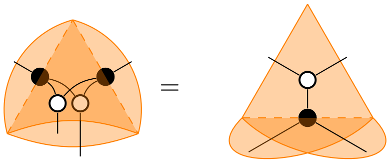
. (9.13)

Let us next give the moves of the tTS, which can be divided into 3 groups. First, there are moves involving only triangles separated by 2-valent edges, which equal the 2-dimensional Pachner moves for the face tensors only, namely



, (9.14)

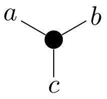
and the same for the 1-3 Pachner move. Second, there are moves involving only 3-valent edges separated by 2-gon faces. In terms of cell complexes, these moves are Poincaré dual to the moves above. In terms of tensor-network diagrams they look the same apart from that we have to use full circles instead of empty circles. Finally, the *corner fusion move* involves both face and edge tensors. It merges two triangles with two shared 3-valent edges into a single triangle,



. (9.15)

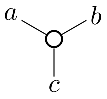
On the left-hand side, there are two triangles adjacent to the same two dashed 3-valent edges, one in the front and one at the back. On the right-hand side, there is only one triangle and one edge at the bottom. This edge is adjacent to two further faces as indicated by two 2-gon stub faces. Note that the non-dashed orange lines on both sides form the same 2-cellulation.

Despite omitting technical details so far, we can already give a meaningful model of this tTS, which happens to be the most well-known model of intrinsic topological order: The *toric code* [95] can be represented as a tTS model in the following way. The bond dimension is 2, and accordingly we will label the basis configurations by $0, 1 \in \mathbb{Z}_2$. The edge tensor is given by a δ-tensor,



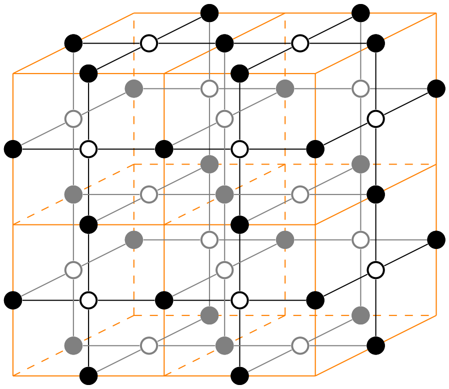
$$= \begin{cases} 1 & \text{if } a = b = c \\ 0 & \text{otherwise} \end{cases} \quad . \tag{9.16}$$

The face tensor is a tensor which ensures that the global $\mathbb{Z}_2$ parity is even,

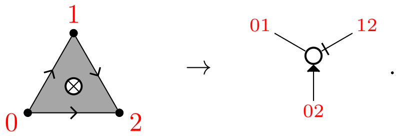


$$= \begin{cases} 1 & \text{if } a + b + c = 0 \mod 2 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{9.17}$$

It is easy to see that each tensor satisfies the 2-2 and 1-3 Pachner move, and both tensors together fulfil the move in Eq. (9.15). In fact, there is a global prefactor of $1/2$ missing in the 1-3 Pachner move for face tensors, which is not crucial, but will be fixed in the following discussion. For the toric code, it is particularly easy to define face and edge tensors with arbitrarily many indices: The δ-tensor just forces all indices to be equal, and the $\mathbb{Z}_2$-tensor is 1 if all index labels sum to $0 \bmod 2$, and zero otherwise. With this, we can define the path integral on arbitrary cellulations. For example, on a patch of a cubic lattice (in orange), the tensor network (in black/gray) looks like:

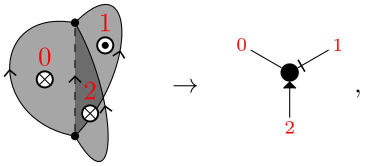

. (9.18)

Let us now take care of the technical details that we have neglected so far, namely (1) distinguishing the tensor indices, (2) introducing orientation and Hermiticity, and (3) adding weight matrices. In order to distinguish indices, we add directions to all edges that are non-cyclic around every triangle, just like a branching structure apart from that we are not working with triangulations. This then allows us to distinguish the three indices of the triangle tensor variable,



. (9.19)

In order to distinguish the indices of the edge tensors, we also need to assign non-cyclic edge directions to all edges of the Poincaré dual lattice. This allows us to label the three faces adjacent to an edge (corresponding to the three indices of the edge tensor) by 0, 1, and 2,



, (9.20)

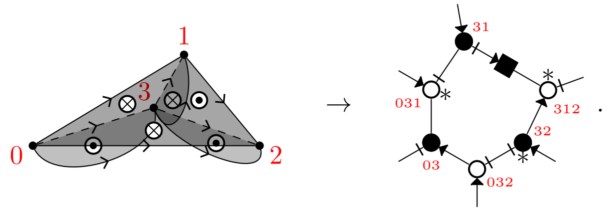
We indicate the dual direction by putting a circle with a cross or a dot on the respective face. A dot means that the dual direction is pointing towards the viewer, and a cross means that it is pointing into the paper.

The dual directions also fulfil a second task, namely to determine whether a triangle is positively or negatively oriented: A triangle is positive if its 01 edge points clockwise relative to the underlying orientation when looking along its dual direction. So in the triangle tensor in Eq. (9.19), where the dual orientation points into the paper, is positive. Dually, an edge is positively oriented if the direction of its adjacent 0 face is counterclockwise when looking along the direction of the edge. Due to Hermiticity, negatively oriented triangles or edges are assigned the complex conjugate of the triangle or edge tensor. Note that what we call positive or negative is only a convention. We have chosen this such that every bond in our diagrams carries exactly one arrow marking if the tensors it connects are either both complex conjugated or both not. If only one of the tensors is complex conjugated, the bond will have either no or two arrow markings.

Next, let us discuss the weight matrices. There are 4 such weight matrices called *corner weights*, which we label 00, 01, 10, and 11,

00 : ─□◂─ ,   10 : ─⊡◂─ ,   01 : ─■◂─ ,   11 : ─◘◂─ .   (9.21)

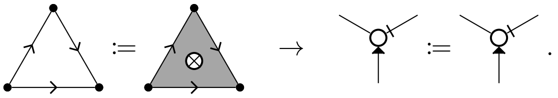
A *corner* is a pair of a volume and an adjacent vertex. Imagine a corner weight inserted at a bond that connects a face and an edge. Let us say that the ij corner weight ($i, j \in \{0,1\}$) contributes to the corner formed by the i vertex of the edge and the j volume of the face. Here, the dual direction of the face is such that it points from what we call the 0 volume to what we call the 1 volume. Now, we require that in a valid tensor-network diagram representing a cellulation, every corner must have exactly one weight contributing to it. Note that every corner corresponds to a loop of bonds in the diagram alternating between face and edge tensors. As an example, consider the following corner between the 3 vertex and the volume above, enclosed by three faces and three edges,



(9.22)

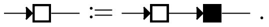
As shown, a 01 corner weight between the 31 edge and the 312 face contributes to this corner. Alternatively, we could for example have inserted a 10 corner weight between the 03 edge and the 032 face.

The index markings, orientation, and edge weights also need to incorporated into the moves of the 3D face-edge tTS. To consisely state these moves, consider the following *suspension mapping* from the 2D tTS to the 3D face-edge tTS. Topologically, it corresponds to taking the suspension of a 2-manifold, which we also discussed in Section 4.14. Combinatorially, we take a 2-dimensional cellulation and map it to a 3-dimensional cellulation by mapping every triangle to a triangle, and every edge to a 2-valent edge. The so-obtained 3-cellulation is degenerate in that it contains only two volumes x and y whose boundary consists of the complete 2-triangulation, and which therefore may not have the topology of a 3-ball. Formally, the mapping looks trivial since we are using the same shapes for the 2D-tTS tensor and face tensors of our 3D face-edge tTS,



(9.23)

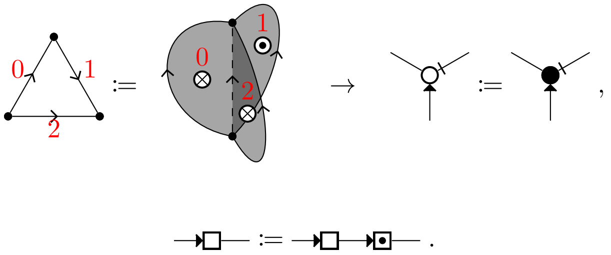
The only non-trivial aspect is the mapping of the vertex-weight. For example, the 0 vertex weight of the 2D tTS is mapped to a product of the 00 and 01 corner weights of the 3D face-edge tTS,



(9.24)

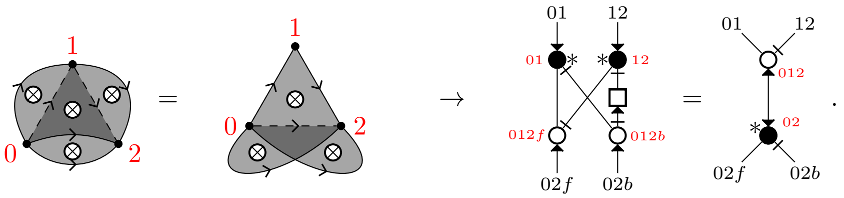
This is because a vertex of the 2-triangulation corresponds to two corners (one with the x volume and one with the y volume) of the 3-cellulation. Under this mapping, all moves of the (simplified) 2D tTS yield moves of the 3D face-edge tTS.

The Poicaré dual of the suspension mapping is the *dual suspension mapping*. This adds two vertices x and y, maps every triangle to a 3-valent edge connecting x and y, every edge to a perpendicular 2-gon face, and every l-valent vertex to a banana-like volume enclosed by l 2-gon faces, as shown for $n = 3$ in Eq. (4.227). So the resulting cellulation consists only of 3-valent edges and 2-gon faces. In terms of tensor-network diagrams, the mapping is given by



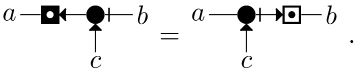
(9.25)

As shown, the 0 vertex weight is mapped to a product of the 00 and 10 corner weights, as every vertex yields two corners of the banana-volume with either x or y. Again, we take all the 2D-tTS moves under this mapping as moves of the 3D face-edge tTS. Next, we also have to equip the corner fusion move from Eq. (9.15) with differenc choices of (dual) edge orientations and corner weights. One such choice is given by



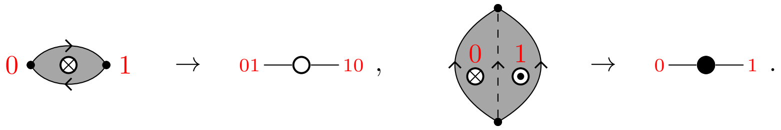
(9.26)

We impose this move for all edge directions and dual edge directions. Finally, there are weight commutation moves. As usual, all corner weights are Hermitian and commute among each other. Further, the corner weights can be moved from one side of a triangle or 3-valent edge to another, for example,



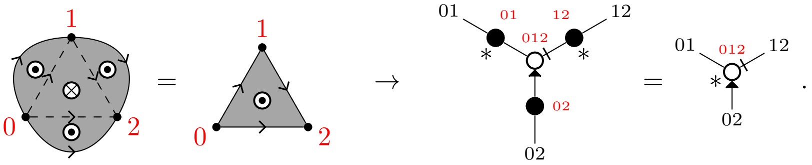
(9.27)

Instead of imposing all of the above moves for all different choices of edge directions and weight matrices, one can define an equivalent simplified tTS. This simplified tTS only contains one version of the corner fusion move, namely the one depicted in Eq. (9.26). Apart from that, it is based on two copies of the tensor variables and moves of the simplified 2D tTS in Section 4.7, via the (dual) suspension mapping. Explicitly, we introduce an 2-gon face and a 2-valent edge with cyclic (dual) edge directions,



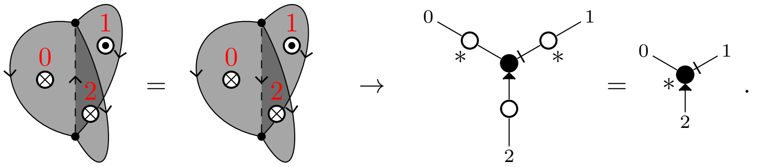
(9.28)

In addition to the corner fusion move and the moves inherited from the 2D tTS, there are moves that effectively change the orientation of edges and dual edges. For example the dual orientation of a triangle can be changed by surrounding it with 2-valent edges, which we call the *dual triangle flip move*,



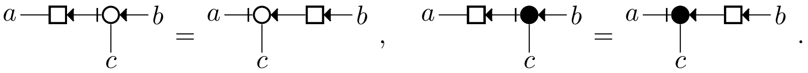
(9.29)

Dually, we can flip the direction of an edge by surrounding it with cyclic 2-gons, which we call the *edge flip move*,



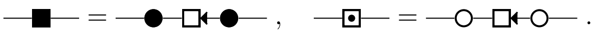
(9.30)

We do *not* include the weight commutation moves of the 2D tTS via the (dual) suspension mapping. Instead, we impose the following two weight commutation moves for only the 00 corner weight,



(9.31)

In addition, we use moves defining the 01 and 10 corner weights from the 00 corner weight, by "conjugating" them with a cyclic 2-valent edge or 2-gon face,

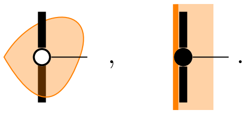


(9.32)

So in total, the tensor variables of the simplified tTS are given by the triangle, 3-valent edge, 2-gon and 2-valent edge in Eqs. (9.19), (9.20), and (9.28), the 00, 01, and 10 corner weights in Eq. (9.21). The moves are given by two copies of the 2D-tTS moves in Eqs. (4.12), (4.88), (4.94), (4.96), and (4.97) under the (dual) suspension mappings of Eqs. (9.23) and (9.25), as well as Eqs. (9.26), (9.29), (9.30), (9.31), and (9.32).

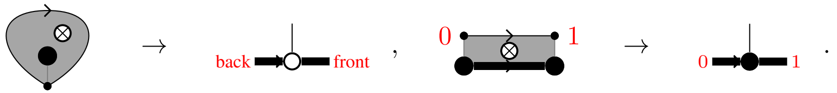
9.3 Anyons for the 3D face-edge tTS

In this section, we will show how to implement anyons inside the 3D face-edge tTS, by constructing a *3D anyon face-edge tTS*. Anyon worldlines are actually ribbons, that is, embedded 1-manifolds with a smooth choice of normal vector at every point. Inside a spacetime 3-cellulation, such a ribbon can be implemented by a sequence of volume-vertex pairs, such that: For every pair of two consecutive pairs, either the volumes are the same and the vertices differ by an edge, or the vertices are the same and the volumes are adjacent to the same face. Intuitively speaking, such a ribbon is a sequence of edges together with a sequence of faces (which are edges in the dual lattice) right next to it. In the according tensor networks, we replace all the tensors on the faces and edges of the anyon worldline by a different *anyon tensor variable*. These anyon tensors have two more indices each, which are contracted with the preceeding and succeeding anyon tensor in the worldline. More precisely, it suffices to equip 1-gon faces and 1-valent edges with anyons,



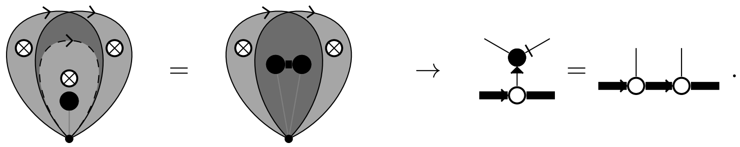
(9.33)

We will call these new tensor variables the *anyon face* and the *anyon edge* tensors. Taking into account (dual) edge directions, the tensors are defined as



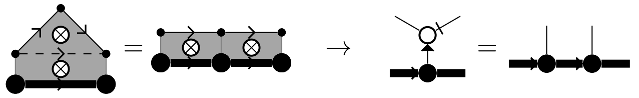
(9.34)

Let us now discuss the moves that allow us to change the cellulation of the anyon worldline. First of all, two consecutive anyon face tensors can be shrinked to a single anyon face tensor attached to a 3-valent edge,



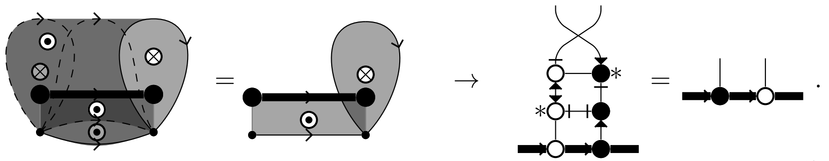
(9.35)

In other words, the anyon face tensor defines a boundary of the 2D tTS model obtained from the dual suspension mapping. Second, two consecutive anyon edge tensors in an anyon worldline can be shrinked to a single anyon edge attached to a triangle,



(9.36)

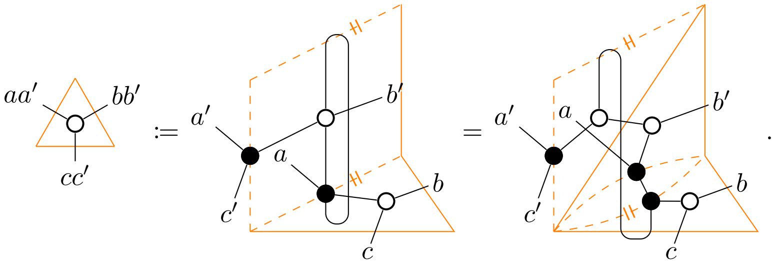
In other words, the edge anyon tensors define a boundary of the 2D tTS model obtained from the suspension mapping. Third, we need a move that exchanges an anyon face and anyon edge tensor, namely



(9.37)

On the left-hand side, there is an anyon face on the left with an anyon edge attached to its right. Around the anyon edge wraps a tube which is rectangle that is self-glued at an edge at the bottom. The left open circle of the tube is attached to the anyon face. The edge inside the bottom of the tube is 4-valent, and additionally adjacent to the anyon edge at the interior of the tube. The tube rectangle may be divided into two triangles, and the 4-valent edge into two 3-valent edges. On the right-hand side, there is only an anyon edge on the left with an anyon face attached to its right.

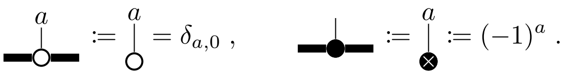
Similar to anyons in the 3D tTS discussed in Section 6.2, anyons of the 3D face-edge tTS as presented above are in one-to-one correspondence with boundaries of a compactified $1+1$-dimensional model. Topologically, the underlying compactification mapping is the same as in Section 6.2, mapping from 2-manifolds to 3-manifolds by taking the cartesian product with a circle. Microscopically, the mapping from the 2D tTS to the 3D face-edge tTS is slightly different: Each 2D-tTS triangle becomes a triangle of the 3D face-edge tTS, and its 0-vertex becomes a 3-valent edge. We take the product of the 01 edge of the triangle with a looping edge, yieling a 4-gon face as well turning the edge into a 4-valent edge. These can be split up into two triangles and two 3-valent edges, respectively,



(9.38)

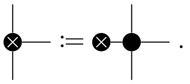
The two edges marked with a double-tick are identified, such that the vertical edges wrap around the non-contractable loop.

Let us now give two simple examples for model of the 3D anyon face-edge tTS, where the underlying bulk is given by the toric code in Eq. (9.17). For both examples, the indices connecting the anyon tensors will be trivial, that is, have bond dimension 1. The first model is given by



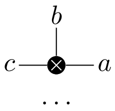
(9.39)

The anyon face tensor is just the same as the 1-gon face tensor without anyon, but the anyon edge tensor has a minus sign compared to the 1-gon edge tensor without anyon. It is easy to see that the moves in Eqs. (9.35), (9.36), (9.37) hold for this assignment. As mentioned earlier, we can define arbitrary n-valent anyon edges from only the 1-valent anyon edge above. For the present model, this is especially simple since the anyon indices are trivial and we do not need to distinguish the indices of the edge tensors in the bulk. So, for example, a 3-valent anyon edge tensor can be obtained by contracting one index of the 4-valent edge with the 1-valent anyon edge,



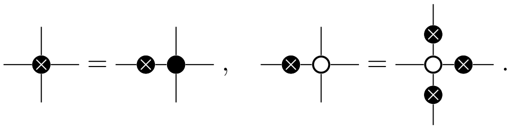
(9.40)

In general, the n-valent anyon edge tensor obtained this way is given by



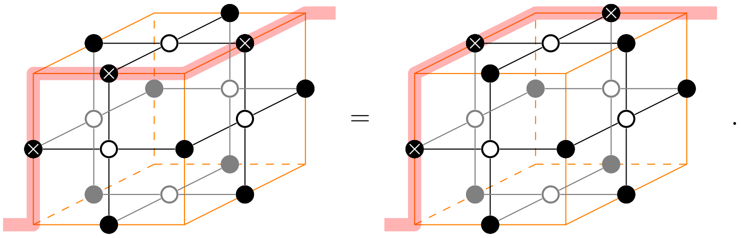
$$= \begin{cases} 1 & \text{if } a = b = c = \ldots = 0 \\ -1 & \text{if } a = b = c = \ldots = 1 \\ 0 & \text{otherwise} \end{cases} . \tag{9.41}$$

Note that this is possible because in this model it does not matter at which index we contract with the face anyon tensor. So for this specific model, anyon worldlines are just sequences of edges in the cellulation. They are implemented in the tensor-network path integral by simply replacing every n-valent edge tensor on the worldline by the n-valent anyon edge tensor. The moves in Eqs. (9.35), (9.36), and (9.37) then reduce to equations like the following,



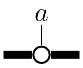
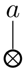
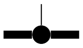
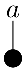
(9.42)

These equations can be used to freely move the worldlines through the bulk, for example,

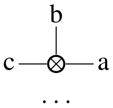


(9.43)

The second anyon-tTS model we want to consider is dual to the first model,

 :=  := $\delta_{a,1}$,  :=  . (9.44)

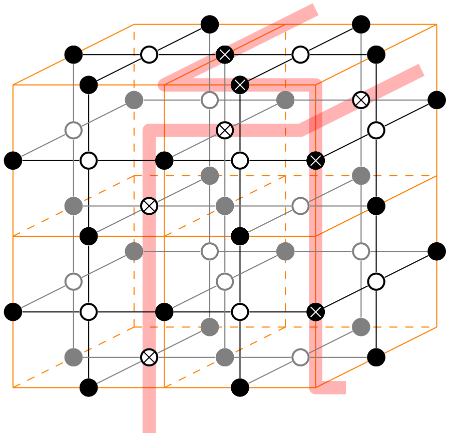
The anyon edge tensor is the same as the 1-valent bulk edge tensor without anyons, but the anyon face tensor equals the 1 basis vector instead of the 0 basis vector. Again, we can easily define anyon n-gon anyon tensors from the 1-gon anyon tensor,



$$= \begin{cases} 0 & \text{if } a+b+c+\ldots = 0 \mod 2 \\ 1 & \text{otherwise} \end{cases} . \tag{9.45}$$

Anyon worldlines are then closed paths of edges in the Poincaré dual lattice, and the implemented in the path integral by replacing every n-gon face tensor on the worldline by the n-gon anyon face tensor.

An example for a spacetime configuration containing anyon worldlines of both models inside a section of a cubic lattice (with worldlines marked in semi-transparent red) is:



. (9.46)

Physically, the first anyon model corresponds to what is known as the e anyons in the toric code, and the second model are the m anyons. The fact that e and m anyons are just paths of edges in the (Poincaré dual) cellulation is related to the fact that both anyons are bosons. f anyons, which are the combination of an e and an m anyon, are fermions, and live on ribbons.

9.4 Face-edge tTS and weak Hopf algebras

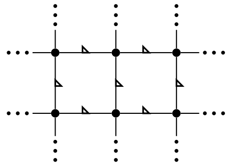
Just like the 2D tTS or 2D atTS, the 3D face-edge tTS is closely related to a well-known algebraic structure, namely *weak Hopf algebras*. To this end, recall the definitions of (associative) (co-)algebras from Section 4.12. A *weak Hopf algebra* consists of 1) a unital associative algebra, 2) a co-algebra and 3) a linear map called the *antipode*. The algebra and co-algebra interact with each other through a rather long list of axioms.

The most important axiom is the *bi-algebra law*. There is a TS mapping from the weak-Hopf-algebra TS to the 3D face-edge tTS, which maps the algebra multiplication to the face tensor in Eq. (9.19), and the coalgebra multiplication to the edge tensor in Eq. (9.20). The bi-algebra law under this mapping directly becomes the corner-fusion move in Eq. (9.26). For the details of this TS mapping, we refer the reader to Section 5.9.6 of Ref. [21]. The weak Hopf algebra obtained this way satisfies some extra conditions: Both the algebra and the coalgebra are special †-Frobenius algebras, as well as *-algebras.

In this context, the compactification mapping for anyons in Eq. (9.38) becomes the *quantum double* of the weak Hopf algebra. More precisely, the quantum double is a *semi-triangular (weak) Hopf algebra*, and the compactified 2D-tTS model corresponds to the algebra part (without the coalgebra) of the quantum double.

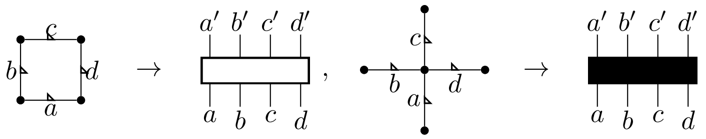
9.5 Commuting-projector Hamiltonian for the 3D face-edge tTS

In this section, we will see how to construct a commuting-projector Hamiltonian from a 3D-face-edge-tTS model. We will construct this commuting-projector Hamiltonian on a square lattice,



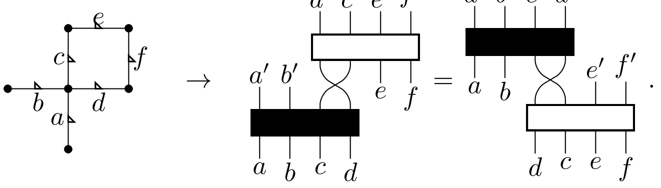
, (9.47)

but it can be defined on any 2-cellulation with edge directions and dual edge directions. There is one qudit associated to every edge of the lattice. There are two types of projectors defining the Hamiltonian, namely *plaquette projectors* and *vertex projectors*, each acting on the qudits at the edges adjacent to a fixed plaquette or vertex,



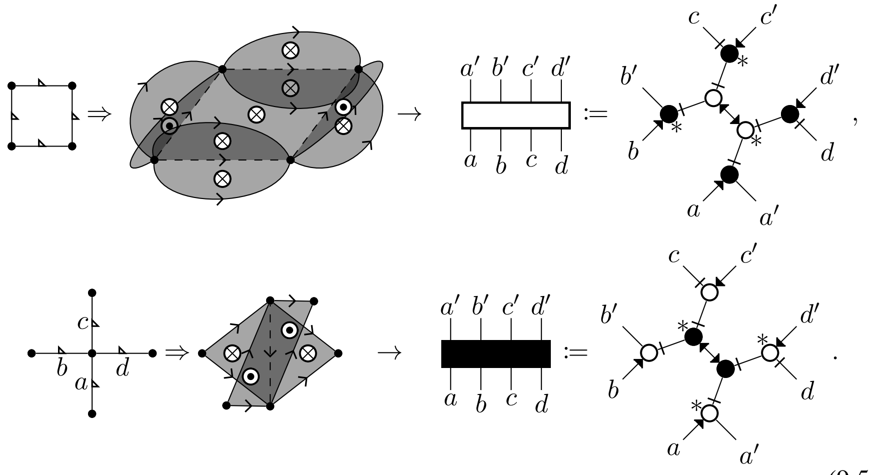
(9.48)

Both these operators need to be Hermitian projectors. Additionally, they need to satisfy four commutativity moves, corresponding to a plaquette projector commuting with the vertex projector at its south west, south east, north west, or north east. For example, for south west, we get a move,



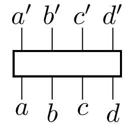
(9.49)

The construction of a commuting-projector Hamiltonian from a 3D-face-edge-tTS model can be formalized as a TS mapping from the commuting-projector TS to the 3D face-edge tTS. This TS mapping is given by

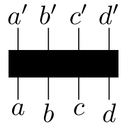


(9.50)

Via this mapping, a brick-layer circuit of plaquette and vertex projectors is mapped to a spacetime 3-cellulation. For example, if we first apply the vertex projectors for all vertices, then the plaquette projectors for all plaquettes, and so on, we get a cubic spacetime lattice. Applying this mapping to the toric-code model Eq. (9.17), we find



$$= \frac{1}{2}(\mathbb{1} - Z_0 Z_1 Z_2 Z_3)\ ,$$



$$= \frac{1}{2}(\mathbb{1} - X_0 X_1 X_2 X_3)\ . \tag{9.51}$$

Up to adding a term proportional to the identity matrix, we obtain a Hamiltonian

$$H = \sum_i A_i + \sum_j B_j\ , \tag{9.52}$$

with

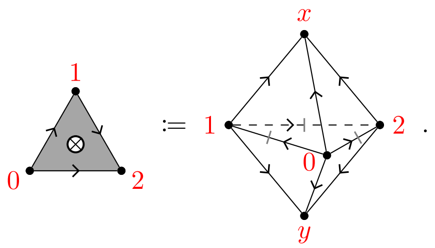
$$A = -Z_0 Z_1 Z_2 Z_3\ , \qquad B = -X_0 X_1 X_2 X_3\ . \tag{9.53}$$

Here, i (j) runs over the plaquettes (vertices) of the square lattice, and A_i (B_j) denotes A (B) acting on the qubits at the four edges adjacent to i (j). This commuting-projector Hamiltonian is the toric code in its original form as defined in Ref. [95].

9.6 Equivalence of the 3D face-edge tTS and 3D tTS

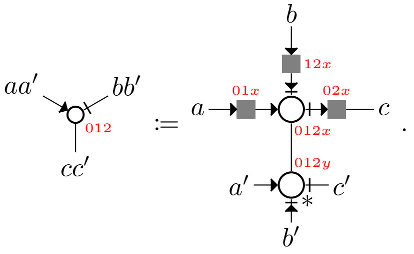
In this section, we will show that the 3D face-edge tTS is equivalent to the standard 3D tTS from Chapter 4. While the standard 3D tTS relies on a known theorem that Pachner moves implement discrete topological invariance, there is no such theorem for the moves of the 3D face-edge tTS. Thus, the proof of equivalence to the standard 3D tTS also has the purpose of showing that the 3D face-edge tTS is indeed a “topological” TS.

As usual we show equivalence by giving two weakly inverse mappings from the 3D tTS to the 3D face-edge tTS and back. Let us start with mapping from the 3D face-edge tTS to the 3D tTS. In order to construct this mapping, we actually need to enrich our diagrams with square roots for all the edge weights, as defined around Eq. (4.135). Geometrically, the mapping refinines a 3-cellulation to a triangulation such that each face and each edge of the original cellulation corresponds to a patch of 3-cells of the refined cellulation. This refinement proceeds in two steps. The first step consists in taking the *stellar subdivision* of every 3-cell. That is, we add a vertex x at its center, an edge spanned by every vertex of the 3-cell and x, a face spanned by every edge of the 3-cell and x, and a volume spanned by every face of the 3-cell and x. For each face of the original 3-cellulation, there is then a “diamond”-like volume, spanned by the face and the central vertices x and y of the two adjacent 3-cells. Specifically, we choose x and y such that the dual orientation points from x to y. For a triangle face, we get,



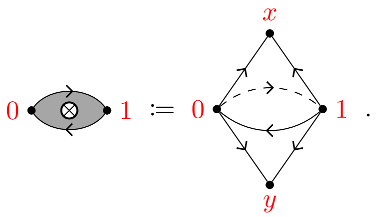
(9.54)

Note that on the left-hand side, the vertex x would be in front, and y behind the face. This volume can be triangulated by two tetrahedra,



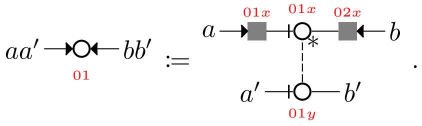
(9.55)

As shown, we include square-root edge weights for all the “equator” edges, and we will see in a moment why this is. We could apply the same mapping to other faces, for example, the cyclic 2-gon is mapped to



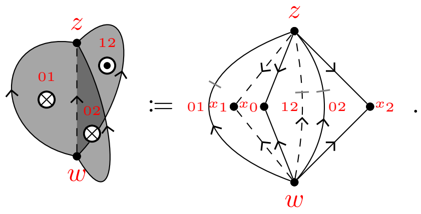
(9.56)

This volume can be decomposed into two 01 triangle flippers,



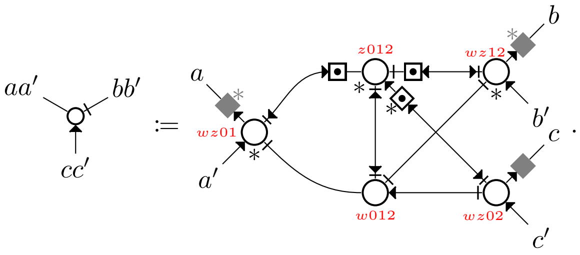
(9.57)

The second step of the mapping consists in “inflating” each edge: For an edge wz with adjacent volumes labelled 0, 1, and 2 and according faces 01, 12, and 23 (for a 3-valent edge), we add one edge connecting w and z inside every adjacent face. Then, add triangles spanned by the edge inside the ij face, and the central vertex of either the i 3-cell and the j 3-cell. After this, we have replaced the wz edge with a volume whose boundary is formed by all these new triangles. For example, the volume corresponding to a 3-valent edge is



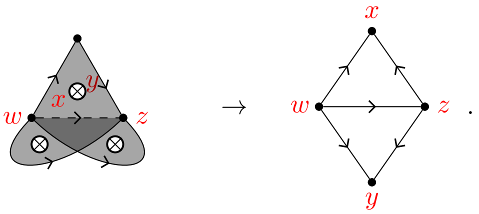
(9.58)

As shown, all the edges introduced in the last step carry square root edge weights. The volume above has the following triangulation in network notation,



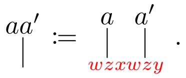
(9.59)

If an edge is adjacent to a face in the original cellulation, the two corresponding volumes share a pair of triangles,



(9.60)

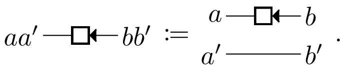
On the left, the edge is the dashed edge in the center, the face is the triangle on top, x is the volume in front and y the volume behind the triangle. Thus, the face-edge bond dimension is mapped to two triangle bond dimensions, that is, every index in a face-edge network corresponds to a pair of indices in a triangle network,



(9.61)

The direction of the edge separating the two triangles in Eq. (9.60) is aligned with the direction of the edge on the left-hand side. The ordering of the two indices is determined by the dual direction of the face on the left-hand side.

At last we check that the edge weights of the refined cellulation are distributed correctly over the tensors of the 3D face-edge tTS. There are two types of edges in the refined cellulation. First, edges corresponding to the wz edge in Eq. (9.60), separating a triangle pair. These edges are part of exactly two of the refined volumes, and each of these two volumes carry a square root edge weight. These two square root edge weights combine to the edge weight for each such edge. Second, edges like the wx, wy, zx, or zy edge in Eq. (9.60) correspond to corners of the original cellulation. Thus, we can incorporate the corresponding edge weights into the corner weights of the face-edge tTS. For example, the 00 corner weight is mapped to an edge weight at the wx edge of the triangle pair shown in Eq. (9.60),



(9.62)

In order to prove that the above recipe defines a valid TS mapping, we would have to give derivations for all the mapped moves. This is a straight-forward and purely combinatorial procedure. However, it is quite tedious and lengthy, thus we only give a quick argument for why the mapping is well-defined: The mapping is constructed such that all mapped moves are retriangulations. As it is known that any retriangulation corresponds to a sequence of Pachner moves, it is clear that all mapped moves can be derived.

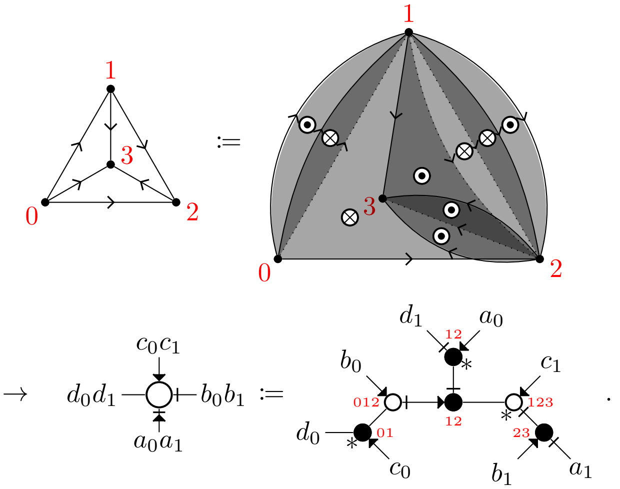
Let us now consider the converse mapping from the 3D tTS to the 3D face-edge tTS. Also this mapping has a geometric intuition in terms of a refining. This time, we want to refine a triangulation into a cellulation such that every tetrahedron maps to its own set of edges and faces. To this end, we first split each triangle into two triangles separated by a pillow-like volume, such that every n-valent edge becomes $2n$-valent.

Then, we replace every such $2n$-valent edge by n 4-valent edges which are cyclically connected by n trivial (non-cyclic) 2-gons. Like this, each original volume is mapped to one face for each of its faces, and one 4-valent edge for each of its edges.

Applying this to the tetrahedron we get a network consisting of 4 triangles and 6 4-valent edges. However, the mapping can be simplified: It suffices to only explicitly represent the 01 and 12 edge of each triangle with an edge, such that the triangle bond dimension variable is mapped to two copies of the face-edge bond dimension variable,

$$\overset{ab}{|} := {}_{01}\overset{a}{|}\quad {}_{12}\overset{b}{|}\ . \tag{9.63}$$

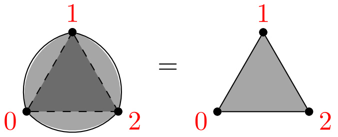
With this, a tetrahedron maps to the following,



(9.64)

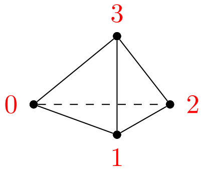
Let us sketch how the 3D-tTS moves can be derived from the 3D-face-edge-tTS moves. When applying the mapping to a triangulation, we obtain a face-edge tensor network where each face tensor corresponds to a triangle and each edge tensor to an edge. However, each edge or face can be represented by none or more than one tensor. If there are n edge tensors at an edge that are connected by indices, we can think of them together as a single $n + 2$-index edge tensor. The decomposition into 3-valent tensors does not matter due to the 2D-tTS axioms for the edge tensors. The edge directions and dual edge directions in the resulting network can be changed arbitrarily using the moves in Eqs. (4.94), (4.97), and (4.96) for either the edge or face tensors. Thus, we can neglect these directions in the following considerations.

In order to derive the 3D-tTS Pachner moves, let us first define some auxiliary moves that can be derived from the moves of the face-edge tTS. To this end, let us denote by $C(012|02)$ the corner fusion move depicted in Eq. (9.26) from left to right. Let us also denote by $P_2(012|023)$ the 2-2 Pachner move for the face tensors, as depicted in Eq. (4.11), from left to right. Next, consider the following move,

$T(012)$:  , (9.65)

which equates two triangles separated by a pillow-like volume and adjacent to the same three edges on the left, and a single triangle on the right-hand side. $T(012)$ can be derived by using the triangle cancellation move to bring one triangle of the corner fusion move in Eq. (9.26) from the right to the left.

Next, with respect to a standard tetrahedron,

 , (9.66)

let us define the following move $X(012|3)$. The left-hand side consists of a network where four triangles and all six edges are represented by 3-valent tensors. On the right-hand side, only the triangles 013, 123, 023, and the edges 03, 13, 23, are represented by tensors. In other words, we go from the left to the right by removing the tensor at the 012 face and the adjacent edges. This move can be derived via the moves above,

$$\begin{aligned} X(012|3) = \overline{T(123)} \rightarrow \overline{C(123|12)} \rightarrow P_2(012|123) \\ \rightarrow C(013|03) \rightarrow T(023) \ , \end{aligned} \tag{9.67}$$

where the bars denote the move in the opposite direction.

Last, we consider a move $Y(12|03)$ defined as follows. On the left-hand side consists of a network where all faces and edges of the tetrahedron are represented by triangle tensors and 3-valent edge tensors, except for the 12 edge. This edge is 2-valent and thus the 012 and 123 face tensors are directly connected via a bond. On the right-hand side, only the triangles 013, 023, and the edge 03 are represented by 3-index tensors. That is, we go from the left to the right by removing both the 012 and the 123 face. This move can be derived by

$$Y(12|03) = P_2(012|123) \rightarrow C(013|03) \rightarrow T(023) \ . \tag{9.68}$$

Let us now apply the mapping in Eq. (9.64) to the 2-3 Pachner move. Before this, we apply $\overline{Y(03|12)}$ to Eq. (9.64), such that every triangle of the tetrahedron is represented by a tensor. Thus, in the Pachner move, each interior triangle will be duplicated, and each space-bondary triangle is represented by exactly one triangle. Next, we apply $C(012|02)$ to every interior duplicated triangle pair, to map it to a single triangle. Next, we can apply the moves derived in Eq. (9.67) and (9.68) to remove all interior triangles and edges on the left and right, which yields an equation between twice the same network. We have thus found a derivation of the 2-3 Pachner move from the 3D-face-edge-tTS moves. In fact, this prescription does not only work for the 2-3 Pachner move, but to all retriangulations.

Chapter 10

Universality mapping with topological boundary

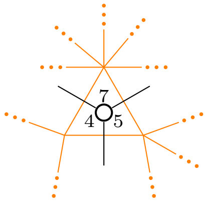
In Chapter 9, we have seen that there are different ways to represent cellulations by tensor-network diagrams, yielding nD tTS that are different from "the" nD tTS discussed in Section 4. When ambiguous, we will refer to the nD tTS from Section 4 as *nD standard tTS*. The tTS discussed in Chapter 9 turned out to be equivalent to the nD standard tTS. An important question arises: Are all nD tTS equivalent to the nD standard tTS, or are there some that potentially capture more general phases? In this chapter we will find that the nD standard tTS is *universal* under one condition, namely that the models exhibit a topological boundary. By universal, we roughly mean that the nD standard tTS is at least as general as any other nD tTS.

10.1 General tTS

In order to better explain universality, we need to first better explain what we mean by a general nD tTS. We have already seen a few examples for general nD tTS, namely the 2D, 3D and 4D standard tTS, the 2D edge tTS, and 3D face-edge tTS. In general, a nD tTS is a prescription that associates tensor networks to cellulations of n-manifolds. This prescription has to be local, that is, the structure of the tensor network at one place can only depend on the combinatorics of the cellulation in a constant-size neighborhood. Here, the size is measured in terms of the combinatorial distance of the cellulation. Apart from being finite, the prescription can be arbitrarily complicated. A tTS has to satisfy moves that allow us to arbitrarily deform the cellulation while preserving its topology. In particular, equivalence classes of tensor-network diagrams under moves need to be in one-to-one correspondence to equivalence classes of manifolds under homeomorphism. Note that if the tTS is very complicated, then the moves can involve large (but constant-size) tensor-network diagrams.

Let us illustrate this general notion by giving an an hoc example for a 2D tTS that is more complicated, and in fact not diagrammatically equivalent the 2D standard tTS. We will call this 2D tTS the *2D neighborhood-sentitive tTS*. Just like the 2D tTS, the

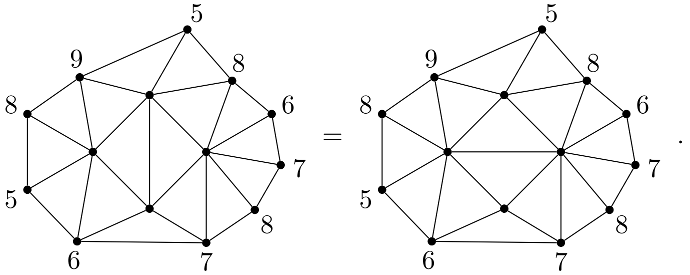
2D neighborhood-sensitive tTS associates one tensor to each triangle of a triangulation, with neighboring triangles sharing a bond. However, these tensors are not all the same, but copies of different tensor variables. This tensor variable depends on the *valencies* of the three corner vertices, that is, the numbers of triangles adjacent these vertices. Diagrammatically, we will draw the tensors in the same way as for the 2D standard tTS, just that we add labels indicating the valencies, for example,



. (10.1)

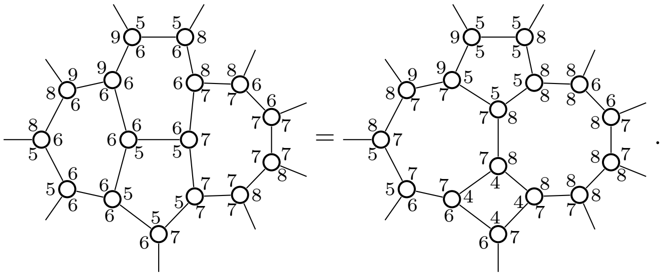
A model of this tTS consists of one tensor for every triple of numbers, such as $(4, 5, 7)$ in the example above. Note that since this is just a toy example, we do not impose a branching structure or an orientation. General triangulations can have vertices with arbitrarily high valencies, so in order to represent any triangulation we would need an infinite number of different tensors. However, we can without loss of generality restrict to a finite set of different adjacencies, see Appendix A of Ref. [25].

Just like for the 2D standard tTS, the moves of the 2D neighborhood-sensitive tTS are based on the Pachner moves. However, the 2-2 Pachner move changes the valency of the four vertices at the corners. Thus, if we apply a 2-2 Pachner move to a triangulation, this changes not only the tensors at the two triangles directly involved in the move, but also those at the triangles adjacent to the four vertices at the corners. For example, for the Pachner move below, the tensors at all the depicted triangles change, not only the two in the middle,



. (10.2)

Here, the labels at the boundary vertices indicate their valencies. In order to implement such a move as a tensor-network equation, we have to include all the tensors that are changed, namely,



. (10.3)

We need to impose one such move for 1) every choice of valencies of the four corner vertices of the 2-2 Pachner move, and 2) every choice of valencies of the vertices connected to these corner vertices by an edge. So a 2D-neighborhood-sensitive-tTS model is determined by a large set of 3-index tensors fulfilling a very large set of very complicated tensor-network equations. Even though large, we can restrict to a finite set of moves by restricting to some large-enough maximal valency.

In general, the tensor variable at a triangle may depend on the combinatorics of the triangulation in an even larger neighborhood, say, of combinatorial distance 10. There may also be additional bonds, for example connecting a tensor at one triangle to all those sharing a common vertex. Or, we might have tensors at different places of the triangulation, for example one at every edge depending on the valencies of the two end vertices of the edge. As one can see, there is an unlimited amount of possibilities to construct arbitrarily complicated tTS.

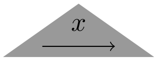
Each of the more complicated tTS might allow for more general models that capture phases which are not captured by the 2D standard tTS. Fortunately, there is a way to argue that simpler tTS, where the geometry of the diagrams and the tensors themselves do not depend the lattice combinatorics in a too large neighborhood, suffice to capture all phases. This is done by fine-graining a triangulation in a fixed way, putting the complicated tTS on the fine-grained lattice, and blocking large chunks of the fine-grained tensor network into a single tensor. Roughly, if the complicated tTS depended on a distance-a lattice neighborhood, and we fine-grain by a factor of λ, the blocked tTS will only depend on a distance-a/λ neighborhood. So by choosing larger and larger fine-graining scales λ, we eventually end up with a tTS that only depends on the underlying triangulation in a minimal way. This will be the idea that we use to attempt to prove universality of the 2D standard tTS in the following section.

10.2 Universality mappings and the corner problem

In this section, we will try to show that the 2D standard tTS is universal, and find that any such attempt must fail due to a problem that we call the *corner problem*. Before we get to that, let us properly introduce the notion of universality. As we mentioned earlier, intuitively, a nD tTS $\mathcal{A}$ is universal if it can emulate any other nD tTS. Formally, $\mathcal{A}$ is universal if for any other nD tTS $\mathcal{B}$, there is a weakly invertible mapping $\mathcal{A} \rightarrow \mathcal{B}$. A construction for such a TS mapping from $\mathcal{A}$ to an arbitrary tTS $\mathcal{B}$ will be called a *universality mapping*. Recall that weakly invertible means that there is a mapping back $\mathcal{B} \rightarrow \mathcal{A}$, such that applying the combination $\mathcal{A} \rightarrow \mathcal{B} \rightarrow \mathcal{A}$ to a tensor-network diagram yields a diagram that is equivalent to the original diagram via the $\mathcal{A}$-moves. Note that we have already used weakly invertible tTS mappings to show the equivalence between the non-simplified and simplified tTS in Sections 4.9, 4.8, 4.7, 4.11, and 5.2, between the 2D edge tTS and 2D tTS in Section 9.1, and between the 3D face-edge tTS and the 3D tTS in Section 9.6. Since both $\mathcal{A}$ and $\mathcal{B}$ are topological, weakly invertible tTS mappings are those that are *topology preserving*, that is, that map a $\mathcal{A}$-diagram representing n-ball to a $\mathcal{B}$-diagram representing an n-ball. By applying the universality mapping to a $\mathcal{B}$-model, we obtain a $\mathcal{A}$-model that represents the same phase.

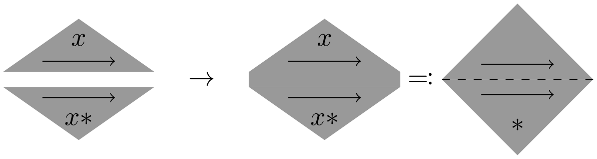
There are two main ways in which a prescription that associates to every $\mathcal{A}$ tensor variable a $\mathcal{B}$ tensor-network diagram can fail to be a TS mapping. First, the mapping might produce *invalid* $\mathcal{B}$-diagrams when applied to an $\mathcal{A}$-diagram, which do not represent a patch of a 2-manifold. For example, if $\mathcal{B}$ is the 2D standard tTS, the mapping might not result in a network with exactly one vertex weight for every vertex. As another example, if $\mathcal{B}$ is the 2D neighborhood-sensitive tTS, it might result in a network with a wrong tensor variable at a triangle, whose labellings do not match the actual valencies of the corner vertices. Second, the mapped moves of $\mathcal{A}$ must be derivable from the moves of $\mathcal{B}$. If the networks on both sides of the $\mathcal{B}$-moves are very large compared to those of $\mathcal{A}$, it might not be possible to use them to derive the mapped moves of $\mathcal{A}$. This problem can be tackled by using large $\mathcal{B}$-networks in the mapping itself. That is, the mapping will assign $\mathcal{B}$-networks that are fine-grained by a scale λ, as we argued at the end of the previous section.

Let us now take for $\mathcal{A}$ the 2D standard tTS and attempt to show that it is universal. That is, we need to construct a weakly invertible mapping from the 2D standard tTS to an arbitrary 2D tTS $\mathcal{B}$. Such a TS mapping is specified by the $\mathcal{B}$-network associcated to the triangle tensor variable. To construct a suitable $\mathcal{B}$-network, we roughly speaking we take a triangular area and fill it with $\mathcal{B}$ diagram that is fine-grained at a length scale λ. To ensure that we can glue arbitrary edges of these triangular-area $\mathcal{B}$ diagrams, we construct this diagram in a multi-step filling-and-cutting process. We start by filling a "flat triangle" with some $\mathcal{B}$-network such that the corners have a combinatorial distance (i.e., minimum number of bonds in a connecting path) of at least λ,



. (10.4)

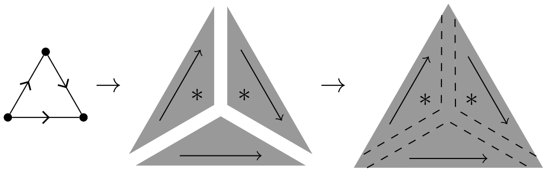
The gray area marked by x represents an arbitrary but enough fine-grained $\mathcal{B}$-network representing a patch of 2-manifold, with an arbitrary configuration of open indices at the boundary. Next, we put a mirrored and complex conjugated copy of the triangle near its bottom edge, fill the space between them, and cut the resulting patch again,



. (10.5)

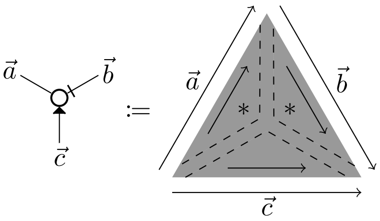
This way, we get a new triangle patch, drawn without the x, whose edges can be glued together. Here we assume that the bottom and top half are complex conjugates of another, which is in fact only possible if the $\mathcal{B}$-network in the middle, viewed as an operator from the bottom to the top indices, is Hermitian and positive semi-definite. So we already encounter a first problem, which we will ignore though since a more substantial problem will appear below.

Next, we combine three of the triangle-shaped network patches into a single triangle-shaped network by placing them next to each other and filling the gaps between them with some $\mathcal{B}$-network as follows,



. (10.6)

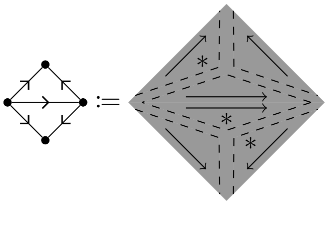
For a negatively oriented triangle, we take the mirrored and complex conjugated $\mathcal{B}$-network. Thus, by construction, the Hermiticity of the 2D tTS follows from the Hermiticity of $\mathcal{B}$. The branching structure of the reference triangle on the left is important to make an unambiguous choice of such a filling. We now associate this $\mathcal{B}$-network to the triangle tensor variable,



. (10.7)

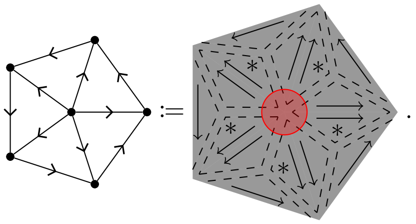
The $\mathcal{B}$-network on the right has many open indices along its open boundary, and their number is scaling linearly with λ. The open indices along each of the three edges are ordered according to the direction of the edge and grouped together to yield one of the indices $\vec{a}$, $\vec{b}$ and $\vec{c}$, respectively. The bond dimension of the 2D-tTS model is the product of all the bond dimensions of the indices along one edge, and therefore grows exponentially in λ. Note, that in the bottom left corner (as well as in the other corners), the filling between the triangles can give rise to additional open indices. For each such index, we have to make a choice of whether to assign it to $\vec{a}$ or $\vec{c}$, and analogously for the other corners.

In order to test whether this prescription actually defines a TS mapping, we need to first check whether every valid 2D-tTS network patch is mapped to a valid $\mathcal{B}$-network patch. If we apply the mapping to two neighboring triangles,



, (10.8)

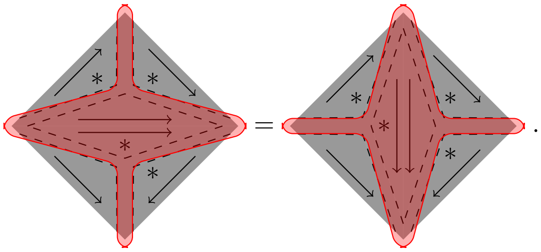
this is the case per construction as we just re-glue what we have cut apart in Eq. (10.5). However, the situation is different when we consider a patch of triangles around a vertex, e.g.,



(10.9)

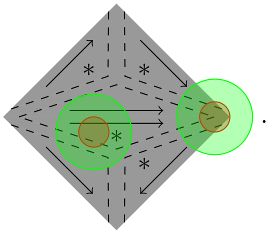
The network in the red marked region around the central vertex does not originate from a cut that is re-glued. We thus have no guarantee that the network in the red marked region is valid and represents a disk-like patch of 2-manifold. Note that this network also depends on the arbitrary choice of whether the open indices in the corners of Eq. (10.7) are associated to $\vec{a}$, $\vec{b}$ or $\vec{c}$.

We also need to test whether the mapped Pachner moves can be derived from the $\mathcal{B}$-moves. For example, the mapping of the 2-2 Pachner move in Eq. (4.11) yields



(10.10)

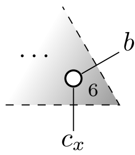
We notice that the $\mathcal{B}$-network only changes within the red shaded area. If we want to change the $\mathcal{B}$-network within some region by $\mathcal{B}$-moves, the latter need to act within this region enlarged by a margin of constant size measured in the combinatorial distance. For example, in order to perform changes within one of the red circles in the following picture, we might need to apply moves within the respective larger green regions,



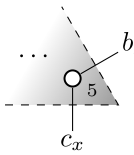
(10.11)

For any red region away from the boundary, the green region is fully contained inside the overall $\mathcal{B}$-network, if we choose the fine-graining scale λ large enough. Consequentially, changes in these regions can be performed using the $\mathcal{B}$-moves, no matter how complicated the latter are. However, this does not hold for changes right at the boundary. Thus, the fine-graining procedure above is a valid mapping for an arbitrary tTS $\mathcal{B}$ only if the two $\mathcal{B}$-networks of every mapped move differ only in regions distant from the boundary. As we can see in Eq. (10.10), the present construction almost succeeds in doing so: Most of the regions on which the two networks differ are located in the bulk of the network. The problem only arises at the corners of Eq. (10.10), hence we refer to the encountered obstruction as the *corner problem.*

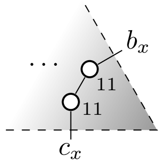
To be more concrete, let us now show that the corner problem actually appears when we try to construct a topology-preserving mapping from the 2D standard tTS to the 2D neighborhood-sensitive tTS. Such an attempted mapping associates to the triangle a planar 2D-neighborhood-sensitive-tTS network. For different possible choices of the mapping, the bottom right corner could look like



, or

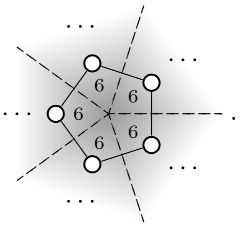


, or



, (10.12)

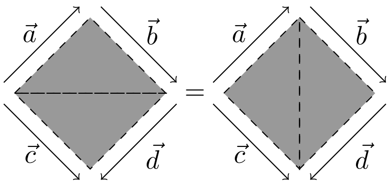
or b_x and c_x could be separated by more bonds and the valency labels in the corner could take other values. The point is that, regardless of the specific choice we make for the mapping, no choice can result in valid 2D-neighborhood-sensitive-tTS networks for all input 2D-standard-tTS networks. This is due to the fact that the 2D standard tTS allows for l-valent vertices for *any* l whereas the tensors at the corner of the mapping have fixed adjacency labels. As a result, for every choice of the mapping, there exists a small patch of the 2D tTS with a vertex of adjacency l, such that the adjacency l does not match the adjacency labels of the network obtained from the mapping. For example, for the first case in Eq. (10.12) and an $l = 5$ vertex, we obtain a 5-gon plaquette, but the surrounding tensors have valency label 6,



. (10.13)

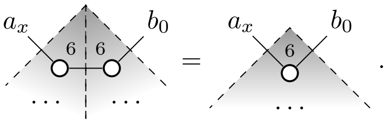
Since valency labels 6 do not match the actual valency 5, the resulting network is invalid.

Another problem with the attempted mapping is that the mapped moves of the 2D tTS cannot be derived from the 2D-neighborhood-sensitive-tTS moves. This problem again originates from the fact that the tensors of the 2D standard tTS do not carry any information about the adjacency of vertices, whereas this is the case for the 2D neighborhood-sensitive tTS. For example, applying the mapping to the 2-2 Pachner move yields



. (10.14)

If the mapping is given, for example, by the first case in Eq. (10.12), the upper corner of this move looks like



(10.15)

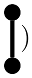
However, this move cannot be derived from the moves of the 2D neighborhood-sensitive tTS: It changes the number of bonds in the plaquette containing a_x and b_0, and thus the valency of the vertex corresponding to this plaquette, by one. So in order to be derivable from the moves of the 2D neighborhood-sensitive tTS, either the 6 on the right would have to be a 5, or the 6s on the left would have to be 7s. Similar arguments apply for any other choice of the mapping.

10.3 2D universality mapping with boundary

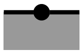
In Section 5.3, we have seen that any 2D-tTS model admits a topological boundary constructed via the cone extension mapping. Thus, this fixed-point ansatz is only suitable for topological phases that admit a topological boundary. Note that topological boundaries are roughly the same as *gapped boundaries*, which is the more common notion in condensed-matter physics. Following the argumentation in Section 3.6, we conjecture that all robust gapped boundaries are obtained from topological ones via defect networks. This implies that in physical dimensions, all gapped boundaries are topological for an appropriate choice of unit cell, since fracton-like phases only exist in $3+1$ dimensions and higher.

For an arbitrary tTS $\mathcal{B}$, there is no obvious choice of a mapping constructing a boundary. If $\mathcal{B}$ does indeed not have such a boundary mapping, then it cannot be emulated by the 2D tTS. In this section, we will show that the converse is true as well: We will construct a universality mapping from the 2D standard tTS to any 2D boundary tTS $\mathcal{B}$. Thereby, we will make use of the topological boundary to resolve the corner problem that appeared in Section 10.2.

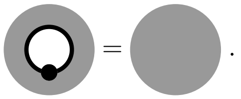
To make this mapping fully formal, we do not only require $\mathcal{B}$ to be a 2D boundary tTS, but we also need some additional defects and topology-changing moves. Namely, the overall extended manifold type that $\mathcal{B}$ describes is

$a:(2,\varnothing)\,,\quad b:(1,\bullet)\,,\quad c:(0,$  $)\,.$ (10.16)

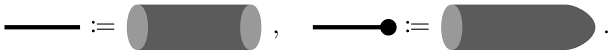
That is, we include one more region c corresponding to point defects inside the boundary, like,


. (10.17)

There is one additional topology-changing move, namely

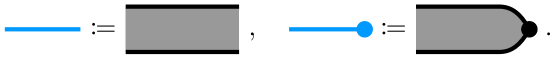

(10.18)

Note that these extra conditions are always fulfilled, but we impose them to be able to work on a purely diagrammatic level. To see this, consider the following mapping from the 1D tTS to $\mathcal{B}$ by taking the cartesian product with a circle,



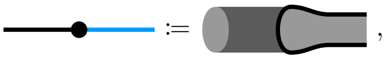
(10.19)

As shown, this can be extended to 1-manifolds with boundary by filling in the circle with a disk. Further, we consider the mapping from the 1D tTS to $\mathcal{B}$ by taking the cartesian product with an interval,



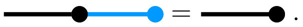
(10.20)

As shown, this can be extended to 1-manifolds with boundary using the boundary point defect. Further, we can define the following domain wall between the circle-compactification and the interval-compactification mappings,



(10.21)

where the right-hand side corresponds to a tube where we cut out a half-disk with physical boundary on the right side. Applying all these compactifications to the move in Eq. (10.18) yields

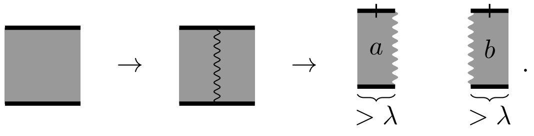


(10.22)

To interpret the algebraic structure of this move, we note that 1D-tTS models correspond to projectors. After applying the full support convention, this projector is the d-dimensional identity matrix. The $0+0$-dimensional boundaries are vectors x and w, and the domain wall is a matrix M. The above equation is then of the form $Mx = w$, so a boundary defect fulfilling Eq. (10.18) exists if w is in the image of M. If the $\mathcal{B}$-model is robust, then the 1D-tTS model is trivial, and we have $d = 1$. In this case, x trivially exists. Any $\mathcal{B}$-model decomposes into a direct sum of robust models, so it suffices to demand that every robust component has a topological boundary.

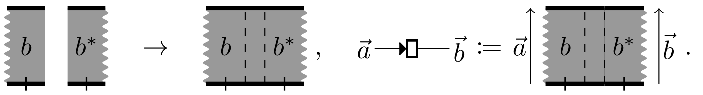
Further, we also need to equip the 2D tTS with both vertex weights and edge weights, c.f. Section 4.5. Note that the boundary point defect will be very closely related to the vertex weights of the 2D tTS.

After discussing additional topology-changing moves and the weight matrices, it is now time to construct the universality mapping. We start by filling a rectangle with $\mathcal{B}$-network, with physical boundary at the top and bottom, and space boundary on the left and right. Then we cut this network into two pieces a and b along the middle separating the space boundaries,



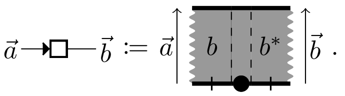
(10.23)

As depicted, we do this such that the two physical-boundary intervals are separated by at least a combinatorial distance λ for both a and b. Next, we take two copies of b, one reflected and complex conjugated, and fill the space between them. This is what we associate to the edge weight,



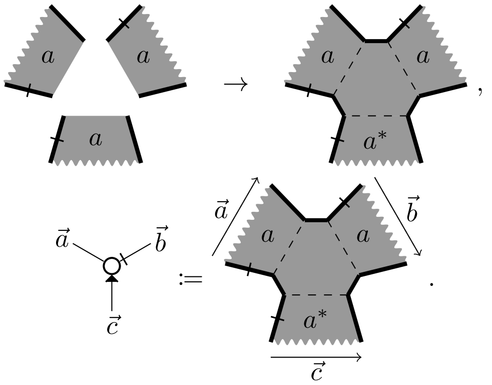
(10.24)

Next, we consider the same filling with a boundary point defect at one of the boundaries. This is what we associate to the 0 vertex weight,



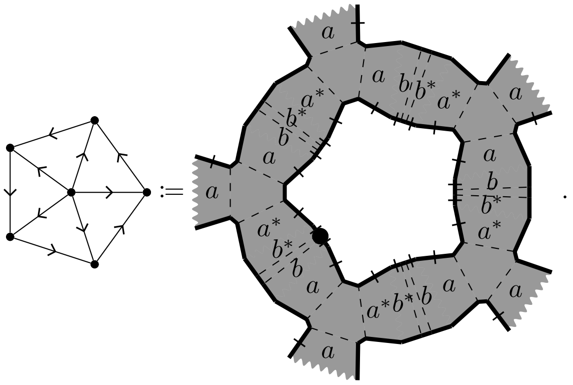
(10.25)

Last, we configure three of the a halves around a circle, and fill the space between them. The result is what we assign to the triangle tensor,



(10.26)

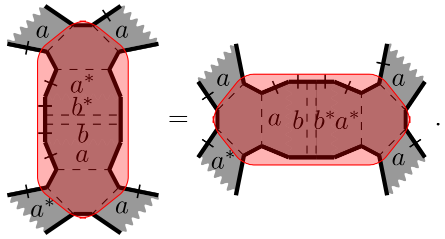
With this choice, the network around any vertex is always valid, irrespective of the configuration of surrounding triangles, for example



(10.27)

This is because we are just re-gluing patches of $\mathcal{B}$-networks in the same way they have been cut before. If we apply the mapping to a triangulation, we obtain a manifold with a puncture at each vertex of the triangulation. Every such puncture carries exactly one boundary defect and can thus be removed using the move in Eq. (10.18). Thus, the mapping is topology-preserving.

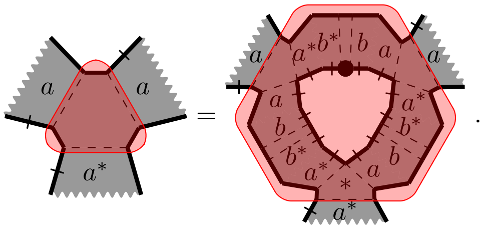
The next thing that we have to check is whether the mapped Pachner moves can be derived from the $\mathcal{B}$-moves. For the mapping of the 2-2 Pachner move in Eq. (4.12), we get



(10.28)

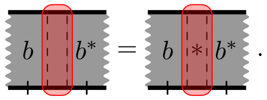
The $\mathcal{B}$-networks on both sides differ only within the red shaded area. Even though this area involves parts of the physical boundary, it is well isolated from the space boundary. More precisely, the combinatorial distance to the space boundary can be made arbitrarily large by choosing a larger and larger fine-graining scale λ. Thus, no matter how complicated the moves of $\mathcal{B}$ are, we can use them to transform the two sides into each other.

For the 1-3 Pachner move in Eq. (4.56), we get



(10.29)

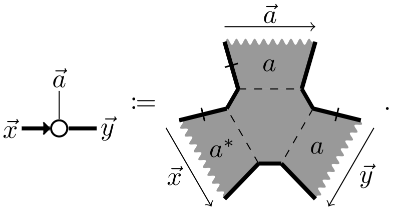
Again, the $\mathcal{B}$-network only changes in the red shaded region which is separated from the open indices. To transform both sides into each other we also need the move in Eq. (10.18), in order to remove the puncture on the right-hand side. Next, the Hermiticity of the edge weight becomes,



(10.30)

Again, the two sides differ only within the red shaded region that is separated from the boundary by a distance λ.

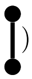
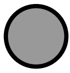
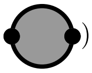
We have seen that the 2D standard tTS is a universal fixed-point ansatz for general 2D tTS models for which a boundary exists. In addition to that, the 2D standard boundary tTS from Section 4.10 is universal for all general 2D boundary tTS. That is, any 2D boundary tTS can be emulated by the 2D standard boundary tTS. The corresponding universality mapping is the universality mapping in Eq. (10.26) combined with the cone-extension boundary in Eq. (5.116).



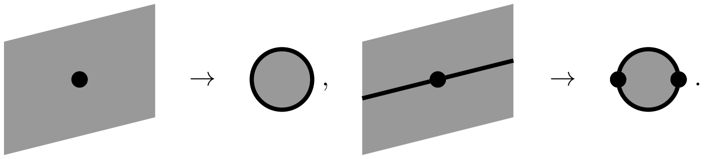
(10.31)

10.4 3D universality mapping with boundary

In this appendix, we generalize the universality mapping with a topological boundary from Section 10.3 to $2+1$ dimensions. Again, to work on a purely diagrammatic level, the target 3D tTS $\mathcal{B}$ needs to be equipped with some defects and topology-changing moves in addition to the boundary. Namely, $\mathcal{B}$ describes a tTS of the following extended manifold type,

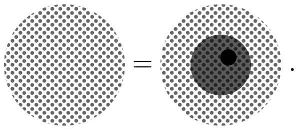
$a:(3,\varnothing)$, $b:(2,$●$)$, $c:(1,$  $)$, $d:(0,$  $)$, $e:(0,$  $)$. (10.32)

That is, we have a bulk (a) with boundary (b), and a boundary anyon (c) as in Section 6.3. In addition, d corresponds to point defects inside the boundary, and e to point defects on the boundary-anyon worldlines,



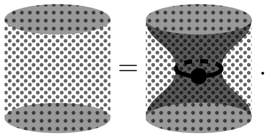
(10.33)

There are two topology-changing moves. The first move is a 3-handle attachment decorated with a point defect,



(10.34)

On the left-hand side, there is a space-boundary 3-ball. On the right-hand side, there is a 3-ball with a smaller 3-ball removed. The inside is physical boundary decorated with a point defect, whereas the outside is space boundary. The second move is a 2-handle attachment decorated with a boundary anyon and a boundary-anyon point defect,



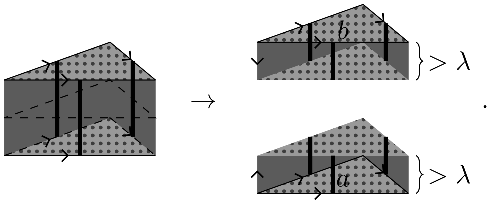
(10.35)

On the left-hand side, there is a cylinder of $\mathcal{B}$-network with physical boundary on the top and the bottom and an space boundary on the side. On the right-hand side, there is a solid torus whose boundary is split into a physical-boundary annulus on the inside and a space-boundary annulus on the outside. The physical boundary on the inside is decorated with a looping boundary-anyon worldline that carries one point defect.

Let us now discuss how restrictive the assumption of the existence of further boundary defects is. To this end, we apply compactification mappings to relate these defects and moves to lower-dimensional ones. Let us start with the topology-changing move in Eq. (10.34). We consider a mapping from the 1D tTS consisting in taking the cartesian product with the disk, which is the link of d in Eq. (10.32). Boundaries of this 1D-tTS model are in one-to-one correspondence with d point defects. Applying this mapping to Eq. (10.22) yields Eq. (10.34). Thus, analogous to Section 10.3, this condition can be automatically fulfilled for any robust phases, as well as for non-robust phases if every robust component has a topological boundary. Next, consider the move in Eq. (10.35). As we have seen in Section 6.3, boundary anyons are in one-to-one correspondence with boundaries of an interval-compactified 2D tTS model. Point defects inside the boundary-anyon worldlines are in one-to-one correspondence with point defects on the boundary of the 2D tTS model, like the defect c in Eq. (10.16). If we apply the compactification mapping to Eq. (10.18), we precisely obtain the move in Eq. (10.35), which in the reverse direction can be imagined as squeezing the latter in the vertical direction. Thus, this condition can be fulfilled if every robust component of the interval-compactified 2D tTS model has a topological boundary. Since conjecturially all topological phases in $1+1$ dimensions are trivial and therefore have a topological boundary, this is always the case.

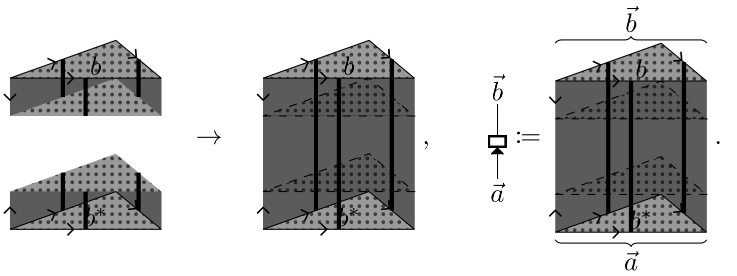
In order to work on a purely diagrammatic level, we also consider a variation of the 3D standard tTS that includes vertex weights as well as face weights, c.f. Section 4.5.

After introducing all additional boundary defects, topology-changing moves, and weight matrices, we are now ready to construct the universality mapping from the 3D standard tTS to $\mathcal{B}$. We start with a triangle prism with space boundary on the bottom and top triangle and physical boundary on the sides. We then fill this with $\mathcal{B}$-network, and include boundary-anyon worldlines connecting the centers of the two 01 edges, the two 12 edges, as well as the two 02 edges of both triangles,



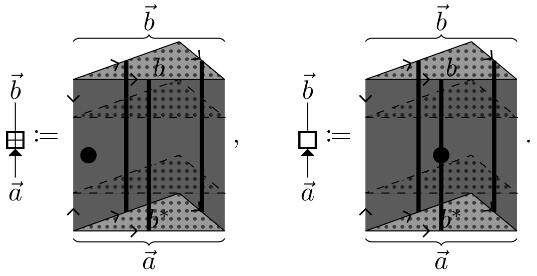
(10.36)

As depicted, we then cut this triangle prism into two halves a and b, such that for each half the combinatorial distance between the two space-boundary triangles is at least λ. Next, we fill the space between two b prisms, one of them reflected and complex conjugated,



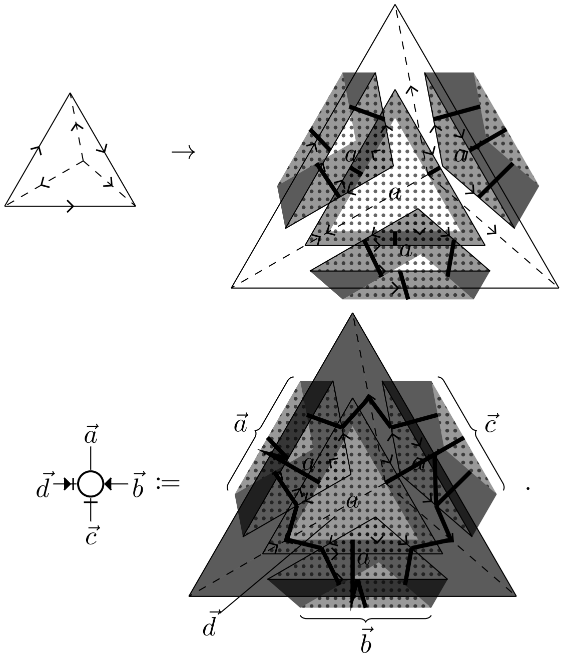
(10.37)

As shown, the obtained $\mathcal{B}$-network is what we associate to the edge weight. If we decorate the filling with a point defect, either on a boundary-anyon worldline, or on the physical boundary, we get the edge and vertex weights. For example, the 0 vertex weight and 02 edge weight are obtained by



(10.38)

Finally, the tetrahedron tensor is obtained by arranging four of the a halves from Eq. (10.36) around a sphere and filling the space between them. The obtained network is what we assign to the tetrahedron tensor,



(10.39)

If we apply the mapping to a triangulation of a 3-manifold, every tetrahedron gets replaced by the $\mathcal{B}$-network in Eq. (10.39), and every triangle is replaced by the $\mathcal{B}$-network in Eq. (10.37). This results in a valid $\mathcal{B}$-network since we just re-glue a and b patches in the same way as we cut them in Eq. (10.36). However, this $\mathcal{B}$-network does not represent the original 3-manifold, but one where the neighbourhood of all the vertices and edges of the triangulation has been removed, with the 3-manifold terminating at a physical boundary. Along each edge of the triangulation, there is a tube of vacuum surrounded by a physical boundary, and around each such tube wraps a boundary anyon. At every edge, there is one edge weight which is mapped to a point defect on the boundary anyon worldline. We can use the move in Eq. (10.35) backwards to fill the tube of vacuum with a cylinder of 3-manifold. Then we are left with a 3-ball of vacuum around every vertex. At every vertex there is one vertex weight that maps to a point defect on the corresponding physical-boundary sphere. We can use the move in Eq. (10.34) to fill this ball of vacuum with a ball of 3-manifold. We thus see that the mapping yields a valid $\mathcal{B}$-network representing the same 3-manifold.

Also, it is easy to see that the mapped Pachner moves yield equations between $\mathcal{B}$-networks that differ only at places of distance $> \lambda$ from the space boundary. By choosing a larger and larger fine-graining scale λ for the mapping, we are eventually guaranteed that the two networks are related by $\mathcal{B}$-moves. In addition, we need the move in Eq. (10.35) in the derivation of the mapped 2-3 Pachner move, and 4 times Eq. (10.35) plus once Eq. (10.34) in the derivation of a mapped 1-4 Pachner move.

Chapter 11

The vertex tTS: Beyond topological boundaries?

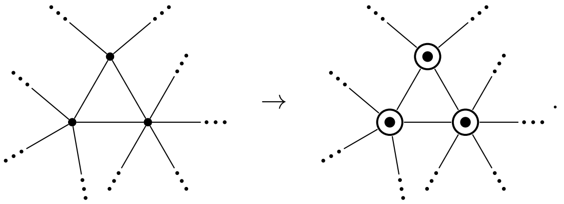
In Chapter 10, we have seen that in order to construct a universality mapping for the 2D or 3D standard tTS, we need a topological boundary. We have also seen that there are more complicated tTS, such as the 2D neighborhood-sensitive tTS, that could potentially capture more general phases than the 2D standard tTS. In this chapter, we will present a new kind of nD tTS, the *nD vertex tTS*, which is more complicated than the nD standard tTS, but less complicated than, for example, the 2D neighborhood-sensitive tTS. For this tTS, we manage to show a universality mapping without assuming the existence of a topological boundary. This tTS is thus a natural candidate to model topological phases without topological boundary, such as chiral topological phases.

We will start by introducing the vertex tTS in Section 11.1. Then we show that trying to emulate it with the nD standard tTS leads to a corner problem in Section 11.2. In Sections 11.3 and 11.4, we show that the cone-extension and commuting-projector mappings fail for the vertex tTS. In Section 11.5, we show that the vertex tTS is universal. Finally, in Section 11.7 we discuss some paths that might lead to fixed-point models for chiral phases.

11.1 The vertex tTS

In this section, we introduce the vertex tTS. There is a version of this tTS in any spacetime dimensions. We will first focus on $1+1$ dimension, and in the end show how the definition straight-forwardly generalizes to higher dimensions.

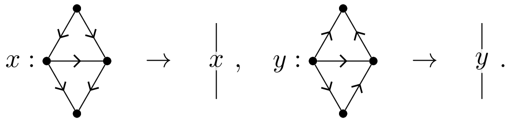
As the name indicates, the 2D vertex tTS associates one tensor to each vertex of a triangulation. The edges of the triangulation correspond to bonds of the tensor network, such that the tensor-network diagram looks like the drawing of the triangulation itself, for example,



(11.1)

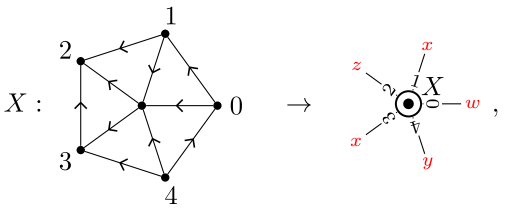
Note that the vertices in a triangulation can have different *valencies*, that is, different numbers of adjacent edges. So for each valency, there is a different tensor variable with the corresponding number of indices. For example, the three tensors in the network above are obviously different, since they have 4, 5, and 6 indices, respectively. Naively, we would need an infinite amount of tensors to represent triangulations with arbitrary valencies, but we can restrict to a finite set of adjacencies as shown in Appendix A of Ref. [25].

To be precise, we have to take care of a few technical details. First, we need to equip the triangulation with a branching structure and an orientation. Next, the bond dimension variable of an index depends on the *star* of the associated edge, that is, the configuration of the two adjacent triangles including the edge directions. Since there are many such stars (9 in fact), we will choose labels for them and decorate the bonds with these labels instead of choosing different line styles. For example, we could choose



(11.2)

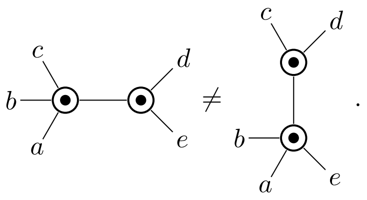
The tensor variable at a vertex depends on its *star*, that is, the configuration of triangles containing the vertex, including their edge directions. In order to incorporate this into the tensor-network notation, we have to name every possible star, as well as every edge inside the star adjacent to the central vertex, for example,



(11.3)

where w and z label bond dimension variables we have not introduced so far. Note that if we restrict to some maximal valency, there is a finite number of different stars, but nonetheless, this number will be fairly large. In this section we show how to systematically construct the tensor variables and moves of the vertex tTS, but we will not write them out explicitly as this would be too tedious. For this reason, we will drop the tensor labels and index markings in the discussion below.

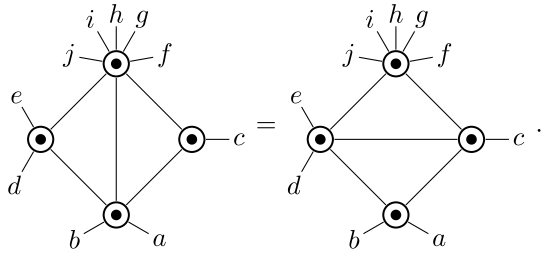
Let us now describe the moves of the vertex tTS. First of all, there are some rather trivial moves: If the star has rotation symmetries, then we demand the associated tensor to have the same symmetries as index permutation moves. If the symmetry also contains a reflection, then by Hermiticity we need to not only permute indices but also complex conjugate. The “real” moves correspond to maps between different planar tensor-network diagrams. While for the 2D standard tTS, any equation between two planar diagrams with disk topology defines a move, this not the case for the 2D vertex tTS. For example, the following does *not* define a move of the 2D vertex tTS,



(11.4)

It is impossible for this to define a move since it is not compatible with the constraint that all plaquettes in a vertex-tTS diagram must be triangles: Imagine applying this move to some larger vertex-tTS diagram. On the left-hand side of the equation, the right-most tensor corresponds to the corner of a triangle in the larger diagram, which has edges d and e. Applying the move inserts an additional edge between d and e, so the triangle becomes a 4-gon, and thus yields an invalid network. Thus, the equation above cannot define a vertex-tTS move.

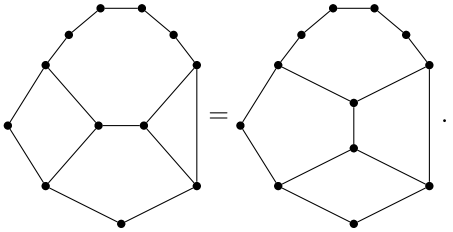
From the considerations above, we see that all moves of the vertex tTS must satisfy the following condition: For each pair of consecutive open indices, the numbers of bonds separating them along the boundary on the left and on the right side have to be equal. In Eq. (11.4), d and e are separated by zero bonds on the left, but by one bond on the right. A straight-forward way to obtain moves that do obey the condition is to take Pachner moves and represent each involved vertex by a tensor. For example, a 2-2 Pachner move yields



(11.5)

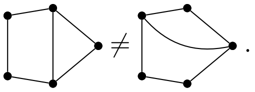
We can write down one such move for each matching quadruple of stars for the vertices at the corners. As such, this yields an infinite set of moves, which becomes finite after limit the maximum valency of vertices via Appendix A of Ref. [25]. Though finite, the set of moves will be quite large, and spelling it out would be too tedious for this thesis.

To get a better feeling for the allowed moves, it is instructive to go to the dual lattice, where n-valent vertices become n-gon faces, and triangles become 3-valent vertices. Neglecting the branching structure, the Pachner move in Eq. (11.5) becomes



(11.6)

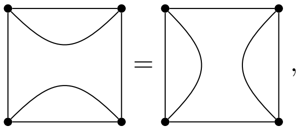
This is a 3-valent re-cellulation where the number of internal edges adjacent to each individual boundary vertex (which is either 0 or 1) does not change, such that all vertices remain 3-valent. Intuitively speaking, the move does not change the vicinity of the boundary. This is the key property that allows us to construct a universality mapping for the vertex tTS in Section 11.5. When we reconsider Eq. (11.4) in this picture, it becomes



(11.7)

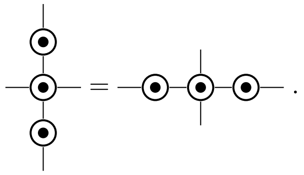
We immediately see that this recellulation does *not* define a move, since there are vertices with no adjacent internal edge on the left side but one adjacent internal edge on the right side, and vice versa.

The dual representation makes it easier to come up with more general recellulations fulfilling the conditions for the vertex-tTS moves. For example, the recellulation



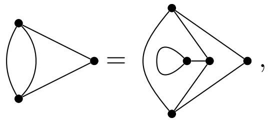
(11.8)

defines the following valid move of the vertex tTS,



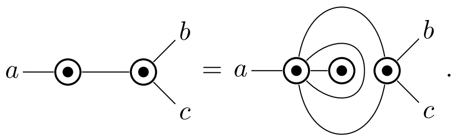
(11.9)

Or, the recellulation



(11.10)

corresponds to the following move,



(11.11)

As these examples show, it is allowed that the tensor-network diagrams of a move do not contain any triangle plaquettes. The fact that on the left-hand side of the last equation the boundary bonds separating a and b as well as a and c are the "two sides of the same bond" is not a problem.

Let us now describe the generalization of the vertex tTS to n dimensions. This is straight-forward apart from one subtlety, namely that the vertex-tTS networks in higher dimensions describe triangulations with not only a branching structure, but also a *dual branching*, as defined below. This dual branching can be canonically chosen in the case of oriented 2-triangulations discussed above. To describe the dual branching, we first need to introduce some basic terminology. The *link* of an x-simplex X in an n-dimensional triangulation is defined as the triangulation of an $n-x-1$-sphere formed by the $n-x-1$-simplices that together with X span an n-simplex. The *star* $\star(X)$ is the configuration of all n-simplices containing X. We also consider all sub-simplices of those n-simplices as part of the star, which is sometimes called "closed star" in the literature. The sub-simplices containing X are called the *internal* simplices. We will usually think of the link and star as simplicial complexes on their own rather than as sub-complexes of the triangulation. Note that X can be contained in an n-simplex multiple times, in which case the star contains multiple copies of this n-simplex, and analogous reasoning applies to the link. As such, the link and star contain the same combinatorial information, as the latter is simply spanned by the former together with an x-simplex. However, if the triangulation has a branching structure, then we will also equip the edges of the star with directions, including the edges that are not part of the link.

A *dual branching* of an x-simplex X is an identification of the star of X with its canonical representative. If this star does not have any symmetries, then there is only one possible dual branching, otherwise there are as many as there are elements of the symmetry group of the star. Hence, there is a unique dual branching for the edges in a 2-dimensional oriented triangulation, as the two adjacent triangles cannot have any symmetries. Note, that reflection symmetries are not allowed if the triangulation is equipped with an orientation.

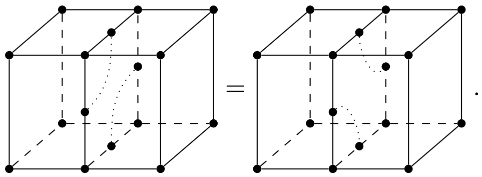
An *x-dual-branched triangulation* is one where all the y-simplices with $y \geq x$ carry a dual branching. More precisely, we equip the star of an z-simplex for $z < x$ in a x-dual-branched triangulation with the dual branchings of its internal z-simplices with $z \geq x$. Consequently, also the standard representatives of stars should be equipped with such dual branchings. Thus, in order to define the dual branching we should proceed inductively, starting defining standard representatives for the $n-1$-simplex stars, then defining standard representatives for the $n-2$-simplex stars containing dual branchings of the internal $n-1$-simplices therein, and so on.

The nD vertex tTS represents 1-dual-branched, branched triangulations as a network with one tensor at every vertex, depending on the star of that vertex, including the 1-dual-branching of the internal simplices of the star. At every edge, there is a bond, whose dimension is allowed to depend on the star of that edge.

The moves correspond to Pachner moves of the triangulations, and consist of the tensors at all the vertices involved in the Pachner move. Even though we are not aware of an official explicit proof in the literature, the consensus seems to be that it suffices to restrict to triangulations with a finite but large enough set of allowed stars, also in higher dimensions. A consequence of this is that the vertex tTS is determined by a finite set of generating tensors and axioms, also in higher dimensions. Even though it suffices to impose Pachner moves, there is a larger family of moves that can be consistently added. To construct these moves, we go from the triangulation to its dual cellulation, where vertices become n-cells whose boundary is dual to a $n-1$-dimensional triangulation. Then, every pair of n-sphere cellulations, that do not differ in the vicinity of the boundary, defines a move. Not differing in the vicinity of the boundary means that the interior part of the link of all boundary vertices remains the same.

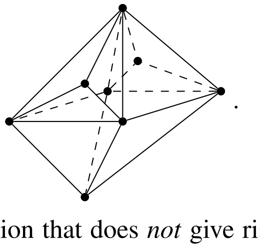
Let us now consider the case of $n = 2+1$ to be more specific. Possible vertex links are two-dimensional triangulations such as the tetrahedron, the octahedron, or a cube with diagonal edges dividing all 6 faces. The corresponding stars can be obtained by adding a central vertex spanning tetrahedra with all triangles of the link. Each star can be equipped with a branching structure in many ways. We also have to equip each star with a 1-dual-branching, meaning that the configuration of tetrahedra surrounding every internal edge is identified with a canonical configuration. For example, if there are l surrounding tetrahedra with edge directions allowing for full $\mathbb{Z}_l$ rotation invariance, then there are l possible choices for the dual branching of that edge. The dual branching of internal triangles is trivial since the two adjacent tetrahedra cannot have any symmetries due to the orientation of the 3-manifold.

The moves are Pachner moves, such as the Pachner move in Eq. (4.18), with five tensors on both the left and right. There is one such move for each choice of stars of the involved vertices, as long as these stars fit together. More generally, consistent moves are dual to recellulations that do not change the vicinity of the boundary. An example of such a move (not drawing branching and dual branching) is



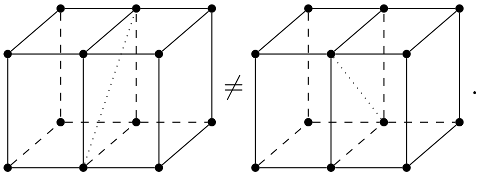
(11.12)

It consists of two vertex tensors on each side, with three indices contracted between them. The dual 3-cells correspond to vertices of the original triangulation whose link is given by



(11.13)

An example for a recellulation that does *not* give rise to a move of the 3D vertex tTS is



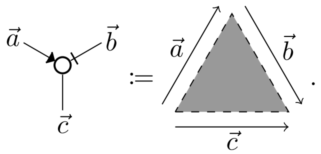
(11.14)

Even though both cellulations have the same boundary, some of the vertices are adjacent to an interior edge on the left but not on the right (or vice versa).

11.2 Failure of mapping from simplicial tTS

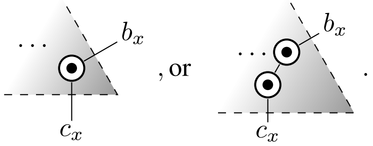
In Section 10.2, we have seen that when we attempt to apply a universality mapping to map to an arbitrary 2D tTS $\mathcal{B}$ from the 2D standard tTS, we potentially encounter the so-called corner problem. In this section we demonstrate that this problem actually occurs for the 2D vertex tTS and thus show that there cannot be any topology-preserving mapping from the 2D standard tTS to the 2D vertex tTS. More precisely, we show that any attempted mapping does neither yield valid networks, nor is compatible with the moves of the 2D vertex tTS.

For simplicity, we assume that the 2D standard tTS is equipped neither with edge nor vertex weights, but the problems discussed below arise independently of this. The attempted mapping associates to the triangle some planar 2D-vertex-tTS diagram with disk topology, and we may place this diagram on the area of a tetrahedron shaded in gray in the following,



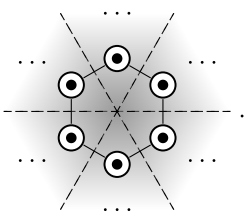
(11.15)

Here, $\vec{a}$ corresponds to a sequence $a_0, \ldots, a_x$ of open indices on the boundary of the vertex-tTS network, ordered according to the arrow direction, and $\vec{b}$ and $\vec{c}$ are given analogously. Consider the bottom right corner and the vertex-tTS network connecting the index components b_x and c_x. In this corner, this network could look like,



(11.16)

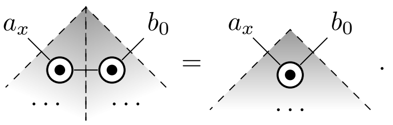
There could also be more bonds separating b_x and c_x. However, no matter what specific network we choose for the mapping, applying the mapping around an l-valent vertex in a triangulation will yield an invalid network for $l > 3$. For example, for a 6-valent vertex and the first of the mappings in Eq. (11.16), we obtain



(11.17)

This network is invalid as it contains a plaquette that is a 6-gon rather than a triangle.

A further problem is that the Pachner moves cannot be derived from the vertex-tTS moves. If we consider the top corner of the move in Eq. (10.14) for the first mapping candidate in Eq. (11.16) we obtain



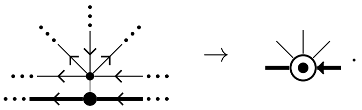
(11.18)

This move cannot be derived from 2D-vertex-tTS moves, as a_x and b_0 are two consecutive open indices that are separated by one bond on the left-hand side, but by no bond on the right-hand side.

The failure of the mapping is analogous, and even more severe, in higher dimensions. For example, for the 3D vertex tTS, a hypothetical mapping assigns to the tetrahedron a ball-topology tensor network, which we may picture as filling the volume of a tetrahedron with the tensor network. In the corner of this tetrahedron we have a fixed configuration of 3D-vertex-tTS tensors. Applying this mapping around a vertex in a 3-triangulation with some link l will not result in a valid 3D-vertex-tTS network for some (and in fact almost all) l. Actually, the problem appears not only at the corners of the tetrahedron, but already along the edges.

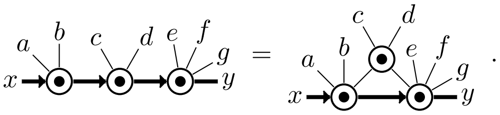
11.3 Failure of cone extension mapping

In this section we discuss why there is no analogue of the cone extension mapping from Section 5.3 for the vertex tTS. In order to be able to talk about a hypothetical boundary cone mapping, we need to extend the vertex tTS with a boundary first, and the simplest way to do so is the following *2D vertex boundary tTS*. This tTS associates tensors to the boundary vertices of a triangulation of a 2-manifold with boundary, for example



(11.19)

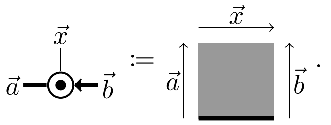
The boundary Pachner moves are implemented by moves such as



(11.20)

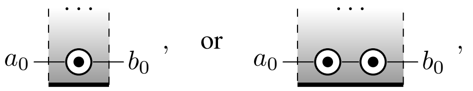
Note that using the universality mapping from Section 10.3, this 2D vertex boundary tTS is equivalent to the much simpler 2D standard boundary tTS. So the vertex boundary tTS is useless for the task of capturing new phases, and we only introduce it to demonstrate the failure of the cone extension mapping. Further, if there was a cone extension mapping from the vertex boundary tTS to the vertex tTS, then we could combine it with the universality mapping in Section 10.3 to obtain a weakly invertible mapping from the 2D standard tTS to the 2D vertex tTS. As such a mapping does not exist according to Section 11.2, also the cone extension mapping cannot exist. It is nonetheless instructive to give a direct argument for why there is no cone extension mapping, which we will do in the following.

Assume there was a topology-preserving mapping from the 2D vertex boundary tTS to the 2D vertex tTS. Consider the network that this mapping associates to a boundary vertex tensor with one adjacent bulk edge. This is a planar network whose topology is a disk with half-space, half-physical boundary. We may place this diagram on a rectangular area, such that three of the sides correspond to one index each, and one side corresponds to physical boundary,



(11.21)

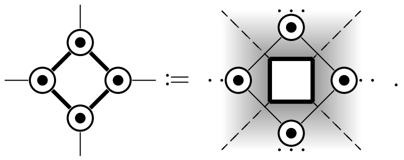
Here, $\vec{a}$ and $\vec{b}$ correspond to the sequences of open indices $a_0, \ldots, a_x$ and $b_0, \ldots, b_x$, respectively, of the network on the right which are ordered according to the arrow directions. We again consider different possible realizations of the mapping. For example, along the physical boundary, from the index components a_0 to b_0, the vertex-tTS network on the right could look like



(11.22)

or like a network with more bonds separating a_0 and b_0.

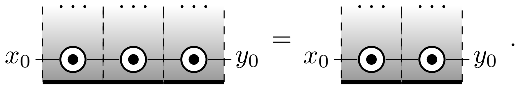
We again show that for every possible choice of the mapping, there exist vertex-boundary-tTS networks that result in invalid vertex-tTS networks under the mapping. To this end, we consider a network with a boundary circle consisting of 4 or more boundary vertices. For, e.g., the first case in Eq. (11.22) we obtain



(11.23)

The resulting network has a non-triangular 4-gon plaquette and is hence invalid. For the second case in Eq. (11.22) we would even obtain an 8-gon.

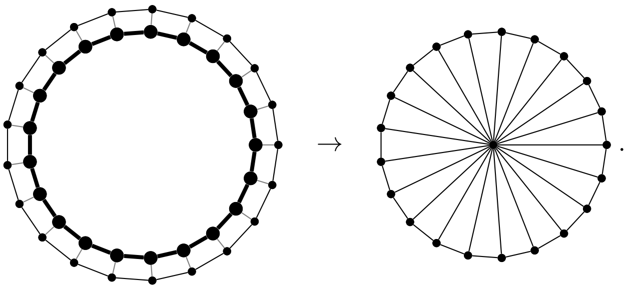
Moreover, there is a problem when we apply the mapping to vertex-boundary-tTS moves such as Eq. (11.20). For example, for the first case in Eq. (11.22), we obtain



(11.24)

This equation cannot be derived from the moves of the vertex tTS as x_0 and y_0 are consecutive open indices that are separated by two bonds on the left-hand side but by only one bond on the right-hand side.

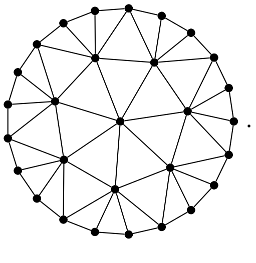
Let us provide some more intuition for why the cone extension mapping works for the 2D standard tTS, but does not for the 2D vertex tTS. The networks of both the 2D standard and the 2D vertex tTS represent triangulations. However, for the 2D standard tTS we can have x-valent vertices for arbitrarily high x, whereas for the vertex tTS we have $x < l$ for some constant l. In terms of triangulations, the boundary cone mapping fills a boundary circle consisting of b edges by dividing the enclosed b-gon into triangles in a pizza-like manner, e.g. for $b = 21$, we have



(11.25)

On the left is a circle of boundary edges surrounded by 2-manifold on the outside, and a hole in the middle. The right side denotes filling that hole with a triangulation of a disk. The essence of why the boundary cone mapping works is that the network associated to the right side has a one-dimensional structure, in particular, the distance between any of the triangles to the boundary is a constant independent of b. This construction does not work for the 2D vertex tTS, as there is a b-valent vertex on the right, so the corresponding vertex-tTS network is invalid for $b > l$.

If we want to close the hole generated by the boundary by a vertex tTS network, we need to fill the interior of the b-edge circle by a triangulation with some maximum adjacency l, such as for $l = 7$,



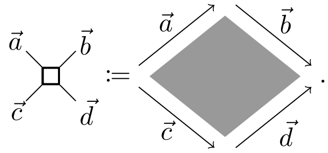
(11.26)

Closing off the boundary with such a triangulation cannot be formalized as a tTS mapping though, as the combinatorial distance between some vertex in the middle and the boundary becomes arbitrarily large when increasing b. To see this, define *removing a layer* of a triangulation of a disk as removing all the triangles that contain a boundary edge or a boundary vertex. Removing a layer cannot decrease the size b of the boundary more than by a constant factor, depending on the maximum allowed adjacency l. Thus, we need to remove at least $\sim \log(b)$ layers until nothing is left. On the other hand, the number of layers we have to remove is proportional to the combinatorial distance of some vertices in the middle to the boundary. So this distance grows unboundedly with increasing b. Note that the tensor network in Eq. (11.26) has a hyperbolic geometry with a constant negative curvature, similar to a *MERA* state which is usually used for models that have not a topological but only a conformal invariance.

In higher dimensions, the cone mapping fails in the analogous way, and this failure is even more severe in some sense: Namely, there is the additional problem that closing the boundary with a cone is not a topological operation but leads to singularities if the boundary is not a $n-1$-sphere. In particular, the link of the vertex at the center of the cone is not necessarily an $n-1$-sphere. Instead, the topology of this link is that of the original boundary, or at least its connected components. This was not a problem for the nD standard tTS as it can be defined on n-manifolds with singularities, but the nD vertex tTS cannot by construction.

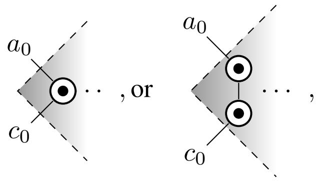
11.4 Failure of the commuting-projector mapping

In this section, we argue that there also is no analogue of the commuting-projector mapping from Section 4.4 for the vertex tTS. We assume that such a mapping exists and then show that it does neither yield valid vertex-tTS networks, nor is it compatible with the vertex-tTS moves. In $1+1$ dimensions, the hypothetical mapping associates to the projector a planar disk-topology vertex-tTS network which we might two onto a diamond-shaped area,



(11.27)

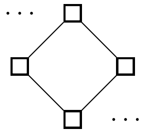
Here, $\vec{a}$ corresponds to a sequence $a_0, \ldots, a_x$ of open indices ordered according to the direction of the arrow on the right, and the same holds for $\vec{b}$, $\vec{c}$, and $\vec{d}$. For some choices of the hypothetical mapping, the left corner of the network connecting the index components a_0 and c_0 looks like



(11.28)

while other choices of mapping can have more bonds separating a_0 and c_0. The same holds for the other corners with the index pairs (a_x, b_0), (c_x, d_0), and (b_x, d_x).

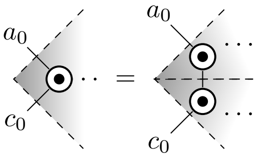
The plaquettes in commuting-projector-TS networks can be arbitrary large l-gons, whereas in the vertex-tTS networks only triangle plaquettes are allowed. However, applying any hypothetical mapping to l-gon plaquettes of a commuting-projector-TS network yields vertex-tTS networks with m-gon plaquettes, such that m gets arbitrarily large when l does. For example, a 4-gon plaquette of the commuting-projector TS,



, (11.29)

yields an m-gon plaquette of the 2D-vertex-tTS network with $m \geq 4$ for any choice of the mapping in Eq. (11.28). Thus, no matter what vertex-tTS network we take on the right hand side in Eq. (11.27), the mapping does not yield valid networks.

Moreover, the moves of the commuting-projector TS can map between l-gon plaquettes for different l, whereas the moves of the vertex tTS always map triangle plaquettes to triangle plaquettes. Therefore, the mapped commuting-projector-TS moves cannot be derived from the vertex-tTS moves. For example, the move in Eq. (4.36) for the first case in Eq. (11.28) in the left corner yields



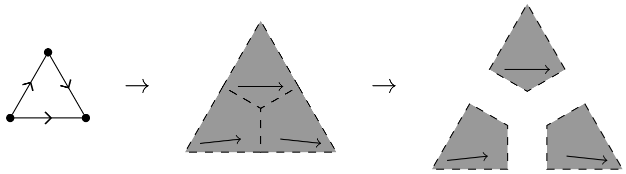
. (11.30)

The resulting move cannot be derived by the vertex-tTS moves, as the consecutive open indices a_0 and c_0 are separated by 0 bonds on the left, but by 1 bond on the right.

11.5 2D universality mapping

In Section 11.2, we have seen that the corner problem described in Section 10.2 actually appears when trying to map the 2D vertex tTS from the 2D standard tTS. In this section, we look at the converse direction and argue that *any* 2D tTS $\mathcal{B}$ can be mapped from the 2D vertex tTS using a universality mapping. That is, we show that the 2D vertex tTS is a universal fixed-point ansatz for $1+1$-dimensional topological order that can emulate any other ansatz. This can be done without running into a corner problem and without any extra conditions on $\mathcal{B}$ like a topological boundary.

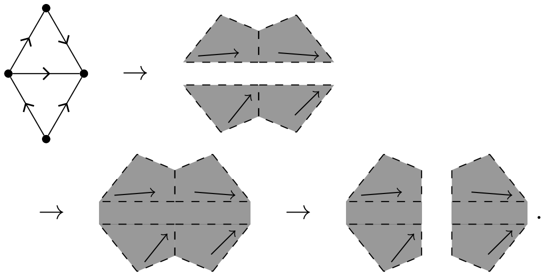
In order to construct the mapping we resort to a cutting and gluing procedure just like in Chapter 10. We start by filling the area of a branching-structure triangle with $\mathcal{B}$-network, such that the corners have combinatorial distance λ and cut this network into 3 kite-shaped parts,



, (11.31)

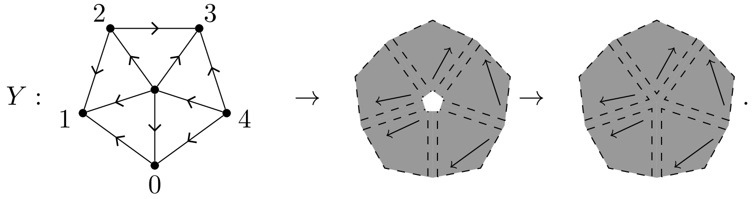
which form the building blocks of our construction. Note, that all three kite-networks are different as the arrows indicate. The cuts of the triangle that define the detailed structure of the building blocks can be chosen unambiguously using the branching structure of the reference triangle on the left. We associate the orientation-reversed and complex conjugated network and decomposition to the counter-clockwise triangle.

To construct the $\mathcal{B}$-network associated to the l-index vertex tensor, we eventually want to glue together l kite-networks around a vertex. Before that, we need to choose a consistent way to glue the kite-networks along the edges. We do this for each star of the edges, i.e., every configuration of two triangles adjacent to the edge. We decompose both triangles into kite-networks, pick the two pairs of kites adjacent to the edge, and place them next to the edge such that they are separated by a gap along the edge. We then fill this gap with some $\mathcal{B}$-network, and then cut the network along a line perpendicular to the edge, for example,



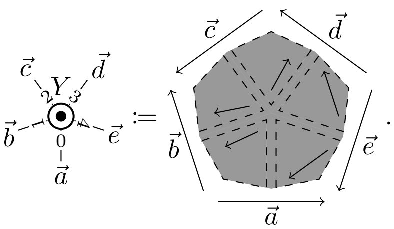
(11.32)

Now, for each star of a vertex, we decompose the surrounding triangles into kite-networks, and keep only the kites nearest to the vertex. We fill the gaps between the kite networks according to Eq. (11.32), such that only a small gap around the central vertex remains. At last, we fill this remaining gap with $\mathcal{B}$-network, e.g., for some star Y we get



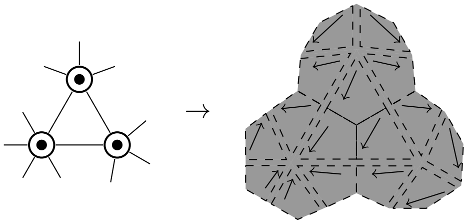
(11.33)

The $\mathcal{B}$-network constructed in this way is used associated to the vertex tensor, for example,



(11.34)

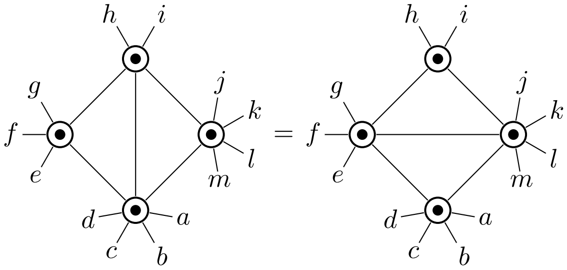
We now verify our claim that this prescription defines a valid mapping. To this end, we consider any patch of vertex-tTS network and the associated $\mathcal{B}$-network, for example,



. (11.35)

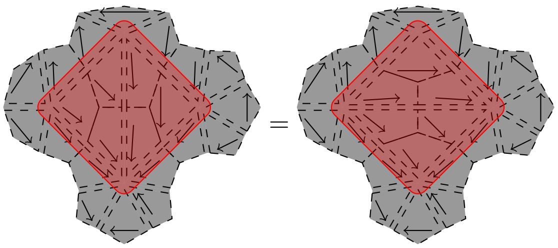
At all the points and lines where the different $\mathcal{B}$-network patches meet, we just re-glue parts in the same way they have been cut before. Thus, the resulting $\mathcal{B}$-network is valid everywhere.

Next we check that the 2D-vertex-tTS can be derived from the $\mathcal{B}$-moves. We consider a move of the vertex tTS, such as



. (11.36)

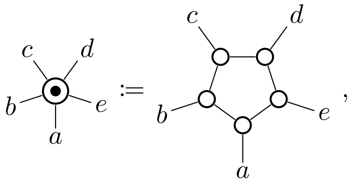
After applying the mapping, this yields the following equation between $\mathcal{B}$-networks,



. (11.37)

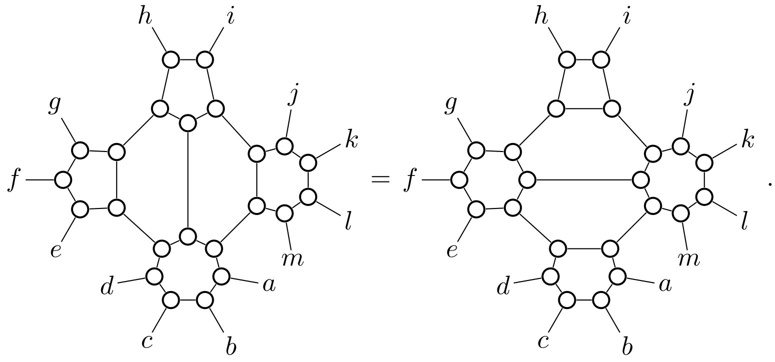
We observe that the two $\mathcal{B}$-networks only differ inside the red shaded region. This region is separated from the space boundary by a combinatorial distance that scales linearly with the fine-graining scale λ. Thus, no matter how complicated the moves of $\mathcal{B}$ are, we can use them to transform the network on the left-hand side to the network on the right-hand side (or vice versa), if we choose a sufficiently large fine-graining scale λ.

Having provided an argument that *any* 2D tTS can be emulated by the 2D vertex tTS via a universality mapping, let us now consider two concrete examples. First, we consider the case of $\mathcal{B}$ being the 2D standard tTS. We can choose a mapping where we replace the n-index vertex tensor with a cycle of n triangle tensors, such as



(11.38)

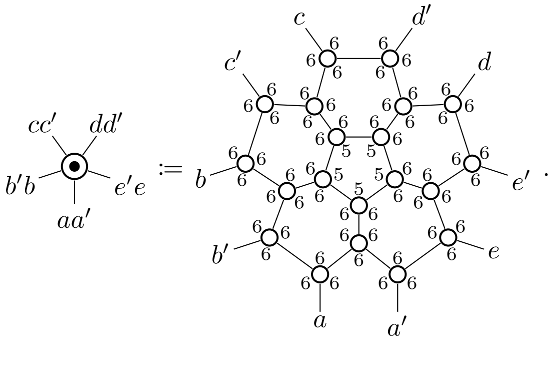
neglecting index markings. If we apply the mapping to, for example, the vertex-tTS move in Eq. (11.36), we obtain



(11.39)

This equation corresponds to a retriangulation of a disk, which can be performed by a sequence of Pachner moves. As a consequence, the mapped 2D-vertex-tTS move can be derived from the 2D-standard-tTS moves. The same holds true for all other 2D-vertex-tTS moves.

As a second example, we take for $\mathcal{B}$ the 2D neighborhood-sensitive tTS presented in Section 10.1. In this case, we need a slightly larger network,



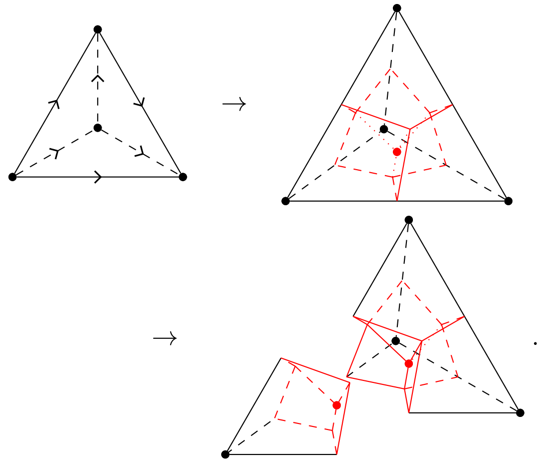
(11.40)

Note that we cannot take a mapping of the same form as Eq. (11.38), because for this mapping, the retriangulation resulting from, e.g., Eq. (11.36) requires Pachner moves involving triangles that contain boundary vertices or edges. If we use the mapping in Eq. (11.40) instead, the corresponding retriangulation can be performed with Pachner moves acting only on triangles that are distant from the boundary.

11.6 3D universality mapping

In this section, we will generalize the vertex-tTS universality mapping from Section 11.5 to $2+1$ dimensions. In the end, we will also sketch the mapping in arbitrary spacetime dimensions n. The mapping for $n=3$ proceeds in four steps.

In step one, we fill the branching-structure tetrahedron with a $\mathcal{B}$-network at a fine-graining scale λ. We then cut the tetrahedron into 4 "kite" volumes, each of which is the convex hull of one of the corner vertices and the centers of each adjacent edge, triangle, and of the tetrahedron itself. Geometrically, a kite is the same as a deformed cube, where one cube vertex is a corner of the tetrahedron and the cube vertex on the opposite side is the center of the tetrahedron. The cut can be chosen unambiguously using the branching structure of the original tetrahedron. The following picture shows the edges of the cut in red, as well as what it looks like removing one of the kites,

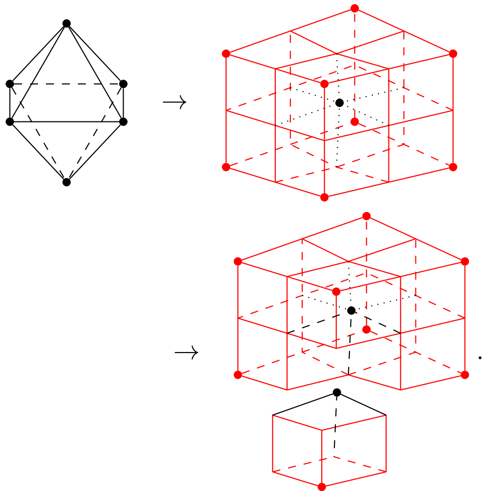


(11.41)

In step two, for each star of a triangle, consider the three adjacent kites of each of the two adjacent tetrahedra, separated by a gap where the triangle is. We then fill the gap and choose a way to cut the result into three pieces, in the same way as we cut the triangle into 2-dimensional kites. The filling can be chosen unambiguously using the orientation of the 3-manifold, and the cut can be chosen using the branching structure on the triangle.

In step three, for each star of an edge, consider the two adjacent kites for each adjacent tetrahedron, with gaps between all of the tetrahedra. Fill the gaps along the triangles between the kites of different tetrahedra using the choices from step two, such that a gap along the edge itself remains. Then, fill this gap with $\mathcal{B}$-network, and cut it into two pieces along the perpendicular plane going through the center of the edge. Note that in order to fill the gap unambiguously we make use of the dual branching of the edge which removes any symmetries of its star, and the branching structure of the edge is needed to unambiguously define the cut.

Finally, in step four, for each star of a vertex, consider the adjacent kite of each adjacent tetrahedron, with gaps between the different tetrahedra. Fill the gaps between the tetrahedra along the triangles and the edges using the choices from step two and step three, such that only a gap at the vertex itself remains. Then, fill the remaining gap with $\mathcal{B}$-network. Note that the choice of filling does not matter since it is distant from the boundary of the surrounding kites, so we do not need a dual branching of the vertices. The so-obtained volume can be identified with the volume dual to the vertex in the dual cellulation. E.g., if the link is an octahedron, then the surrounding kites yield a volume which looks like a cube,



(11.42)

The obtained volume is what we associate to the vertex tensor with the same star. Each index of the vertex tensor corresponds to an internal edge of the star. Each such index is mapped to the composite of all the indices on the face of the volume that is dual to this internal edge. In the above example, the open indices at each face of the cube form a composite index. Due to the dual branching of the internal edge dual to the face, we can choose an unambiguous ordering of indices when forming their composite.

We need to show that the $\mathcal{B}$-network obtained by applying the mapping to a triangulation is valid. To construct this $\mathcal{B}$-network we replace each vertex by the $\mathcal{B}$-network filling its dual 3-cell, and then glue all these 3-cells together. In doing so, we simply re-glue patches of $\mathcal{B}$-network in the same way they have been cut before in the construction. More precisely, at each face of the dual cellulation we re-glue the cut made in step three of the construction. At every edge of the dual cellulation we re-glue the cut from step two, and at every vertex of the dual cellulation we re-glue the cut from step one. We thus see that the resulting $\mathcal{B}$-network is valid everywhere.

Next we need to check whether the mapped moves of the vertex liquid can be derived from the $\mathcal{B}$-moves. Recall that the moves of the vertex liquid are dual to recellulations, where both sides agree at the boundary as well as at the immediate vicinity of the boundary, as shown in Eq. (11.12). Now decompose all the volumes of the recellulation into kites. We will find that all the kites that have contact with the boundary of the cellulation remain the same. The two cellulations only differ at kites in the interior, which are separated from the space boundary by a certain distance. This distance can be made arbitrarily large by increasing the fine-graining scale λ. Thus, no matter what the $\mathcal{B}$-moves are, they can be used eventually for deriving the mapped 3D-vertex-tTS moves if we pick a large enough λ.

At this point, the generalization to $n > 3$ is straight-forward: First consider the intersection of an n-dimensional triangulation with its dual cellulation, yielding a decomposition of all x-simplices and all dual x-cells into kites, which are deformed cubes. Then, the construction proceeds in $n+1$ steps $0, \ldots, n$. In the ith step, for every star of an $n-i$-simplex, we consider the $n-i+1$ adjacent kites of each adjacent n-simplex (for $i=0$ there are none), with gaps in between the kites at different tetrahedra. We fill the gaps at the y-simplices ($n-i<y<n$) between the kites at different n-simplices using the result of step $n-y$, such that only a gap along the $n-i$-simplex remains. We fill this gap with valid $\mathcal{B}$-network, and choose a way to cut the resulting network into $n-i$ parts such that the $n-i$-simplex itself is cut into $n-i$-dimensional kites. When filling, we take the combinatorial distance between different corners to be at least λ.

11.7 Fixed-point models for new phases?

In the previous sections we have developed a framework to describe fixed-point ansatzes that do not necessarily have a topological boundary and are more general on a diagrammatic level than the nD standard tTS equivalent to established state-sum constructions. Thus, the crucial question that presents itself is, whether there are phases that can be captured by our new fixed-point ansatz, but not by any of the established ones. Unfortunately, we cannot give a definitive affirmative answer to this question yet. However, in this section we will list several indications for why believe that this indeed is the case. Most importantly, *chiral* phases cannot be described by established state-sum constructions and are therefore a hot candidate to be represented by a vertex-tTS model.

11.7.1 Chiral phases, commuting-projector models, and topological boundaries

Let us now discuss where we would expect vertex-tTS models to describe topological phases that are not captured by the standard tTS. This is not the case in $1+1$ dimensions, since it is believed that no non-trivial topological phases exist in this dimensions. If we add global symmetries or fermionic degrees of freedom, then non-trivial topological phases do exist. But even then, these phases have been classified to a relatively satisfactory level of rigor, without assuming the existence of a fixed-point model in the first place, see for example Refs. [127, 40, 88]. Since all these phases have gapped/topological boundaries, they are captured by (symmetric or fermionic versions of) the 2D standard tTS and there is no need to consider the (symmetric or fermionic) 2D vertex tTS. However, in $2+1$ dimensions, there is an important class of intrinsic topological phases of matter without gapped/topological boundary, namely *chiral* phases.

It has been conjectured that intrinsic topological phases in $2+1$ dimensions are classified by the UMTC $\mathcal{M}$ describing their anyon content together with a number $c \in \mathbb{Q}$ known as the *chiral central charge* [135]. In other words, it is believed that topological phases are in one-to-one correspondence with 3D atTS models as described in Section 8.4, apart from the choice of the number c. $\mathcal{M}$ determines $c \mod 8$, and phases for the same $\mathcal{M}$ but different c are related by stacking with the so-called E_8 *phase*, which is an invertible intrinsic topological phase with trivial anyon content and $c = 8$. While there is no rigorous argument for this claim, it is supported by many examples for non-fixed-point models that host topological phases corresponding to a large number of UMTCs. For example, the *Kitaev honeycomb model* in Ref. [96] is a non-fixed-point microscopic qudit model that can still be analyzed analytically by mapping it to a free fermion model. In different parameter regimes, this model represents the toric code phase, and also the Ising-anyon phase. Furthermore, many UMTCs can be described by *Chern-Simons theories* [140], which are not fixed-point models but microscopic quantum field theories. By a *chiral phase*, we mean a phase described by a 3D-atTS model (or UMTC $\mathcal{M}$) that does not arise from a 3D standard tTS model (or that is not a Drinfeld center of a unitary fusion category). Note that a sufficient but not necessary condition for being chiral is that $c \neq 0$. Also note that there are many different notions of chiral phases in the physics literature, some of which are equivalent to ours, while others are not. Examples for chiral phases include the Ising-anyon phase in Ref. [96], and all “non-doubled” Chern-Simons theories.

Chiral phases have several properties that make them incompatible with established fixed-point ansatzes, while still being compatible with the 3D vertex tTS. Most importantly, chiral phases do not have gapped/topological boundaries. Again, topological boundaries are conjectured to be classified by 3D-boundary-atTS models, or Lagrangian algebras of the UMTC, as discussed in Section 8.5. Such Lagrangian algebras are known to be in one-to-one correspondence with fusion categories whose Drinfel’d center is $\mathcal{M}$, such that chiral phases are precisely the ones without topological boundaries. Also in the study of non-fixed-point microscopic models for chiral phases, it turns out that no matter what boundary we use, it will always be gapless. In contrast, established fixed-point ansatzes always have at least one “standard” topological boundary via the cone extension mapping, as we have seen in Section 5.3.

Furthermore, the non-zero chiral central charge of chiral phases is closely related to a non-zero *thermal Hall conductance*. It has been argued in Ref. [89] that chiral topological phases with non-zero thermal Hall conductance cannot be realized as quantum spin systems with commuting-projector Hamiltonians. This parallels, but is different from the no-go theorem of Ref. [87] proving the vanishing of the *electric Hall conductance* for commuting-projector models. In contrast, established fixed-point ansatzes always allow for commuting-projector Hamiltonians via a commuting-projector mapping, as we have seen in Section 4.4.

Moreover, there seems to be an obstruction to represent ground states of chiral Hamiltonians with tensor-networks [60], and certainly the classification of phases via *MPO-injective PEPS* [39] is restricted to non-chiral phases. In contrast, established fixed-point ansatzes always have tensor-network representations for their ground states. Namely, restricting the tensor network of a boundary-tTS model to the vicinity of the boundary yields such a tensor-network representation of a ground state. Since established fixed-point ansatzes always have a standard topological boundary via the cone extension mapping, they also have a standard ground-state tensor-network representation.

All of the above restrictions can be circumvented by using the vertex tTS instead of the standard tTS. As argued in Sections 11.3 and 11.4, the constructions for topological boundaries as well as commuting-projector models fail for vertex-tTS models. Furthermore, we have seen in Sections 11.5, and 11.6 that the vertex tTS is a universal fixed-point ansatz. That is, if there are any fixed-point models for chiral phases, then there must also be vertex-tTS models of these phases.

We would like to stress once more that the failure of the commuting-projector and the cone extension mapping is on a purely diagrammatic level. As such, this does not concretely imply that there exist vertex-tTS models that do not possess commuting-projector Hamiltonians or topological boundaries. However, given how well the properties of chiral phases fit to the vertex tTS, it is tempting to believe that such models exist.

Last, we would like to remark that vertex-tTS models also provide a chance to unify two different worlds of classifications of topological phases: The interacting many-body classification using fixed-point models on the one hand, and the free-fermionic classification using band structure topology on the other hand. Indeed, many of the free-fermion phases are chiral, and the existence of "gapless modes" at the boundary is commonly used as indication if not as definition for a non-trivial topological phase. Note that both kinds of phases can already be captured by a non-fully-extended-TQFT framework describing their co-dimension 2 defects, c.f. Ref. [11]. In our formalism, these descriptions correspond to the fermionic/spin and symmetric/homological variants of the 3D atTS discussed in Section 8.4. However, it would be nice to "fully extend" this unified description to the level of microscopic models.

11.7.2 Chiral anomaly

In this section, we discuss how to implement the *chiral anomaly* in potential chiral vertex-tTS models. A chiral anomaly is present if $c \neq 0 \mod 24$, which is "usually", but not strictly always the case for chiral phases. As we discussed in Section 8.4.4, there are two ways to implement a chiral anomaly in the corresponding 3D atTS models: The lazy way is to consider all tensors and axioms only up to global scalar prefactors, which is fine since our path integrals have a pure-state quantum mechanical interpretation and are therefore only "physical" up to a normalization. More precisely, the moves hold up to global phase factors of the form $e^{2\pi i \frac{c}{24} x}$, where x depends on the move. The more insightful way is to add a 4-dimensional internal region $\underline{f}$ that is only defined up to cobordism. This suggests that also for a hypothetical vertex-tTS model, the chiral anomaly should be implemented by either global phase prefactors for the moves, or by the addition of a 4-dimensional bulk.

In Chern-Simons theory, the chiral anomaly is also known as *framing anomaly*. This means that the partition function associated to a manifold does not only depend on its topology, but changes by a phase factor when the metric is changed. As different triangulations for combinatorial manifolds play a similar role to different metrics for continuum manifolds, the situation is analogous to hypothetical vertex-tTS models. However, for Chern-Simons theory, the chiral anomaly is usually dealt with differently: We let the action depend on a cohomological extra structure, similar to an orientation or a spin structure, namely a *Atiyah 2-framing*. This way, the phase factors when changing the metric are canceled.

Inspired by the situation in Chern-Simons theory, let us discuss how the chiral anomaly can be implemented in a hypothetical 3D vertex-tTS model describing a chiral phase. It is known that Atiyah 2-framings are equivalent to P_1-*structures*, see for example Ref. [7]. Here, P_1 denotes the first Pontryagin class, which we discussed in Section 2.7. In general, for a degree-i characteristic class X, an X-structure is a $i-1$-cochain F such that $dF = X[M]$, or if X is a Poincaré dual cellular $n-i$-cycle, F is a $n-i+1$-cycle. For example, orientations are ω_1-structures and spin structures are ω_2-structures. The situation is slightly degenerate for P_1-structures in $2+1$ dimensions, since P_1 is of degree 4, and therefore non-existent on a 3-manifold. In this case, F can be an arbitrary 0-cycle, and $dF = P_1[M]$ is implemented when we consider the equivalence classes of cellulations with 0-cycles. Namely, we can think of a sequence of Pachner moves mapping a 3-triangulation X_1 to another 3-triangulation X_2 as a 4-triangulation Y whose boundary consists of X_1 and X_2. P_1 is a 0-cycle on this 4-triangulation, and it determines the change of F during the sequence of Pachner moves,

$$\Delta F = \int_Y P_1 \; . \tag{11.43}$$

We will sketch later how this is precisely implemented for a triangulation. First we will show how the P_1-structure F can be used to implement the chiral anomaly. Namely, let R describe a Pachner move including the combinatorics of a large-enough but constant-size neighborhood. For every such R we denote the change of F by $\Delta F(R)$. Then, the global prefactor in the corresponding tensor-network equation is given by

$$e^{2\pi i \frac{c}{24}\Delta F(R)} \; . \tag{11.44}$$

In other words, we can make the 3D-vertex-tTS-model dependent on the P_1-structure F in a rather trivial way by simply adding a prefactor

$$e^{2\pi i \frac{c}{24}F(v)} \tag{11.45}$$

to the tensor associated to a vertex v. This F-dependent vertex-tTS model then obeys topological invariance without global prefactors.

Let us now describe how the P_1-structure is connected to the 4-dimensional internal region $\underline{f}$ that we use to model the chiral anomaly in Section 8.4.4. As mentioned, the change of F in a Pachner move is related to the value of P_1 on an according 4-cellulation. $e^{2\pi i \frac{c}{24}P_1}$ on the other hand can be interpreted as an all-scalar 4D-vertex-tTS model. So we can view the hypothetical 3D-vertex-tTS model as the boundary of the all-scalar 4D-vertex-tTS model. Then the global prefactors of the 3D Pachner moves precisely cancel the scalars in the bulk, and the combined model is invariant under 3D boundary Pachner moves without global prefactors. Note that is is known that $\int_Y P_1$ for a 4-manifold Y depends only on the cobordism class of Y, namely

$$\int_M P_1 = 3\sigma(M) \; , \tag{11.46}$$

where σ is a cobordism invariant known as the *signature* of a 4-manifold. All oriented cobordism classes are generated by disjoint unions of the complex projective plane

$\mathbb{C}P(2)$ or its orientation-reversed version, for which we have,

$$\int_{\mathbb{C}P(2)} P_1 = 3\sigma(\mathbb{C}P(2)) = 3\;. \tag{11.47}$$

So the value of the all-scalar 4D vertex-tTS model on $\mathbb{C}P(2)$ equals $e^{2\pi i \frac{c}{8}}$, which is in accordance with the anomaly in the corresponding 3D-atTS-models discussed in Section 8.4.4.

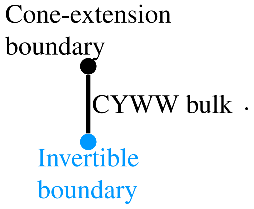
Let us now describe more accurately how the P_1-structure can be implemented on triangulations. As we have argued in Section 2.7, P_1 is a cellular characteristic cycle, and its value on a vertex in a branched and dually branched triangulation depends on the star of this vertex. For a 4-triangulation coming from a sequence of 3D Pachner moves, we need to redistribute the value of P_1 associated to the vertices such that there is one value for each move R. While it is a priori not obvious that this is possible, arguments similar to those in Ref. [68] ensures this. Instead of assigning a $\mathbb{Z}$-value to every star of a vertex, we can work with the differences of $\mathbb{Z}$-values when performing Pachner moves on every star. Imposing Pachner-move invariance, one then finds that this difference is a sum, where every summand only depends on the effect of the Pachner move on the star of an edge. Now, given a 3-triangulation, we make every vertex star into an edge star inside a 4-triangulation. $\Delta F(R)$ is then given by the sum of the difference values of all involved edge stars. The topological invariance of the all-scalar 4D-vertex-tTS model is in accordance with the fact that the accumulated phase factors $\Delta F(R)$ for sequences of moves connecting the same two triangulations have to be equal.

In summary, we want to stress that the Pontryagin class P_1 is naturally formulated as an all-scalar 4D-vertex-tTS model, and cannot be represented as an all-scalar 4D-standard-tTS model. In this context, it seems very natural that models with a chiral anomaly should also be described by 3D-vertex-tTS models.

11.7.3 Chiral anomaly and Crane-Yetter-Walker-Wang models

In the previous section, we have argued that the chiral anomaly in a hypothetical chiral 3D-vertex-tTS model can be implemented by viewing this model as the boundary of a physically trivial all-scalar 4D tTS model. Besides being physically trivial, the 4D tTS model is also invariant under arbitrary surgery operation, and thus invertible as discussed in Section 5.5. In this section, we will see that chiral phases can be represented by the standard tTS if we drop the requirement that the 4D tTS model is physically trivial. Namely, for any 3D atTS model/UMTC, there exists a 4D-standard-tTS model such that 1) the model is invertible in the bulk, and 2) the collection of "boundary anyons" inside the $2+1$-dimensional cone-extension boundary is described by the original 3D-atTS model/UMTC. This model is the Crane-Yetter-Walker-Wang (CYWW) model, which we already mentioned in Section 4.15. More precisely, the CYWW model can be defined for an arbitrary *unitary braided fusion category*, and we only consider the case where this braided fusion category is *modular*, that is, a UMTC. In our framework, this model is simply a TS mapping from the (robust) 3D atTS to the (invertible) 4D tTS.

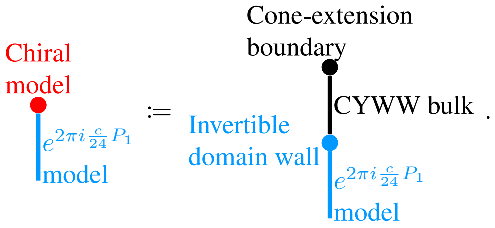
Of course, this construction does *not* directly solve the problem of finding a fixed-point description for a chiral phase: Comparing a standalone $2+1$-dimensional model (which we want) to a boundary of a $3+1$-dimensional model (which we have) only makes sense if the $3+1$-dimensional model is in a trivial phase. In this case we can disentangle the $3+1$-dimensional bulk and obtain a standalone $2+1$-dimensional chiral model. Formally, this argument can be phrased as follows: In order to show that a model is in a trivial phase we need to find an invertible boundary, as discussed in Section 5.4. Assume the modular CYWW model has such an invertible boundary. Then we can consider the mapping from the 3D tTS to the 4D tTS given by taking the cartesian product with an interval between cone-extension and invertible boundary,



(11.48)

In other words, we consider the 3D tTS model obtained from compactifying the 4D tTS model between invertible and cone-extension boundary.

In fact, such an invertible boundary cannot hold without global prefactors if the input UMTC has a chiral central charge $c \mod 8 \neq 0$. This is because the invariant of the CYWW model on a closed 4-manifold Y equal to the number the 3D-atTS model assigns to Y, namely $e^{2\pi i \frac{c}{24}\int_Y P_1}$. If we we want the topological moves of the invertible boundary to hold without global prefactors, we can turn the invertible boundary into an invertible domain wall to the physically trivial all-scalar $e^{2\pi i \frac{c}{24}P_1}$ 4D-vertex-tTS model discussed in Section 11.7.2. In this case, the compactified 3D-tTS model is at the boundary of the all-scalar $e^{2\pi i \frac{c}{24}P_1}$-model as discussed in the previous section,



(11.49)

Note that, unfortunately, the hypothetical invertible boundary/domain wall has the same "problem" as the hypothetical chiral tTS model itself: If the UMTC is chiral, it cannot be described by a standard domain-wall tTS, but needs to be described by some vertex-variant of the domain-wall tTS. Otherwise, the compactification would result in a 3D standard tTS model as well, which cannot be the case. So finding an invertible domain wall/boundary that disentangles a modular Walker-Wang model is not necessarily easier than directly finding a 3D-vertex-tTS model for a chiral phase. However, this approach offers significantly more guidance as opposed to free search for chiral vertex-tTS models, since it aims to realize one specific UMTC.

An interesting development in this light is the finding that some modular chiral CYWW models can be disentangled via so-called *quantum cellular automata (QCAs)*.

Specifically, such QCAs have been found for many abelian UMTCs using the *stabilizer formalism*, starting from the three-fermion UMTC in Ref. [74], later generalized to odd-prime stabilizers in Ref. [75] and to even more UMTCs in Ref. [128]. A QCA is a unitary that maps local operators, acting non-trivially only in some real-space support, to local operators on the same support enlarged by a constant-size margin. The QCAs disentangle the CYWW models in the sense that each projector of the corresponding commuting-projector Hamiltonian is mapped onto an on-site projector $(\mathbb{1}-Z)/2$, where Z is the (generalized) Pauli-Z operator, on a single qubit or qudit. Ref. [74] argues that since the QCA disentangles a chiral CYWW model, it must be a *non-trivial* QCA in the sense that it cannot be written as a local unitary circuit. To this end, it is noted that such a circuit could be terminated near the cone-extension boundary in a light-cone like fashion, yielding a commuting-projector for the boundary alone which is chiral, contradicting the arguments in Ref. [89]. Even though the QCAs in the references above are defined on regular spatial grids, we conjecture that they can be extended to arbitrary spatial triangulations/cellulations. In fact, for the case of the 3-fermion CYWW model, a constructive existence proof for the QCA on arbitrary cubulations has been presented in Ref. [65].

As such, a QCA is *not* a microscopic realization of an invertible domain wall. However, it has been shown in Ref. [119] that any abstract QCA has a concrete representation as conjugation with a *simple PEPO*. If we close all open indices on the disentangled side of this PEPO with the $|0\rangle$ vector, and contract all open indices on the entangled side with the indices coming from the bulk CYWW tensor-network path integral, we indeed obtain a boundary for the corresponding CYWW tensor-network path integrals. The topological invariance of this boundary follows from the fact that it is invariant under applying the local ground-state projector at any point. The latter corresponds to removing/attaching a diamond-like cell of the CYWW spacetime bulk from the boundary. Showing full topological invariance under boundary Pachner moves would require a little more work. Furthermore, the simpleness condition of this PEPO resembles the invertibility condition for the boundary. Unfortunately though, the simpleness condition is a global condition where all open PEPO indices are physical. Thus the condition is unaffected by stacking a standalone 3-dimensional tensor-network path integral onto the 3-dimensional PEPO, and thus cannot guarantee invertibility of the resulting boundary. The simple QCA PEPO in Ref. [119] is obtained by acting with a fixed rank-1 tensor-network superoperator on an arbitrary (as long as the result is non-zero) choice of 3-dimensional PEPO. Depending on this choice, we might indeed obtain a PEPO corresponding to an invertible boundary.

11.7.4 Renormalization and infinite-bond-dimension tensors

A strong but non-rigorous argument for the existence of chiral vertex-tTS models is the fine-graining procedure for tensor-network path integrals discussed in Section 3.5. By blocking larger and larger patches of a non-fixed-point path integral, we obtain path integrals with a smaller and smaller correlation length. So if the fine-graining procedure converges in the sense of Section 3.5, then the result must indeed be a fixed-point model. If chiral phases are topological in the sense of topological invariance, which their description as atTS models strongly suggests, then the resulting fixed-point model will be a tTS model. Then the universality mapping discussed in Section 11.5 and Section 11.6 implies that the fixed-point model can be described as a 3D-vertex-tTS model.

So all in all, this seems like a relatively safe procedure to arrive at a vertex-tTS model for a chiral phase. The weakest point in the argumentation might be the assumption that the fine-graining procedure converges. It seems hard to imagine that there is convergence at all, given the variety of different invertible domain walls S_b we are allowed to apply at each fine-graining scale λ. In particular, we would expect the non-fixed-point model to have an approximate notion of topological invariance, which implies that it is possible to truncate the bond dimension when fine-graining. However, it is well conceivable that the limit of the sequence does not converge to a tensor with *finite* bond dimension. So one might consider fixed-point models consisting of tensors with infinite bond dimension. As we describe in Section 3.5 and around Eq. (3.22), such infinite tensors need to be normalizable for a norm that is compatible with tensor products and contractions. Note that in principle, taking finite versus infinite tensors is independent of taking vertex versus standard-tTS models. In principle, there can be non-chiral tTS models with infinite tensors as well as chiral tTS models with finite tensors. The cone extension boundary for standard-tTS models and the universality of vertex-tTS models work on a purely diagrammatic level, and thus hold equally for finite tensors and for infinite tensors. So even if we allow infinite tensors, we cannot avoid going to a vertex tTS if we want to represent chiral phases, or other phases without topological boundary.

An interesting example for an infinite-bond-dimension tTS model is the standard state-sum construction [41] for the $2+1$-dimensional group cohomology SPT phases for the non-trivial (discontinuous) $U(1)$ 3-cocycles. This model has been studied and found to have a non-zero electric Hall conductance in Ref. [56]. At first glance, the state sum looks like a 3D-standard-tTS model, and evades the no-go theorem in Ref. [87] not by being a vertex-tTS model, but by the fact that the latter only rules out *finite-dimensional* commuting-projector models. However, there seem to be problems with formally interpreting this model as an infinite-bond-dimension 3D-standard-tTS model. After going to the Fourier basis, the homogeneous $U(1)$ 3-cocycle becomes a tensor with 4 $\mathbb{Z}$-valued indices satisfying a decay condition. However, at each vertex there will be a $\mathbb{Z}$-multiplication tensor that does not obey any decay condition. Potentially, this can be resolved by transferring some of the decay from the cocycle tensors to the $\mathbb{Z}$-multiplication tensors. If we do so, however, the state-sum has the form of a 3D-vertex-tTS model, and the modified $\mathbb{Z}$-multiplication tensor at a vertex does not define a commutative algebra anymore and prevents us from reshaping the model into a 3D-standard-tTS model.

11.7.5 Chiral topological codes

An promising development for searching for chiral tTS models or fixed-point path integrals is the existence of chiral topological subsystem codes. These are $2+1$-dimensional Pauli error-correcting codes that have the characteristics of a chiral topological phase. Already more than a decade ago, Ref. [27] presented a subsystem code that appears to belong to a 3-fermion phase. Recently, subsystem codes based on arbitrary (including chiral) abelian anyon theories have been constructed in Ref. [63] using a mechanism of "gauging out" anyons. In Chapter 12 we show how to interpret any code, given as a circuit of measurements and channels, as a tensor-network path integral. For common Pauli codes, this tensor-network path integral is simply the circuit of $+1$-postselected Pauli measurements. Now, subsystem codes per definition do not include a schedule of measurements, and if we choose the "wrong" schedule we end up with a path integral describing a non-chiral phase. However, we believe that for some of the codes in Ref. [63] it is possible to choose a schedule such that the logical dimension matches that of the associated chiral phase. It thus seems tempting to believe that the corresponding path integral does indeed represent a chiral topological phase. It will be very interesting to see whether these path integrals do genuinely represent chiral phases, and whether one can show their discrete topological invariance.

Chapter 12

Dynamic fault-tolerant protocols from fixed-point path integrals

In this chapter, we will discuss a concrete application of the framework developed in the main part of this thesis, and the field of TQFT and fixed-point models in general This application is topological quantum computation, as mentioned in the introduction.

12.1 General formalism

12.1.1 Topological quantum computation

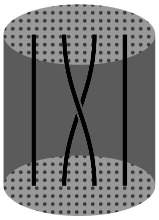
In topological quantum computation, logical information is encoded into the ground state space of a topologically ordered model on a topologically non-trivial spatial configuration. A computation is performed by "adiabatically deforming" the spatial configuration changing its topology. In total, we get a non-trivial spacetime configuration that encodes the computation. It is in principle possible to perform computation by only changing the topology of some bare spatial manifold. However, the set of accessible logic gates becomes much richer if we introduce defects, such as boundaries, anyons, domain walls, twist defects, and so on as discussed in Chapters 5 and 6. We refer to such defects as *computational defect* to stress that they serve a very different purpose from the *syndrome defects* that we will later use to transform path integrals into sequences of measurements. Computational defects are also necessary for implementing computation in practice where we need to faithfully embed the topological manifolds into the Euclidean space we happen to live in. In our framework, the space and spacetime configurations are then extended manifolds of a certain type, c.f. Section 2.4, and we will refer to these as the *computational extended manifold*. Topological order has been shown to be robust under arbitrary local perturbations [31]. The idea behind topological quantum error correction (QEC) is that this robustness carries over to the error-correcting protocols.

There is two main levels on which topological quantum computation is studied: On the one hand, people engineer concrete microscopic error-correcting protocols consisting of local measurements, channels, and a classical decoders. On the other hand, one can just assume that there is a way to implement the adiabatic process as an error-corrected protocol, and study the logical gates performed by an operation. Our formalism can be used to systematically study both.

While this chapter is about how to obtain concrete error-correcting protocols from tTS models or fixed-point path integrals, let us first to describe how tTS models can be used to calculate the logical operation performed by an topological process. Since topological quantum computation happens only inside the ground-state space (possible enhanced by defects), the result of a computation can be obtained by simply evaluating the tensor-network path integrals discussed in the main body of this thesis. Let us give a simple example for this. As we said, the set of defects that we use for a protocol of topological quantum computation form an extended manifold type. Let us consider the following extended manifold type,

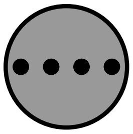
$$a:(3,\varnothing)\ ,\quad b:(0,\bullet)\ ,\quad c:(1,\bigcirc)\ . \tag{12.1}$$

That is, extended manifolds are 2-manifolds with boundary and embedded, normally framed, anyon worldlines. Now, the computational extended manifolds are extended manifolds with a final and initial spatial configuration, for example,



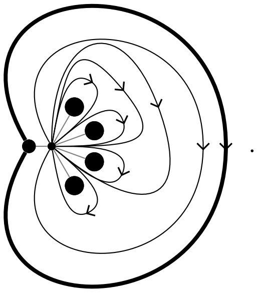
. (12.2)

The picture shows a cylinder with space boundary on the bottom and top connected by four anyon worldlines, and physical boundary at the sides. It corresponds to a process where two out of four anyons on a disk are exchanged. Both initial and final space boundary are given by the following extended manifold,



. (12.3)

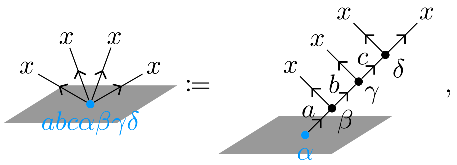
Now imagine we are given an arbitrary 3D tTS model with boundary and anyons as described in Sections 5.1 and 6.2. Then we calculate the logic operation by simply cellulating the spacetime manifold and evaluating the tTS model. First, we find a (minimal) extended cellulation of the initial and final space boundary,



(12.4)

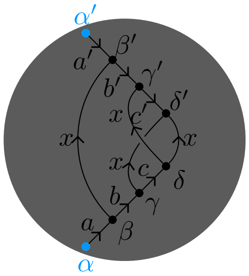
This extended cellulation consists of one boundary edge, four anyon vertices, and three bulk triangles. Next, we need to choose a basis for the ground state space on this minimal extended cellulation. To this end, we take the cartesian product of the cellulation with a single edge. This yields a 3-dimensional cellulation consisting of 9 tetrahedra, 2 boundary triangles, and four anyon edges. The resulting weight defines the 1D tTS model obtained by compactification with Eq. (12.3). In other words, the logical ground state space is the same as the space of point defects whose link is given by Eq. (12.3). The 1D-tTS model is a projector, and we choose a basis for the support of this projector. Finally, we choose a cellulation for the computational extended manifold in Eq. (12.2) itself. Evaluating the path integral on this cellulation yields an operator from the initial to the final space boundary. Conjugating this operator with the choice of basis yields the desired logical operation. For example, if we use a Fibonacci model from the end of Section 4.14 for a, its standard cone-extension boundary for b, and the τ anyon for c, then we get a 2-dimensional logical space. The logic gate implemented this way thus acts on one logical qubit, and its precise form depends on the chosen basis of the logical space.

Note that the logical gate can also be computed using 3D atTS models rather than 3D tTS models by gluing the computational extended manifold from the generators rather than cellulating it. In many cases, these computations turn out to be less tedious. In the above example, we can emulate fusion vertices with the link in Eq. (12.3) by



(12.5)

where x is the name of the anyon in Eq. (12.2). With this, the logical operation in Eq. (12.2) equals the evaluation of the 3D (anyon+boundary) atTS on the following extended manifold,



. (12.6)

We do this by gluing together the generating extended manifolds in Sections 8.4.3 and 8.5.

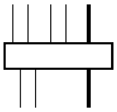
A computational extended manifold together with topological phases or fixed-point models assigned to all the topological defects gives us a way to perform a logical gate that we would hope to be fault-tolerant due to the robustness of topological order. However, it is non-trivial to turn this computational extended manifold into an actual microscopic fault-tolerant topological QEC protocol. Like fixed-point models for topological phases, topological QEC protocols can be described by discrete (tensor-network) path integrals in spacetime. Namely, quantum error-correcting protocols on the lowest level are are mixed-state circuits of quantum channels and measurements. However, the two path integrals are of a very different nature: Whereas QEC circuits are path integrals with a real time component on a mixed-state level, fixed-point path integrals are in imaginary time and on a pure-state level. In this chapter, we present a powerful picture for topological QEC, at whose core is the relation between the two types of discrete path integrals. Concretely, there exists a history of "trivial" measurement outcomes (often $+1$ for Pauli based codes) such that the QEC circuit becomes a fixed-point path integral. The other histories of measurement outcomes then correspond to the same path integral including different pattern of topological defects such as anyons. We will call these defects *syndrome defects* to distinguish them from the computational defects. The path integral is invariant under certain local deformations of syndrome defects, giving rise rich equivalences between different syndrome-defect pattern. The corrections correspond to the insertion of additional syndrome defects, which are chosen by the classical decoder to ensure that the total pattern of defects is equivalent to the trivial one. The correspondence between QEC circuits and topological path integrals with syndrome defects provides a single simple criterion for topological fault tolerance.

The formalism presented in this chapter has two major practical applications. The first application is to systematically analyze existing codes. In particular, the correspondence to path integrals can be used to assign a topological phase to any topological code. This phase determines the logical dimension on different topologies, the possible boundary conditions, anyons, or other sorts of defects that can be introduced, as well as the logical operations that can be performed. Codes within the same phase can be seen as distinct microscopic representations of one another, which we do in Section 12.2. To illustrate this, we focus on recently developed *Floquet codes* [79, 92, 53, 2]. These are specified by a sequence of gauge checks measured in a fixed schedule. Since the checks are non-commuting, they can be analyzed using the formalism of *subsystem codes* [100, 27, 32]. However, due to the fixed schedule, they manage to protect a certain number of logical qubits even though the subsystem formalism predicts less or none at all. Lately, there has been a quest to better understand the relation between stabilizer, subsystem, and Floquet codes, and between different Floquet codes among each other. Our formalism helps to establish direct relations between different codes, often by finding that they to belong to a common phase. Concretely, we find that the stabilizer toric code, the subsystem toric code, and the CSS Floquet code correspond to the same path integral on different spacetime lattices, and thus belong to a single code family in our spacetime perspective. This can be seen as a spacetime analogue of viewing stabilizer toric codes on different spatial lattices as part of a single code family. We also find that the underlying path integrals of the CSS Floquet code and the honeycomb Floquet code are equal up to a local change of basis. All four codes belong to the toric code phase. They only differ by the microscopic representation of the underlying path integral, as well as by the locations of the defect segments corresponding to the non-trivial measurement outcomes.

The second application is to systematically construct new codes, which we do in Section 12.3. Roughly speaking, we start with a fixed-point path integral and interpret it as a non-unitary circuit. Then we turn each non-unitary operator into an instrument that measures the absence or presence of a syndrome defect. The circuit of instruments then defines a fault tolerant dynamic code. By making use of the rich mathematical theory of fixed-point models developed in the main part of this thesis, this yields a great variety of new dynamic topological codes. First, we can start from different models of different tTS (or fixed-point families), corresponding to different phases. Further, we can leverage the exact combinatorial topological invariance of tTS models, to put the fixed-point path integrals on arbitrary spacetime lattices. Finally, even if the path integral and lattice are fixed, they can be turned into a non-unitary circuit in various ways by choosing different causal orderings. An interesting feature of our approach that goes beyond much of the quantum error-correction literature is that there is no necessity for the resulting codes to be based on Pauli/Clifford measurements or operations. Concretely, we illustrate the capability of finding new codes through two examples. First, we construct a Floquet version of the $3+1$-dimensional toric code that uses only 2-body XX and ZZ measurements. The code lives on a triangulation with 4-colorable vertices, with a qubit on every left-handed tetrahedron. In each of 8 rounds we perform measurements on the qubits adjacent to each edge of a certain type. Second, we present a non-Pauli dynamic code based on the double-semion Turaev-Viro path integral. We sketch a presentation of this code as a circuit of common 2 and 3-qubit gates and measurements. Due to its relatively large spacetime overhead it merely serves as an illustrative example rather than as a practical QEC code.

12.1.2 Dynamic codes

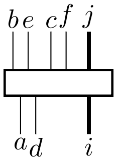
In this section we review topological QEC as a dynamic processes, or more technically, as a circuit executed in spacetime. This might be at the same time the most low-level and the most general point of view. An error-correcting process needs to be able to filter out noise introduced into the system by extracting entropy. Thus, the corresponding circuits are circuits of quantum channels rather than unitaries. It is useful to consider channels that simultaneously act on classical and quantum degrees of freedom, even though these can always be embedded into purely quantum channels. Mathematically, such a quantum/classical hybrid channel is a tensor where every classical or quantum degree of freedom, either at the input or the output of the channel, corresponds to one index. More precisely, a d-dimensional qudit is represented by a pair of d-dimensional indices, one for the ket and one for the bra part, whereas a classical d-state degree of freedom is just a d-dimensional index. For example, a channel with one quantum and one classical input, and two quantum and one classical output can look like:



, (12.7)

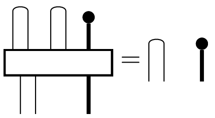
where the time direction is from bottom to top, like everywhere in this chapter. As shown, we draw classical indices with thick lines.

A proper channel needs to fulfill two conditions: First, it needs to be *completely positive*: For every fixed configuration of the classical indices, block all ket indices and all bra indices such that the tensor becomes a matrix. This matrix has to be non-negative, for example,



$$\rightarrow [M^{ij}]_{(abc),(def)} \geq 0 \quad \forall i, j \ . \tag{12.8}$$

Second, it needs to be *trace preserving*: When closing all quantum output (double-)indices with a trace, and all classical output indices with a sum, we obtain a trace and sum at all classical and quantum input indices, for example,

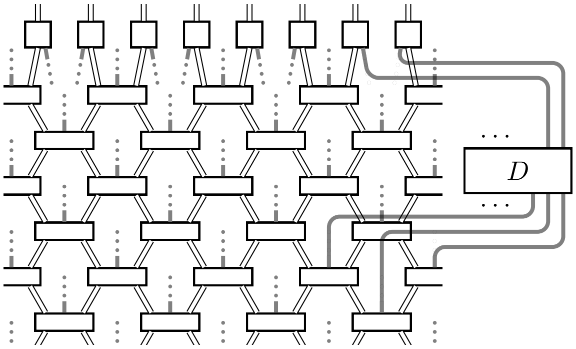


. (12.9)

Here the black dot is the δ-tensor in Eq. (3.3) with one index, that is, a vector with all entries equal to 1.

In *topological* QEC, we demand the circuit to be geometrically local. Only then it is fair to assume that also the noise occurring in the process is local. The great achievement of topological QEC is fault tolerance with respect to arbitrary local noise, and thereby any noise that is physically possible. We will refer to this type of QEC, where the complete circuit of quantum/classical channels is geometrically local, as *fully local QEC*. Topological QEC has the additional property of being uniform in spacetime, or at least to scale in a uniform way. Fully local topological QEC is not only of practical but also of fundamental physical interest since it might provide a model for the process of cooling a topologically ordered material. Fully local QEC can also be viewed as self-correction using engineered dissipation, formulated in discrete time. Examples for fully local QEC circuits are given by cellular automaton decoders [101]. While fault tolerant fully local decoders are known exist in $4+1$ dimensions, the situation is unclear in $3+1$ and $2+1$ dimensions.

Since the feasibility of fully local topological QEC in low dimensions is an open question, we consider *quantum-local QEC* as a second type of topological QEC, where only the quantum part of the circuit is assumed to be local. Quantum-local QEC consists of a geometrically local circuit of channels with additional open classical inputs and outputs. These inputs and outputs are then coupled to a purely classical *decoder* that is not implemented by a classical circuit in the same spacetime, but treated as a black box that can be evaluated instantly and without noise. In practice, the efficiency of this decoder is of course still of great importance, and any reasonable decoder should be executable in at most a polynomially larger spacetime. An example for this is minimum-weight-perfect-matching decoding of the toric code: The quantum parts, namely the stabilizer measurements and corrections, are local, while the classical decoding algorithm has more-than-constant runtime even if we allowed for instant non-local communication. From a fundamental point of view, quantum-local QEC is not scalable with a fault tolerant threshold. This is because for large enough system sizes the quantum circuit has to wait for the results of the decoder, and during this waiting time additional errors accumulate. Nonetheless, quantum-local QEC might have a practical impact, since current implementations of qubits are by orders of magnitude larger, slower, and noisier, than classical information technology. A toy example for quantum-local QEC in a $1+1$-dimensional spacetime looks like



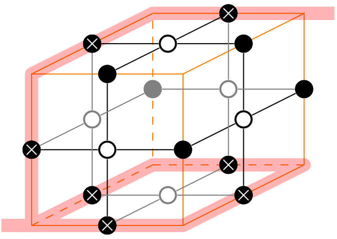
, (12.10)

where we have semi-transparently drawn some of the classical bonds connecting the circuit and the decoder D, and omitted the remaining ones. Note that for a real topological error-correcting circuit we would need at least $2+1$ spacetime dimensions. The shown example has a special layout where we first apply only hybrid channels without classical inputs for a time $T \sim L$. Physically, such hybrid channels are known as *instruments* in quantum information theory and describe measurements, which are 2-qu-d-it measurements in the example above. We will refer to the recorded classical outputs/measurement results as *spacetime syndrome*. Then, at time T, we perform a constant-depth *correction* layer of quantum channels with an additional classical input, which are single-qu-d-it operators in the example above. The inputs to these correction channels are obtained from applying the decoding algorithm D to the spacetime syndrome.

Note that in general, corrections could also be applied in every time step like measurements, and not only after a time $T \sim L$. This might be necessary for example for topological error correction based on non-abelian phases. However, for all examples considered in this paper, a layout as in Eq. (12.10) works.

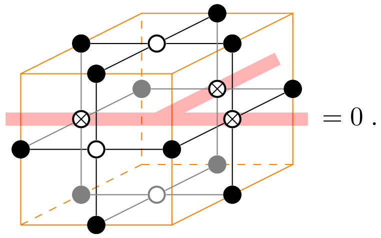
12.1.3 From anyons to 1-form symmetries

Before we get to describing the core formalism in the next section, let us first introduce an example for syndrome defects in a fixed-point path integral. We will in fact use these syndrome defects for all the examples in Section 12.2. The underlying fixed-point path integral will be a 3D face-edge tTS, namely the toric code as defined in Section 9.2. The syndromes defects are the e and m anyons defined around Eq. (9.46). For their usage as syndrome defects, we will give a slightly different interpretation to these anyons, namely as *projective 1-form symmetries*. A 3D tTS with a (projective) 1-form symmetry assigns tensor networks to 3-cellulations that are decorated with a G-valued 2-cocycle, which we refer to as the *1-form symmetry defects*. Here G is an abelian group, and we may equivalently use a 1-cycle instead of a 2-cocycle. The tensors at a point are allowed to depend on the 2-cocycle (or 1-cycle) within a constant-size neighborhood. In addition to Pachner moves, the path integral is invariant under homology moves that locally change the 2-cocycle or 1-cycle. In our example, the e and m anyons are the 1-form symmetry defects, where we represent e as a 1-cycle and m as a 2-cocycle, with $G = \mathbb{Z}_2 \times \mathbb{Z}_2$. The dependence of the path integral on the 1-form symmetry is as simple as possible: At every edge/face carrying an e/m anyon worldline, we replace the δ/$\mathbb{Z}_2$-tensor with the e/m anyon tensor defined in Eqs. (9.41) and (9.45). A homology move is shown in Eq. (9.43). At the end of the day, the only thing that changes when we interpret e and m anyons as 1-form symmetries, is that now instead of worldline configurations as in Eq. (9.43), we can put e anyons on every 1-cycle, such as



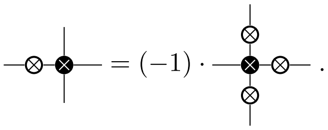
$$. \qquad (12.11)$$

The e and m anyons/1-form symmetries have one additional property that is important for their usage as syndrome defects: Namely, consider the path integral where the configuration of defects is not a 1-cycle or 2-cocycle, but only a 1-chain or 2-cochain. That is, the (co-)cycle condition is violated at some vertex or volume. Then the path integral around this point evaluates to 0. For example, the tensor network around a cube evaluates to 0 if the number of adjacent m worldline segments is odd,



(12.12)

That is, the syndrome defects obey both an equivalence relation Eq. (9.43) and a constraint Eq. (12.12). Note that the 1-form symmetry is *projective* since the homology moves only hold up to a global prefactor: If we move an m anyon worldline past an e anyon worldline, we get a factor of -1,



(12.13)

Note that in a general model anyons can only be interpreted as 1-form symmetries if the anyon fusion rules form a group (which is then abelian). On the other hand, syndrome defects are by far not limited to anyons. Topological defects of other dimensions can be used as well, such as the membrane defects in the $3+1$-dimensional toric code.

12.1.4 From path integrals to circuits

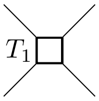
As discussed in Section 12.1.1, topological quantum computation effectively executes the imaginary time evolution of some topological phase on some spacetime manifold, possibly including computational defects. However, in the real world, we can only perform real-time evolution. Real-time evolution can be described by a tensor-network path integral as well, namely as a unitary circuit. However, the tensors of the imaginary-time path integrals are not at all unitaries, and therefore it is impossible to execute the Euclidean path integral in the real world. In this paper, we will understand how topological QEC is precisely a way to solve this problem. That is, topological QEC constructs a real-time path integral that is equal to a given imaginary-time fixed-point inside the ground-state subspace. As argued in Section 12.1.2, the resulting real-time path integrals will in fact not be unitary but circuits of quantum/classical hybrid channels.

Let us now try to formalize the relation between the imaginary-time path integrals and a corresponding fault-tolerant real-time fully local uniform QEC circuit on a high level. To this end, we consider the "transfer" operator corresponding to executing one time period of the circuit. The following three conditions should hold: (1) There is a set of highest-magnitude eigenvalues, which are contained in an interval that shrinks exponentially with the system size L. (2) The remaining eigenvalues are separated from the highest-magnitude subset by a gap that shrinks at most polynomially with L. The dimension of the high-magnitude subspace equals the ground state degeneracy of the Euclidean fixed-point path integral of the phase. (3) In order to perform computation, we consider a circuit that varies in time. The circuit acts like the imaginary-time path integral when restricted to the high-magnitude subspace, and decouples from the orthogonal complement, up to an error exponentially small in L.

Note that for the transfer operator of the imaginary-time evolution, the gap in (2) does not shrink at all, but is constant. However, for a real-time fully local QEC process, this gap must shrink at least like L^{-1} due to the finite propagation speed of information. Namely, if we insert an "error" operation of size $\sim L$ into the circuit, then it takes time $\sim L$ to correct this error and return to the steady state. In contrast, a gapped operator returns to the steady state from any starting point at a system-size independent rate. For a quantum-local circuit, a similar relation could be formulated, though one might need to consider the transfer operator for an extended time T instead of only one time period.

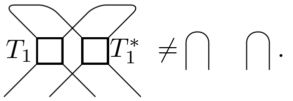
Unfortunately, the three conditions formulated above are neither simple to verify for a generic circuit, nor do they directly help with constructing such QEC circuits. Furthermore, it is unclear whether fully local QEC is possible in physical spacetime dimensions $n \leq 4$. Below, we will describe an explicit general method to construct topological QEC circuits from topological fixed-point path integrals. More precisely, we will only construct the quantum part of the QEC circuit, which is then coupled to a classical decoder. Which classical decoder works depends on the nature of the syndrome defects used, but for all cases in this chapter a global classical decoder will be necessary to achieve fault tolerance. For all examples studied in this chapter, a *minimum-weight perfect matching* decoder will do the job.

We start by putting the path integral on some regular lattice and choosing a time direction. Then we interpret the tensor network as a geometrically local circuit of operators, where each operator corresponds to a single tensor, or a patch of a few tensors. The indices of each tensor or patch are divided into input and output in accordance with the chosen time direction. This can always be done, however the resulting operators, like



, (12.14)

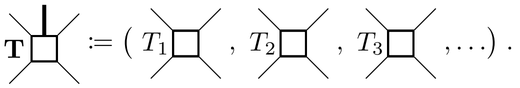
are not in general unitaries, or equivalently, stacking two copies does not result in a channel that is normalized as in Eq. (12.9),



. (12.15)

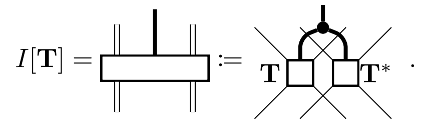
In fact, it does never happen that all operators are unitary, since the operator corresponding to a full layer of imaginary-time evolution is a projector of low rank, and thus not a unitary.

Even though T_1 does not define a channel, it can always occur as part of an instrument. To this end, we choose further tensors $T_2, T_3, \ldots$, that we combine into one single tensor using an additional classical output index,



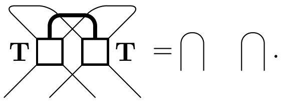
(12.16)

We then use this tensor to define an instrument,



(12.17)

The small dot on the right denotes a δ-tensor as defined in Eq. (3.3).The normalization condition in Eq. (12.9) of this instrument reduces to the following condition for T:



(12.18)

In other words, we are looking for tensors $T_2, T_3, \ldots$, such that the collection $\mathbf{T} = (T_1, T_2, T_3, \ldots)$ forms an isometry.

We now turn the fixed-point path integral into a circuit of instruments as in Eq. (12.17). If we happen to always get the trivial measurement outcome corresponding to T_1, then we have successfully executed the imaginary-time fixed-point path integral. However, if some of the outcomes are non-trivial, we have performed another path integral including some tensors $T_2, T_3, \ldots$, and need to apply corrections. In order to know how to correct non-trivial outcomes, also these must correspond to an exactly solvable fixed-point path integral of some sort. This is where we use the syndrome defects, such the e and m anyons: We choose $T_2, T_3, \ldots$ such that each of these tensors corresponds to a piece of fixed-point path integral that includes one or more segments of syndrome defect. Then every configuration of classical outputs corresponds to a topological path integral with a pattern of syndrome defects. The corrections are then implemented by classically controlled operations in the circuit that insert additional segments of syndrome defects depending on the classical control. This motivates the following definition.

Definition 27. *A* fixed-point path integral code *is a uniform geometrically local circuit of quantum channels with additional classical inputs and outputs, such that the following holds:*

- *When fixing a configuration of classical inputs and outputs, the circuit becomes a mixed-state tensor-network path integral. This path integral is a stack of two copies of the same (pure-state) path integral, with one of them complex conjugated.*
- *This path integral is (in the same fixed-point phase as) a fixed-point path integral for a topological phase, including a pattern of syndrome defects. This pattern only depends locally on the classical inputs and outputs.*

In order to turn a fixed-point path integral code into a full error-correcting protocol, we have to couple the classical inputs and outputs to a classical decoder D. Very vaguely speaking, the resulting process is error correcting if D yields a total defect pattern (formed by the outputs and inputs) that is equivalent to the trivial one. If there is noise, the total defect pattern does not fulfill the local constraints, so instead we take the closest defect pattern that does. More concretely, let us give a decoder that works if the syndrome defects are i-form symmetries, which is the case for all examples given in this paper. This can be viewed as a generalization of decoding the toric code in the presence of measurement errors [57].

Proposition 7. *A fixed-point path-integral code whose syndrome defects are (co-)cycles can be turned into a complete fault tolerant process as follows. The overall circuit layout is that of Eq.* (12.10)*, where we first record measurement outcomes for a time $T \sim L$ (L is the linear system size), and then perform corrections at time T. Thereby, we need to insert enough controlled operations at time T to be able to close off any measured defect pattern. The classical decoder D is given as follows:*

1. *Consider the (co-)chain(s) corresponding to the recorded spacetime syndrome by definition of the fixed-point path integral code. Choose a minimum-weight fix turning the (co-)chain(s) into (co-)cycle(s). Thereby, treat the time-like boundary at time T as "open", such that (co-)cycles can freely terminate there. In contrast, treat the initial time-like boundary at time* 0 *as "closed", such that (co-)cycles are not allowed to terminate there.*

2. *Consider the endpoints of the (co-)cycle(s) at time T. Choose a set of defect segments at time T that together with the fixed (co-)cycle(s) in the spacetime forms homologically trivial (co-)cycle(s). This set of defects determines the input to the classically controlled correction operations.*

Let us give a rough argument for why this process has a fault tolerant threshold under local noise, a detailed proof will appear elsewhere. If we perform the circuit without noise, then the classical outputs correspond to a defect pattern consisting of (co-)cycle(s). Otherwise the path integral evaluates to zero as in Eq. (12.12), and the corresponding configuration of outcomes is measured with probability zero. However, if we perturb the circuit by adding (weak) noise, the (co-)cycle(s) are (slightly) broken. We find that (1) the probability that they are broken everywhere inside a connected region is exponentially small in the size of that region, (2) two cycles of different homology classes differ inside a region of at least size $\sim L$, and (3) the number of connected regions of size L is at most exponential in L. Thus for weak enough noise, the probability for the minimum-weight fix to yield the wrong cohomology class is exponentially small in L.

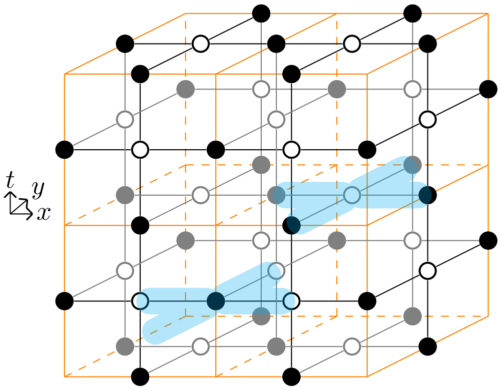
For all examples in this chapter except for Section 12.3.1, the syndrome defects are anyon worldlines. In this case, the correction operators closing the spacetime string net pattern at time T are known as *string operators*. Fixing the string net pattern in the presence of noise means pairing up the string endpoints in spacetime. A polynomial-time algorithm solving this problem is known as *minimum weight perfect matching* [61].

12.2 Known codes in terms of path integrals

In this section, we consider four different examples of fixed-point path-integral codes, which we all find to be equivalent to existing codes, namely the stabilizer toric code, subsystem toric code, CSS Floquet code, and honeycomb Floquet code. The first three examples are all based on the toric-code path integral introduced in Section 9.2, which we put on different spacetime lattices with different choices of time direction. The fourth example differs from the previous ones only by a local change of basis of the tensor-network path integral.

12.2.1 Stabilizer toric code

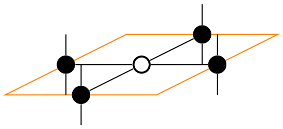
As a first example let us consider the first of all topological error-correcting codes, namely the toric code on a square lattice [95, 57]. The underlying tensor-network path integral is the toric-code path integral from Section 9.2 on a cubic lattice, whose unit vectors we call x, y, and z. The time direction t is coincident with z,



, (12.19)

where the background cubic lattice is in orange and the tensor-network diagram is in black. We now view the tensor-network path integral as a circuit of operators, where each operator corresponds to one or a few tensors. There are two types of operators, as marked above in semi-transparent blue. Both operators act on 4 qubits that correspond to t-directed bonds in the tensor-network diagram. Specifically, there is an operator T_1 at each xy face, and an operator V_1 at every t edge,

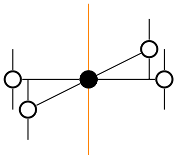
$T_1 :=$



,

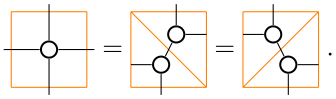
(12.20)

$V_1 :=$



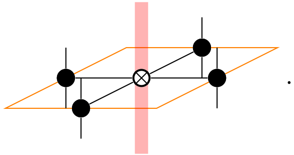
.

Note that these diagrams are identical to well-known ZX diagrams for the vertex and plaquette terms of the toric code [71, 93, 30]. In order to get the decomposition, we need to split up all the 4-index $\mathbb{Z}_2$ tensors at the xt and yt faces into two 3-index $\mathbb{Z}_2$-tensors,



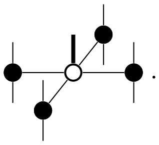
(12.21)

As shown, this splitting up can be represented geometrically as dividing each plaquette into two triangles. After this, V_1 corresponds to a t edge together with the adjacent triangles. As shown (see also Eq. (4.3)), there are two different ways to split up the plaquette/tensor. As we will discuss more later, these correspond to different orderings in which V_1 at neighboring t edges act on the same qubit. Dually, we need to split each 4-index δ-tensor at a x or y edge into two 3-index δ-tensors. Geometrically, this corresponds to splitting a 4-valent edge into two 3-valent edges separated by a 2-gon face yielding a configuration as shown in Eq. (9.13). After this, T_1 corresponds to a xy face together with the adjacent 3-valent edges. Neither T_1 nor V_1 are unitary, which is not a surprise given that the path integral represents an imaginary, and not a real time evolution. In fact, T_1 is the projector onto the $+1$ eigenspace of the Pauli operator $Z_0Z_1Z_2Z_3$, and V_1 the projector onto the $+1$ eigenspace of $X_0X_1X_2X_3$. To fix this, we define a second projector T_m corresponding to a xy face carrying a segment of m worldline,

$T_m :=$ 

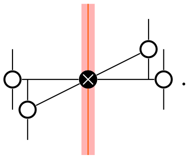
(12.22)

This way, T_1 is extended to an isometry $\mathbf{T}$,

$\mathbf{T} := (T_1, T_m) =$ 

(12.23)

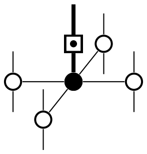
$\mathbf{T}$ defines an instrument $I[\mathbf{T}]$ via Eq. (12.17), which is in fact just a projective $Z_0Z_1Z_2Z_3$ measurement. Dually, we can define an operator V_e carrying an e anyon segment along a t edge,

$V_e :=$ 

(12.24)

This gives rise to an isometry $\mathbf{V}$,

$$\mathbf{V} \coloneqq (V_1, V_e) = \qquad . \tag{12.25}$$



$\mathbf{V}$ yields an instrument $I[\mathbf{V}]$, which is just a projective $X_0X_1X_2X_3$ measurement. The presence of the *Hadamard* matrix,

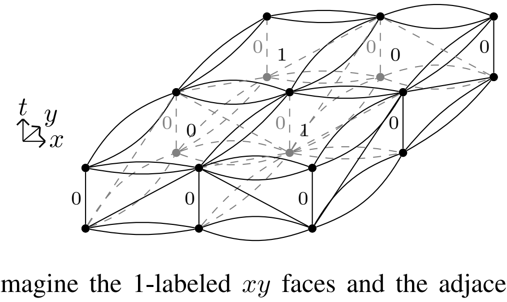
$$-\!\boxed{\cdot}\!- \coloneqq H \coloneqq \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} , \tag{12.26}$$

seemingly spoils the duality between Eq. (12.25) and Eq. (12.22). This is explained by the fact that in Eq. (12.17) we use a δ-tensor for both $I[\mathbf{T}]$ and $I[\mathbf{V}]$.

As mentioned earlier, the way in which we divide each plaquette into two triangles and each 4-valent edge into two 3-valent edges determines the ordering in which we act on the qubits. A straight-forward way to choose an ordering is to checkerboard-number each vertex and t edge with x/y-coordinate $ax+by$ by $a+b \mod 2$. Then we first act with all the 0-labeled V operators (which act on mutually disjoint quadruples of qubits) and then with all the 1-labeled ones. Dually, we can checkerboard-number cubes and xy faces according to their x/y coordinate and act with 0-labeled T operators first. One full period of the circuit then consists of four rounds of instruments,

$$\rightarrow I[\mathbf{T}]_0 \rightarrow I[\mathbf{T}]_1 \rightarrow I[\mathbf{V}]_0 \rightarrow I[\mathbf{V}]_1 \rightarrow \ . \tag{12.27}$$

Geometrically, this corresponds to dividing each xt and yt plaquette along a diagonal edge connecting a 0 vertex at time T to a 1 vertex at time $T-1$. Dually, we also split each 4-valent edge by inserting a 2-gon that is adjacent to a 0 cube at time T and a 1 cube at time $T-1$. The following shows a section of this modified cubic lattice (this time in black instead of orange),

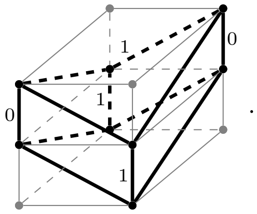


$$. \tag{12.28}$$

Here we should imagine the 1-labeled xy faces and the adjacent edges being bent slightly towards the positive t direction. Note that, since all the operators commute, applying them in any order defines a valid spacetime cellulation. For example, consider the alternative ordering

$$\rightarrow I[\mathbf{T}]_0 \rightarrow I[\mathbf{V}]_0 \rightarrow I[\mathbf{T}]_1 \rightarrow I[\mathbf{V}]_1 \rightarrow \ . \tag{12.29}$$

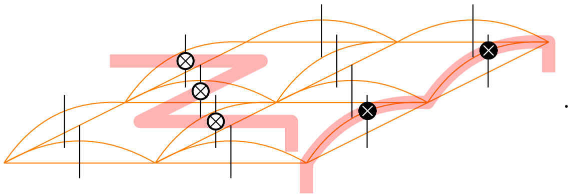
This yields a modified cellulation where the 1 xy faces are replaced by faces formed by four diagonals. That is, the 0 cubes remain the same, but the 1 cubes have their 0 vertices shifted by $+1$ in t direction. The following shows the original (gray) and modified (black) 1-cubes:



(12.30)

The cellulation representation does not only apply to different ordering of measurements, but also to decomposing a stabilizer measurement into CX gates acting on an ancilla. Also, we may put the toric code on different spatial cellulations. In this case the spacetime cellulation is based on the cartesian product of the spatial cellulation with the regular 1-dimensional lattice. The representation of the circuit as a spacetime cellulation also has a practical application: The spacetime cellulation provides a natural lattice on which the classical decoder performs minimum-weight matching, since individual Pauli-X or Pauli-Z errors at specific moments in time correspond to different faces and edges in this lattice.

Let us briefly describe the general decoder of Proposition 7 for the present code. We first record the spacetime syndrome until the time $T \sim L$ that we assume to be after the plaquette measurements, where the associated spatial slice of the lattice is a square lattice. This syndrome is a subset of t edges and xy faces forming an e 1-chain and a m 2-cochain inside the (modified) cubic spacetime lattice. The classical decoder D now finds a minimum-weight set of edges (faces), such that flipping these edges (faces) fixes the e 1-chain (m 2-cochain) to a 1-cycle (2-cocycle). It is important to note that the flipped edges (faces) can be any edges (faces) of the modified cubic lattices and not only t edges (xy faces). Thereby the fixed e 1-cycle (m 2-cocycle) is allowed to terminate at the square-lattice spatial slice at time T. To perform the corrections, we insert a 2-gon and a 2-valent edge in between every time-T x and y edge and the adjacent time-$(T+\frac{1}{2})$ xt and yt face. Then D chooses any subset of the inserted 2-valent edges (2-gon faces), such that these edges (faces) together with the fixed spacetime e 1-cycle (m 2-cocycle) forms a homologically trivial 1-cycle (2-cocycle) that does not terminate on the spatial slice at time T. At every chosen 2-valent edge (2-gon face), we put an e (m) worldline, such that the overall defect configuration is equivalent to the trivial one, for example,



(12.31)

As shown, adding these the worldlines corresponds to inserting 2-index δ and $\mathbb{Z}_2$-

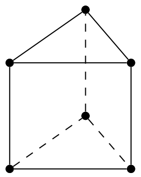
tensors into the circuit, which are single qubit Pauli operators,

$$\otimes = X\ , \qquad \boldsymbol{\otimes} = Z\ . \tag{12.32}$$

Note that in the absence noise, the measured e 1-chain (m 2-cochain) is a 1-cycle (2-cocycle) with probability 1. Since e (m) is only supported on t edges (xy faces), the measured syndrome consists of a subset of infinite (dual) lines in t direction. Thus, future measurement outcomes are determined by the past history. In particular, if we start with the ground state, we deterministically measure the trivial syndrome. This property is not necessary for fault tolerance and is the key qualitative difference between the stabilizer toric code and the subsystem and Floquet versions thereof that we will look at in the following.

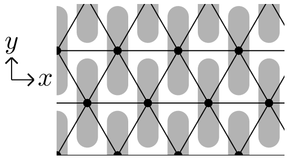
12.2.2 Subsystem toric code

The *subsystem toric code* is a topological subsystem code developed in Ref. [32], that only involves 3-body measurements. From the path integral point of view, it can be derived from the stabilizer toric code on a regular triangular spatial lattice by a simple modification of the spacetime cellulation. So we start with a spacetime cellulation consisting of triangle-prism volumes,



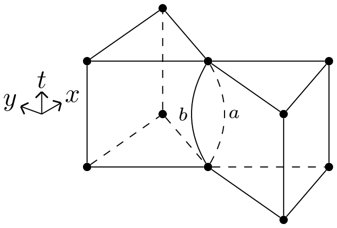
. (12.33)

This way, the operators T_1 at the xy faces are already 3-qubit, but the operators V_1 at to the t edges are 6-qubit. However, V_1 can be split up into two 3-qubit operators V_1^a and V_1^b by the following trick. We choose one spatial direction x aligned with one third of the edges, and refer to the orthogonal direction as y. Then at every space vertex, we pair up the two adjacent triangles whose centers are located in the positive and in the negative y direction,



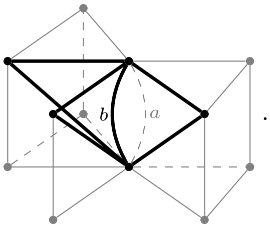
. (12.34)

In the spacetime cellulation, we get a pair of triangle prisms adjacent to each t edge. We simply split up each 6-valent t edge into two 3-valent edges a and b, such that the two adjacent prisms become one single volume,



$$. \tag{12.35}$$

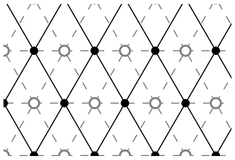
As for the stabilizer toric code we split each rectangle face into two triangles. Then we define V_1^a and V_1^b as the operators corresponding to a and b together with the three adjacent triangles, for example



$$. \tag{12.36}$$

We also introduce versions V_e^a or V_e^b where a or b carries an e anyon, and the according instruments $I[\mathbf{V}^a]$, $I[\mathbf{V}^b]$.

In total, the QEC circuit consists of $X_0X_1X_2$ and $Z_0Z_1Z_2$ measurements on different triples of qubits, which are located at the edges of a triangular lattice. After drawing a square lattice (black) over the triangular lattice (gray) as follows,



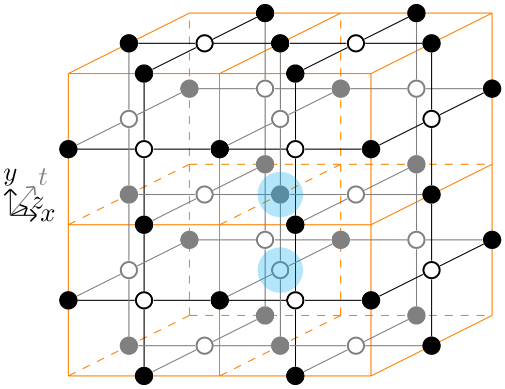
$$, \tag{12.37}$$

we recover the subsystem code as presented in Ref. [32] with qubits on the edges and vertices, and measurements at the corners.

The general decoding procedure of Proposition 7 is similar to the stabilizer toric code. The spacetime lattice in which we find a minimum-weight fix of the e 1-chain (m 2-cochain) is now the modified lattice with volumes as in Eq. (12.35). The spatial slice of the lattice is a triangular lattice, and the correction works in the same way. The crucial qualitative difference to the toric code is that the edges where V_e measurements are performed come in a, b pairs forming little loops that support small 1-cycles. Thus even in the absence of noise measurements are non-deterministic and the results (x, y) and $(x+1, y+1)$ (mod 2) at an a, b pair both occur with probability $\frac{1}{2}$. If we start with the ground state, then the measured e 1-cycle is a random subset of a, b-loops.

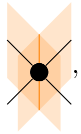
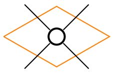
12.2.3 CSS Floquet code

As a third example, we consider the recently discovered CSS Floquet code [92, 53, 2]. This code performs 6 rounds of 2-body XX or ZZ measurements. To obtain this code from our path-integral picture, we start with the toric code path integral on a cubic spacetime lattice, just as for the stabilizer toric code. The only difference is that instead of z, we choose $t = x + y + z$ as the time direction,

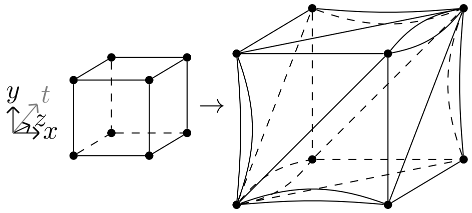


(12.38)

The operators of the circuit are now individual tensors at the edges and faces as marked in blue above. Traversing the path integral in the t direction gives a natural direction to each tensor, acting as 2-qubit operators

$T_1 :=$  , $\quad V_1 :=$  . (12.39)

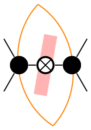
Neither T_1 nor V_1 are unitaries. In fact they are projectors onto the $+1$ subspace of 2-qubit operators Z_0Z_1 and X_0X_1. Our construction proceeds by defining versions T_m and V_e of these operators including an anyon worldline segment. To this end, we slightly modify the cubic lattice. We split each face into two triangles by diagonal 2-valent edges along the $x + y$, $x + z$, or $y + z$ direction, respectively. Dually, we split each 4-valent edge into two 3-valent edges separated by a 2-gon, such that the 2-gons are perpendicular to the $x + y$, $x + z$, and $y + z$ directions. A volume of this slightly modified cubic lattice thus looks like



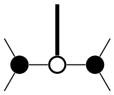
. (12.40)

Note that each edge of the cube gives rise to a 2-gon face, but this 2-gon is not always part of the boundary of the modified cube. In this modified lattice, T_1 now corresponds to a 2-gon together with the two adjacent 3-valent edges. T_m is the same with an m

anyon worldline perpendicular to the 2-gon,

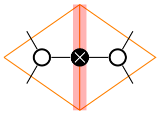
$T_m \coloneqq$  . (12.41)

Together with T_1, we obtain an isometry $\mathbf{T}$,

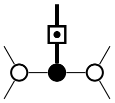
$\mathbf{T} \coloneqq (T_1, T_m) =$  . (12.42)

The according instrument $I[\mathbf{T}]$ is just a Z_0Z_1 measurement.

Dually, V_1 now consists of a 2-valent diagonal edge together with the two adjacent triangles. V_e is the same with an e anyon worldline segment along the diagonal edge,

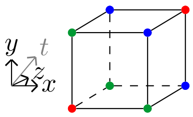
$V_e \coloneqq$  , (12.43)

yielding an isometry

$\mathbf{V} \coloneqq (V_1, V_e) =$  . (12.44)

The according instrument $I[\mathbf{V}]$ is a X_0X_1 measurement.

In principle, the spacetime cellulation fully specifies the combinatorics of the circuit formed by the instruments defined above. However, it is instructive to express the circuit in a more conventional form as a sequence of measurements acting on qubits located on a fixed spatial lattice. We start by decomposing the circuit into rounds of operators acting in parallel. Within one $t = x + y + z$ period there are three different levels of vertices, which we will label 0/red, 1/green, and 2/blue, respectively,

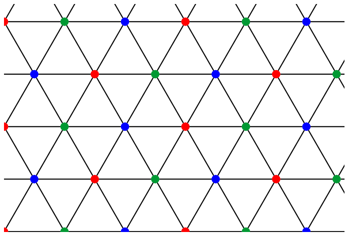


(12.45)

Accordingly, there are three levels of edges, 01, 12, and 20, and three levels of faces, 012, 120, and 201. So one t period of the circuit consists of 6 rounds of instruments:

$$\begin{aligned}&\to I[\mathbf{T}]_{01} \to I[\mathbf{V}]_{012} \to I[\mathbf{T}]_{12}\\ &\quad\to I[\mathbf{V}]_{120} \to I[\mathbf{T}]_{20} \to I[\mathbf{V}]_{201} \to \;.\end{aligned} \tag{12.46}$$

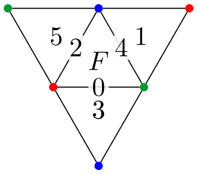
An appropriate spatial lattice on which the circuit acts can be obtained by projecting the 3-dimensional cubic lattice along the t direction. This yields a 2-dimensional regular triangular lattice such that the vertices of each triangle have different numbers/colors,



. (12.47)

The spacetime faces become rhombi in this spatial lattice, consisting of two triangles.

Each qubit corresponds to a time-like continued string of bonds in the tensor network/circuit diagram. The goal is to arrive at a circuit that consists only of 2-body measurements without any swap operations. This fully determines the time-like strings by the way the inputs and outputs are paired in Eq. (12.42) and Eq. (12.44). Geometrically, these time-like strings are sequences of adjacent faces and edges. In the space projection, there is one such sequence for every triangle F as follows,



. (12.48)

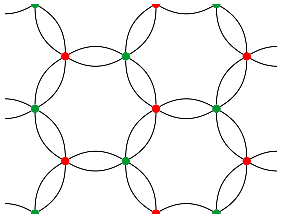
Here the labels 0, 2, 4 correspond to projections of edges, and the labels 1, 3, 5 at triangles correspond to projections of faces formed by this triangle together with F. Then the sequence $0-1-2-3-4-5$ is the time-like string within one t period.

As we have seen, there is one qubit associated to each triangle. For each edge, the instrument $I[\mathbf{T}]$ acts on the qubits at the two triangles adjacent to its projection. For each face, $I[\mathbf{V}]$ acts on the qubits at the two triangles contained in its projection. Note that the instruments $I[\mathbf{T}]_{01}$ act on the same pairs of qubits as the instruments $I[\mathbf{V}]_{120}$, and analogous for cyclic permutation of the numbers/colors. Taking into account that $I[\mathbf{T}]$ is a Z_0Z_1 measurement and $I[\mathbf{V}]$ is a X_0X_1 measurement, we can rewrite Eq. (12.46) as

$$\begin{aligned} &\rightarrow ZZ_{01} \rightarrow XX_{20} \rightarrow ZZ_{12} \\ &\quad \rightarrow XX_{01} \rightarrow ZZ_{20} \rightarrow XX_{12} \rightarrow \ . \end{aligned} \tag{12.49}$$

After going to the dual hexagonal lattice, we recover the CSS Floquet code as introduced in Refs. [92, 53, 2].

Let us briefly describe the general decoding procedure in Proposition 7 for the present code. The spacetime lattice in which we fix the measured e 1-chain (m 2-cochain) is the rotated modified cubic lattice. If we choose the correction time T after the $I[\mathbf{T}]_{01}$ measurements, the spatial slice of the lattice looks like



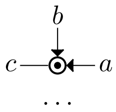
. (12.50)

The fixed spacetime e 1-cycle (m 2-cocycle) terminates at a 0-cycle (2-cocycle) on this spatial lattice. It is closed in a homologically trivial way by inserting 2-valent edges (2-gon faces) potentially carrying e (m) anyon worldlines similar to Eq. (12.31). The static toric code on this spatial lattice also coincides with the *instantaneous stabilizer group* of the code at time T. In contrast to the stabilizer and subsystem toric code, the edges (dual edges) where measurements potentially yield e (m) anyon worldlines are not aligned with the t direction. Furthermore, the graph formed by these edges (dual edges) is much more connected. In the absence of noise, any 1-cycle (2-cocycle) supported these edges (dual edges) is measured with equal probability. So as for the subsystem toric code the measurement results are non-deterministic, but now they are even more fluctuating and may include homologically non-trivial loops. This is not a problem for decoding though, since these homologically non-trivial loops are recorded and can be corrected.

We have seen that the CSS Floquet code and the stabilizer toric code are both based on the cubic-lattice toric code path integral, but with different time directions. If we superimpose the two cubic lattices such that the time directions align, the path integrals are different. Nonetheless, they are in the same fixed-point phase as defined in Section (3.7). That is, they are related by locally applying a set of tensor-network equations. These tensor-network equations are just the moves of the 3D face-edge tTS in Section 9.2. Since these moves can perform arbitrary local changes of the cellulation, they can map the cubic lattice onto the rotated cubic lattice. So the time evolutions of the two codes postselected to the trivial 0 (or $+1 \in \{\pm 1\}$) measurement outcomes are locally equivalent. In both codes, the non-trivial 1 (or $-1 \in \{\pm 1\}$) outcomes correspond to e or m anyon worldline segments. However, the positions of these segments in spacetime are different for the two codes. The subsystem toric code, as well as the honeycomb Floquet code discussed in the following section, are related in the same way.

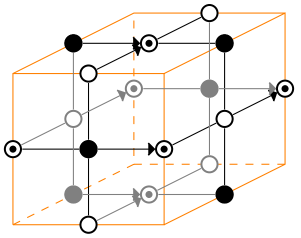
12.2.4 Honeycomb Floquet code

In this section, we consider the honeycomb Floquet code introduced in Ref. [79]. The underlying tensor-network path integral will be referred to as the *honeycomb path integral*. It has the same geometry as the cubic-lattice toric-code path integral used in previous sections. However, it involves a third kind of tensor,



$$= \begin{cases} (-1)^{\frac{a+b-c+\dots}{2}} & \text{if } a+b+c+\ldots = 0 \mod 2 \\ 0 & \text{otherwise} \end{cases} . \tag{12.51}$$

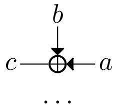
We will refer to this tensor as $\mathbb{C}$-*tensor* since it is related to the two-dimensional real algebra of complex numbers. Note that the tensor depends on a choice of arrow direction at each index, which we indicate by an arrow at the incoming indices. The honeycomb path integral has δ-tensors at every z edge and every xy face, $\mathbb{Z}_2$-tensors at every x edge and yz face, and $\mathbb{C}$-tensors at every y edge and xz face,



(12.52)

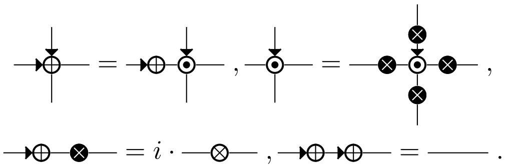
The arrow directions of the $\mathbb{C}$-tensors are chosen to point in the positive x and z direction, respectively.

The honeycomb path integral can be decorated with two types of topological syndrome defects, located on 1-cycles and 2-cocycles. To this end, we define a charged version of the $\mathbb{C}$-tensor,



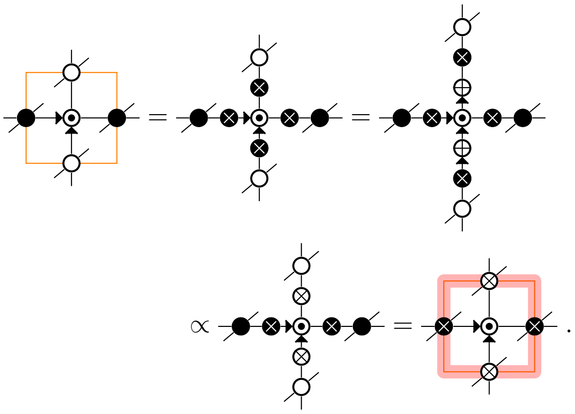
$$= \begin{cases} i^{a+b-c+\dots} & \text{if } a+b+c+\ldots = 1 \mod 2 \\ 0 & \text{otherwise} \end{cases} . \tag{12.53}$$

Then, at every edge of the 1-cycle, we replace the associated tensor with the charged version thereof. We do the same for the faces of the 2-cocycle. The tensors satisfy tensor-network equations such as:



(12.54)

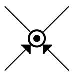
Using these equations, we can locally change the 1-cycle and 2-cocycle. For example, we can add the boundary of a xz face to the 1-cocycle as follows,



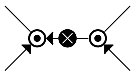
(12.55)

Note that global prefactors are irrelevant due to the quantum mechanical interpretation of the path integral. Also note that the honeycomb path integral and its defect configurations are only defined on a fixed cubic background lattice.

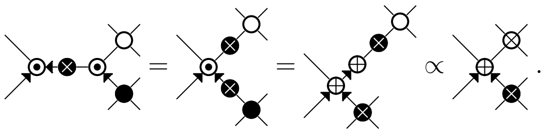
We will now turn the honeycomb path integral into a circuit of instruments using the same time direction as in Section 12.2.3 when discussing the CSS Floquet code. δ-tensors and $\mathbb{Z}$-tensors become operators T_1 and V_1 yielding instruments $I[\mathbf{T}]$ and $I[\mathbf{V}]$, which are Z_0Z_1 and X_0X_1 measurements as before. The operator W_1 corresponding to a single $\mathbb{C}$-tensor,

$W_1 :=$  $= \frac{1}{2}(1+Y_0Y_1)$, (12.56)

will be complemented by another tensor

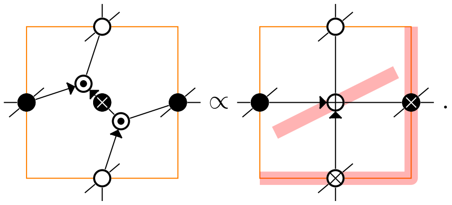
$W_x :=$  $= \frac{1}{2}(1-Y_0Y_1)$, (12.57)

such that the instrument $I[\mathbf{W}]$ defined by (W_1, W_x) is a projective Y_0Y_1 measurement. We now need to turn a configuration of measurement outcomes into a configuration of defects on the cubic lattice. To this end, we apply tensor-network equations to move the "charge" from the middle 2-index tensors in Eq. (12.41), Eq. (12.43), or Eq. (12.57) to the tensors of the original honeycomb path integral. For example, for the operator W_x, this results in a "charge" at the neighboring δ and $\mathbb{Z}_2$-tensors, but also at the $\mathbb{C}$-tensor itself,

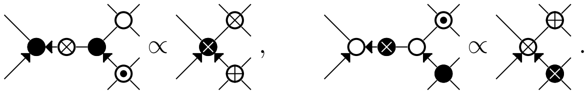


(12.58)

So the W_x measurement outcome corresponds to three different syndrome defect segments being present. For a W_x operator located at a xy face, we get 1-cycle defect segments on two adjacent edges, but also a 2-cocycle defect at the xy face itself,


(12.59)

Note that also in the CSS Floquet in Section 12.2.3, a V_e operator at a face corresponds to an e defect segment at a diagonal edge, which is equivalent to e defect segments at two boundary edges like above. However, in this case there is no m defect segment at the face itself. Dually, for a W_x measurement outcome at an edge, we obtain 2-cocycle defects at two adjacent faces as well as a 1-cycle defect at the edge itself. The same is also true for T_m and V_e instead of W_x,


(12.60)

All in all, we find that the condition of Definition 27 still holds, just that now each measurement outcome corresponds to multiple defect segments of different types.

Let us now look at the combinatorics of the resulting circuit. The overall geometry is as for the CSS Floquet code in Eq. (12.46), just that the type of measurement now depends on the orientation of the edge or face and not on the time step:

$$\begin{aligned}
&\rightarrow (I[\mathbf{T}]_{z01}, I[\mathbf{V}]_{x01}, I[\mathbf{W}]_{y01})\\
&\quad\rightarrow (I[\mathbf{T}]_{xy012}, I[\mathbf{V}]_{yz012}, I[\mathbf{W}]_{xz012})\\
&\qquad\rightarrow (I[\mathbf{T}]_{z12}, I[\mathbf{V}]_{x12}, I[\mathbf{W}]_{y12})\\
&\quad\rightarrow (I[\mathbf{T}]_{xy120}, I[\mathbf{V}]_{yz120}, I[\mathbf{W}]_{xz120})\\
&\qquad\rightarrow (I[\mathbf{T}]_{z20}, I[\mathbf{V}]_{x20}, I[\mathbf{W}]_{y20})\\
&\rightarrow (I[\mathbf{T}]_{xy201}, I[\mathbf{V}]_{yz201}, I[\mathbf{W}]_{xz201}) \rightarrow \;.
\end{aligned} \tag{12.61}$$

After projecting the cubic lattice along time as in Eq. (12.47), x, y, and z refer to the three different directions of edges in the resulting triangular lattice. The measurements at $x01$ and $yz120$ (and analogous pairs) act on the same pair of qubits and are in fact the same type of measurement. We thus find that the circuit repeats already after three rounds, yielding

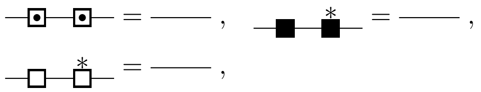
$$\begin{aligned}
&\rightarrow (ZZ_{z01}, XX_{x01}, YY_{y01})\\
&\quad\rightarrow (ZZ_{z20}, XX_{x20}, YY_{y20})\\
&\rightarrow (ZZ_{z12}, XX_{x12}, YY_{y12}) \rightarrow \;.
\end{aligned} \tag{12.62}$$

After going to the dual hexagonal lattice, we obtain the honeycomb code as presented in [79].

It has been argued in Ref. [79] that the honeycomb Floquet code is closely related to the toric code since the instantaneous stabilizer group of the former is equivalent to the latter. Here we will make this relation precise by showing that the underlying path integrals are in the same fixed-point phase. The sequence of tensor-network equations transforming the toric-code path integral into the honeycomb path integral (or vice versa) is as follows: We first insert a resolution of the identity, $\mathbb{1} = GG^{-1}$ at every bond. G is an invertible matrix that depends on the bond within a unit cell, but not on the unit cell. Then we contract each 4-index tensor with the four surrounding matrices G or G^{-1}, yielding a new 4-index tensor at that place. Note that this is just a complicated way of saying that the two tensor networks are equivalent up to a basis change at every bond. The matrices G are built from the Hadamard matrix H in Eq. (12.26), together with the following two matrices,

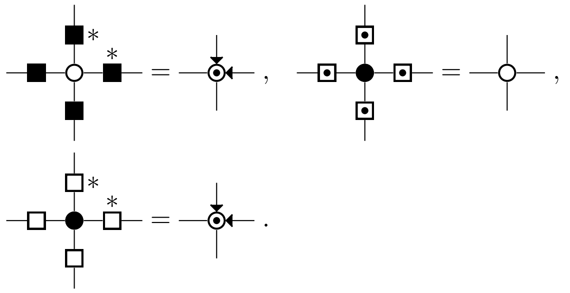
$$-\!\!\blacksquare\!\!- := S := \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} , \quad -\!\!\square\!\!- := U := HSH \,. \tag{12.63}$$

H, S, and U are all unitary,



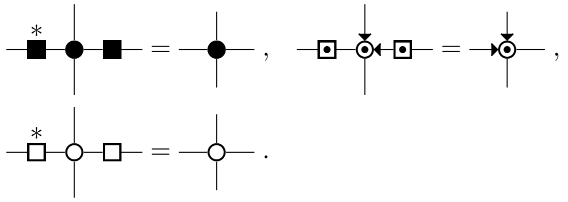
(12.64)

where the $*$ denotes complex conjugation. H, S, and U, together with the 4-index δ, $\mathbb{Z}_2$ and $\mathbb{C}$ tensors satisfy two types of equations. First, adding H to all indices exchanges δ and $\mathbb{Z}_2$, and the same holds with S, $\mathbb{Z}_2$, and $\mathbb{C}$, as well as with U, δ, and $\mathbb{C}$,



(12.65)

Due to Eq. (12.64), each H, S, or U matrix can be either on the right or on the left-hand side. Furthermore, two S matrices adjacent to a δ-tensor can be canceled, and the same for H and $\mathbb{C}$, as well as for U and $\mathbb{Z}_2$:



(12.66)

With this, we are now ready to find matrices G that transform the toric-code path integral into the honeycomb path integral. Each bond inside a unit cell can be specified by the involved edge a (either x, y, or z), the involved face b (either xy, xz, or yz), and the direction $\pm$ of the bond $a \to b$ relative to the x, y, or z direction. Thus, we need to specify 12 different matrices $G(a, b, \pm)$. As an ansatz, we set $G(a, b, -) := G(a, b, +)^*$, and impose that every G is some product formed by H, S, and U. For each edge a, there are different choices for the two matrices $G(a,\dots,+)$, such that the toric-code tensor together with the surrounding G matrices yields the according honeycomb tensor. For example, for $a = x$, we want to transform a toric-code δ-tensor into a honeycomb $\mathbb{Z}_2$-tensor. First, any of the two matrices $G(a,\dots,+)$ may or may not contain S, since each G appears at two ($\pm$) indices and can be annihilated using the first of Eq. (12.66). Then, both matrices $G(a,\dots,+)$ need to contain H in order to transform the δ-tensor into a $\mathbb{Z}_2$-tensor via the second of Eq. (12.65). Finally, each of $G(a,\dots,+)$ may or may not contain U, due to the third of Eq. (12.66). This yields a set of possible choices for either matrix $G(a,\dots,+)$ which we denote by $(S)H(U)$. The following table shows all potential $G(a,\dots,+)$ for edges a or $G(\dots,b,+)$ for faces b,

a/b	toric code	honeycomb	potential $G(a/\dots,\dots/b,+)$
x	δ	$\mathbb{Z}_2$	$(S)H(U)$
y	δ	$\mathbb{C}$	$(S)U(H)$
z	δ	δ	$(S)\mathbb{1}(S)$
xy	$\mathbb{Z}_2$	δ	$(U)H(S)$
xz	$\mathbb{Z}_2$	$\mathbb{C}$	$(U)S(H)$
yz	$\mathbb{Z}_2$	$\mathbb{Z}_2$	$(U)\mathbb{1}(U)$

. (12.67)

In order to find $G(a,b,+)$, we write out all potential $G(a,\dots,+)$ and $G(\dots,b,+)$ and take any common element. A solution is given by

a	$-$	b	$G(a,b,+)$
x	$-$	xy	H
x	$-$	xz	SH
y	$-$	xy	UH
y	$-$	yz	U
z	$-$	xz	S
z	$-$	yz	$\mathbb{1}$

. (12.68)

So we have found that the toric code and honeycomb path integrals are in the same fixed-point phase. Next we notice that Eq. (12.65) and Eq. (12.66) still hold after we replace all δ, $\mathbb{Z}$ and $\mathbb{C}$-tensors with their charged versions, at least up to a phase prefactor. Thus, the toric code path integral with a configuration of e and m defects is locally equivalent to the honeycomb path integral with the same configuration of 1-cycle and 2-cocycle defects. However, the CSS and honeycomb Floquet codes differ by the selection of defect segments corresponding to a measurement in the circuit: A face measurement in the CSS Floquet code corresponds to the presence of an e defect segment at a diagonal edge, or an equivalent pair edges of the original lattice. A face measurement in the honeycomb Floquet code corresponds to the same e defect segments and an additional m defect segment at the face itself. The analogous holds for the edge measurements.

Note that the reduced time periodicity is closely related to the fact that the honeycomb path integral on a cubic lattice allows for a smaller choice of unit cell, namely one consisting of only one δ, one $\mathbb{Z}_2$, and one $\mathbb{C}$-tensor. For example, consider an xy plane containing xy faces, and the honeycomb path integral restricted to this plane including the tensors at all xy faces, x edges, and y edges. When shifting this plane by $\frac{1}{2}z$, the tensors at z edges, xz faces, and yz faces form exactly the same tensor network on the Poincaré dual lattice, that is, shifted by $\frac{1}{2}x+\frac{1}{2}y$. So instead of $\{x,y,z\}$, we can choose $\{x,y,\frac{1}{2}x+\frac{1}{2}y+\frac{1}{2}z\}$ as a unit cell. When we instead consider these two planes shifted by $\frac{1}{2}z$ for the toric code, we swap δ and $\mathbb{Z}_2$ tensors in addition to going to the dual lattice. This has the same effect as inserting a duality domain wall (exchanging e and m anyons) in between the two planes. So with the new unit cell, the phase of both path integrals is a toric code with a rigid stack of duality domain walls perpendicular to z. The according exchange of e and m after one code cycle has already been observed in the honeycomb code [79]. The halved unit cell is also responsible for the weak breaking of translation symmetry in the closely related Kitaev honeycomb model Hamiltonian [96].

Let us briefly discuss the decoding of Proposition 7 for the honeycomb Floquet code. Each spacetime syndrome maps to a 1-chain and 2-cochain in the cubic spacetime lattice. The novel feature compared to earlier examples is that a single measurement outcome corresponds to both 1-chain and 2-cochain segments. Apart from this, the decoding proceeds as usual by finding a minimum-weight fix of the spacetime syndrome and then closing the syndrome at a spatial slice at time T such as in Eq. (12.50). Defects along 1-chains or 2-cochains can be introduced by applying Pauli X, Y, or Z operators to the corresponding places in the circuit. Let us discuss when the syndrome 1-chain is broken at a specific vertex v, by looking at the measurements for which the boundary of the corresponding defect segments contains v. These include the measurements at the 6 faces in the cubic lattice for which v is the temporally first or last vertex, like for the CSS Floquet code. However, they also include the measurements at the 6 edges indicent to v themselves. This is in accordance with the fact that a detection cell is formed by 12 measurements in the honeycomb Floquet code [118], but only 6 in the CSS Floquet code [92].

12.3 New codes from tensor-network path integrals

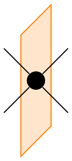
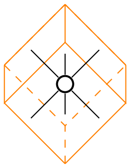
In this section we use our path integral framework to construct two new dynamic error-correcting codes. First, we introduce a generalization of the CSS Floquet code to $3+1$ dimensions. Then we construct a non-Pauli dynamic code based on the double-semion string-net model.

12.3.1 Floquet toric code in 3+1D

In this section, we use our method to construct a new Floquet code, namely a Floquet version of the toric code in $3+1$ dimensions. The $3+1$-dimensional toric code can be described by a Euclidean tensor-network path integral on arbitrary 4-dimensional cellulations. It is given by a sum over all cellular 2-cocycles, that is, configurations of $\mathbb{Z}_2$ variables on all the faces, such that at every volume the sum of variables at its boundary faces is 0 (mod 2). So the tensor network consists of one δ-tensor at every face and one $\mathbb{Z}_2$-tensor at every volume, with bonds shared between pairs of adjacent face and volume.

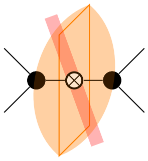
We can also introduce defects inside the path integral. We will use two types of syndrome defects, namely line-like m defects, as well as membrane-like e defects. The e defects are placed on 2-cycles, and at every face of the 2-cocycle we replace the δ-tensor by the charged δ-tensor in Eq. (9.41). The m defects are placed on 3-cocyles, and at every volume of the 3-cocycle we replace the $\mathbb{Z}_2$-tensor by the charged $\mathbb{Z}_2$-tensor in Eq. (9.45).

To construct the Floquet code, we take a 4-dimensional hypercubic lattice spanned by the four unit vectors w, x, y, z, and choose $t = w + x + y + z$ as the time direction. The operators of the circuit are individual 4-index δ-tensors at the faces, and 6-index $\mathbb{Z}_2$-tensors at the cubes. The diagonal t direction allows for a natural interpretation of these tensors as operators by dividing their indices into inputs and outputs,

$T_1 :=$  , $V_1 :=$  . (12.69)

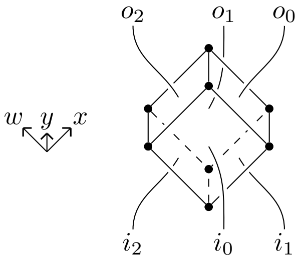
The 4 indices of T_1 connect it to the four cubes adjacent to the face (in orange), whose position in the drawing should not be taken literal due to the 4-dimensional nature. The 6 indices of V_1 connect it to the 6 faces of the cube.

T_1 is a projector onto the $+1$ eigenspace of Z_0Z_1, whereas V_1 is a projector onto the intersection of the $+1$ eigenspaces of X_0X_1 and X_1X_2. Since both T_1 and V_1 are non-unitary, we turn them into instruments using additional operators that include defect segments. In addition to T_1 at a face we define another operator T_m that includes a line segment of m defect. To this end, we modify the lattice slightly by replacing each 4-valent face by a pair of 3-valent faces separated by a pillow-like volume, whose boundary is formed by these two faces only. T_1 then corresponds of the pillow-like volume together with the two faces, and T_m is the same with the pillow-like volume carrying an m anyon segment,

$T_m :=$  . (12.70)

The drawn m worldline goes perpendicular to the pillow volume connecting the two adjacent 4-cells, and its positioning in the drawing should not be taken literal. T_1 and T_m together yield a Z_0Z_1 measurement $I[\mathbf{T}]$ as usual.

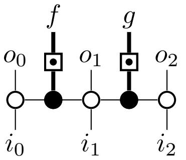
To turn V_1 into an instrument, we define three new operators, V_{1e}, V_{e1}, and V_{ee}, corresponding to the absence or presence of two different e membrane defect segments. To this end, we divide the cube into three volumes along two internal 2-valent faces g and f. V_{1e}, V_{e1}, or V_{ee} then correspond to an e membrane segment being present at either g, f, or at both. Specifically, in a cube with faces labelled like



, (12.71)

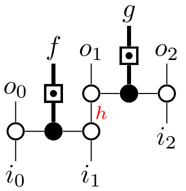
we choose f to have the same boundary as i_0 together with o_0, and g with the same boundary as i_2 and o_2. The isometry resulting from combining the different V operators is given by

$$\mathbf{V} := (V_1, V_{e1}, V_{1e}, V_{ee}) =$$



(12.72)

$$=$$



.

As shown, $\mathbf{V}$ can be split up, such that $I[\mathbf{V}]$ consists of two consecutive measurements X_0X_1 and X_1X_2. Geometrically, this corresponds to adding another internal 2-valent face h whose boundary is that of i_0, i_1, and o_0 together (or equivalently i_2, o_1, and o_2). Note that h does not carry a potential e defect segment.

The definition of $\mathbf{V}$ above depends on a choice of labelling the bottom faces of a cube by i_0, i_1, and i_2, and of the top faces by o_0, o_1, and o_2. In principle, any choice would yield a valid error-correcting circuit. However, we aim to obtain a circuit consisting of projective 2-qubit measurements acting on a fixed set of qubits without intermediate swap gates. The following choice achieves this goal, as will come clear later in this section. First, we divide the cubes into 8 different sorts labelled by p_{abc}, and there will be a different choice for each sort. $p \in \{0, 1\}$ labels whether the cube appears at an even (0) or odd (1) time step in the circuit. abc labels the three spanning directions $\{a, b, c\} \subset \{w, x, y, z\}$ of the cube, ordered according to w, x, y, z. With this, i_0, i_1, and i_2 are the bottom faces with spanning directions ab, bc, and ca, respectively. If $p = 0$, then o_0, o_1, and o_2 are the top faces with spanning directions bc, ca, and ab. If $p = 1$, then they are given by ca, ab, and bc. For example, Eq. (12.71) shows the labelling for a 1_{wxy} cube.

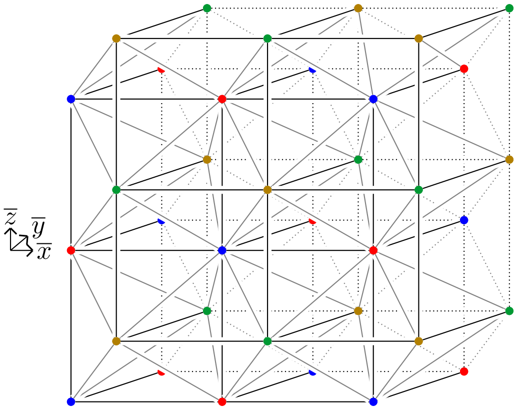
In principle, the combinatorics of the circuit formed by the instruments $I[\mathbf{T}]$ and $I[\mathbf{V}]$ is fully specified by the 4-dimensional cellulation. However, it is instructive to give a more conventional description of this circuit in terms of measurements acting on fixed qubits on a spatial lattice. We start by decomposing the circuit into rounds of instruments that are applied in parallel. To this end, we notice that there are four different levels of vertices in the 4-dimensional cubic lattice along the t direction, which we label/color by 0/red, 1/green, 2/blue, and 3/yellow. Accordingly, there are four levels of faces, 012, 123, 230, and 301, and four levels of volumes, 0123, 1230, 2301, and 3012. So one time period of the circuit consists of 8 rounds of instruments,

$$\begin{aligned}&\rightarrow I[\mathbf{T}]_{012} \rightarrow I[\mathbf{V}]_{0123} \rightarrow I[\mathbf{T}]_{123} \rightarrow I[\mathbf{V}]_{1230}\\ &\quad\rightarrow I[\mathbf{T}]_{230} \rightarrow I[\mathbf{V}]_{2301} \rightarrow I[\mathbf{T}]_{301} \rightarrow I[\mathbf{V}]_{3012}\ .\end{aligned} \tag{12.73}$$

To get an appropriate spatial lattice, we project the 4-dimensional cubic lattice onto 3-dimensional space along the $t = w + x + y + z$ axis. To this end, we choose new basis vectors

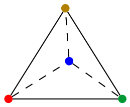
$$\begin{aligned}\overline{x} &= \frac{1}{2}w + \frac{1}{2}x - \frac{1}{2}y - \frac{1}{2}z\ ,\\ \overline{y} &= \frac{1}{2}w - \frac{1}{2}x + \frac{1}{2}y - \frac{1}{2}z\ ,\\ \overline{z} &= \frac{1}{2}w - \frac{1}{2}x - \frac{1}{2}y + \frac{1}{2}z\ ,\end{aligned} \tag{12.74}$$

orthogonal to t. The projected 0 and 2 vertices then form a cubic lattice A with unit vectors $\overline{x}$, $\overline{y}$, and $\overline{z}$. The 1 and 3 vertices form a second cubic lattice B shifted by $\frac{1}{2}(\overline{x} + \overline{y} + \overline{z})$, such that the vertices of A are the centers of the cubes of B and vice versa. Within A, 0 and 2 vertices alternate in a checkerboard manner, and the same for 1 and 3 vertices within B. The projected edges have length $\sqrt{\frac{3}{4}}$ and connect each B vertex with the 8 corner vertices of the corresponding A cube, and vice versa. The edges of the A and B lattice themselves are not projected edges of the 4-dimensional cubic lattice. The following depicts a section of the lattice with four layers of vertices in $\overline{y}$ direction, projected edges in gray, edges of A and B in black, and edges connecting vertices of the two back layers dotted:

 . (12.75)

The edges of A and B together with all the projected edges define a triangulation where each tetrahedron has one 0, one 1, one 2, and one 3 vertex. The projections of spacetime faces are rhombi consisting of two triangles. The projections of the spacetime cubes are (rhombic) cubes consisting of 6 tetrahedra, 3 left-handed and 3 right-handed ones. If a cube is adjacent to a face, then one of the right-handed tetrahedra contains one of the triangles of the face.

As usual, qubits can be identified by following the timeline of the bonds in the tensor-network/circuit diagram. There is one such timeline for every tetrahedron F that is right-handed relative to the vertex ordering 0123,

 . (12.76)

Let $F_{i,i+1,i+2}$ be the spacetime face whose projection is spanned by the $(i,i+1)$ and $(i+1,i+2)$ edges of the tetrahedron, where all numbers are understood mod 4. Let $F_{i,i+1,i+2,i+3}$ be the spacetime cube whose projection is spanned by the $(i,i+1)$, $(i+1,i+2)$, and $(i+2,i+3)$ edges of the tetrahedron. Then, within a fixed t-period, the timeline of bonds is given by the following sequence of adjacent faces and cubes,

$$\begin{aligned} F_{012} - F_{0123} - F_{123} - F_{1230} - F_{230} \\ - F_{2301} - F_{301} - F_{3012} - \ . \end{aligned} \tag{12.77}$$

To go from the face $F_{i,i+1,i+2}$ to the face $F_{i+1,i+2,i+3}$ inside the projection of the cube $F_{i,i+1,i+2,i+3}$, we have to either rotate left or right when looking in the direction $i \rightarrow i+3$. Since the tetrahedron is right-handed relative to the orderings 0123 and 2301 but left-handed for 1230 and 3012, we rotate right for $i=0$ and $i=2$, and right for $i=1$ and $i=3$. This fits our choice of labeling the faces of each cube by i_0, …, o_2, which we have discussed in the paragraph after Eq. (12.72): As can be seen in Eq. (12.71), in order to go from i_x to o_x we turn either right or left in the spatial projection of the cube when looking from bottom to top. We turn left for even time steps ($p=0$ which we identify with $i=1$ or $i=3$), and right for odd time steps ($p=1$, which is $i=0$ or $i=2$).

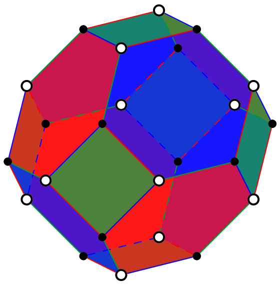
As we have seen, there is one qubit associated to every right-handed tetrahedron. Each instrument $I[\mathbf{T}]$ at a spacetime face acts on the qubits at the two right-handed tetrahedra adjacent to the two triangles that are contained in the projection of the face. Alternatively, these two right-handed tetrahedra are the ones adjacent to the diagonal $(i,i+2)$ edge of a $(i,i+1,i+2)$ face, which is an edge of the A or B cubic lattice. Each instrument $I[\mathbf{V}]$ at a spacetime cube acts on the qubits at the three right-handed tetrahedra contained in the projection of the cube. Alternatively, these two right-handed tetrahedra are the ones adjacent to the diagonal $(i+3,i)$ edge of a $(i,i+1,i+2,i+3)$ cube.

So in total we obtain the following dynamic code. Consider two shifted cubic lattices A and B together with all length-$\sqrt{\frac{3}{4}}$ edges connecting A and B, defining a triangulation whose vertices are 4-colorable as 0, 1, 2, or 3. There is one qubit at every right-handed tetrahedron. The sequence of measurements consists in 8 rounds,

$$\begin{aligned} ZZ_{02} \rightarrow (XX,XX)_{30} \rightarrow ZZ_{13} \rightarrow (XX,XX)_{01} \\ \rightarrow ZZ_{20} \rightarrow (XX,XX)_{12} \rightarrow ZZ_{31} \\ \rightarrow (XX,XX)_{23} \rightarrow \ . \end{aligned} \tag{12.78}$$

In each round we measure either Z_0Z_1 on the two adjacent right-handed tetrahedra adjacent to all edges of the specified type, or we measure X_0X_1 and X_1X_2 on the three right-handed tetrahedra adjacent to all edges of that type. Note that the rounds 0 and 4 (numbered starting from 0), as well as 2 and 6 in Eq. (12.78) are identical.

This Floquet code can be generalized to arbitrary triangulations with 4-colored vertices. In every round, we measure Z_0Z_1, Z_1Z_2, $\ldots$ $Z_{i-1}Z_i$ on the set of right-handed tetrahedra adjacent to the specified type of edges in the lattice, or the same for X instead of Z. The Poincaré dual to such a lattice has 4-colorable volumes and is used in the definition of the 3-dimensional color code [28]. However, our code involves only half of the qubits. The dual lattice of the triangulation depicted in Eq. (12.75) is known as *bitruncated cubic honeycomb* [137]. The volumes are bitruncated cubes,



. (12.79)

The drawn volume is dual to a 3 vertex. The blue shaded 6-gon faces are dual to 23 edges, and the red shaded 6-gon faces to 30 edges. The green shaded 4-gon faces are dual to 13 edges. The red, green, and blue edges are dual to 123-triangles, 230-triangles, and 301-triangles, respectively. The overall lattice also contains faces dual to 01 edges, 12 edges and 02 edges, as well as edges dual to 012 triangles, but none of these are contained in the boundary of the 3 volume shown above. There are qubits on all the full vertices, and none at the empty vertices. The measurements in the dual lattice take place on the faces and involve the qubits at the vertices. For example, the ZZ_{13} measurements take place simulteneously on all green 4-gon faces shown above.

Let us briefly look at the decoding procedure from Proposition 7 for the present code. The spacetime syndrome measured over some time $T \sim L$ consists of one outcome at every face and two outcomes at every cube of the hypercubic lattice. The syndrome yields a e 2-chain and an m 3-cochain inside the 4-dimensional modified hypercubic lattice, supported on the pillow-like volumes and the dividing f and g faces. The boundary of the m 3-cochain is a (0-dimensional) 4-cocyle, and the boundary of the e 2-chain is a (1-dimensional) 1-cycle. We then use the classical decoder D to find a low-weight fix that turns e into a 2-cycle and m into a 3-cocycle. For closing off the e 2-cycle and m 3-cocycle, we choose T to be after a round of $I[\mathbf{T}]_{123}$ instruments. The corresponding spatial slice of the modified hypercubic lattice at this time is obtained by (1) taking only the 123 faces in the lattice in Eq. (12.75), and (2) replacing every face by two copies separated by a pillow-like volume. The non-pillow volumes of this spatial slice are rhombic dodecahedra, each formed by the four 0123 cubes adjacent to a 0 vertex in Eq. (12.75). Each 123 face in Eq. (12.75) has two adjacent qubits, so there is one qubit for every face of the spatial slice. The m 3-cocycle restricted to this spatial slice is again a 3-cocycle, that is, a collection of rhombic dodecahedra and pillow volumes. We close this 3-cochain by a 2-cochain, and apply a Pauli-X operator to the qubits at each face of this 2-cochain. The e 2-cycle restricted to the spatial slice becomes a 1-cycle, that is, a collection of edges. We close this 1-cycle by a 2-chain, and apply a Pauli-Z operator to the qubits at each face of this 2-cochain. Note that the e part of the syndrome could also be corrected by a local cellular automaton shinking the corresponding 1-cycle in each time step using a mechanism similar to *Toom's rule* [101].

12.3.2 Dynamic double-semion string-net code

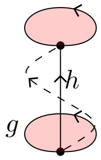
In this section we will give an example for a non-Pauli fixed-point path integral code, which is based on the double-semion Turav-Viro/Dijkgraaf-Witten model [58, 130], the state-sum version of the double-semion string-net model [105, 82]. Note that there are in fact non-Pauli as well as Pauli stabilizer codes for this phase (and any Abelian non-chiral anyon model) [52, 110, 62]. Here, we present a dynamic non-Pauli and non-stabilizer code. This code can be seen as somewhere between stabilizer and Floquet codes, since the anyon worldlines forming the spacetime syndrome move in a fixed direction, but this direction does not coincide with the t direction. Apart from this, our code has some similarities to recent protocols for syndrome extraction for the non-Abelian double-Fibonacci string-net model presented in Ref. [126]. The goal here is not to produce a particularly practical code, but rather to demonstrate the applicability of our framework beyond the toric-code phase.

The underlying path integral is the ordinary cohomology model discussed in Section 4.16.1 for $G = \mathbb{Z}_2$ with the $p = 1$ group 3-cocycle in Eq. (4.304),

$$\omega_{a,b,c} := \omega(a,b,c) = e^{\frac{2\pi i}{4} p\overline{a}(\overline{b}+\overline{c}-\overline{b+c})} = (-1)^{abc}\,. \qquad (12.80)$$

Recall that this model is a state sum with $\mathbb{Z}_2$-elements at all edges that sum to zero around all triangles, and a weight ω associated to every tetrahedron as shown in Eq. (4.294). The state sum can also be written as a tensor network with one δ-tensor at every edge, one 3-index $\mathbb{Z}_2$ tensor at every face, and one 3-index ω tensor at every tetrahedron.

As syndrome defects we will use anyon worldlines as usual. Recall the cohomology models for the 3D anyon tTS in Section 6.2.3. Here for the sake of this section, we simplify the drawings a bit by omitting the anyon edges and vertices. Instead, we will fill the looping edges adjacent to the anyon vertices with red 1-gons, such as,



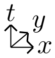
$$\rightarrow \quad \rho_{g,h} := \rho^g(h)\,. \qquad (12.81)$$

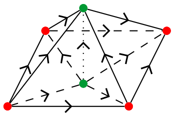
Also note that for the double-semion model the free index labels μ and ν have dimension at most 1, so we omit them. We have classified and computed all anyon worldlines for $G = \mathbb{Z}_l$ around Eq. (6.41). For $G = \mathbb{Z}_2$ with $p = 1$, there are four different anyons, 1, s, $\bar{s}$, and $s\bar{s}$, and the associated weights are

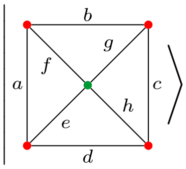
$$\begin{aligned} \rho^1_{g,h} &= \delta_{g,0} \ , \\ \rho^s_{g,h} &= \delta_{g,1} i^h \ , \\ \rho^{\bar{s}}_{g,h} &= \delta_{g,1} (-i)^h \ , \\ \rho^{s\bar{s}}_{g,h} &= \delta_{g,0} (-1)^h \ . \end{aligned} \tag{12.82}$$

We now consider this path integral on a triangulation consisting of two cubic lattices A and B with unit vectors x, y, and z, shifted relative to each other by $\frac{1}{2}x+\frac{1}{2}y+\frac{1}{2}z$. Each tetrahedron is formed by one A edge, one nearby B edge, as well as four length-$\sqrt{\frac{3}{4}}$-edges connecting A vertices with nearby B vertices. So this is the same as the lattice depicted in Eq. (12.75), just that we color all A vertices red and all B vertices green. The branching structure can be chosen such that for every directed edge with associated vector $ax+by+cz$, we have $a+b+c>0$.

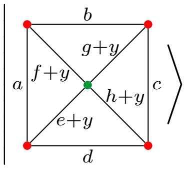
We turn the path integral into a circuit of operators choosing $t=z$ as the time direction. There are two kinds of operators in the circuit which correspond to different volumes as follows. For every t edge, there is an operator T_1 consisting of the four adjacent tetrahedra, acting on 8 qubits (here with coloring for a t edge of B),



$T_1 :=$  ,

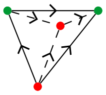
T_1 

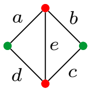
$$= \sum_y \omega_{e,f,f+y}\,\omega_{f,f+y,g+y}\,\omega_{h,h+y,g+y}\,\omega_{e,h,h+y} \tag{12.83}$$

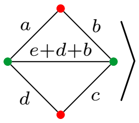
P_{cocycle}  .

P_{cocycle} acting on a triangle with edge labels a, b, and c is the projector onto the *cocycle subspace*, spanned by the configurations that fulfil $a+b=c$. Here and in the following, we also use P_{cocycle} for the product of P_{cocycle} on all the triangles that are currently acted on. As shown, T_1 contains the ω-tensors of the involved tetrahedra, and the $\mathbb{Z}_2$-tensors at the internal and bottom faces. The δ-tensors at the edges of the lattice are split between the adjacent volumes.

For every x or y edge of A or B there is an operator V_1 consisting of the tetrahedron spanned by this edge and the y or x edge of B or A whose center is shifted by $\frac{1}{2}t$,

$V_1 :=$  ,

$V_1 \Bigg|$ 

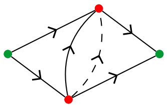
$\Bigg\rangle = \omega_{d,a,b} P_{\text{cocycle}} \Bigg|$ 

$\Bigg\rangle$. (12.84)

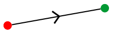
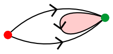
Neither T_1 nor V_1 are unitary since we have

$$T_1 = P_{\text{cocycle}} T_1 = T_1 P_{\text{cocycle}} = P_{\text{cocycle}} T_1 P_{\text{cocycle}} \ , \tag{12.85}$$

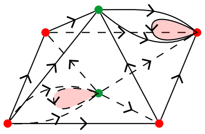
and the same for V_1 instead of T_1. So the support of T_1 and V_1 is contained in the cocycle subspace of the involved triangles. Restricted to this cocycle subspace, V_1 is indeed unitary,

$V_1^\dagger V_1 = P_{\text{cocycle}} =$  . (12.86)

On the right, we have depicted the corresponding volume that arises from gluing the tetrahedron with a reflected copy like in Eq. (4.255). This is not the case for T_1, whose support is contained in but not equal to the cocycle subspace. We will now show how to extend T_1 to an isometry that is fully supported on the cocycle subspace, and later extend both T_1 and V_1 to the full Hilbert space using a different method. To this end, we slightly modify the spacetime lattice to incorporate anyon worldlines running along the $x+y+t$ direction. We consider all the edges aligned with the $x+y-t$ direction. We split every such edge into two edges separated by a 2-gon perpendicular to the $x+y+t$ direction. Then we insert an anyon 1-gon into each such 2-gon, at the vertex with the smaller t component, for example,

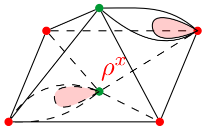
 $\rightarrow$  . (12.87)

The T_1 volume then gets two anyon 1-gons at its boundary, which we connect using an anyon tube along the $x+y+t$ edge,

 . (12.88)

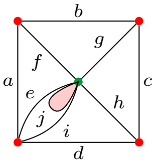
With this, we can replace T_1 by a collection of partial isometries $\mathbf{T} = (T_x)_{x\in\{1,s,\bar{s},s\bar{s}\}}$,

$T_x :=$



,

$T_x \Big|$

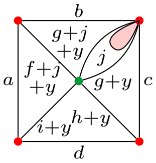


$\Big\rangle$

$$= \sum_y \rho^x_{j,g}\omega_{i,h,h+y}\omega_{h,h+y,g+y} \qquad (12.89)$$

$$\omega_{e,f,f+j+y}\omega_{f,f+j+y,g+j+y}\omega_{i+y,g+j+y,j}\omega_{e,g,j}\omega_{e,j,g}$$

$P_{\text{cocycle}} \Big|$



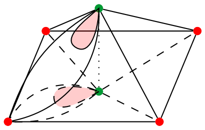
$\Big\rangle$.

Here we have used a cellulation of the volume with one anyon tube and 7 tetrahedra. $\mathbf{T}$ is indeed an isometry when restricted to the cocycle subspace,

$$\mathbf{T}^\dagger\mathbf{T} = \sum_x T_x^\dagger T_x = P_{\text{cocyc}} \, . \qquad (12.90)$$

In order to see this, we compute $T_x^\dagger T_x$ by gluing Eq. (12.89) with a time-reflected copy and using the topological invariance, yielding a projector,

$T_x^\dagger T_x =$



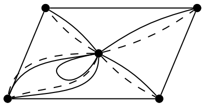
, (12.91)

where the bottom and top 1-gon are connected via a tube segment along the t edge. [1] Then we compute the sum over all tube segments,

$$\rho^1_{g,h} + \rho^s_{g,h} + \rho^{\bar{s}}_{g,h} + \rho^{s\bar{s}}_{g,h} = \delta_{h,0} \, . \qquad (12.92)$$

Setting h to 0 geometrically corresponds to removing the anyon tube and the t edge in Eq. (12.91), and identifying the loop edges at the top and bottom. So we obtain the following volume of solid-torus topology:

$\mathbf{T}^\dagger\mathbf{T} = \sum_x T_x^\dagger T_x =$



$= P_{\text{cocycle}}$. (12.93)

[1] This is a projector since gluing two copies of this volume stacked on top of each other yields the same volume, which corresponds to the equation $P^2 = P$.

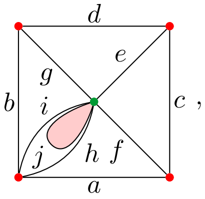
For the last equation we have used that this spacetime volume can be obtained from gluing one volume as in Eq. (12.86) for every pair of neighboring triangles. With this, using $\mathbf{T}$ in Eq. (12.17) defines an instrument $I[\mathbf{T}]$ restricted to the cocycle subspace.

We will now discuss how to extend $I[\mathbf{T}]$ and $I[\mathbf{V}]$ to full instruments also outside the cocycle subspace. The first step is to choose arbitrary extensions $\widetilde{\mathbf{T}}$ and $\widetilde{\mathbf{V}}$ to the full Hilbert space. [2] However, the circuit consisting of the extended instruments $I[\widetilde{\mathbf{T}}]$ and $I[\widetilde{\mathbf{V}}]$ clearly violates Definition 27. This can be fixed by introducing a new channel C to the circuit, with the following task: C measures whether the cocycle constraint is violated at any of the triangles, and maps back to the cocycle subspace if yes. Roughly speaking, this works because (1) $\widetilde{\mathbf{T}}$ and $\widetilde{\mathbf{V}}$ still preserve the cocycle subspace,

$$\begin{aligned}\widetilde{T}_x \circ P_{\text{cocycle}} = T_x = P_{\text{cocycle}} \circ T_x \ ,\\ \widetilde{V}_x \circ P_{\text{cocycle}} = V_x = P_{\text{cocycle}} \circ V_x \ ,\end{aligned} \tag{12.94}$$

and (2) P_{cocycle} consists of the same triangle terms for each isometry.

Concretely, it suffices to apply a channel C before every $I[\widetilde{\mathbf{T}}]$ instrument. The space that $\widetilde{\mathbf{T}}$ acts on is given by



(12.95)

and C acts on that same space. C is the product of one 3-qubit channel C^t for each of the 5 different triangles,

$$C^t_{c,e,f} \rightarrow C^t_{d,e,g} \rightarrow C^t_{f,a,h} \rightarrow C^t_{g,b,i} \rightarrow C^t_{h,i,j} \ . \tag{12.96}$$

Each instrument C^t acts on the qubits at the three edges of the triangle, as indicated by the labels which refer to Eq. (12.95). The 3-qubit instrument $C^t_{a,b,c}$ is defined as follows. First we measure $x = a + b + c \mod 2$, which is the same as a $Z_0Z_1Z_2$ measurement just that we label the outcome with $x \in \{0,1\}$ instead of ± 1. Then we apply a classically controlled operation $c \rightarrow c + x$, which is the same as a CNOT after turning the classical bit x into a qubit. In other words, $C^t_{a,b,c}$ fixes the cocycle condition by flipping the edge c, and C pushes potential cocycle constraint violations into the anyon 1-gon. It is easy to see that C (1) maps everything into the cocycle subspace,

$$C = (P_{\text{cocycle}} \otimes P_{\text{cocycle}}) \circ C \ , \tag{12.97}$$

and (2) acts as the identity inside the cocycle subspace,

$$C \circ (P_{\text{cocycle}} \otimes P_{\text{cocycle}}) = P_{\text{cocycle}} \otimes P_{\text{cocycle}} \ . \tag{12.98}$$

[2] In general, this might also involve enlarging the output dimension by adding new measurement outcomes. This is not necessary in the present case though.

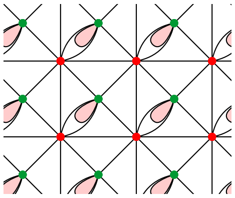
With this, the complete QEC circuit consists of 6 rounds of channels/instruments. First we apply $I[\widetilde{\mathbf{T}}]$ for every t edge of A whose center is within a fixed xy plane of the B lattice, and apply the according operator C before that. Then we apply $I[\widetilde{\mathbf{V}}]$ at all x and all y edges of B inside this xy plane. We then shift the xy plane by $\frac{1}{2}t$ and perform the same instruments with A and B exchanged. In total we obtain

$$\begin{aligned} &\rightarrow C_{At} \rightarrow I[\widetilde{\mathbf{T}}]_{At} \rightarrow (I[\widetilde{\mathbf{V}}]_{Bx}, I[\widetilde{\mathbf{V}}]_{By}) \\ &\quad\rightarrow C_{Bt} \rightarrow I[\widetilde{\mathbf{T}}]_{Bt} \rightarrow (I[\widetilde{\mathbf{V}}]_{Ax}, I[\widetilde{\mathbf{V}}]_{Ay}) \rightarrow \ . \end{aligned} \tag{12.99}$$

Let us now show that this circuit defines a valid path-integral QEC circuit according to Definition 27. To this end, we use the tensor-network equations Eq. (12.97), Eq. (12.94), and Eq. (12.98) transform the circuit in Eq. (12.99) into the circuit

$$\begin{aligned} &\rightarrow I[\mathbf{T}]_{At} \rightarrow (I[\mathbf{V}]_{Bx}, I[\mathbf{V}]_{By}) \\ &\quad\rightarrow I[\mathbf{T}]_{Bt} \rightarrow (I[\mathbf{V}]_{Ax}, I[\mathbf{V}]_{Ay}) \rightarrow \ . \end{aligned} \tag{12.100}$$

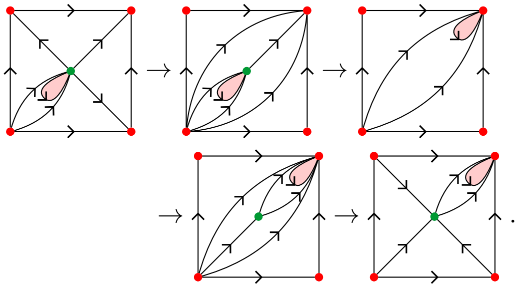
Specifically, applying Eq. (12.97) to all channels C_{At}/C_{Bt} inserts P_{cocycle} on all triangles of the corresponding spatial cut of the lattice (here coloring like before C_{Bt}),



. (12.101)

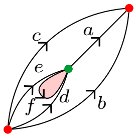
Then applying Eq. (12.94) moves P_{cocycle} to different spatial cuts. Finally, applying Eq. (12.98) removes all the channels C_{Bt}/C_{At}. The remaining P_{cocycle} can be absorbed into the following $I[\mathbf{T}]_{Bt}$/$I[\mathbf{T}]_{At}$ using Eq. (12.85). The transformation implies that the circuit in Eq. (12.99) is in the same fixed-point phase as the circuit in Eq. (12.100). Since for the circuit in Eq. (12.100), every spacetime syndrome corresponds to a fixed-point path integral with anyon worldlines, the circuit in Eq. (12.99) fulfils Definition 27 as well.

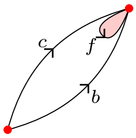
Depending on how we map the circuit onto a fixed set of qubits, $I[\widetilde{\mathbf{T}}]$ acts on at least 10 qubits. So in order to implement it in practice we should decompose it into smaller gates. Surely, any gate can be written as a circuit using a small fixed universal gate set, but this circuit might be approximate and finding it might be hard for such a large operator. However, a first decomposition can be obtained by decomposing the volume in Eq. (12.89) into tetrahedra or at least smaller volumes. Let us give such a decomposition as a sequence of spatial lattices that we get from gluing these smaller volumes step by step,



(12.102)

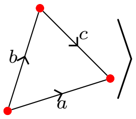
In the first step we glue two tetrahedra, applying twice a 5-qubit operators U_1. The same happens in the last step with an operator R_1. U_1 and R_1 are the same as V_1 shown in Eq. (12.84) except that the involved edges have different directions. In the second step, the volume we glue can be cellulated with an anyon tube together with two tetrahedra, defining an operator S_x acting on 6 qubits,

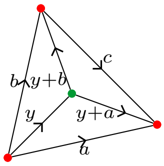
$S_x \Big|$  $\Big\rangle = \delta_{d+a,b}\delta_{e+a,c}$

$\rho^x_{f,a}\omega_{e,f,a}\omega_{e,a,f}P_{\text{cocycle}} \Big|$  $\Big\rangle$.

(12.103)

In the third step, we glue a tetrahedron at a single face, yielding a 6-qubit operator W_1,

$W_1 \Big|$  $\Big\rangle$

$= \sum_y \omega_{y,y+b,c} P_{\text{cocycle}} \Big|$  $\Big\rangle$.

(12.104)

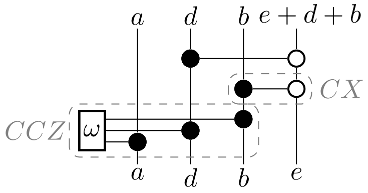
As discussed before, we now arbitrarily extend $\mathbf{U}$, $\mathbf{R}$, $\mathbf{S}$, and $\mathbf{W}$ into isometries $\widetilde{\mathbf{U}}$, $\widetilde{\mathbf{R}}$, $\widetilde{\mathbf{S}}$, and $\widetilde{\mathbf{W}}$ supported on the full Hilbert space. Then, we replace the instrument $I[\widetilde{\mathbf{T}}]$ by a sequence of up-to-6-qubit instruments

$$(I[\widetilde{\mathbf{U}}], I[\widetilde{\mathbf{U}}]) \rightarrow I[\widetilde{\mathbf{S}}] \rightarrow I[\widetilde{\mathbf{W}}] \rightarrow (I[\widetilde{\mathbf{R}}], I[\widetilde{\mathbf{R}}]) \ . \qquad (12.105)$$

To extend the operators, we essentially just remove the P_{cocycle} terms from the corresponding definitions. This way, V_1 in Eq. (12.84) becomes a unitary

$$\widetilde{V}_1 \, |d,a,b,e\rangle = \omega_{d,a,b} \, |d,a,b,e+d+b\rangle \ , \qquad (12.106)$$

acting trivially on the label c. This unitary can be written as a circuit of controlled-X and controlled-controlled-Z gates,

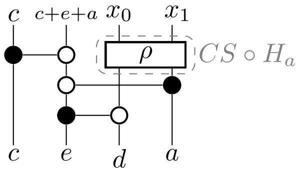


. (12.107)

S_x in Eq. (12.103) is a map from 6 to 3 qubits. Since there are 4 anyons and thus 4 measurement results x, we need to measure one further qubit to turn $\mathbf{S}$ into an isometry on the full Hilbert space. In order to fulfil Definition 27, the measurement outcome for this further qubit must be deterministic inside the cocycle subspace. This can be done by measuring the cocycle constraint, e.g., on the (a, c, e) triangle in Eq. (12.103). Using $\omega_{e,f,a}\omega_{e,a,f} = 1$ and $f = d + e$ inside the cocycle subspace, we obtain an isometry

$$\widetilde{S}_x \left|c, e, d, a\right\rangle = \rho^x_{d+e,a} \left|c, c+e+a\right\rangle \ , \tag{12.108}$$

acting trivially on b and f. $\widetilde{S}_x$ be expressed as a circuit,

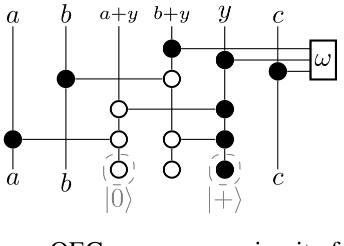


. (12.109)

Here we have split $x \to (x_0, x_1)$ into two qubits using $1 \to (0, 0)$, $s \to (1, 0)$, $\bar{s} \to (1, 1)$, and $s\bar{s} \to (0, 1)$. So the qubits labeled x_0, x_1, and $c + e + a$ are measured after applying the above isometry. ρ is a 2-qubit gate which in fact equals a Hadamard on the a qubit followed by a controlled-S gate. The operator W_1 in Eq. (12.104) becomes an isometry

$$\widetilde{W}_1 \left|a, b, c\right\rangle = \sum_y \omega_{y,y+b,c} \left|a, b, c, y, y+b, y+a\right\rangle \ . \tag{12.110}$$

$\widetilde{W}_1$ can be written as a circuit,



. (12.111)

We have thus decomposed our QEC process as a circuit of common 2 or 3-qubit gates. For a practical implementation it might again be useful to write this circuit in terms of measurements and unitaries acting on qubits on a fixed spatial lattice. This is straightforward, but might involve auxiliary qubits and swap operations.

12.4 Discussion and outlook

In this chapter we have proposed a perspective on topological quantum error correction based on topological fixed-point path integrals. Our approach provides a unified view on topological stabilizer, subsystem, and Floquet codes, as demonstrated in Section 12.2. In particular, we have seen that the stabilizer toric code, subsystem toric code, and CSS Floquet code can be considered the same code on different spacetime lattices. The approach can also describe topological QEC codes that are not based on Pauli/Clifford operations as we have demonstrated in Section 12.3.2. As summarized in Definition 27 and Proposition 7, we have given a simple unified criterion for when a circuit of measurements forms a fault-tolerant topological error-correcting code. Namely that, for every spacetime history of measurement outcomes, we obtain a topological fixed-point path integral including syndrome defects.

Our framework provides a way to systematically construct new codes. To this end, we start with some known fixed-point path integral, and possibly apply some tensor-network equations to obtain another path integral in the same fixed-point phase. Then we interpret this path integral as a circuit of operators by setting a time direction. Dressing every operator with segments of syndrome defects, we obtain a circuit of instruments with the desired properties. We have demonstrated this at hand of two examples in Section 12.3. First, we have presented a Floquet version of the $3+1$-dimensional toric code, by considering the tensor-network path integral on a hypercubic lattice and traversing it in the $t = x + y + z + w$ direction. The model has qubits living on the right-handed tetrahedra of a triangulation with 4-colored vertices. The code cycles through 8 rounds, in each of which we perform 2-body measurements among the qubits adjacent to edges of a certain type. Second, we have constructed a Floquet code based on the double-semion string net. This code is not designed to be particularly practical for implementation, but is decomposed into a sequence of common 2 or 3-qubit gates.

Ref. [30] proposes a similar perspective based on the ZX calculus. In that reference, it was independently recognized that the tensor-network diagrams for the stabilizer toric code and CSS Floquet code are the same, just traversed in a different direction. In addition to this, our work provides a clear physical interpretation of the tensor networks as topological fixed-point path integrals including topological defects. We also give a neat geometric interpretation of the phaseless ZX diagrams as cellulations, the ZX rules as topological invariance, and the *Pauli webs* or detection cells as volumes and vertices. As can be seen from Ref. [30] also fusion-based topological quantum computing [17] is described by our formalism. This holds true for topological measurement-based quantum computing [120] in general. A relation between the fusion-based model and Floquet CSS codes has also been pointed out in Ref. [117]. In contrast to all of the above examples, our formalism is not limited to the ZX calculus or stabilizer framework, but works for arbitrary tensor-network path integrals, as demonstrated in Section 12.3.2. [3]

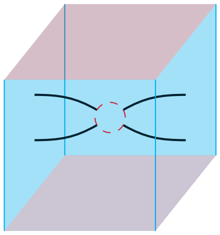
The framework can be generalized in various directions. First, topological state-sum path integrals do not cover all zero-correlation length path integrals, and similarly

[3] Even though any tensor can be written as a ZX diagram, it can be beneficial to work with elementary operations that are not elementary ZX tensors.

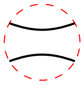
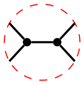
not all gapped phases. Exemptions can be obtained from topological path integrals by inserting a rigid network of topological defects, which we refer to as *foliation defects*. To this end, we choose some cubic "superlattice" with a potentially larger unit cell than the topological path integral. Then (in 2+1 dimensions) we introduce domain walls at all superlattice faces, which meet at 1-dimensional foliation defects along the edges, which in turn meet at the vertices. Examples for this in 2+1 dimensions seem to yield topological path integrals again after choosing a larger unit cell, and thus correspond to a "weak breaking of translation symmetry", as we have seen in Section 12.2.4. In 3+1 dimensions however, topological defect networks can describe fracton phases [3], and potentially more if we also insert foliation defects perpendicular to time [139]. Floquet codes based on fracton phases have been presented in Refs. [53, 144].

A second straight-forward generalization is to consider spacetime lattices that change with time. By changing the topology of the spatial configuration, we obtain circuits that do not only fault-tolerantly store, but also process logical information. Both storing and processing of logical information becomes much more versatile if we equip the topological path integral with computational defects such as boundaries, domain walls, or other sorts of interfaces and defects. For example, we can then perform computation via braiding with anyons or via lattice surgery with boundaries.

Another direction is to consider path integrals where the defects that we use for error correction (such as anyons) do not possess abelian fusion rules. In this case the scheme of Proposition 7 outlined in Eq. (12.10) cannot work, since there is not necessarily a unique way to perform a correction. For example, consider a path integral QEC circuit based on the double-Fibonacci phase, and assume we measure the following spacetime syndrome on a torus,


, (12.112)

with the left and right, as well as front and back identified. There are two ways of fixing the syndrome inside the red dashed circle, namely


, 
, (12.113)

which correspond to different logical operations acting on the ground space on a torus. There is no way to find out which superposition of these logical operations will correctly undo the error that occured. A decoding strategy that has been tested successfully is based on a hierachical decomposition of the lattice into *colonies* [51, 125]. A different strategy that might work is to "continuously" apply small corrections in every timestep instead of one large correction after a large time $T \sim L$. That is, in every time step, we choose a new low-weight fix of the spacetime syndrome in all of its past.

Then we consider the set of string operators that could be used to close the repaired spacetime syndrome in a cohomologically trivial way inside the current spatial cut. We pick a low (e.g., minimum) weight representative from this set. Then, we apply only a single segment of this closing string operator near each of its endpoints.

Even though it is not entirely settled how to design a classical decoder for non-abelian phases, our formalism readily allows us to construct circuits that extract syndrome information. In fact, the method in Section 12.3.2 straight-forwardly generalizes to arbitrary string-net models, or in other words, block-diagonal 3D-tTS models. A very interesting observation is that the circuit given in Refs. [29, 35] for fault-tolerantly performing a logical non-Clifford gate in $2+1$ dimensions corresponds to a non-Abelian phase when viewed as a path integral in our formalism. Specifically, the microscopics in Ref. [35] directly yield a cohomology model with $G=\mathbb{Z}_2^3$ and $\omega(a,b,c)=(-1)^{abc}$ on a cubic lattice. This gives hope that just-in-time decoders can also be used to fault-tolerantly decode more general abelian phases. Since the syndrome in these circuits does *not* correspond to the anyons, their microscopics is considerably simpler compared to the circuits resulting from generalizing Section 12.3.2.

Chapter 13

Conclusion and outlook

In this thesis, we have provided a systematic formalism for fixed-point descriptions of topological phases. Namely, we have shown that each family of fixed-point descriptions can be described as a tTS, that is, a set of tensor-network equations for a set of tensor-valued variables. Then each fixed-point description within one family was a solution to these equations. The equations were direct implementations of combinatorial topological invariance in discrete spacetime, either by recellulation for microscopic fixed-point models, or more generally by gluing for extended TQFT. It suffices to take one particularly coarse and simple "standard" ansatz for how to associate tensors to spacetime, namely with one tensor per n-cell and one index per $n-1$-cell. We have seen that this is not the only possible ansatz, and have given an intricate example with tensors at the edges and faces giving rise to an algebraic structure closely related to weak Hopf algebras. However, we have shown that the standard ansatz is universal if we assume the existence of a topological boundary. We have found a more complicated but natural ansatz that is universal even without a topological boundary, and therefore provides a route towards microscopic fixed-point models for chiral phases. We have developed the general method of block-diagonalization to transform the coarse ansatzes into more fine-tuned ones that turn out equivalent to the established higher-categorical structures. Our ansatz thus provides a way to derive these higher-categorical structures from simpler principles rather than just postulating them underpinned with non-rigorous physics arguments. It also gives a way to systematically find the many extra properties and structures that bare "n-categories" need to be equipped with.

We have tried to fit as many classifications as possible into the framework of this thesis. This includes the classification of intrinsic topological order up to spacetime dimension 4, and its boundaries and domain walls. Further, it includes a large range of lower-dimensional topological defects, namely anyons, boundary anyons, corners, twist defects, torus ground states, fusion events, condensation events, line and membrane defects in $3+1$-dimensional spacetime, as well as membrane-fusion line defects. Moreover, we have shown how to classify topological phases in models with symmetry, including time-reversal, as well as models with fermionic degrees of freedom in dimensions up to $2+1$. Last, we have defined atTS, which model topological phases through higher-level invariants, and seem to be roughly equivalent to extended TQFT.

There are still many more scenarios to which the formalism could be applied, such as combining fermions or symmetries with defects, boundaries, or atTS, being more explicit in $3+1$ dimensions, as well as studying many more types of defects. Most of these scenarios can be straight-forwardly generalized from the presented examples. Let us in the following point out a few aspects that might be not so straight-forward.

First, we have not discussed block-diagonalization for symmetric or fermionic models. 2D spin or G-cycle tTS models can be fully classified up local basis changes, just like 2D-tTS models in Section 4.13. In terms of algebras, 2D G-cycle tTS models are semi-simple algebras with an automorphism representation, which in the block-diagonal basis is given by a G-set action on the irrep labels together with a projective G-set representation. 2D spin tTS models correspond to super-algebras, which are classified by direct sums of the tensor product of the full-matrix algebra with either the trivial algebra or the Clifford algebra Cl_1. We now use these 2-dimensional classifications in higher-dimensional block-diagonalizations such as in Section 4.14. As a result, for example, the irreps in the fermionic case can be divided into trivial irreps and Cl_1 irreps, which are called q-type and m-type in Ref. [1]. The irreps in the symmetric case are equipped with G-set 2-cocycles, c.f. 5.1.3.

When discussing defects in models with symmetries or fermions, there is a further thing to keep in mind: The links of extended manifold types are decorated with the corresponding homological extra structure, that is, a G-cycle or spin structure η. If the link has non-trivial homology, then every homology class defines a different type of defect. As an example consider anyons, that is defects in $2+1$ dimensions with circle link. If we have a symmetry, then we need to equip the circle with a G 0-cycle, and there is one cohomology class for every value of the sum $g \in G$ over the 0-cycle. The defects with a fixed g are known as *symmetry defects*. For fermions, there are two different spin structures on the circle, namely the bounding and the non-bounding one. The defects with non-bounding circle as link are known as *vortices* in the fermionic model.

The different cohomology classes or spin structures on a circle also in the symmetric and fermionic versions of the atTS: In the corresponding extended manifold types in Eqs. (8.21) and 8.39, there are different versions of the space region b for different cohomology classes or spin structures. After block diagonalization, the resulting structure for symmetries is closely related to *G-crossed braided tensor categories*, c.f. Ref. [13].

We also did not discuss the 4D atTS, that is, the almost-fully extended TQFT description of $3+1$-dimensional topological order. This atTS is determined by an extended manifold type with an internal 4-region and space regions of dimension up to 2 with arbitrary links. The generating space-region links of the (block-diagonal) 4D atTS are just given by the generating extended manifolds of the (block-diagonal) 3D atTS given in Section 8.4.3, and the generating tensors correspond to the generating axioms of the 3D atTS. That is, most notably, the generating tensors contain a "pentagonator" and a "hexagonator". However, we do not know at this point how to systematically find the generating axioms. Surely, the (block-diagonal) 4D tTS is contained in the (block-diagonal) 4D atTS, so the 3-3 Pachner move determines one axiom.

One important discussion that we have omitted for length and time reasons is how to obtain atTS models from tTS models. In the framework of extended TQFT, a fully-extended TQFT trivially yields an almost-fully extended TQFT by restriction. However, if we define the fully-extended TQFT via a microscopic (tTS) model, extracting the corresponding atTS model becomes a non-trivial operation. In fact, it is given by a TS mapping that maps an nD-atTS extended cellulation to a n-cellulation with space boundary by simply removing all the space regions. We have explicitly done this in Section 8.3 for $1+1$ dimensions, where we replace a 2-manifold with point defects by a 2-manifold with 1-gon boundary "holes". Analogously, in $2+1$ dimensions, we replace a 3-manifolds with an embedded string-net by a 3-manifold with a "tunnel system" removed. This TS mapping becomes a non-trivial calculation if we start and end with the block-diagonal form, but in fact we have already done most of this computation. The atTS irreps are precisely the isomorphism classes of irreducible anyons that we computed in Section 6.2.2. In Section 6.7, we have chosen a basis for the fusion events of three anyon worldlines, which we need to do for all triples of anyons. Then, for example, the F-tensor defining the 3D atTS is obtained by evaluating a cellulation of the corresponding extended manifold using four fusion vertices and one bulk tetrahedron. We see that atTS models and defects are closely related. However, they are still distinct notions. For example, in $3+1$ dimensions, the membrane labels for membrane-nets embedded into 4-manifolds do *not* correspond to the irreducible membrane defects as we study in Section 6.9. Instead, they correspond to the robust sectors in the according compactification with the circle.

Apart from a framework that captures all different algebraic classifications of phases of matter, we have also studied some concrete examples. Most notably, we have provided a prescription that allows us to explicitly compute microscopic models for arbitrary types of defects of spacetime dimensions 0, 1, and 2 in n-dimensional twisted gauge theories. The prescription is efficient in the sense that the only non-trivial steps are decomposing finite G-sets into irreducible orbits, finding trivializing group $i-1$-cochains for i-coboundaries, and block-diagonalizing twisted group algebras for subgroups of G. The rest is plugging into explicit formulas.

Finally, we have found an application of tensor-network path integrals for fault-tolerant topological quantum computation in Chapter 12. On the one hand, they can be used to systematically predict the logical gate performed by an arbitrary topological process with defects as we sketch in Section 12.1.1. Second, by measuring syndrome defects, we can turn the path integrals into explicit error-correcting protocols given by a circuit of channels and syndrome measurements. So far we have done this only for the simplest phases of matter, and an interesting question will be how to apply it to more involved phases in both theory and practice. More future directions of this application were discussed in Section 12.4.

Acknowledgements

First of all, I would like to thank my supervisor, Jens Eisert. Thanks Jens, for many fun discussions and for a tremendous amount of trust and freedom to pursue my own research. Thanks for providing me with funding for seven years, for shielding me from administrative duties so I could focus on my research, and for always being reachable for anything urgent. Thanks for having created and maintaining this wonderful and gigantic research group — being able to interact with such a large number of interesting people has been a true enrichment. Also thanks to Annette, Marianne, Christiane, Claudia, and Felix, for keeping this group running over the years.

Next, I would like to thank all my collaborators. Thanks, Caro, for sharing my interest in topological order, and accompanying me on my endeavor to understand topological phases in terms of spacetime tensor networks. Thanks, Alex, for coming with me on a fun adventure on tensor types and how to redefine monoidal category theory without a flow of time. Thanks, Julio, for getting me on board for a wonderful project on bulk-to-boundary anyon condensation. Thanks, Bobak, Eric, and Seth, for a fun collaboration on efficient simulation of models with qubit permutation symmetries.

Thanks also to all my colleagues, friends, and climbing buddies in the department, and to all the people that I met on visits or conferences and with whom I had enlightening discussions throughout the years. It is impossible to not forget anyone, but I will give it a shot: Frederik Hahn, Dave Aasen, Dominik Hangleiter, Juani Bermejo-Vega, Ryan Sweke, Jonathan Conrad, Laura Baez, Max Geier, Guanyu Zhu, Margarita Davydova, Ben Brown, Dom Williamson, Sergio Acero, Jonas Haferkamp, Marek Gluza, Daniel Litinski, Fernando Pastawski, Nathan Walk, Ville Lahtinen, Nick Tarantino, Christian Klöckner, Nat Tantivasadakarn, Xiao-Gang Wen, Tyler Ellison, Yu-An Chen, Ryan Thorngren, Alexis Schotte, Norbert Schuch, Laurens Lootens, Caroline de Groot, Anna Francuz, Sal Pace, Alex Jahn.

Most importantly, a fruity thanks to my obst-office mates Alex and Markus, for a wonderful 7 years of laughter, biking, sailing, acrobatics, juggling, climbing, hiking, sternis, plants, bird feeders, organic experiments, pull-up and paper-ball-throwing competitions, tea, stickers, erster-games, and tasty fruit.

Last, but not least, thanks to my family, Irmi, Berti, Lena, Moni, Johnny, Matilda, Robert, Traudi, Uli, Kathi, Roland, Flo, Ingrid, and to my partner Lara, for always being there for me unconditionally in all my endeavors.

Selbstständigkeitserklärung

Name: Andreas Bauer

Ich erkläre gegenüber der Freien Universität Berlin, dass ich die vorliegende Dissertation selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe. Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht. Diese Dissertation wurde in gleicher oder ähnlicher Form noch in keinem früheren Promotionsverfahren eingereicht.
Mit einer Prüfung meiner Arbeit durch ein Plagiatsprüfungsprogramm erkläre ich mich einverstanden.

I declare to the Freie Universität Berlin that I have completed the submitted dissertation independently and without the use of sources and aids other than those indicated. The present thesis is free of plagiarism. I have marked as such all statements that are taken literally or in content from other writings. This dissertation has not been submitted in the same or similar form in any previous doctoral procedure.
I agree to have my thesis examined by a plagiarism examination software.

Date: ____________ Signature: ____

Bibliography

[1] David Aasen, Ethan Lake, and Kevin Walker. “Fermion condensation and super pivotal categories”. In: *J. Math. Phys.* 60 (2019), p. 121901. DOI: 10.1063/1.5045669. arXiv: 1709.01941.

[2] David Aasen, Zhenghan Wang, and Matthew B. Hastings. “Adiabatic paths of Hamiltonians, symmetries of topological order, and automorphism codes”. In: *Phys. Rev. B* 106 (8 2022), p. 085122. DOI: 10.1103/PhysRevB.106.085122. arXiv: 2203.11137.

[3] David Aasen et al. “Topological defect networks for fractons of all types”. In: *Phys. Rev. Research* 2 (4 2020), p. 043165. DOI: 10.1103/PhysRevResearch.2.043165. arXiv: 2002.05166.

[4] Alexander Altland and Martin R. Zirnbauer. “Novel Symmetry Classes in Mesoscopic Normal-Superconducting Hybrid Structures”. In: 55 (1997), p. 1142. DOI: 10.1103/PhysRevB.55.1142. arXiv: cond-mat/9602137.

[5] Eric R. Anschuetz et al. “Efficient classical algorithms for simulating symmetric quantum systems”. In: *Quantum* 7 (2023), p. 1189. DOI: 10.22331/q-2023-11-28-1189. arXiv: 2211.16998.

[6] M. Atiyah. “Topological quantum field theories”. In: *Inst. Hautes Études Sci. Publ. Math.* 68 (1988), pp. 175–186.

[7] Michael Atiyah. “On framings of 3-manifolds”. In: *Topology* 29.1 (1990), pp. 1–7.

[8] John C. Baez and James Dolan. “Higher-dimensional Algebra and Topological Quantum Field Theory”. In: 36 (1995), pp. 6073–6105. DOI: 10.1063/1.531236. arXiv: q-alg/9503002.

[9] Benjamin Balsam. *Turaev-Viro invariants as an extended TQFT II*. 2010. arXiv: 1010.1222.

[10] Benjamin Balsam. *Turaev-Viro invariants as an extended TQFT III*. 2010. arXiv: 1012.0560.

[11] Maissam Barkeshli et al. “Classification of (2+1)D invertible fermionic topological phases with symmetry”. In: *Phys. Rev. B* 105 (2022), p. 235143. DOI: 10.1103/PhysRevB.105.235143. arXiv: 2109.11039.

[12] Maissam Barkeshli et al. "Codimension-2 defects and higher symmetries in (3+1)D topological phases". In: *SciPost Phys.* 14 (2023), p. 065. DOI: 10.21468/SciPostPhys.14.4.065. arXiv: 2208.07367.

[13] Maissam Barkeshli et al. "Symmetry Fractionalization, Defects, and Gauging of Topological Phases". In: *Phys. Rev. B* 100 (2019), p. 115147. DOI: 10.1103/PhysRevB.100.115147. arXiv: 1410.4540.

[14] John W. Barrett and Bruce W. Westbury. "Invariants of Piecewise-Linear 3-Manifolds". In: *Trans. Amer. Math. Soc.* 348 (1996), pp. 3997–4022. DOI: 10.1090/S0002-9947-96-01660-1. arXiv: hep-th/9311155.

[15] T. Barthel, C. Pineda, and J. Eisert. "Unitary circuits for strongly correlated fermions". In: *Phys. Rev. A* 80 (2009), p. 042333. DOI: 10.48550/arXiv.0905.0669. arXiv: 0905.0669.

[16] Bruce Bartlett et al. *Extended 3-dimensional bordism as the theory of modular objects*. 2014. arXiv: 1411.0945.

[17] Sara Bartolucci et al. "Fusion-based quantum computation". In: *Nat Commun* 14 (2023), p. 912. DOI: 10.1038/s41467-023-36493-1. arXiv: 2101.09310.

[18] A. Bauer. *Generalized topological state-sum constructions and their universality*. 2019. arXiv: 1909.03031.

[19] A. Bauer. *Quantum mechanics is *-algebras and tensor networks*. 2020. arXiv: 2003.07976.

[20] A. Bauer, J. Eisert, and C. Wille. "A unified diagrammatic approach to topological fixed point models". In: *SciPost Phys. Core* 5 (3 2022), p. 38. DOI: 10.21468/SciPostPhysCore.5.3.038. arXiv: 2011.12064.

[21] A. Bauer, J. Eisert, and C. Wille. *Towards a mathematical formalism for classifying phases of matter*. 2019. arXiv: 1903.05413.

[22] A. Bauer and A. Nietner. *Tensor types and their usage in physics*. 2022. arXiv: 2208.01135.

[23] Andreas Bauer. "Disentangling modular Walker-Wang models via fermionic invertible boundaries". In: *Phys. Rev. B* 107 (2023), p. 085134. DOI: 10.1103/PhysRevB.107.085134. arXiv: 2208.03397.

[24] Andreas Bauer. *Topological error correcting processes from fixed-point path integrals*. 2023. arXiv: 2303.16405.

[25] Andreas Bauer, Jens Eisert, and Carolin Wille. "Towards topological fixed-point models beyond gappable boundaries". In: *Phys. Rev. B* 106 (12 2022), p. 125143. DOI: 10.1103/PhysRevB.106.125143. arXiv: 2111.14868.

[26] Salman Beigi, Peter W. Shor, and Daniel Whalen. "The Quantum Double Model with Boundary: Condensations and Symmetries". In: *Commun. Math. Phys.* 306.3 (2011), pp. 663–694. DOI: 10.1007/s00220-011-1294-x. arXiv: 1006.5479.

[27] H. Bombin. "Topological Subsystem Codes". In: *Phys. Rev. A* 81 (2010), p. 032301. DOI: 10.1103/PhysRevA.81.032301. arXiv: 0908.4246.

[28] H. Bombin and M. A. Martin-Delgado. "Exact Topological Quantum Order in D=3 and Beyond: Branyons and Brane-Net Condensates". In: *Phys.Rev.B* 75 (2007), p. 075103. DOI: 10.1103/PhysRevB.75.075103. arXiv: cond-mat/0607736.

[29] Hector Bombin. *2D quantum computation with 3D topological codes*. 2018. DOI: 10.48550/arXiv.1810.09571. arXiv: 1810.09571.

[30] Hector Bombin et al. *Unifying flavors of fault tolerance with the ZX calculus*. 2023. arXiv: 2303.08829.

[31] S. Bravyi and M. B. Hastings. "A short proof of stability of topological order under local perturbations". In: *Commun. Math. Phys.* 307 (2011), p. 609. DOI: 10.1007/s00220-011-1346-2. arXiv: 1001.4363.

[32] Sergey Bravyi et al. "Subsystem surface codes with three-qubit check operators". In: *Quant. Inf. Comp.* 13 (11&12 2013), pp. 0963–0985. arXiv: 1207.1443.

[33] J. C. Bridgeman. 2020. DOI: 10.5281/zenodo.4277499. URL: github.com/JCBridgeman/UnitarySphericalFusionData.

[34] Jacob C. Bridgeman, Laurens Lootens, and Frank Verstraete. "Invertible bimodule categories and generalized Schur orthogonality". In: (2022). DOI: 10.1007/s00220-023-04781-y. arXiv: 2211.01947.

[35] Benjamin J. Brown. "A fault-tolerant non-Clifford gate for the surface code in two dimensions". In: *Sci. Adv.* 6 (2020), p. 4929. DOI: 10.1126/sciadv.aay4929. arXiv: 1903.11634.

[36] Edgar H. Brown. "The Cohomology of BSO_n and BO_n with Integer Coefficients". In: *Proc. Amer. Math. Soc.* 85.2 (1982), pp. 283–288. URL: http://www.jstor.org/stable/2044298.

[37] O. Buerschaper et al. "A hierarchy of topological tensor network states". In: *J. Math. Phys.* 54 (2013), p. 012201. DOI: 10.1063/1.4773316. arXiv: 1007.5283.

[38] Alex Bullivant and Clement Delcamp. "Tube algebras, excitations statistics and compactification in gauge models of topological phases". In: *J. High Energ. Phys.* 2019 (2019), p. 216. DOI: 10.1007/JHEP10(2019)216. arXiv: 1905.08673.

[39] Nick Bultinck et al. "Anyons and matrix product operator algebras". In: *Ann. Phys.* 378 (2017), pp. 183–233. DOI: 10.1016/j.aop.2017.01.004. arXiv: 1511.08090.

[40] X. Chen, Z.-C. Gu, and X.-G. Wen. "Local unitary transformation, long-range quantum entanglement, wave function renormalization, and topological order". In: *Phys. Rev. B* 82 (2010), p. 155138. DOI: 10.1103/PhysRevB.82.155138. arXiv: 1004.3835.

[41] X. Chen et al. "Symmetry protected topological orders and the group cohomology of their symmetry group". In: *Phys. Rev. B* 87 (2013), p. 155114. DOI: 10.1103/PhysRevB.87.155114. arXiv: 1106.4772.

[42] J. I. Cirac et al. "Matrix product states and projected entangled pair states: Concepts, symmetries, and theorems". In: (2020). arXiv: 2011.12127.

[43] Iris Cong, Meng Cheng, and Zhenghan Wang. *Topological Quantum Computation with Gapped Boundaries*. 2016. arXiv: 1609.02037.

[44] Philippe Corboz and Guifre Vidal. "Fermionic multi-scale entanglement renormalization ansatz". In: *Phys. Rev. B* 80 (2009), p. 165129. DOI: 10.1103/PhysRevB.80.165129. arXiv: 0907.3184.

[45] L. Crane and Dd N. Yetter. "A categorical construction of 4D TQFTs". In: *Quantum Topology*. Ed. by Louis Kauffman and Randy Baadhio. Singapore: World Scientific, 1993. DOI: 10.1142/9789812796387_0005. arXiv: hep-th/9301062.

[46] Louis Crane, Louis H. Kauffman, and David N. Yetter. *State-Sum Invariants of 4-Manifolds I*. 1994. arXiv: hep-th/9409167.

[47] T. Cubitt, Perez-Garcia, and M. M. Wolf. "Undecidability of the spectral gap". In: *Nature* 528 (2015), p. 207. arXiv: 1502.04135.

[48] Shawn X. Cui. "Higher Categories and Topological Quantum Field Theories". In: *Quantum Topol.* 10.4 (2019), pp. 593–676. DOI: 10.4171/QT/128. arXiv: 1610.07628.

[49] Shawn X. Cui and Zhenghan Wang. "State Sum Invariants of Three Manifolds from Spherical Multi-fusion Categories". In: *J. Knot Theory Ramif.* 26 (14 2017). DOI: 10.1142/S0218216517501048. arXiv: 1702.07113.

[50] E. Evand D and Y. Kawahigashi. "On Ocneanu's theory of asymptotic inclusions for subfactors, topological quantum field theories and quantum doubles". In: *Int. J. Math.* 06 (1995), pp. 205–228. DOI: 10.1142/S0129167X95000468.

[51] Guillaume Dauphinais and David Poulin. "Fault-Tolerant Quantum Error Correction for non-Abelian Anyons". In: *Commun. Math. Phys.* 355 (2 2017), pp. 519–560. DOI: 10.1007/s00220-017-2923-9. arXiv: 1607.02159.

[52] Guillaume Dauphinais et al. "Quantum Error Correction with the Semion Code". In: *New J. Phys.* 21 (2019), p. 053035. DOI: 10.1088/1367-2630/ab1ed8. arXiv: 1810.08204.

[53] Margarita Davydova, Nathanan Tantivasadakarn, and Shankar Balasubramanian. *Floquet codes without parent subsystem codes*. 2022. arXiv: 2210.02468.

[54] Clement Delcamp. "Excitation basis for (3+1)d topological phases". In: *J. High Energ. Phys.* 128 (2017). DOI: 10.1007/JHEP12(2017)128. arXiv: 1709.04924.

[55] Clement Delcamp and Apoorv Tiwari. “On 2-form gauge models of topological phases”. In: *J. High Energ. Phys.* 64 (2019). DOI: 10.1007/JHEP05(2019)064. arXiv: 1901.02249.

[56] Michael DeMarco and Xiao-Gang Wen. *A commuting projector model with a non-zero quantized Hall conductance*. 2021. arXiv: 2102.13057.

[57] Eric Dennis et al. “Topological quantum memory”. In: *J. Math. Phys.* 43 (2002), pp. 4452–4505. DOI: 10.1063/1.1499754. arXiv: quant-ph/0110143.

[58] Robbert Dijkgraaf and Edward Witten. “Topological Gauge Theories and Group Cohomology”. In: *Commun. Math. Phys.* 129 (1990), pp. 393–429. DOI: 10.1007/BF02096988.

[59] Christopher L. Douglas and David J. Reutter. *Fusion 2-categories and a state-sum invariant for 4-manifolds*. 2018. arXiv: 1812.11933.

[60] J. Dubail and N. Read. “Tensor network trial states for chiral topological phases in two dimensions and a no-go theorem in any dimension”. In: *Phys. Rev. B* 92.20 (2015), p. 205307. DOI: 10.1103/PhysRevB.92.205307. arXiv: 1307.7726.

[61] Jack Edmonds. “Paths, Trees, and Flowers”. In: *Canadian Journal of Mathematics* 17 (1965), pp. 449–467. DOI: 10.4153/CJM-1965-045-4.

[62] Tyler D. Ellison et al. “Pauli Stabilizer Models of Twisted Quantum Doubles”. In: *PRX Quantum* 3 (1 2022), p. 010353. DOI: 10.1103/PRXQuantum.3.010353. arXiv: 2112.11394.

[63] Tyler D. Ellison et al. *Pauli topological subsystem codes from Abelian anyon theories*. 2022. arXiv: 2211.03798.

[64] Pavel Etingof et al. *Tensor categories*. Vol. 205. American Mathematical Soc., 2016. DOI: 10.1090/surv/205.

[65] Lukasz Fidkowski, Jeongwan Haah, and Matthew B. Hastings. “An Exactly Solvable Model for a $4+1D$ Beyond-Cohomology Symmetry Protected Topological Phase”. In: *Phys. Rev. B* 101 (2020), p. 155124. DOI: 10.1103/PhysRevB.101.155124. arXiv: 1912.05565.

[66] Julio C. Magdalena de la Fuente, Jens Eisert, and Andreas Bauer. “Bulk-to-boundary anyon fusion from microscopic models”. In: *J. Math. Phys.* 64.11 (2023), p. 111904. DOI: 10.1063/5.0147335. arXiv: 2302.01835.

[67] M. Fukuma, S. Hosono, and H. Kawai. “Lattice Topological Field Theory in Two Dimensions”. In: *Commun. Math. Phys.* 161 (1994), pp. 157–176. DOI: 10.1007/BF02099416. arXiv: hep-th/9212154.

[68] A. A. Gaifullin. “Local formulae for combinatorial Pontrjagin classes”. In: *Izv. Math.* 68 (5 2004), pp. 861–910. DOI: 10.4213/im502. arXiv: math/0407035.

[69] Alexander A. Gaifullin and Denis Gorodkov. “An explicit local combinatorial formula for the first Pontryagin class”. In: *Russ. Math. Surv.* 74 (2019), p. 6. DOI: 10.1070/RM9920.

[70] D. Gaiotto and A. Kapustin. “Spin TQFTs and fermionic phases of matter”. In: *Int. J. Mod. Phys. A* 31 (2016), p. 1645044. DOI: 10.1142/S0217751X16450445. arXiv: 1505.05856.

[71] Craig Gidney. *A Pair Measurement Surface Code on Pentagons*. 2022. arXiv: 2206.12780.

[72] Richard Z. Goldstein and Edward C. Turner. “A formula for Stiefel-Whitney homology classes”. In: *Proc. Amer. Math. Soc.* 58 (1976), pp. 339–342. DOI: 10.1090/S0002-9939-1976-0415643-5.

[73] J. Haah. “Bifurcation in entanglement renormalization group flow of a gapped spin model”. In: *Phys. Rev. B* 89 (2014), p. 075119. DOI: 10.1103/PhysRevB.89.075119. arXiv: 1310.4507.

[74] J. Haah, L. Fidkowski, and M. B. Hastings. “Nontrivial quantum cellular automata in higher dimensions”. In: *Commun. Math. Phys.* (2022). DOI: 10.1007/s00220-022-04528-1. arXiv: 1812.01625.

[75] Jeongwan Haah. “Clifford Quantum Cellular Automata: Trivial group in 2D and Witt group in 3D”. In: *J. Math. Phys.* 62 (2021), p. 092202. DOI: 10.1063/5.0022185. arXiv: 1907.02075.

[76] Jeongwan Haah. “Local stabilizer codes in three dimensions without string logical operators”. In: *Phys. Rev. A* 83 (2011), p. 042330. DOI: 10.1103/PhysRevA.83.042330. arXiv: 1101.1962.

[77] Alexander Hahn and Ramona Wolf. “Generalized string-nets for unitary fusion categories without tetrahedral symmetry”. In: *Phys. Rev. B* 102 (2020), p. 115154. DOI: 10.1103/PhysRevB.102.115154. arXiv: 2004.07045.

[78] M. B. Hastings and Xiao-Gang Wen. “Quasi-adiabatic Continuation of Quantum States: The Stability of Topological Ground State Degeneracy and Emergent Gauge Invariance”. In: *Phys.Rev. B* 72 (2005), p. 045141. DOI: 10.1103/PhysRevB.72.045141. arXiv: cond-mat/0503554.

[79] Matthew B. Hastings and Jeongwan Haah. “Dynamically Generated Logical Qubits”. In: *Quantum* 5 (2021), p. 564. DOI: 10.22331/q-2021-10-19-564. arXiv: 2107.02194.

[80] Seung-Moon Hong. *On symmetrization of 6j-symbols and Levin-Wen Hamiltonian*. 2009. arXiv: 0907.2204.

[81] Yuting Hu, Nathan Geer, and Yong-Shi Wu. “Full Dyon Excitation Spectrum in Generalized Levin-Wen Models”. In: *Phys. Rev. B* 97 (2018), p. 195154. DOI: 10.1103/PhysRevB.97.195154. arXiv: 1502.03433.

[82] Yuting Hu, Yidun Wan, and Yong-Shi Wu. “Twisted quantum double model of topological phases in two dimensions”. In: *Phys. Rev. B* 87 (2013), p. 125114. DOI: 10.1103/PhysRevB.87.125114. arXiv: 1211.3695.

[83] Ling-Yan Hung and Yidun Wan. “String-Net Models with Z_N Fusion Algebra”. In: *Phys. Rev. B* 86 (2012), p. 235132. DOI: 10.1103/PhysRevB.86.235132. arXiv: 1207.6169.

[84] Kansei Inamura and Kantaro Ohmori. *Fusion Surface Models: 2+1d Lattice Models from Fusion 2-Categories*. 2023. arXiv: 2305.05774.

[85] Corey Jones and David Penneys. "Operator algebras in rigid C*-tensor categories". In: *Commun. Math. Phys.* 355 (2017), pp. 1121–1188. DOI: 10.1007/s00220-017-2964-0. arXiv: 1611.04620.

[86] Alexander Kirillov Jr. and Benjamin Balsam. *Turaev-Viro invariants as an extended TQFT*. 2010. arXiv: 1004.1533.

[87] Anton Kapustin and Lukasz Fidkowski. "Local commuting projector Hamiltonians and the quantum Hall effect". In: *Commun. Math. Phys.* 373 (2020), pp. 763–769. DOI: 10.1007/s00220-019-03444-1. arXiv: 1810.07756.

[88] Anton Kapustin, Nikita Sopenko, and Bowen Yang. "A classification of phases of bosonic quantum lattice systems in one dimension". In: *J. Math. Phys.* 62 (2021), p. 081901. DOI: 10.1063/5.0055996. arXiv: 2012.15491.

[89] Anton Kapustin and Lev Spodyneiko. "Thermal Hall conductance and a relative topological invariant of gapped two-dimensional systems". In: *Phys. Rev. B* 101 (2020), p. 045137. DOI: 10.1103/PhysRevB.101.045137. arXiv: 1905.06488.

[90] Anton Kapustin and Ryan Thorngren. "Topological Field Theory on a Lattice, Discrete Theta-Angles and Confinement". In: *Adv. Theor. Math. Phys.* 18 (5 2014), pp. 1233–1247. arXiv: 1308.2926.

[91] R. M. Kashaev. "On realizations of pachner moves in 4d". In: *J. Knot Theory Ramif.* 24.13 (2015), p. 1541002. arXiv: 1504.01979.

[92] Markus S. Kesselring et al. *Anyon condensation and the color code*. 2022. arXiv: 2212.00042.

[93] Aleks Kissinger. *Phase-free ZX diagrams are CSS codes (...or how to graphically grok the surface code)*. 2022. arXiv: 2204.14038.

[94] A. Kitaev. "Unpaired Majorana fermions in quantum wires". In: *Phys.-Usp.* 44 (2001), p. 131. DOI: 10.1070/1063-7869/44/10S/S29. arXiv: cond-mat/0010440.

[95] A. Yu. Kitaev. "Fault-tolerant quantum computation by anyons". In: *Ann. Phys.* 303 (2003), pp. 2–30. DOI: 10.1016/S0003-4916(02)00018-0. arXiv: quant-ph/9707021.

[96] Alexei Kitaev. "Anyons in an exactly solved model and beyond". In: *Ann. Phys.* 321 (2006), pp. 2–111. DOI: 10.1016/j.aop.2005.10.005. arXiv: cond-mat/0506438.

[97] Alexei Kitaev and Liang Kong. "Models for gapped boundaries and domain walls". In: *Commun. Math. Phys.* 313 (2012), pp. 351–373. DOI: 10.1007/s00220-012-1500-5. arXiv: 1104.5047.

[98] J. Kock. "Frobenius Algebras and 2D Topological Quantum Field Theories". In: *London Math. Soc. Student Texts* 59 (2004). DOI: 10.1017/CBO9780511615443.

[99] Christina V. Kraus et al. “Fermionic Projected Entangled Pair States”. In: *Phys. Rev. A* 81 (2010), p. 052338. DOI: 10.1103/PhysRevA.81.052338. arXiv: 0904.4667.

[100] David Kribs, Raymond Laflamme, and David Poulin. “A Unified and Generalized Approach to Quantum Error Correction”. In: *Phys. Rev. Lett.* 94 (2005), p. 180501. DOI: 10.1103/PhysRevLett.94.180501. arXiv: quant-ph/0412076.

[101] Aleksander Kubica and John Preskill. “Cellular-automaton decoders with provable thresholds for topological codes”. In: *Phys. Rev. Lett.* 123 (2019), p. 020501. DOI: 10.1103/PhysRevLett.123.020501. arXiv: 1809.10145.

[102] G. Kuperberg. “Involutory Hopf algebras and 3-manifold invariants”. In: *Internat. J. Math.* 2.1 (1991), pp. 41–66. DOI: 10.1142/S0129167X91000053. arXiv: math/9201301.

[103] Tian Lan and Xiao-Gang Wen. “Topological quasiparticles and the holographic bulk-edge relation in 2+1D string-net models”. In: *Phys. Rev. B* 90 (2014), p. 115119. DOI: 10.1103/PhysRevB.90.115119. arXiv: 1311.1784.

[104] Lev D. Landau. “On the Theory of Phase Transitions”. In: *Zh. Eksp. Teor. Fiz.* 7 (1937), pp. 19–32.

[105] M. A. Levin and X.-G. Wen. “String-net condensation: A physical mechanism for topological phases”. In: *Phys. Rev. B* 71 (2005), p. 045110. DOI: 10.1103/PhysRevB.71.045110.

[106] Michael Levin. “Protected edge modes without symmetry”. In: *Phys. Rev. X* 3 (2013), p. 021009. DOI: 10.1103/PhysRevX.3.021009. arXiv: 1301.7355.

[107] Norman Levitt and Colin Rourke. “The existence of combinatorial formulae for characteristic classes”. In: *Transactions of the American Mathematical Society* 239 (1978), pp. 391–397. DOI: 10.1090/S0002-9947-1978-0494134-6.

[108] Chien-Hung Lin and Michael Levin. “Generalizations and limitations of string-net models”. In: *Phys. Rev. B* 89 (2014), p. 195130. DOI: 10.1103/PhysRevB.89.195130.

[109] Jacob Lurie. *On the Classification of Topological Field Theories*. 2009. arXiv: 0905.0465.

[110] Julio Carlos Magdalena de la Fuente, Nicolas Tarantino, and Jens Eisert. “Non-Pauli topological stabilizer codes from twisted quantum doubles”. In: *Quantum* 5 (Feb. 2021), p. 398. DOI: 10.22331/q-2021-02-17-398. arXiv: 2001.11516.

[111] Richard Mandelbaum. “Four-dimensional topology: an introduction”. In: *Bull. Am. Math. Soc.* 2 (1980), pp. 1–159. DOI: 10.1090/S0273-0979-1980-14687-X.

[112] J.P. May. *A Concise Course in Algebraic Topology*. Chicago Lectures in Mathematics. University of Chicago Press, 1999. ISBN: 9780226511832. URL: https://www.math.uchicago.edu/~may/CONCISE/ConciseRevised.pdf.

[113] John W. Milnor and James D. Stasheff. *Characteristic Classes*. Vol. 76. Annals of Mathematics Studies. Princeton University Press, 1974. ISBN: 9781400881826. URL: https://books.google.de/books?id=vJbKCwAAQBAJ.

[114] Chetan Nayak et al. "Non-Abelian Anyons and Topological Quantum Computation". In: *Rev. Mod. Phys.* 1083 (2008), p. 80. DOI: 10.1103/RevModPhys.80.1083. arXiv: 0707.1889.

[115] "Operator Algebras, Topology and Subgroups of Quantum Symmetry – Construction of Subgroups of Quantum Groups". In: *Adv. Stud. Pure Math.* 31 (2001), pp. 235–263. DOI: 10.2969/aspm/03110235.

[116] U. Pachner. "P. L. homeomorphic manifolds are equivalent by elementary shellings". In: *Europ. J. Comb.* 12.2 (1991), pp. 129–145. DOI: 10.1016/S0195-6698(13)80080-7. URL: http://www.sciencedirect.com/science/article/pii/S0195669813800807.

[117] Stefano Paesani and Benjamin J. Brown. *High-threshold quantum computing by fusing one-dimensional cluster states*. 2022. arXiv: 2212.06775.

[118] Adam Paetznick et al. "Performance of planar Floquet codes with Majorana-based qubits". In: *PRX Quantum* 4 (2023), p. 010310. DOI: 10.1103/PRXQuantum.4.010310. arXiv: 2202.11829.

[119] Lorenzo Piroli and J. Ignacio Cirac. "Quantum Cellular Automata, Tensor Networks, and Area Laws". In: *Phys. Rev. Lett.* 125 (2020), p. 190402. DOI: 10.1103/PhysRevLett.125.190402. arXiv: 2007.15371.

[120] Robert Raussendorf, Jim Harrington, and Kovid Goyal. "Topological fault-tolerance in cluster state quantum computation". In: *New Journal of Physics* 9 (2007), p. 199. DOI: 10.1088/1367-2630/9/6/199. arXiv: quant-ph/0703143.

[121] N. Reshetikhin and V.G. Turaev. "Invariants of 3-manifolds via link polynomials and quantum groups". In: *Invent. Math.* 103 (1991), pp. 547–597. DOI: 10.1007/BF01239527.

[122] Eric Rowell, Richard Stong, and Zhenghan Wang. "On classification of modular tensor categories". In: *Comm. Math. Phys.* 292 (2009), pp. 343–389. DOI: 10.1007/s00220-009-0908-z. arXiv: 0712.1377.

[123] Shinsei Ryu et al. "Topological insulators and superconductors: ten-fold way and dimensional hierarchy". In: 12 (2010), p. 065010. DOI: 10.1088/1367-2630/12/6/065010. arXiv: 0912.2157.

[124] M. B. Sahinoglu. "A tensor network study of topological quantum phases of matter". PhD thesis. Universität Wien, 2016.

[125] Alexis Schotte, Lander Burgelman, and Guanyu Zhu. *Fault-tolerant error correction for a universal non-Abelian topological quantum computer at finite temperature*. 2022. arXiv: 2301.00054.

[126] Alexis Schotte et al. "Quantum error correction thresholds for the universal Fibonacci Turaev-Viro code". In: *Phys. Rev. X* 12 (2 2022), p. 021012. DOI: 10.1103/PhysRevX.12.021012. arXiv: 2012.04610.

[127] Norbert Schuch, David Perez-Garcia, and Ignacio Cirac. "Classifying quantum phases using matrix product states and projected entangled pair states". In: *Phys. Rev. B* 84.16 (2011). DOI: 10.1103/physrevb.84.165139. arXiv: 1010.3732.

[128] Wilbur Shirley et al. "Three-dimensional quantum cellular automata from chiral semion surface topological order and beyond". In: *PRX Quantum* 3 (3 2022), p. 030326. DOI: 10.1103/PRXQuantum.3.030326. arXiv: 2202.05442.

[129] N. E. Steenrod. "Products of Cocycles and Extensions of Mappings". In: *Ann. Math.* 48.2 (1947), pp. 290–320. ISSN: 0003486X. DOI: 10.2307/1969172.

[130] V. G. Turaev and O. Y. Viro. "State sum invariants of 3-manifolds and quantum 6j-symbols". In: *Topology* 31 (1992), pp. 865–902. DOI: 10.1016/0040-9383(92)90015-A.

[131] Vladimir G. Turaev. "Modular Categories and 3-Manifold Invariants". In: *Int. J. Mod. Phys. B* 6 (11-12 1992), pp. 1807–1824.

[132] R. Verresen, R. Moessner, and F. Pollmann. "One-Dimensional Symmetry Protected Topological Phases and their Transitions". In: *Phys. Rev. B* 96 (2017), p. 165124. DOI: 10.1103/PhysRevB.96.165124. arXiv: 1707.05787.

[133] K. Walker and Z. Wang. "(3+1)-TQFTs and Topological Insulators". In: *Front. Phys.* 7 (2012), p. 150. DOI: 10.1007/s11467-011-0194-z. arXiv: 1104.2632.

[134] Kevin Walker. *A universal state sum*. 2021. arXiv: 2104.02101.

[135] Xiao-Gang Wen. "A theory of 2+1D bosonic topological orders". In: *Natl. Sci. Rev.* 3 (2016), pp. 68–106. DOI: 10.1093/nsr/nwv077. arXiv: 1506.05768.

[136] Xiao-Gang Wen. "Topological Orders in Rigid States". In: *Int. J. Mod. Phys. B* 4 (2 1990), p. 239. DOI: 10.1142/S0217979290000139.

[137] Wikipedia. *Bitruncated cubic honeycomb*. URL: https://en.wikipedia.org/wiki/Bitruncated_cubic_honeycomb.

[138] D. J. Williamson and Z. Wang. "Hamiltonian models for topological phases of matter in three spatial dimensions". In: *Ann. Phys.* 377 (2017), pp. 311–344. DOI: 10.1016/j.aop.2016.12.018. arXiv: 1606.07144.

[139] Dominic Williamson. *Spacetime topological defect networks and floquet codes*. KITP Conference: Noisy Intermediate-Scale Quantum Systems: Advances and Applications. 2022. DOI: 10.26081/K69H2K.

[140] Edward Witten. “Quantum field theory and the Jones polynomial”. In: *Comm. Math. Phys.* 121 (3 1989), pp. 351–399. DOI: 10.1007/BF01217730.

[141] Wenjie Xi et al. *A lattice realization of general three-dimensional topological order*. 2021. arXiv: 2110.06079.

[142] David Yetter. “TQFT’s from homotopy 2-types”. In: *J. Knot Theory Ramif.* 02 (1993). DOI: 10.1142/S0218216593000076.

[143] B. Zeng et al. *Quantum information meets quantum matter – from quantum entanglement to topological phase in many-body systems*. Berlin: Springer, 2015.

[144] Zhehao Zhang, David Aasen, and Sagar Vijay. *The X-Cube Floquet Code*. 2022. arXiv: 2211.05784.