Exploiting Symmetry in Variational Quantum Machine Learning

Johannes Jakob Meyer,[1,*] Marian Mularski,[1,2] Elies Gil-Fuster,[1,3] Antonio Anna Mele,[1]
Francesco Arzani,[1] Alissa Wilms,[1,2] and Jens Eisert[1,3,4]

[1] *Dahlem Center for Complex Quantum Systems, Freie Universität Berlin, Berlin 14195, Germany*
[2] *Porsche Digital GmbH, Ludwigsburg 71636, Germany*
[3] *Fraunhofer Heinrich Hertz Institute, Berlin 10587, Germany*
[4] *Helmholtz-Zentrum Berlin für Materialien und Energie, Berlin 14109, Germany*

(Received 17 June 2022; revised 26 September 2022; accepted 18 January 2023; published 17 March 2023)

Variational quantum machine learning is an extensively studied application of near-term quantum computers. The success of variational quantum learning models crucially depends on finding a suitable parametrization of the model that encodes an inductive bias relevant to the learning task. However, precious little is known about guiding principles for the construction of suitable parametrizations. In this work, we holistically explore when and how symmetries of the learning problem can be exploited to construct quantum learning models with outcomes invariant under the symmetry of the learning task. Building on tools from representation theory, we show how a standard gateset can be transformed into an equivariant gateset that respects the symmetries of the problem at hand through a process of gate symmetrization. We benchmark the proposed methods on two toy problems that feature a nontrivial symmetry and observe a substantial increase in generalization performance. As our tools can also be applied in a straightforward way to other variational problems with symmetric structure, we show how equivariant gatesets can be used in variational quantum eigensolvers.

DOI: 10.1103/PRXQuantum.4.010328

The advent of quantum devices that come close to outperforming classical computers in certain computational tasks [1–3] has sparked a wealth of investigations into the capabilities of *noisy intermediate-scale quantum* (NISQ) devices [4]. These endeavors are aimed at exploiting the computational power of quantum computers without quantum error correction that make use of relatively short quantum circuits. A particular emphasis is put on *hybrid* quantum classical approaches that use the quantum device to implement subroutines in otherwise largely classical algorithms [5].

Important areas of applications intended for these hybrid approaches include problems in quantum chemistry, classical combinatorial optimization, and machine learning [6,7]. *Symmetries* have always played a very important role in the analysis of physical systems, due to their frequent admitting of conserved quantities. Combining this with Noether's theorem informs us that conserved quantities correspond to symmetries of the underlying system. In quantum chemistry, for example, solutions to ground-state problems or electronic structure calculations have to respect conserved quantities such as the global particle number or the parity of the fermion number and hence admit nontrivial symmetries.

The usefulness of symmetries is not limited to the domain of physical systems, however, and their role in the context of classical machine learning must not be understated. Symmetries have been pivotal in the development of successful models for image recognition that sparked a revolution in learning with artificial neural networks [8]. More recent breakthroughs, like the development of the transformer model [9], are also linked to a better understanding of the underlying symmetries of the learning task. Consequently, the rigorous treatment of symmetries in machine learning has recently culminated in the creation of the subfield of *geometric deep learning* [10].

In this work, we explore how symmetries of a learning task can be exploited to create quantum learning models whose output is invariant under symmetry transformations on the level of the data. The fact that classical data have to be embedded into the Hilbert space of a quantum system makes it difficult to directly transfer ideas from the classical world to the construction of variational quantum

[*] jj.meyer@outlook.com

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

learning models. As our main contribution, we show how to embed data in a way that enables us to build invariant learning models for which we give both a general blueprint and mathematical tools for their design.

In Sec. II, we exhibit data embeddings that give rise to meaningful representations of practically relevant data symmetries on the level of the Hilbert space, both for discrete and continuous symmetries. The trainable parts of variational quantum learning models are usually constructed using gates from a standard gateset. In Sec. III we show how the representation on the level of the Hilbert space can be exploited to symmetrize the generators of such a gateset to give rise to an equivariant gateset. We especially detail which pitfalls should be avoided when constructing an equivariant gateset. In Sec. IV we show how these building blocks can be combined to construct variational quantum data reuploading models that make invariant predictions and thus include a meaningful inductive bias. A visual summary of our approach is given in Fig. 1.

To elaborate on how our approaches are applied and to showcase the increase in performance, we conduct a number of numerical experiments in Sec. V. We consider the classification of games of tic tac toe as a paradigmatic learning task with nontrivial symmetry and show a significant increase in generalization performance that even generically holds for random constructions in the equivariant gateset. We further consider a learning task with similar symmetry properties that is inspired by a problem from the automotive industry and which can, as soon as the quantum computational resources become available, be scaled up to address the actual task. For this task, concerned with the classification of the criticality of driving scenarios for autonomous vehicles, we reach the same conclusions as for the tic-tac-toe example. We also argue that the idea of equivariant gatesets itself is not limited to applications in quantum machine learning. This is why we additionally provide numerical investigations into ground-state problems of the transverse-field Ising model, the Heisenberg model and a toy model based on the longitudinal transverse-field Ising model that exhibits the same symmetries as the tic-tac-toe problem. For these kinds of problems, equivariant ansätze usually help, but the benefit is less apparent than in the case of quantum machine learning models.

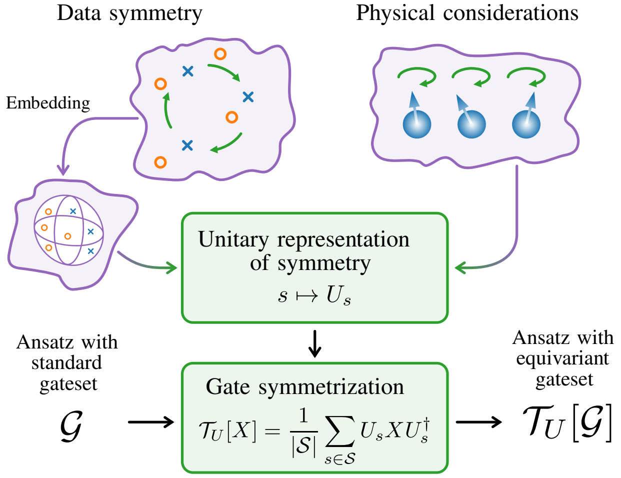

Data symmetry
Physical considerations
Embedding
Unitary representation of symmetry
$s \mapsto U_s$
Ansatz with standard gateset
$\mathcal{G}$
Gate symmetrization
$\mathcal{T}_U[X] = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} U_s X U_s^\dagger$
Ansatz with equivariant gateset
$\mathcal{T}_U[\mathcal{G}]$

FIG. 1. A unitary representation of a symmetry group $\mathcal{S}$ can arise from data symmetries when the data points are suitably encoded or alternatively from physical considerations of a variational problem. We can use such a representation to replace gates with their equivariant counterparts and thus symmetrize generic ansätze used in the construction of learning models and variational algorithms.

There is already a substantial amount of literature on the application of symmetries in the context of variational quantum algorithms. Explicit constructions of ansätze and gadgets for quantum chemistry problems can be found in Refs. [11–18]. Another approach has been chosen by the authors of Refs. [19,20] where symmetries were imposed through classical postprocessing. An approach in the same direction as that we outline in this work has been proposed in Ref. [21], where it has been suggested to simply remove generators that violate the symmetry or fix the parameters of the gates.

Here, we are reaching the same goal via a distinctly different route, as we propose to alter the set of generators used in the construction of the ansatz. Further works that treat symmetry-enhanced variational methods are found in Refs. [22–24]. Another possibility outside of altering the ansatz itself to enforce symmetries is to enforce them through the optimizer via the use of additional terms in the cost function that penalize asymmetric states as in Refs. [25–27]. In a different direction, Mernyei *et al.* [28] proposed a quantum model that closely follows the geometric deep learning blueprint [10] by having first a symmetry equivariant quantum "layer," followed by a symmetry invariant classical "aggregator." In particular, they proposed a model for graph-based learning tasks, so the symmetry they considered is the permutation (or relabeling) of the graph nodes. Also specifically for graphs, Verdon *et al.* [29] included a recipe for building permutation invariant models inspired by *graph convolutional neural networks* [10]. Zheng *et al.* [30] looked at the construction of group-equivariant ansatz states to learn quantum states. Glick *et al.* [31] proposed a construction of covariant quantum kernels that evaluate data that already comes with a group structure.

I. PRELIMINARIES

A. Variational quantum algorithms

Variational quantum algorithms [6] are a major research direction in the study of the capabilities of near-term quantum devices to solve practically relevant problems. In these hybrid quantum classical approaches [5], a quantum device uses a *parametrized quantum circuit* (PQC) as an *ansatz* to prepare a parametrized quantum state vector $|\psi(\boldsymbol{\theta})\rangle$. After the preparation of the state, measurements are performed in order to estimate the expectation values of a set of observables $\{O_1, O_2, \dots\}$. These expectation values are then used to compute a *cost function* $C(\boldsymbol{\theta}, \langle O_1\rangle, \langle O_2\rangle, \dots)$ that encodes the problem of interest. Particularly prominent applications are the *variational quantum eigensolver* (VQE) where the cost function is given by the expectation value of a Hamiltonian H whose ground state is to be approximated and the *quantum approximate optimization algorithm* (QAOA) that combines a specific ansatz construction with VQE to approximate solutions to classical optimization problems. As variational quantum algorithms offer an extremely flexible paradigm to tackle problems, such algorithms have been proposed for a wide variety of different problems beyond that [7].

B. Variational quantum learning models

With the current success of machine learning, especially neural-network-based deep learning, the interest into possible gains from applying quantum computers to learning problems has come into focus, sparking the field of *quantum machine learning*. Variational approaches as outlined above can be used to construct *variational quantum learning models* (VQLMs); see, e.g., Refs. [32–34]. In such circuit-based models, the ansatz state does not depend only on variational parameters but also on the input data, and predictions are encoded in expectation values of observables

$$y(\mathbf{x}) = \langle\psi(\boldsymbol{\theta}, \mathbf{x})|O|\psi(\boldsymbol{\theta}, \mathbf{x})\rangle. \tag{1}$$

Because of the no-cloning theorem, it is important for the expressivity of quantum learning models to embed the data into the quantum circuit multiple times, a process dubbed *data reuploading* [35–37]. In the following, we therefore consider VQLMs where a fixed data-embedding unitary $U(\mathbf{x})$ is interleaved with parametrized quantum circuits as

$$|\psi(\boldsymbol{\theta}, \mathbf{x})\rangle = W_L(\boldsymbol{\theta})U(\mathbf{x}) \cdots W_1(\boldsymbol{\theta})U(\mathbf{x})W_0(\boldsymbol{\theta})\,|\psi_0\rangle\,. \tag{2}$$

C. Symmetry groups

Symmetries are commonly captured by *groups*. A group $\mathcal{S}$ is a set of objects $s \in \mathcal{S}$ together with a composition rule "$\circ$"—usually called "multiplication"—so that $s_1 \circ s_2 \in \mathcal{S}$ for all $s_1, s_2 \in \mathcal{S}$. The multiplication is also required to be associative, i.e., $(s_1 \circ s_2) \circ s_3 = s_1 \circ (s_2 \circ s_3)$. Furthermore, there is an identity element e so that $e \circ s = s \circ e = s$ for all $s \in \mathcal{S}$ and that, for all $s \in \mathcal{S}$, there exists a unique inverse element s^{-1} so that $s \circ s^{-1} = s^{-1} \circ s = e$. A simple example of a group is $\mathbb{Z}_2 = \{0, 1\}$ where the composition is given by addition modulo 2. If a restriction of $\mathcal{S}$ to some subset is itself a group under the composition rule of $\mathcal{S}$, we call this subset a *subgroup*.

The concept of groups also extends to continuous sets in the form of *Lie groups*. A real Lie group is a group $\mathcal{S}$ that is also a smooth manifold and the composition rule and inversion are smooth maps. One example of a (noncompact) Lie group is the set of all invertible linear operators $\mathrm{GL}_n(\mathbb{C})$. Lie groups that are a subgroup of $\mathrm{GL}_n(\mathbb{C})$ and can hence be expressed in terms of matrices are called *matrix Lie groups* and we focus on these in the following. An example of a matrix Lie group are the unitary matrices acting on $\mathbb{C}^n$, denoted $\mathrm{U}(n)$.

Lie groups are intimately related to *Lie algebras* that can be seen as the generators of elements of the Lie group. For a matrix Lie group $\mathcal{S}$, we can define the associated algebra as

$$\mathfrak{s} = \mathrm{Lie}(\mathcal{S}) = \{G\colon\ \exp(tG) \in \mathcal{S} \text{ for all } t \in \mathbb{R}\}. \tag{3}$$

This is motivated by the fact that, for two group elements $s_1, s_2 \in \mathcal{S}$, we can understand the group multiplication on the level of the Lie algebra via the Baker-Campbell-Hausdorff formula

$$s_1 s_2 = e^{G_1} e^{G_2} = e^{G_1 + G_2 + [G_1, G_2]/2 + \cdots}. \tag{4}$$

For s_1 and s_2 close to the identity element, the norms of G_1 and G_2 are small, in which case their commutator gives the first correction. It is actually part of the formal definition of a Lie algebra and is called the Lie bracket.

A *representation* of a group is a map $U\colon \mathcal{S} \to \mathrm{Aut}(V)$, $s \mapsto U_s$ that is compatible with the composition rule: $U_{s_1 \circ s_2} = U_{s_1} U_{s_2}$. If the space V is equipped with a scalar product and the linear maps U_s are unitary, we call this a unitary representation.

II. SYMMETRIES INDUCED BY DATA EMBEDDINGS

Symmetries play an important role all across physics and problems that are of interest in near-term applications are no different in this respect. A symmetry group $\mathcal{S}$ acts on an n-qubit Hilbert space $\mathcal{H}$ through a unitary representation U_s for $s \in \mathcal{S}$.

In ground-state problems—as they are encountered in the variational quantum eigensolver [38]—symmetries are usually derived from physical considerations. An example of such a symmetry, which is encountered for example in the Heisenberg model, is that the energy of a state does not change if all spins are flipped, i.e., when the operator $X^{\otimes n}$ is applied. This can be understood as a representation of the symmetry group $\mathbb{Z}_2$. Further symmetries include joint unitary transformations of all spins [SU(2)] or translation symmetry ($\mathbb{Z}_n$). Such symmetries have already been extensively studied in the context of VQE, with a particular focus towards building symmetry-preserving ansätze for quantum chemistry applications [11–18].

But ground-state problems are not the only problems of interest in near-term applications that can have a symmetric structure. Indeed, many learning problems encountered in the context of quantum machine learning also have some symmetry: a prototypical example is the labeling of images. Even if we move all the pixels of a cat photo to the right or rotate it, the photo still depicts a cat. More formally, consider a task where a prediction $y \in \mathcal{Y}$ should be associated with data points $\mathbf{x} \in \mathcal{X}$. We say that the prediction is invariant under a symmetry group $\mathcal{S}$ with representation $V_s \colon \mathcal{S} \to \operatorname{Aut}(\mathcal{X})$ if

$$y(V_s[\mathbf{x}]) = y(\mathbf{x}) \quad \text{for all } \mathbf{x} \in \mathcal{X} \text{ and } s \in \mathcal{S}. \tag{5}$$

Notions like this have recently sparked the subfield of *geometric deep learning* [10] that abstractly reasons about these symmetries and their implementation in learning models, particularly in artificial neural networks. The geometric reasoning also elucidates the early success of convolutional neural networks [8] that use building blocks that respect the translation symmetry present in typical image classification tasks and the success of transformer models [9]. It is also intuitive to expect that building symmetry-invariant models comes with advantages: by definition, a symmetry-invariant model does not see any difference between data points that are related by a symmetry transformation, which effectively reduces the number of possible predictions the learning model can make. Such a model has lower complexity and is thus expected to be easier to train. Furthermore, these models only produce predictions that respect the symmetry already present in the learning problem and are thus expected to generalize better to unseen data.

In the following, we show how to construct quantum circuits that embed classical data in a way that gives rise to a meaningful unitary representation of the symmetry on the level of the Hilbert space that we call an *induced representation*. These embeddings are then used as building blocks in the construction of symmetry-invariant data reuploading models.

A. Introductory example

We first consider an introductory example that is given in Fig. 2. Assume that we have data points $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ and the prediction—for example, a classification—has the symmetries

$$y(x_1, x_2) = y(x_2, x_1) = y(-x_1, -x_2), \tag{6}$$

which correspond to an exchange of the coordinates and a simultaneous sign flip. The associated symmetry group, famously known as Klein's four group, is given by

$$\mathbb{Z}_2 \times \mathbb{Z}_2 = \{(0,0), (1,0), (0,1), (1,1)\} \tag{7}$$

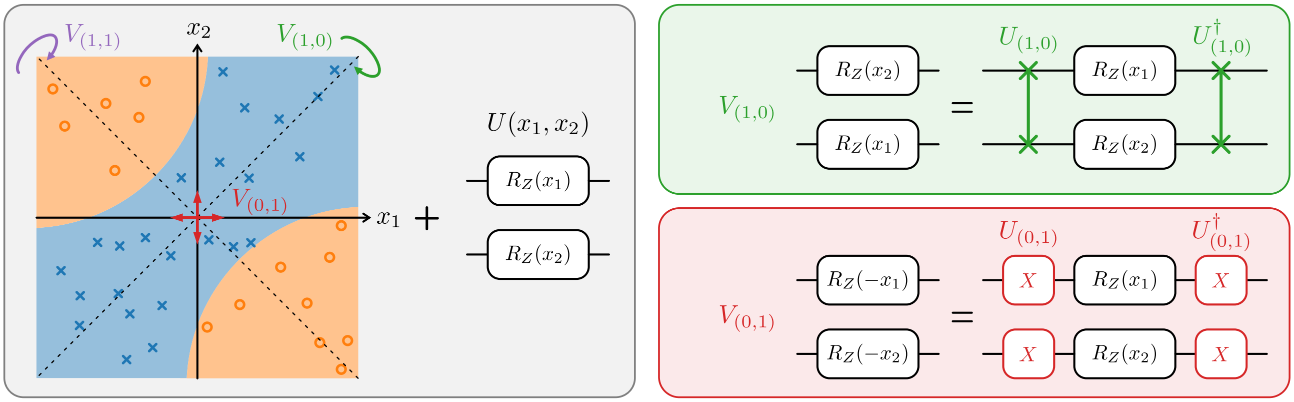
with the group operation being entrywise addition modulo 2. In this abstract definition of the symmetry group, the first component could be understood as the Boolean answer to the question "do we exchange?" The second component could be understood as the Boolean answer to the question "do we flip?" The representation on the level of the data is given by

$$V_{(0,0)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad V_{(1,0)} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \tag{8}$$

$$V_{(0,1)} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \qquad V_{(1,1)} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}. \tag{9}$$

If we use an embedding unitary

$$U(x_1, x_2) = R_Z(x_1) \otimes R_Z(x_2) \tag{10}$$



$\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetric problem + embedding

Induced representation

FIG. 2. Simple example of a classification problem with two features (x_1, x_2) that has a $\mathbb{Z}_2 \times \mathbb{Z}_2$ symmetry generated by a flip about the axis $x_1 = x_2$ (green) and inversion about the center (red). Combining both operations yields a flip about the axis $x_1 = -x_2$ (purple). Together with the simple qubitwise embedding of the coordinates, these induce a representation of $\mathbb{Z}_2 \times \mathbb{Z}_2$ generated by a SWAP operation of the two qubits and a simultaneous application of Pauli X on the two qubits.

then the symmetry operations can be represented on the level of the Hilbert space as

$$\begin{aligned} U(x_2,x_1) &= R_Z(x_2)\otimes R_Z(x_1) \\ &= \text{SWAP}\, U(x_1,x_2)\,\text{SWAP}, \qquad (11)\\ U(-x_1,-x_2) &= R_Z(-x_1)\otimes R_Z(-x_2) \\ &= (X\otimes X)U(x_1,x_2)(X\otimes X), \qquad (12)\end{aligned}$$

where we have used the fact that $XZX = -Z$. Note that we could have equivalently used $(Y\otimes Y)$ in the above example, so the operations are not necessarily unique.

In the example, we see that symmetry operations on the data map directly to unitary transformations of the underlying Hilbert space. We can use this intuition to formalize the concept of an *induced representation* and to define what it means for an embedding to be *equivariant* [39].

Definition 1 (Equivariant embedding): We call a data encoding unitary $U(\mathbf{x})$ *equivariant* with respect to the data symmetry V_s if

$$U(V_s[\mathbf{x}]) = U_s U(\mathbf{x}) U_s^\dagger \qquad (13)$$

for a unitary representation U_s of $\mathcal{S}$. The representation U_s is the *induced representation* of the embedding $U(\mathbf{x})$.

The representation induced by our above example is then

$$\begin{aligned} U_{(0,0)} &= \mathbb{I}\otimes\mathbb{I}, \qquad U_{(1,0)} = \text{SWAP}, \qquad (14)\\ U_{(0,1)} &= X\otimes X, \qquad U_{(1,1)} = \text{SWAP}(X\otimes X). \qquad (15)\end{aligned}$$

In the following, we show how we can construct data encodings that are equivariant with respect to the most important symmetry classes encountered in real-world learning tasks.

B. Permutation symmetries

Most data encountered in contemporary learning tasks, like images or time series data, have a discrete structure. Symmetry transformations of the data, like translating or rotating an image or translating a time series, can thus be captured by a permutation of the different data features. This means that in many relevant scenarios with symmetry, the symmetry group will be a subgroup of the *permutation group* (or *symmetric group*) S_d of the d different data features $\mathbf{x} = (x_1,\ldots,x_d)$.

The elements of the symmetric groups are all possible reshufflings σ of the data features

$$\sigma(x_1,x_2,\ldots,x_d) = (x_{\sigma(1)},x_{\sigma(2)},\ldots,x_{\sigma(d)}). \qquad (16)$$

Every permutation can be constructed by concatenating *transpositions*, which are exchanges of two data features at a time. We write a transposition as τ such that, for all $\sigma\in S_d$, we have $\sigma = \tau_m\cdots\tau_2\tau_1$ for some transpositions $\{\tau_i\}$. Note that a product of transpositions is usually considered to be implemented from right to left.

In the previous section, we already saw how to construct a data encoding that is equivariant with respect to the permutation of two elements, and the construction directly generalizes to higher dimensions, needing d qubits for d-dimensional data. We complement this with a related construction that allows using approximately $\log d$ qubits but still retains equivariance. The natural construction is to embed every data feature via a Pauli rotation on a separate qubit; the choice of the generator is not relevant, so we settle for Z, to get

$$U(x_1,\ldots,x_d) = R_Z(x_1)\otimes\cdots\otimes R_Z(x_d), \qquad (17)$$

where the gate $R_Z(x_i)$ acts on the ith qubit. This encoding is obviously equivariant with respect to the permutations of the qubits and induces a representation of S_d that is given by

$$U_\sigma = U_{\tau_m\cdots\tau_2\tau_1} = \text{SWAP}_{\tau_m}\cdots\text{SWAP}_{\tau_2}\text{SWAP}_{\tau_1}, \qquad (18)$$

where SWAP_τ swaps the two qubits corresponding to the features swapped by τ. This embedding actually allows us to go even further than the permutation group and also equivariantly embed sign flips of the ith coordinate by conjugating with Pauli X on the ith qubit, thus generalizing to $S_d\rtimes\mathbb{Z}_2^d$ [40].

With this strategy, we need as many qubits as data features. A natural question is whether it is possible to do better, namely, if we could find a different encoding gate that embedded d features in less than d qubits and still be equivariant with respect to the permutation group. If we want to add another Pauli word as a generator to the above set of local Z operators, we are limited by the fact that this generator has to be invariant under all SWAP operations between qubits because these should, by definition, only exchange the associated coordinates and not the newly added one. This means that the only operator we can add is $Z^{\otimes d}$, yielding one additional data feature we could in principle encode with this strategy. In this case, the representation for exchanging the ith coordinate with the $(d+1)$th coordinate is a multi-controlled-NOT gate where the ith qubit controls a simultaneous application of $X^{\otimes d-1}$ on the rest of the qubits.

Note that we could in principle employ a different set of mutually commuting Pauli words as any set of independent, mutually commuting Pauli words are interchangeable via Clifford gates [41]. One example would be the set of generators $\{Z\otimes\mathbb{I},\mathbb{I}\otimes Z, Z\otimes Z\}$ that could equivalently be replaced by $\{X\otimes X, Y\otimes Y, Z\otimes Z\}$, which would constitute a more "balanced" encoding strategy as the operators are of similar weight. For this set of operators, the permutations are generated by the transpositions

$$\tau_{12} \leftrightarrow H_{XY} \otimes H_{XY}, \tag{19}$$

$$\tau_{23} \leftrightarrow H_{YZ} \otimes H_{YZ}, \tag{20}$$

$$\tau_{31} \leftrightarrow H_{ZX} \otimes H_{ZX}, \tag{21}$$

where H_{AB} is a generalization of the Hadamard operator $H = H_{ZX}$ given by $H_{AB} = \exp(-i\pi(A+B)/2\sqrt{2})$.

The encoding based on local Z rotations is very intimately related to the instantaneous quantum polynomial (IQP) encoding proposed by Havlíček *et al.* [42]. We can use their inspiration to also include higher monomials of the data features like, for example, $\xi = x_1 x_4$. If we embed such a monomial with a generator $Z_1 Z_4$, where Z_i is a local Z acting on the ith qubit, then we see that a permutation of the features can be realized by a permutation of the qubits. An equivariant embedding of the data is then achieved by simultaneously embedding all possible permutations of monomial like x_1x_2, x_1x_3, x_2x_3, and so on using the operators Z_1Z_2, Z_1Z_3, Z_2Z_3, etc. The number of terms needed for this construction grows rather quickly because all possible permutations have to be considered. On the other hand, this means that we can expect to have to embed fewer terms if we only wish to realize a subgroup of the permutation group. If we consider only translations that map $x_i x_j$ to $x_{i+1}x_{j+1}$, we only need to embed all possible translations of the monomial that is linear in the number of features.

While the above strategy for embedding data is straightforward to implement, we can in principle embed exponentially many data features in an equivariant way. One strategy to do so is to assign every state of the computational basis – understood as the collection of all $|k\rangle$ identified by the binary representation of the integer k – a generator

basis as the collection of all identified by the binary representation of the integer a generator

$$G_k = |k\rangle\langle k|. \tag{22}$$

This generates unitaries $R_k(x_k) = \exp(-ix_k|k\rangle\langle k|)$ that only put a phase on the amplitudes of the kth computational basis state. The transposition of features k and l is then easily seen to be achieved by

$$\tau_{kl} \leftrightarrow |k\rangle\langle l| + |l\rangle\langle k| + \sum_{j\neq k,l} |j\rangle\langle j|, \tag{23}$$

which induces a representation of the permutation group for a number of data features up to 2^n. However, contrary to the simple Z rotations in the example above, these rotations are very difficult to implement, as they mostly correspond to operations controlled on nearly all qubits. Furthermore, as we will see in Sec. III, if all of the available basis states are used, only trivial operations will respect the symmetry of the system.

The above example motivates the claim that the construction of equivariant embeddings should be possible with relatively few qubits, but at the cost of very complicated gates that need to be implemented. This reduces the practicality of such an approach, especially in the NISQ era. The other extreme is the 1-local embedding proposed at the beginning of this section, which essentially only allows us to embed as many data features as there are qubits but which can be implemented very easily. It is therefore an interesting direction for future research what kind of embeddings can be defined that interpolate between these two regimes and allow us to embed more data features than the 1-local embedding while still being sufficiently implementable.

C. Continuous symmetries

While many cases of practical interest display discrete symmetries, we need not constrain ourselves to this case. Especially when looking at possible applications in the natural sciences, we can also encounter continuous symmetries, if, for example, rotating spatial data points around a certain axis do not alter predictions. The framework for dealing with continuous symmetries is given by Lie groups that are continuous groups with a differentiable structure.

Representation theory tells us that any compact Lie group can be expressed as a subgroup of a suitably large unitary group [43], and as any unitary group can be embedded in a larger special unitary group, every Lie group can be represented on a suitably large quantum system. However, it is not *a priori* clear which Lie groups admit embeddings that induce a faithful unitary representation. We show how we can construct an embedding that is equivariant with respect to the special orthogonal group in three dimensions, SO(3), and show how we can extend this to the full orthogonal group O(3). We then outline how the general task can be understood mathematically.

1. SO(3) and O(3)

The group SO(3) is made of all possible rotations of a sphere centered at the origin. For elements of the special orthogonal group, we use lowercase letters to distinguish them from their quantum counterparts. Any element $r \in \mathrm{SO}(3)$ can be constructed by successively applying one of the three canonical rotations about the axes of the coordinate system that we denote as $r_x(\alpha), r_y(\alpha)$, and $r_z(\alpha)$. In the following, we use the parametrization in terms of the Euler angles given by

$$r(\psi,\theta,\phi) = r_z(\psi) r_x(\theta) r_z(\phi). \tag{24}$$

We now define an equivariant embedding of a data point $\mathbf{x} \in \mathbb{R}^3$ as

$$U(\mathbf{x}) = e^{-i(x_1 X + x_2 Y + x_3 Z)/2} = e^{-i\langle \mathbf{x}, \boldsymbol{\sigma}\rangle/2}, \tag{25}$$

where we have defined the vector of Pauli operators $\boldsymbol{\sigma} = (X, Y, Z)$. We can now exploit that this is essentially a mapping from data points to the Bloch sphere, of which we know that conjugation with a Pauli rotation generates the associated rotation of the Bloch sphere to obtain the induced representation. We only need

$$\begin{aligned}\langle r_x(\alpha)\mathbf{x}, \boldsymbol{\sigma}\rangle &= \langle \mathbf{x}, r_x(-\alpha)\boldsymbol{\sigma}\rangle \\ &= R_X(-\alpha)\langle \mathbf{x}, \boldsymbol{\sigma}\rangle R_X(\alpha),\end{aligned} \tag{26}$$

$$\langle r_z(\alpha)\mathbf{x}, \boldsymbol{\sigma}\rangle = R_Z(-\alpha)\langle \mathbf{x}, \boldsymbol{\sigma}\rangle R_Z(\alpha), \tag{27}$$

to deduce that

$$\begin{aligned}U(r(\psi,\theta,\phi)\mathbf{x}) &= e^{-i\langle \mathbf{x}, r(-\phi,-\theta,-\psi)\boldsymbol{\sigma}\rangle/2} \\ &= e^{-i\langle \mathbf{x}, r_z(-\phi)r_x(-\theta)r_z(-\psi)\boldsymbol{\sigma}\rangle/2} \\ &= R_Z(-\psi)R_X(-\theta)R_Z(-\phi)e^{-i\langle \mathbf{x}, \boldsymbol{\sigma}\rangle/2} \\ &\quad \times R_Z(\phi)R_X(\theta)R_Z(\psi).\end{aligned} \tag{28}$$

At this point, we recall that there is a parametrization for arbitrary unitaries $U \in \mathrm{SU}(2)$, also in terms of three angles $U(\psi,\theta,\phi) = R_Z(\psi)R_X(\theta)R_Z(\phi)$. Using this, we arrive at the desired induced representation

$$U(r(\psi,\theta,\phi)\mathbf{x}) = U(-\psi,-\theta,-\phi)U(\mathbf{x})U^\dagger(-\psi,-\theta,-\phi). \tag{29}$$

The full group of all orthogonal (i.e., length-preserving) transformations on $\mathbb{R}^d$, O(3), is made of not only rotations but also reflections about any plane that passes through the origin. Luckily, reflections about different planes can be related by the rotation that maps the planes to one another. This means that one fixed reflection, together with the three canonical rotations, is enough to span the whole group O(3). We pick the plane perpendicular to the vector $(1, 1, 1)$, which realizes an inflection about the origin $\mathbf{x} \mapsto -\mathbf{x}$. As we have seen that we have already exhausted SU(2) to represent SO(3), this cannot be possible using an embedding with only a single qubit. However, we can actually straightforwardly realize this inflection if we add an additional qubit and use the embedding

$$U(\mathbf{x}) = e^{-i(x_1X + x_2Y + x_3Z)/2\otimes X}. \tag{30}$$

The rotations of SO(3) are embedded in the same way as before on the first qubit, but now the inflection can also be realized as

$$\begin{aligned}U(-\mathbf{x}) &= e^{-i(-x_1X - x_2Y - x_3Z)/2\otimes X} \\ &= e^{-i(x_1X + x_2Y + x_3Z)/2\otimes -X} \\ &= (\mathbb{I}\otimes Z)e^{-i(x_1X + x_2Y + x_3Z)/2\otimes X}(\mathbb{I}\otimes Z) \\ &= (\mathbb{I}\otimes Z)U(\mathbf{x})(\mathbb{I}\otimes Z).\end{aligned} \tag{31}$$

In this way, we can generate the entire group O(3).

2. The general case

We have found an encoding that is equivariant with respect to SO(3) transformations, but the immediate question arising from this construction is if it can be generalized to arbitrary Lie groups acting on the data. To this end, we mathematically formalize the process that leads to an equivariant embedding. Note that it is required that the symmetry $\mathcal{S}$ must have nontrivial finite-dimensional unitary representations, which rules out interesting groups like $\mathrm{SO}(3, 1)$ as they are not compact groups.

We assume that the embedding of the data is of the form

$$U(\mathbf{x}) = e^{-i\mathcal{E}(\mathbf{x})}, \tag{32}$$

which means that we effectively embed the data into the Lie algebra of the quantum system $\mathfrak{su}(2^n)$. We further assume that the map from the data to the Lie algebra $\mathcal{E}\colon \mathbb{R}^d \to \mathfrak{su}(2^n)$ is linear. We model the symmetry transformation on the level of the data as a representation of the symmetry Lie group $\mathcal{S}$ given by $V_s\colon \mathbb{R}^d \to \mathbb{R}^d$. Our aim is to find a map $W_s\colon\ \mathfrak{su}(2^n) \to \mathfrak{su}(2^n)$, such that the following diagram commutes:

$$\begin{tikzcd}\mathbb{R}^d \arrow[r, "\mathcal{E}"] \arrow[d, "V_s"'] & \mathfrak{su}(2^n) \arrow[d, "W_s"] \\ \mathbb{R}^d \arrow[r, "\mathcal{E}"] & \mathfrak{su}(2^n)\end{tikzcd}$$

Here, W_s is a transformation that implements the symmetry transformation on the level of the Lie algebra in an equivariant way. This means that it needs to be a conjugation by a unitary

$$W_s[X] = U_s X U_s^\dagger \tag{33}$$

for some $U_s \in \mathrm{SU}(2^n)$ that represents the symmetry group $\mathcal{S}$ acting on the data. The above is nothing but the adjoint action of U_s, which if $U_s = e^H$, is generated by the adjoint action on the level of the Lie algebra

$$U_s X U_s^\dagger = \mathrm{Ad}_{U_s}[X] = e^{\mathrm{ad}_H}[X], \tag{34}$$

where $\mathrm{ad}_H[X] = [H, X]$. Looking back to the data space, we can use the fact that V_s is a matrix representation of a Lie group and that we can write $V_s = e^G$ for some element of the Lie algebra of $\mathcal{S}$, $G \in \mathfrak{s}$. Making the diagram commute then enforces

$$\mathcal{E}e^G \overset{!}{=} e^{\mathrm{ad}_H}\mathcal{E}, \tag{35}$$

which can be quickly verified to be the case if

$$\mathcal{E}G = \mathrm{ad}_H\,\mathcal{E}. \tag{36}$$

In other words, this means that, to construct an equivariant embedding for a given Lie group representation acting on the data, we need to find a subspace of $\mathfrak{su}(2^n)$—which is then the image of $\mathcal{E}$—such that the restriction of the adjoint representation of $\mathfrak{su}(2^n)$ to this subspace is *identical* to the representation of the symmetry Lie algebra $\mathfrak{s}$ acting on the data, G.

This elucidates *why* we are able to construct an embedding equivariant with respect to SO(3). It is because the fundamental representation of the Lie algebra $\mathfrak{so}(3)$ that acts on the data is equal to the adjoint representation of $\mathfrak{su}(2)$. Additional to this, we can also see that there always exists a trivial embedding $\mathcal{E}_0[X] = 0$ that induces a trivial representation of the Lie algebra. From the above reasoning, we also see that it is not enough to find a sub-Lie algebra isomorphic to the one acting on the data; we really have to make sure that there is an appropriate subrepresentation of the adjoint representation of $\mathfrak{su}(2^n)$ identical to the representation of $\mathfrak{s}$ on the data space. As we think that the realization of embeddings equivariant with respect to symmetries more exotic than O(3) is a topic of limited interest, we leave the classification of equivariant embeddings that can be realized in this setup as a topic of further investigation.

III. GATE SYMMETRIZATION

As we have seen above, symmetries arise naturally in variational quantum learning models if an equivariant embedding is used. In these applications, it is paramount to construct the trainable parts of the model in a way that encodes an inductive bias suitable to the problem at hand. The problem we face is that the knowledge of the relation between parametrized quantum circuits and the associated inductive bias is not really understood, leaving us with precariously little to inform our construction of learning models. Symmetries provide a first avenue to the construction of better quantum learning models as they allow us to include knowledge about the underlying data into the model in a meaningful way. They furthermore allow us to reduce the complexity of the ansatz as measured by the number of free parameters and thus also save resources. The same holds for other variational applications, e.g., in the search for ground states. There, the goal is not a better generalization capability but a better expressivity in the relevant parts of the Hilbert space.

In this section, we explain how we can use elementary group theory to construct an *equivariant gateset* from a standard gateset used in ansatz constructions, where, as we will make formal below, we define equivariant as commuting with the symmetry representation. This allows us to take an existing ansatz and make it equivariant by replacing every gate by its equivariant counterpart. We also explore why this approach has its advantages but is no panacea as there are several pitfalls to avoid. This sometimes makes it advisable not to build equivariance with respect to the full symmetry group into the model but to only consider subgroups thereof.

A. Equivariant gatesets

We focus on gates generated by a fixed generator G as

$$R_G(\theta) = \exp(-i\theta G), \tag{37}$$

as they are usually encountered in ansatz constructions for variational approaches. When constructing an ansatz, gates are picked with generators from a fixed gateset $G \in \mathcal{G}$.

We call a gate *equivariant* with respect to the symmetry embodied by a unitary representation U_s of $s \in \mathcal{S}$ if the order of applying the symmetry operation and the gate itself does not matter, so that

$$[R_G(\theta), U_s] = 0 \quad \text{for all } \theta \in \mathbb{R},\ s \in \mathcal{S}. \tag{38}$$

This can only be the case if the generator itself commutes with the representation that is captured by the following proposition.

Proposition 1 (Commuting generators): *For a given gate R_G, we have*

$$[R_G(\theta), U_s] = 0 \quad \textit{for all } \theta \in \mathbb{R},\ s \in \mathcal{S}, \tag{39}$$

if and only if $[G, U_s] = 0$ for all $s \in \mathcal{S}$.

Proof. First, we show that the condition is necessary. For this, consider the expansion of $R_G[\theta]$ to first order

$$\begin{aligned} 0 &= [R_G(\theta), U_s] \\ &= [\mathbb{I} - i\theta G + O(\theta^2), U_s] \\ &= i\theta[G, U_s] + O(\theta^2) && (40) \\ \iff \quad 0 &= [G, U_s], && (41) \end{aligned}$$

as this relation also needs to be valid for infinitesimally small θ. The condition is obviously sufficient as $[G, U_s] = 0$ implies that all powers of G and hence the full exponential commutes with U_s. ■

Luckily for us, there is a straightforward way to ensure that a generator does commute with a given representation—we can make use of the *twirling formula*.

Proposition 2 (Twirling formula [44]): *Let U_s be a unitary representation of $\mathcal{S}$. Then,*

$$\mathcal{T}_U[X] = \frac{1}{|\mathcal{S}|} \sum_{s\in\mathcal{S}} U_s X U_s^\dagger \tag{42}$$

defines a projector onto the set of operators commuting with all elements of the representation, i.e., $[\mathcal{T}_U[X], U_s] = 0$ for all X and $s \in \mathcal{S}$.

The same holds for Lie groups if we replace the uniform average with an integration over the Haar measure μ,

$$\mathcal{T}_U[X] = \int d\mu(s)\, U_s X U_s^\dagger. \tag{43}$$

We use this approach to associate with any gateset $\mathcal{G}$ an *equivariant gateset*

$$\mathcal{T}_U[\mathcal{G}] = \{\mathcal{T}_U[G] \mid G \in \mathcal{G}\}. \tag{44}$$

Note that this approach also directly extends to gates that are not parametrized, as these gates can either be directly symmetrized via the twirling formula or they can be written as a parametrized gate with fixed evolution angle as in Eq. (37).

B. Ansatz symmetrization

We can now use the aforementioned gate symmetrization technique to convert a complete ansatz (or trainable block) to an equivariant ansatz (or equivariant trainable block) by replacing the ansatz's gateset with its equivariant counterpart. The computation of the equivariant gateset can for practical purposes usually be done efficiently beforehand.

To make this more palpable, let us return to our example with an exchange symmetry on two qubits. Assume that an ansatz is made up of local rotation gates generated by the Pauli operators, $\{X, Y, Z\}$, and entangling gates generated by a ZZ interaction, all of which we consider as trainable. In this case, the gateset is

$$\mathcal{G} = \{X_1, Y_1, Z_1, X_2, Y_2, Z_2, Z_1Z_2\}, \tag{45}$$

where the index identifies the qubit the Pauli is acting on. The R_{ZZ} gate already commutes with the SWAP operation, which means that we can focus on the local operations only.

In our example, the symmetry group is $\mathbb{Z}_2 \times \mathbb{Z}_2$, where the symmetries are generated by SWAP (exchange $\leftrightarrow$) and $X \otimes X = X_1X_2$ (sign flip $\pm$). We first consider the symmetrization over the two subgroups only to synergize afterwards.

The generator Z_1Z_2 already commutes with the SWAP operations and thus stays invariant under symmetrization. If we apply the symmetrization with respect to the exchange symmetry to $X_1 = X \otimes \mathbb{I}$, we obtain

$$\begin{aligned}\mathcal{T}_{U_\leftrightarrow}[X_1] &= \mathcal{T}_{U_\leftrightarrow}[X \otimes \mathbb{I}]\\ &= \tfrac{1}{2}[X \otimes \mathbb{I} + \text{SWAP}(X \otimes \mathbb{I})\text{SWAP}]\\ &= \tfrac{1}{2}[X \otimes \mathbb{I} + \mathbb{I} \otimes X]\\ &= \tfrac{1}{2}[X_1 + X_2]\\ &= \mathcal{T}_{U_\leftrightarrow}[X_2].\end{aligned} \tag{46}$$

The symmetrization proceeds analogously for the other operators. We see that X_1 and X_2 map to the same operator upon symmetrization—hence, the symmetrized gateset has a lower cardinality. This means that, as expected, the subspace of symmetry-preserving unitaries is smaller than the full space of unitaries and that upon symmetrization, we reduce the number of parameters, and hence the complexity, of the ansatz. The gateset equivariant with respect to the exchange symmetry is then given by symmetric entangling gates and simultaneous Pauli rotations on the two qubits that share the same angle,

$$\mathcal{G}_\leftrightarrow = \left\{\frac{X_1 + X_2}{2}, \frac{Y_1 + Y_2}{2}, \frac{Z_1 + Z_2}{2}, Z_1Z_2\right\}. \tag{47}$$

Now, let us look at the sign flip symmetry. Again, Z_1Z_2 already commutes with X_1X_2 and stays invariant under symmetrization. The situation is again different for the local gates. It is quite straightforward that X_1 and X_2 commute with X_1X_2, but this is not true for the other gates as

$$\begin{aligned}\mathcal{T}_{U_\pm}[Y_1] &= \mathcal{T}_{U_\pm}[Y \otimes \mathbb{I}]\\ &= \tfrac{1}{2}[Y \otimes \mathbb{I} + (X \otimes X)(Y \otimes \mathbb{I})(X \otimes X)]\\ &= \tfrac{1}{2}[Y \otimes \mathbb{I} - Y \otimes \mathbb{I}]\\ &= 0,\end{aligned} \tag{48}$$

where we have used the fact that $XYX = -Y$. The calculation goes analogously for Y_2 and also for Z_1 and Z_2 because $XZX = -Z$. This means that the sign flip equivariant gateset looks rather different than that for the exchange symmetry, as in this case we are only allowed local Pauli-X rotations

$$\mathcal{G}_\pm = \{X_1, X_2, Z_1Z_2\}. \tag{49}$$

Because of the commuting nature of the group, we can obtain the fully equivariant gateset by applying either symmetrization procedure to the other gateset, which yields

$$\mathcal{G}_{\leftrightarrow,\pm} = \left\{\frac{X_1 + X_2}{2}, Z_1Z_2\right\}. \tag{50}$$

We see that taking into account the full symmetry greatly reduces the amount of available operations, but which comes at a cost of reduced expressivity as we will also detail further in the next section.

C. Pitfalls to avoid

Making an ansatz equivariant through symmetrization can bring considerable advantages but is by no means a panacea. There are important considerations and a series of pitfalls that one should be aware of.

First and foremost, there is always a trade-off between the gain in equivariance and therefore specialization to the relevant subspace and the reduction in expressivity of the ansatz. This is the point where quantum machine learning and ground-state problems arguably differ the most: in machine learning applications, maximal expressivity is almost always a bad thing, as it leads to overfitting and therefore bad generalization performance. Depending on the amount of available data and the specifics of the learning model, the regime of overfitting can also be reached relatively quickly. In this case, equivariant models offer a clear advantage as they not only reduce the expressivity but they do so in a way that ensures that generalization improves. For the preparation of ground states with a given symmetry, the picture looks somewhat different. In this case, there exists no phenomenon of overfitting *per se*, as a lower energy is preferred even if the prepared state does not have the same symmetry as the ground state.

We can conclude that the trade-off between expressivity and equivariance should always be kept in mind. To fine-tune this trade-off, it is also advisable to only realize a subset of all existing symmetries of the problem. In our example above, one could, for example, opt to only respect the exchange symmetry of the problem. Another way to fine-tune expressivity of the model is by including a limited number of explicitly symmetry-breaking gates, as was argued in, e.g., Ref. [24]. This can be especially advisable if we do not know the symmetry sector the ground state is in, so that we have to have the ability to change the sector, which is impossible with purely equivariant circuits.

Complementary to the trade-off between the gain in equivariance and the decrease in expressivity, there is an accompanying change of the loss landscape that has to be taken into account. It is known that the loss landscapes of variational quantum algorithms are typically challenging [45] and can contain exponentially many poor local minima [46,47]. The picture changes when the quantum circuits become sufficiently deep to reach the *overparametrization* regime where the number of free parameters is larger than the dimension of the relevant subspace of the Hilbert space [48,49], in which case all local optima become global. It is suggested in Ref. [47] that enforcing symmetries could make reaching this regime easier and could contribute to more favorable optimization landscapes. It is, however, too early to tell if there are no additional roadblocks that could arise when enforcing symmetries, as in quantum chemistry symmetry breaking was often found to be helpful for optimization [24]. We consider the analysis of the impact of symmetrization on the loss landscape of learning models to be an important topic of further research.

During the symmetrization process itself, there are further pitfalls to avoid: first, we have already seen that the symmetrization procedure can trivialize certain generators. One example is the representation of $\mathbb{Z}_2$ given by $U_0 = \mathbb{I}$ and $U_1 = X$. We can express every single-qubit unitary as a decomposition $V = R_Z(\theta_1)R_Y(\theta_2)R_Z(\theta_3)$ that corresponds to using the gateset $\mathcal{G} = \{Y, Z\}$. However, neither of these generators is equivariant with respect to this representation, i.e., $\mathcal{T}_U[Y] = \mathcal{T}_U[Z] = 0$, leaving us with *no gates*, even though we started from a universal parametrization. This is in contrast to the fact that we have a set of unitaries compatible with the symmetries that is generated as $R_X(\theta)$. If we had chosen, for example, $V = R_X(\theta_1)R_Y(\theta_2)R_Z(\theta_3)$, the trivialization would not have occurred.

Second, depending on the gates that are used to construct the ansatz, it can lose universality. This can happen if the gates involved do not generate the full set of unitaries compatible with the symmetry. In the context of quantum chemistry and fermi-to-qubit mappings, 3-local unitaries are necessary to generate all possible symmetry-preserving transformations [15,50]. If one would naively symmetrize an ansatz built from the customary single- and two-qubit gates, one would lose universality. In Ref. [51] it was even shown that no set of symmetric local gates can achieve universality under global continuous symmetries like U(1) or SU(2). However, if we only consider states within a specific symmetry sector, as is customary in ground-state search problems, this no-go theorem does not hold anymore and universality may still be achieved [52]. The crucial point is that in a fixed symmetry sector, one can generate every unitary up to irrelevant phases—but these phases become relevant when considering more than one symmetry sector. In the case of reuploading models, it is important to note that the data embedding unitaries $U(\mathbf{x})$ are neither required nor expected to commute with the symmetry operations. In learning models, the symmetry of the underlying state is therefore already broken by the embedding of the classical data and such no-go theorems do not apply.

Third, the circuit depth could increase dramatically under symmetrization. This would happen when the symmetry induces interactions that are nonlocal with respect to the underlying hardware and that have to be realized by extensive SWAP chains.

We also want to note that making the ansatz equivariant is not the only way to make use of symmetries. One way to do so is to use *penalty terms*, which are additional parts of the cost function that increase the cost of states that have low symmetry [26,27,53]. This is comparable to the use of regularization terms in classical machine learning. Penalty terms can be constructed for arbitrary symmetry representations and have the charm that they act through the *optimizer* instead of the ansatz, which means that they can be generically included to partially alleviate problems with uninformed ansätze.

IV. INVARIANT REUPLOADING MODELS

In this section, we want to summarize the construction of invariant reuploading models. We repeat the definition of a generic data reuploading model in our sense. We prepare an ansatz state vector consisting of repeated applications of trainable blocks $\{W_i(\boldsymbol{\theta})\}$ and a data-encoding unitary $U(\mathbf{x})$ as

$$|\psi(\boldsymbol{\theta},\mathbf{x})\rangle = W_L(\boldsymbol{\theta})U(\mathbf{x})\cdots W_1(\boldsymbol{\theta})U(\mathbf{x})W_0(\boldsymbol{\theta})\,|\psi_0\rangle\,. \quad (51)$$

A prediction is then obtained from the expectation value

$$y(\mathbf{x}) = \langle\psi(\boldsymbol{\theta},\mathbf{x})|O|\psi(\boldsymbol{\theta},\mathbf{x})\rangle \quad (52)$$

of an observable. We wish to construct the quantum learning model in a way that the prediction is invariant under the action of a symmetry group $\mathcal{S}$ acting through a representation V_s for $s\in\mathcal{S}$ in the sense that

$$y(V_s[\mathbf{x}]) = y(\mathbf{x}) \quad \text{for all } \mathbf{x}, s\in\mathcal{S}. \quad (53)$$

As we have argued in Sec. II, we can do so for quantum learning models if the data embedding $U(\mathbf{x})$ is *equivariant* with respect to the symmetry in the sense that it *induces* a unitary representation of the symmetry group on the Hilbert space according to

$$U(V_s[\mathbf{x}]) = U_sU(\mathbf{x})U_s^\dagger \quad \text{for all } s\in\mathcal{S}. \quad (54)$$

If the data embedding is not equivariant, we have little hope to build invariant quantum learning models as the symmetry is not meaningfully represented on the level of the Hilbert space. Under an equivariant embedding, the parts of the reuploading model transform as

$$U(V_s[\mathbf{x}])W_i(\boldsymbol{\theta})U(V_s[\mathbf{x}]) = U_sU(\mathbf{x})U_s^\dagger W_i(\boldsymbol{\theta})U_sU(\mathbf{x})U_s^\dagger. \quad (55)$$

We can make this construction equivariant by enforcing that the trainable block $W_i(\boldsymbol{\theta})$ is *equivariant* with respect to the unitary representation U_s of $\mathcal{S}$, mathematically formulated as

$$[W_i(\boldsymbol{\theta}), U_s] = 0 \quad \text{for all } \boldsymbol{\theta},\ s\in\mathcal{S}. \quad (56)$$

As argued in Sec. III, the representation U_s gives us a direct recipe to enforce equivariance of the trainable blocks. This is because the *twirling formula*

$$\mathcal{T}_U[X] = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}} U_sXU_s^\dagger \quad (57)$$

is a projector onto all operators that commute with the symmetry representation (a projection onto the *commutant*) and are thus *equivariant*. We use the fact that ansätze are constructed from parametrized gates with generators from a gateset $\mathcal{G}$ to define an *equivariant gateset* $\mathcal{T}_U[\mathcal{G}]$, which consists of the twirled generators of the standard gateset. The equivariant gateset now contains building blocks that can be freely combined to construct equivariant trainable blocks. Combining data embeddings equivariant with respect to the data symmetry with trainable blocks that are equivariant with respect to the induced symmetry on the Hilbert space gives rise to an *equivariant circuit*

$$U(\boldsymbol{\theta}, V_s[\mathbf{x}]) = U_sU(\boldsymbol{\theta},\mathbf{x})U_s^\dagger \quad \text{for all } \boldsymbol{\theta},\mathbf{x}, s\in\mathcal{S}. \quad (58)$$

We have so far shown how to construct equivariant circuits. To reach invariance of the final prediction, we also need an *invariant* initial state vector

$$U_s\,|\psi_0\rangle = |\psi_0\rangle \quad \text{for all } s\in\mathcal{S}. \quad (59)$$

As $U_s = U_{s^{-1}}^\dagger$, this equivalently implies that

$$U_s^\dagger\,|\psi_0\rangle = |\psi_0\rangle \quad \text{for all } s\in\mathcal{S}. \quad (60)$$

If we now apply an *equivariant circuit* to an *invariant initial state*, we obtain an *equivariant ansatz*

$$\begin{aligned}|\psi(\boldsymbol{\theta}, V_s[\mathbf{x}])\rangle &= U(\boldsymbol{\theta}, V_s[\mathbf{x}])\,|\psi_0\rangle\\ &= U_sU(\boldsymbol{\theta},\mathbf{x})U_s^\dagger\,|\psi_0\rangle\\ &= U_sU(\boldsymbol{\theta},\mathbf{x})\,|\psi_0\rangle\\ &= U_s\,|\psi(\boldsymbol{\theta},\mathbf{x})\rangle \quad \text{for all } \boldsymbol{\theta},\mathbf{x}, s\in\mathcal{S}. \end{aligned} \quad (61)$$

Finally, if we combine an equivariant ansatz with an *invariant observable*

$$U_sOU_s^\dagger = O \quad \text{for all } s\in\mathcal{S}, \quad (62)$$

we obtain an *invariant reuploading model*

$$\begin{aligned}y(V_s[\mathbf{x}]) &= \langle\psi(\boldsymbol{\theta}, V_s[\mathbf{x}])|O|\psi(\boldsymbol{\theta}, V_s[\mathbf{x}])\rangle\\ &= \langle\psi(\boldsymbol{\theta},\mathbf{x})|U_s^\dagger OU_s|\psi(\boldsymbol{\theta},\mathbf{x})\rangle\\ &= \langle\psi(\boldsymbol{\theta},\mathbf{x})|O|\psi(\boldsymbol{\theta},\mathbf{x})\rangle\\ &= y(\mathbf{x}) \quad \text{for all } \boldsymbol{\theta},\mathbf{x}, s\in\mathcal{S}. \end{aligned} \quad (63)$$

Note that the conditions for invariance of the observable in Eq. (62) and the condition for the equivariance of the trainable blocks in Eq. (56) are actually the same if spelled out. The reason for this apparent contradiction is that both objects are operators, but they do different things: the trainable block performs a transformation of states, but the initial state and the final observable act like objects that are transformed. The picture becomes more clear if we look at the operations in terms of their action on density matrices, i.e., if we understand them as quantum channels. Then, equivariance of a trainable block manifests as

$$W_i(\boldsymbol{\theta}) U_s \rho U_s^\dagger W_i(\boldsymbol{\theta})^\dagger = U_s W_i(\boldsymbol{\theta}) \rho W_i(\boldsymbol{\theta})^\dagger U_s^\dagger, \qquad (64)$$

whereas invariance of an observable is still realized as in Eq. (62). Both pictures come together in the fact that conjugation with an *invariant* object (in our case the unitary representing the trainable block) constitutes an *equivariant* transformation.

The construction outlined above can be viewed as the "adjoint" or "two-sided" version of the geometric deep learning blueprint where we have to start from an invariant object, the initial state, apply equivariant operations, and then evaluate the expectation value again on an invariant object.

A. Quantifying learning advantages

We have just outlined a blueprint for the construction of invariant quantum reuploading models. It is logical to ask how much the ability of the model to generalize to unseen data actually increases when we replace a regular reuploading model with its invariant counterpart. In this section, we outline how a simple measure of *symmetrization gain* captures the improvement achieved by our procedure.

Expressivity and generalization need to be balanced to obtain a successful learning model. High expressivity allows the model to capture a multitude of correlations in the training data, but makes it prone to overfit on statistical fluctuations. Reducing expressivity reduces this danger, but can reduce the overall performance of the model. Our symmetrization procedure ensures that we reduce the expressivity in a very informed way, as we only remove degrees of freedom of the model that we know are not present in the data.

The generalization capability can be captured by so-called generalization bounds. These bounds are analytical tools that give guarantees on the performance of a model on unseen data given a certain number of data points in the training set. They can thus serve to obtain upper bounds on the number of training samples needed for a specific performance on unseen data.

Consequently, there is a growing interest in rigorously analyzing the generalization benefits of symmetrizing models by computing the improvement of generalization bounds associated with the models. As a first work in this direction, Elesedy and Zaidi [54] quantified the improvement for linear models and found that it is proportional to the number of removed dimensions in the model's output function space when enforcing invariance. As reuploading models are nonlinear, these results cannot be translated directly to our procedure. It is an interesting future direction of research to determine if a similar argument could be made about linear models in Hilbert space and if the mapping of reuploading models to linear models proposed in Ref. [37] can be applied in this context.

While we cannot apply the results of Ref. [54] straightforwardly, we can still build on prior work on generalization bounds for variational quantum learning models that appeared in Ref. [55]. It essentially tells us that the number of parameters of a quantum learning model is a good proxy for generalization performance, even if the parameters are reused in multiple gates. Caro *et al.* [55] proved the following theorem.

Theorem 1 (Theorem 2 of Ref. [55], informal): *For a quantum learning model with T independent and possibly polynomially often reused gate parameters, with high probability over training data of size N, the generalization gap of the empirical risk minimizer is of order*

$$\tilde{O}\left(\sqrt{\frac{T}{N}}\right), \qquad (65)$$

where the tilde hides logarithmic factors.

As our symmetrization procedure can be applied to any sort of reuploading model, how much this changes the model is highly dependent on the parametrization of the trainable layers and how equivariant these already are. Using the above result, we see that the ratio of parameters of a model before and after symmetrization captures how much less training data is required to achieve the same guarantee on the generalization gap. We can rigorously phrase this as a corollary of the above theorem.

Corollary 1 (Symmetrization gain): *Consider a reuploading model $y(\mathbf{x})$ with T independent parameters. We obtain an invariant reuploading model $y'(\mathbf{x})$ by applying the symmetrization procedure outlined in this work that now has T' independent parameters. Defining the symmetrization gain*

$$\gamma = \frac{T}{T'}, \qquad (66)$$

the invariant model achieves the same guarantee on the generalization gap as the original model given only a ratio $\tilde{O}(1/\gamma)$ of the input samples.

We note that the symmetrization gain can be proportional to the size of the quantum system, e.g., when a layer of independent single-qubit gates is symmetrized with a translation symmetry and is consequently replaced by a layer of single-qubit gates with a shared parameter. In this case, the improvement is of order $\gamma = O(n)$, where n is the number of qubits.

It is important to note that any result that quantifies advantages through changes in generalization bounds, be it our statement built on the work of Caro *et al.* [55] or the results of Elesedy and Zaidi [54], suffers from the drawback that generalization bounds usually only capture worst cases and that specific practical instances need not have similar scaling in sample complexity. A reduction in a specific generalization bound does not therefore necessarily translate to an actual improvement in generalization performance if the underlying problem is "easier" than the worst case. This is a necessary feature of generalization bounds as they make general statements about model classes. However, we do think that our notion of symmetrization gain is meaningful, especially since the guarantees of Ref. [55] imply polynomial sample complexities that are widely considered as efficient.

V. NUMERICAL EXPERIMENTS

In this section, we report on the results of numerical experiments we undertake to compare the performance of invariant learning models with their noninvariant counterparts on two selected toy problems with nontrivial symmetry. We observe that invariant learning models under-perform on the training data, but have a much better generalization performance than learning models that do not respect the symmetry considerations. To ensure that the symmetrization procedure proposed in Sec. III helps generically and that we did not cherry-pick a working edge case, we also perform simulations where we randomize over the layouts of the trainable parts of our learning models that confirm our main numerics and show a generic advantage in generalization.

We further investigate the performance of equivariant ansätze for ground-state problems at the hands of the transverse-field Ising model, the Heisenberg model, and a longitudinal transverse-field Ising model that shares the same symmetries as our learning toy problems. For these problems, and in the ansatz constructions we have studied, equivariant ansätze produce better ground-state approximations on average while needing fewer iterations to converge. We further show that equivariant ansätze can mitigate the barren plateaus problem. We note, however, that the applicability picture is not as clear as in the quantum machine learning application and that equivariant ansätze can also have downsides that we discuss in detail.

A. Tic tac toe

To showcase the methods outlined above we start by considering a simple training task based on the well-known *tic-tac-toe game*. The goal will be to train a variational quantum learning model to classify games into the categories "cross won" (×), "circle won" (○), or "draw" (−). This classification problem has a nontrivial symmetry group, as rotating the board and reflecting the board about an axis does not change the outcome. It also has the advantage that it can be realized with equivariant embeddings on a modest number of nine qubits. The symmetries of the tic-tac-toe game are depicted on the left of Fig. 3.

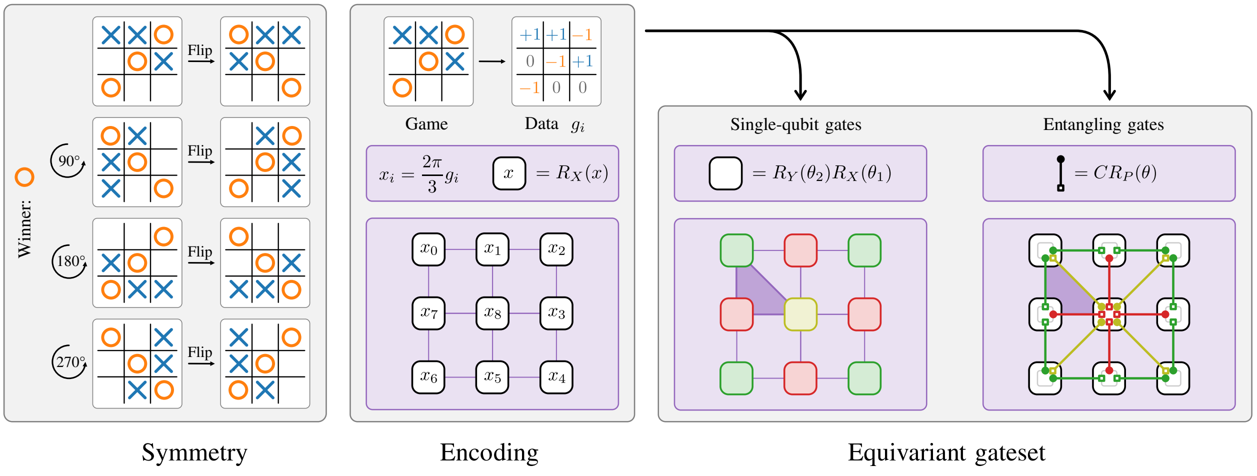


FIG. 3. The winner of a game of tic tac toe is independent of rotations and flips of the board, which means that its symmetry group is given by the dihedral group D_4, as visualized on the left-hand side of the figure. We use a dataset of tic-tac-toe games encoded via Pauli-X rotations on nine qubits, as visualized in the second column of the figure. The three different values of a field are equidistantly encoded by using a multiple of $2\pi/3$ for the rotation angle. Using our symmetrization procedure, we obtain an equivariant gateset made up of single-qubit gates acting identically on the corners (green), edges (red), and the middle (yellow) of the board. The two-qubit gates are given by controlled rotations between corners and edges (green), edges and the middle (red), and the middle and the corners (yellow), where the control qubit is always listed first. Because of the symmetry, it is sufficient to specify the gates on one neighboring trio of corner edge and middle, as visualized by the dark purple triangle.

The symmetry group of this learning task is given by the *dihedral group* D_4, which is equivalent to all operations that map a square to itself. The dihedral group can be generated by a counterclockwise rotation of 90° and a flip about the vertical axis through the center. The group has order 8, so has a total of eight elements. The group induces equivalence classes of fields of the tic-tac-toe board, as corners of the board will always be mapped to corners, edges to edges, and the center will always stay the same. These three equivalence classes will also be mirrored in the equivariant gateset.

We note that the task of labeling games of tic tac toe is easily solved by a classical deterministic algorithm, and in fact, only a finite number of games are even possible. The goal of this numerical example is to showcase a possible end-to-end implementation of our symmetrization procedure and to compare equivariant with standard gatesets. Furthermore, in some of the following sections, we discuss how this example can be taken as a paradigm to tackle other, potentially more useful problems.

1. Dataset

The first process is mapping a game of tic tac toe to classical data. As shown in the second column of Fig. 3, we do so by mapping the nine fields of the board to elements of a vector $\mathbf{g}$ where $+1$ represents a cross, -1 a circle, and 0 an empty field. The labels of the game are encoded in a one-hot fashion in a vector $\mathbf{y} = (y_\circ, y_-, y_\times)$, where a value of $+1$ is assigned to the correct label and -1 to the other two entries. The number of distinct tic-tac-toe games is sufficiently small so that we are able to generate all possible valid games. In our dataset, we also allow for unfinished games, which are labeled as “draw.” The training and test data sets are then constructed by choosing a subset of all possible games of tic tac toe at random, with the constraint that each outcome is equally represented.

2. Learning model

To address the tic-tac-toe learning task, we make use of a data reuploading model with an equivariant embedding as described in Secs. I and II. The equivariant embedding is constructed by encoding the different numerical values that represent a game via a Pauli-X rotation on separate qubits that we view in a planar grid. To distribute the three data features equidistantly, we use a multiple of $2\pi/3$ for the rotation angle, again as shown in the second column of Fig. 3.

The equivariance of the embedding ensures that symmetry transformations are realized by unitary conjugation. For example, a reflection along the vertical axis is implemented by $\mathrm{SWAP}_{0,2}\mathrm{SWAP}_{7,3}\mathrm{SWAP}_{6,4}$ in the numbering of Fig. 3. The advantage of permutation-type symmetries on the level of qubits is that they can easily be understood on a visual level. The qubits lie in equivalence classes under the symmetry operation. Single-qubit gates then have to act on all qubits of the same equivalence class equally. This is how we get the type of equivariant single-qubit layer that is depicted in the third column of Fig. 3, where single-qubit unitaries are applied that share the same parameters when acting on corners, on edges, or on the middle. The same reasoning can also be applied to two-qubit gates. An entangling operation that connects, for example, a corner with the edge next to it has to act in the same way on all pairs of neighboring corners and edges. This is how we obtain the equivariant layers of entangling gates used for our learning models, which perform controlled rotations from corners to neighboring edges, edges to the middle, and from the middle to the corners, as depicted in the fourth column of Fig. 3. We chose CR_Y for parametrized rotations.

The learning model starts with all qubits initialized in the $|0\rangle$ state vector, which is invariant under the problem’s symmetry. Then a number of layers are applied, each made up of one data encoding followed by a sequence of parametrized layers. The default parametrized layer is chosen to be “cemoid,” which corresponds to one application of the single-qubit gates followed by the entangling gates, both visible in Fig. 3. The prediction of a label is obtained in a one-hot encoding by measuring the expectation values of three invariant observables

$$O_\circ = \frac{1}{4}\sum_{i\in\text{corners}} Z_i = \frac{1}{4}[Z_0 + Z_2 + Z_4 + Z_6], \tag{67}$$

$$O_- = Z_{\text{middle}} = Z_8, \tag{68}$$

$$O_\times = \frac{1}{4}\sum_{i\in\text{edges}} Z_i = \frac{1}{4}[Z_1 + Z_3 + Z_5 + Z_7] \tag{69}$$

as $\hat{\mathbf{y}} = (\langle O_\circ\rangle, \langle O_-\rangle, \langle O_\times\rangle)$. A prediction for a given data point is obtained by selecting the class for which the observed expectation value is the largest.

3. Training

We train the learning model using an l_2-loss function, which, for a set of games with associated one-hot label vectors $\mathcal{D} = \{(\mathbf{g}, \mathbf{y})\}$, is given by

$$\mathcal{L}(\mathcal{D}) = \frac{1}{|\mathcal{D}|}\sum_{(\mathbf{g},\mathbf{y})\in\mathcal{D}} \|\hat{\mathbf{y}}(\mathbf{g}) - \mathbf{y}\|_2^2. \tag{70}$$

We run optimizations with 100 epochs, each consisting of 30 steps. At each step, the gradient is computed using 15 data points representing a tic-tac-toe game. The size of the training set is then the product of these two numbers, 15 × 30 = 450. The training set is shuffled after each epoch is completed. The test set comprises 600 randomly chosen games with the same constraint as above and kept fixed throughout the run. The above hyperparameters are chosen empirically.

The quantum learning model is implemented with the PennyLane library [56] for quantum machine learning. Using the PyTorch interface provided by PennyLane, we train the PQC by stochastic gradient descent as implemented by the Adam PyTorch optimizer [57].

4. Results

In all our numerics, we compare invariant with noninvariant models where the parameter sharing indicated in Fig. 3 is not imposed. The noninvariant models therefore have more independent parameters and hence a higher expressivity. To evaluate the performance of the models, we record the classification accuracy for the training and the test sets. The general trend we observe is that invariant models built from equivariant circuits achieve the same or lower accuracy on the training set, but consistently higher accuracy on the test set, which indicates their better generalization capabilities.

We first verify that this is the case for circuits of different sizes. Figure 4 compares the results of invariant and noninvariant models composed of l layers, each of which consists of one data encoding followed by p independent repetitions of the layout "cemoid." We refer to the different architectures with the pair (l, p), and we sweep over the range $(l, p) \in \{1, \ldots, 5\}^2$. Because of limited computational resources, we are unable to record the results for the values $(l, p) \in \{(4, 4), (4, 5), (5, 3), (5, 4), (5, 5)\}$. In this experiment, the parametrized entangling gates are chosen to be $CR_Y(\theta)$, but similar experiments with different controlled Pauli rotations produce comparable results.

The difference in accuracy on the training set should not come across as odd, since invariant models only express a subset of the output mappings of the noninvariant ones. For noninvariant models, high training accuracy and low test accuracy are clear witnesses of the overfitting regime. On the contrary, for invariant models, we see that the price we pay by lowering the expressivity returns a very similar performance on both previously seen and unseen data, hinting at the sweet spot in the bias-variance trade-off. Formally, we say that the empirical generalization gap of invariant models is much smaller than that of noninvariant ones, which confirms our expectations.

As a sanity check, and to make sure that we did not cherry-pick a working example, we study the performance of other variational layouts different from "cemoid." We do so by fixing the number of layers to $l = 3$, where after encoding the data with "t" we take three consecutive random permutations of "cemoid" without contiguous repetitions. An example of such a layout is "tdomiececiodmdmoiec," which we then repeat three times. In total, 20 different layouts are generated and simulated. The summary of results in Fig. 5 reveals that the test performance of the invariant models is consistently higher across different layer layouts, with few exceptions. This confirms

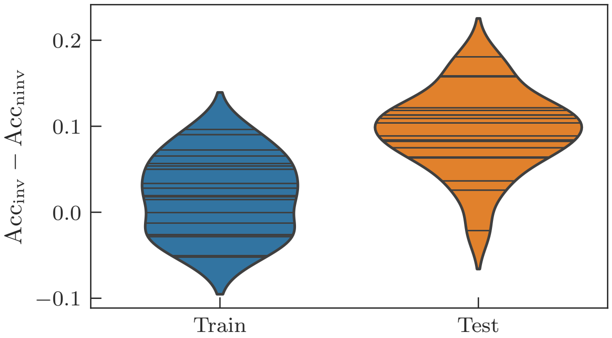


FIG. 4. Violin plot of the difference in achieved mean accuracies between the invariant and noninvariant models in a sweep over different values of l and p as detailed in the text. Positive values correspond to invariant models outperforming noninvariant ones. The mean is averaged over ten runs for each combination of l and p. We see that the mean accuracy on the training data is more or less the same for both invariant and noninvariant models, but that the invariant models clearly outperform the noninvariant ones on the test data. The magnitude of the difference compares well to the absolute test accuracies of both models that, depending on l and p, lie between 0.59 and 0.80 (invariant) and 0.48 and 0.62 (noninvariant), significantly larger than the random guessing probability 0.33. The full training graphs are given in Fig. 14 in the Appendix.

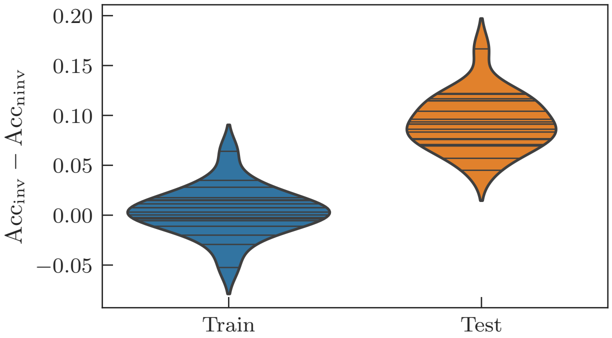


FIG. 5. Violin plot of the difference in achieved accuracies between the invariant and noninvariant models for 20 random parametrizations of the trainable blocks as detailed in the text. Positive values correspond to invariant models outperforming noninvariant ones. The values for every parametrization are averaged over ten repetitions with random parameter initializations. We see that the mean accuracy on the training data is more or less the same for both invariant and noninvariant models, but that the invariant models again clearly outperform the noninvariant ones on the test data. The magnitude of the difference compares well to the absolute test accuracies of both models that lie between 0.68 and 0.80 (invariant) and 0.60 and 0.66 (noninvariant), significantly larger than the random guessing probability 0.33.

our expectations that making learning models invariant helps more or less generically to enhance generalization performance.

B. Classifying autonomous vehicle scenarios

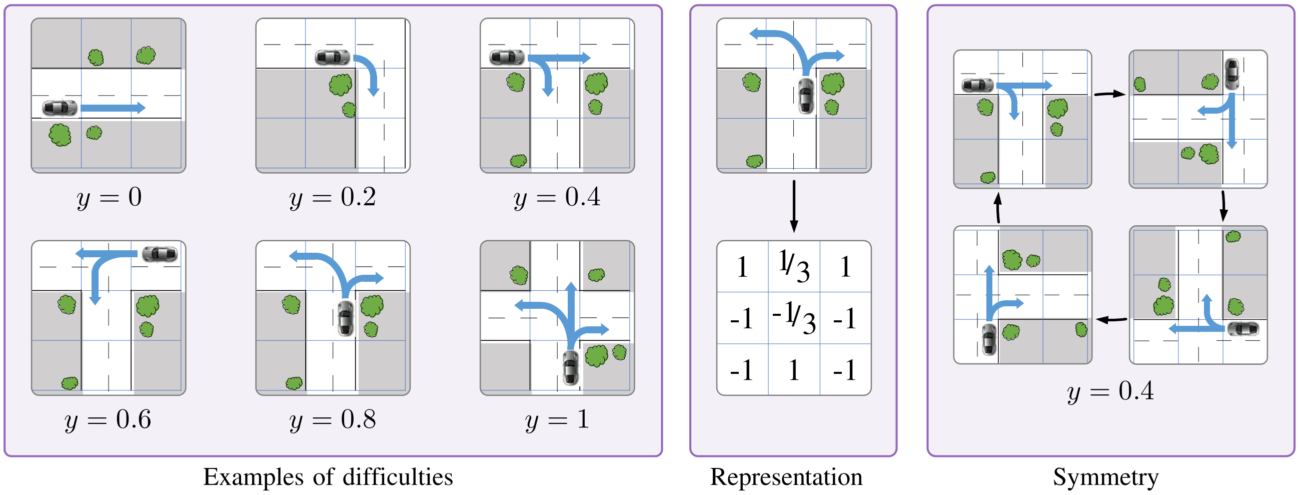
The tic-tac-toe task provides an intuitive example of how we can exploit symmetry in learning, but it does not connect to a relevant real-world learning scenario. In this section, we want to outline how we can use the intuition developed in the tic-tac-toe example and apply it to a toy model of a task that is of actual relevance in the automotive industry and shows a clear way of connecting to a real-world scenario.

Autonomous driving is a future-oriented field of the automotive industry, for which two main challenges interact. First, the vehicle must be able to recognize and automatically evaluate its surroundings to deduce possible driving maneuvers to reach a goal. Second, the automated evaluation performed by the car requires verification and testing along the development cycle of the car. To meet these challenges, scenario-based development is state of the art [58]. We define a scenario as a concatenation of a scene (snapshot of the surrounding) and actions (destination goals and values). Therefore, a scenario is a specific description of a driving situation, taking into account dynamic and static components that are determined by the sensor systems of the vehicle, map data, and others [59]. In the development process of the vehicle different predefined scenarios are classified with respect to their safety relevance according to a criticality metric. This classification will then be used to evaluate the requirements for testing, in the field or in simulation. The Pegasus Project [60] classifies the scenarios into different levels. In this publication, we concentrate on street level 1, which includes geometry and topology of the streets, in order to demonstrate our concept. Driving-maneuver classification is investigated using classical machine learning tools among other approaches [61,62]. We again want to emphasize that, for the purpose of this paper, the original classification task is reduced to a simple version that can be simulated with the computational resources available to us. It is however clear how larger resources in terms of quantum computational hardware would allow us to address parts of the actual question as soon as it becomes available.

1. Dataset

In the following, we derive driving scenarios at different street intersections, deduce the geometric symmetries, and build a classification of the possible actions according to their safety relevance. Since relevant scenarios include behavior at intersections, traffic circles, and traffic jams [63], we consider the following simplified street situations, using a 3×3 grid. Each tile of this grid can be either part of a road or be impassable, and a car is placed on one of the road tiles. With this encoding, we are able to represent different geometries from a straight road to a more complicated intersection. The symmetries of these scenarios are a subset of the tic-tac-toe case as a scenario can always be rotated by 90°(see the right panel of Fig. 6), but not necessarily mirrored in all instances as left turns have a different difficulty than right turns. We translate this into a 3×3 array $\mathbf{x}$, where a road tile is encoded with 1, an impassable tile with -1, the car with $-1/3$, and the orientation of the car with $1/3$. An example of this can be seen in the middle panel of Fig. 6.

We rate the level of difficulty y with regards to a simplified criticality metric as follows: forward: 0; right or left curve: 0.2; forward and right (T crossing, intersection of three streets): 0.4; forward and left (T crossing, intersection of three streets): 0.6; left and right (T crossing,



Examples of difficulties

Representation

Symmetry

FIG. 6. Examples of the various difficulties encountered in the autonomous driving toy model (left), a demonstration of the data representation (middle), and a demonstration of the rotational symmetry of the model (right).

intersection of three streets): 0.8; forward, left, and right (X crossing, intersection of four streets): 1. Examples of the possible difficulties are visualized in the left panel of Fig. 6 alongside a representation of the data encoding and a demonstration of the symmetry.

The different scenarios $\mathbf{x}$ are generated by first hand placing various road layouts (T intersection, left curve, X intersections, …), generating all their images under rotations and reflections and then putting the car on every possible road tile. Additionally, every possible orientation of a placed car is iterated. This process creates the total set of possible scenarios and their associated difficulty $\mathcal{D} = \{(\mathbf{x}, y)\}$.

As for the tic-tac-toe games, training and test data sets are constructed by choosing a subset of scenarios at random, with the constraint that each difficulty level is equally represented.

2. Learning model

The general circuit construction is essentially the same as in the tic-tac-toe case and follows the data reuploading models as described in Secs. I and II. The different numerical values that represent a scene are encoded via a Pauli-X rotation on the separate qubits where the rotation angle is given by a multiple $2\pi/3$ of the array value as in the tic-tac-toe case. The key difference to the tic-tac-toe case is the missing mirror symmetry, rendering the actual symmetry group to $\mathbb{Z}_4$ instead of D_4. This only affects the outer layer and splits it into two distinct sublayers: one where controlled operations are performed clockwise with shared parameters and one where controlled operations are performed counterclockwise with shared parameters.

As in the tic-tac-toe case, all qubits are initialized in the $|0\rangle$ state and layers of data encoding followed by parametrized gates are applied. The default topology for the parametrized blocks is again chosen to be "cemoid," where it is understood that the outer layer splits into two sublayers as explained above. The model's prediction $\hat{y}$ of difficulty is obtained by measuring and normalizing the Pauli-Z expectation value of the middle qubit Z_8:

$$\hat{y} = \frac{\langle Z_{\text{middle}}\rangle + 1}{2} = \frac{\langle Z_8\rangle + 1}{2}. \tag{71}$$

A hard prediction for a given scenario $\mathbf{x}$ is obtained by rounding $\hat{y}$ to the nearest difficulty in $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$.

3. Training

The learning model is trained using an l_2-loss function that is, for a set of games $\mathcal{D} = \{(\mathbf{x}, y)\}$, given by

$$\mathcal{L}(\mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},y)\in\mathcal{D}} (\hat{y}(\mathbf{x}) - y)^2. \tag{72}$$

No epochs are used in this case and optimizations are run with 30 steps. At each step, the gradient is computed for the same 60 scenes. As the limited grid size of 3 × 3 only allows four difficulty 1 (X crossing) scenarios, random copies are created to ensure an equal distribution (ten games of each difficulty). After each step, the accuracy on the training data and the test data, which consisted of 130 unique games, is evaluated by calculating the fraction of correctly classified inputs. The above hyperparameters are again chosen empirically. The numerical experiments are performed using the PennyLane library [56]. For the two-qubit gates, only the CR_Z gate is implemented. All optimizations are executed using the PyTorch L-BFGS optimizer [57].

4. Results

We repeat the experiments we designed for the tic-tac-toe learning task, now with this different dataset. As expected, we observe the same general trend. We repeat that we compare an invariant model with the noninvariant model obtained from the invariant one where the parameter sharing indicated in Fig. 3 is not imposed. The noninvariant models therefore have more independent parameters and hence a higher expressivity. To evaluate the performance of the models, we record the classification accuracy for the training and the test sets.

We again study different hyperparameters for the architecture of the learning model. Recall that we use l layers consisting of a data encoding followed by p repetitions

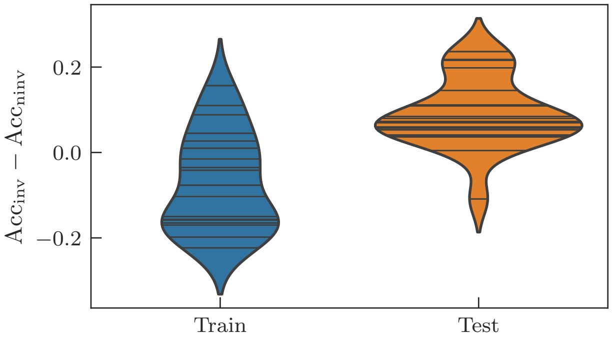


FIG. 7. Violin plot of the difference in achieved mean accuracies between the invariant and noninvariant models in a sweep over different values of l and p as detailed in the text. Positive values correspond to invariant models outperforming noninvariant ones. The mean is averaged over ten runs for each combination of l and p. We see that the mean accuracy on the training data is more or less the same for both invariant and noninvariant models, but that the invariant models clearly outperform the noninvariant ones on the test data. The magnitude of the difference compares well to the absolute test accuracies of both models that, depending on l and p, lie between 0.17 and 0.97 (invariant) and 0.17 and 0.64 (noninvariant), significantly larger than the random guessing probability 0.17. The full training graphs are given in Fig. 15 in the Appendix.

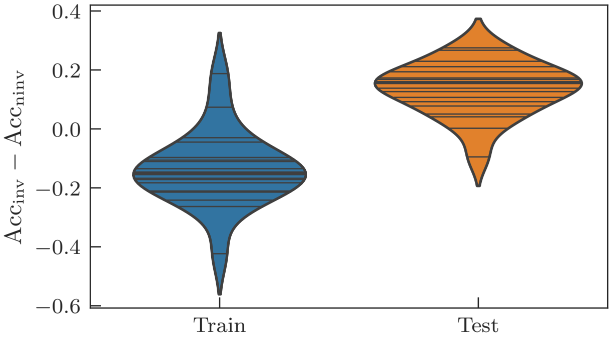


FIG. 8. Violin plot of the difference in achieved accuracies between the invariant and noninvariant models for 20 random parametrizations of the trainable blocks as detailed in the text. Positive values correspond to invariant models outperforming noninvariant ones. For every parametrization, ten random initializations are performed and then averaged. We see that the mean accuracy on the training data is more or less the same for both invariant and noninvariant models, but that the invariant models clearly outperform the noninvariant ones on the test data. The magnitude of the difference compares well to the absolute test accuracies of both models that lie between 0.48 and 0.80 (invariant) and 0.39 and 0.58 (noninvariant), significantly larger than the random guessing probability 0.17.

of the atom "cemoid." We sweep over the range $(l,p) \in \{1,\ldots,5\}^2$ except for the values $(4,4), (4,5), (5,3), (5,4)$, and $(5,5)$. As can be seen in Fig. 7, most invariant learning models perform worse than noninvariant models on the training set and better on the test set. A straightforward explanation would be the position the different models take in the bias-variance trade-off: the overfitting regime for noninvariant models and the sweet spot for invariant models. The full data presented in Fig. 15 in the Appendix further support this intuition as the data show a drop in accuracy for high (l,p) values, where the expressivity of both models is found empirically to be too high for this task. This experiment crystallizes in the statement: invariant models generalize better than noninvariant models.

We also repeat the randomization experiment over the specific spelling of the trainable parts, with the results being reported in Fig. 8. We make circuits with a random layer repeated 3 times. Random layers start with a data encoding followed by three consecutive random permutations of "cemoid," such that there are no repeated letters adjacent. Again, in almost every case the invariant models outmatch the noninvariant ones.

C. Variational quantum eigensolvers

The gate symmetrization procedure proposed in this work only necessitates the existence of a unitary symmetry representation on the level of the Hilbert space. It can therefore also be applied to ground-state problems where conserved quantities of the Hamiltonian yield representations of symmetries. This case is paradigmatically treated through a variational algorithm using the variational quantum eigensolver. In the following, we conduct some numerical experiments that showcase the advantages and disadvantages of generic symmetrization in this context.

1. Transverse-field Ising model

We first consider the *transverse-field Ising model* (TFIM) [64] as a paradigmatic example. The TFIM Hamiltonian with periodic boundary conditions on N spins is given by

$$H_{\mathrm{TFIM}} = -\sum_{i=1}^{N} Z_i Z_{i+1} - g\sum_{i=1}^{N} X_i, \tag{73}$$

where we consider a transverse field strength $g > 0$.

The TFIM Hamiltonian has a $\mathbb{Z}_2$ symmetry as it commutes with the parity operator

$$P = \prod_{i=1}^{N} X_i. \tag{74}$$

The unitary representation is then given by $U_s = P^s$ for $s \in \mathbb{Z}_2$. The eigenvalues of the parity operator are either $+1$ or -1. For $g \to \infty$, the ground-state vector is given by $|+\rangle^{\otimes N}$, which has a parity of $+1$. Using the *adiabatic theorem* and the fact that, for finite system size, the ground-state energy is not degenerate, we can conclude that the parity of the ground state is the same for each $g > 0$. Therefore, if we want to force our ansatz state to encode this symmetry, we require that

$$P\,|\psi(\boldsymbol{\theta})\rangle = +\,|\psi(\boldsymbol{\theta})\rangle\,. \tag{75}$$

Of course, it is not necessary for our ansatz to respect this property for all values of the parameters as long as it finds the correct ground state, but in many cases it can be beneficial to restrict the expressivity of the ansatz into a relevant part of the Hilbert space. If we do this, however, we have to assure that the ansatz does produce a state that is in the same symmetry sector as the true ground state. This can be assured by taking an initial state that has the correct symmetry, for example $|\psi_0\rangle = |+\rangle^{\otimes N}$, and then only performing equivariant gates that can be obtained from the symmetrization procedure of Sec. III. We note again that one has to be attentive to the fact that symmetrization has to be executed with care as it can trivialize certain generators.

In our numerical experiments, we use the QAOA ansatz [65] that can be easily seen to be equivariant with respect to the parity symmetry,

$$|\psi_{\text{QAOA}}(\boldsymbol{\beta},\boldsymbol{\gamma})\rangle = \prod_{m=1}^{p}\prod_{i=1}^{N} e^{-i\beta_m X_i}\prod_{i=1}^{N} e^{-i\gamma_m Z_i Z_{i+1}}|\psi_0\rangle . \tag{76}$$

Here, p is the number of QAOA layers. We compare it to a variant of the QAOA ansatz that we denote as QAOA$'$, where adding an additional mixer term involving Pauli-Y rotations increases the expressivity as

$$|\psi_{\text{QAOA}'}(\boldsymbol{\alpha},\boldsymbol{\beta},\boldsymbol{\gamma})\rangle = \prod_{m=1}^{p}\prod_{i=1}^{N} e^{-i\alpha_m Y_i}\prod_{i=1}^{N} e^{-i\beta_m X_i}\prod_{i=1}^{N} e^{-i\gamma_m Z_i Z_{i+1}}|\psi_0\rangle . \tag{77}$$

As $PY_iP = -Y_i$, the gate symmetrization procedure of Sec. III will remove these Pauli-Y rotations and will yield the standard QAOA ansatz, which has been widely studied and shown [66–69] to provide a faithful ground-state approximation for the TFIM when $p \geq N/2$, while for $p < N/2$, the QAOA ansatz can only reach a variational energy that is above the ground-state energy [67] due to light-cone arguments [66]. This behavior can also be observed in our experiments.

We compare the performance of the two ansätze using the TFIM with $N = 10$ spins and different values for the number of QAOA layers p. We optimize using the L-BFGS optimizer until convergence is reached. We perform the optimization for 20 random initializations of the circuit parameters and average the results to arrive at the statistics shown in Fig. 9. For all values of p, the QAOA ansatz needs fewer iterations to converge. For small $p < N/2$, we observe that the QAOA ansatz does not converge to a good approximation of the ground state, which is in line with the aforementioned previous findings. In this regime, it is outperformed by the nonequivariant ansatz, highlighting the trade-off between expressivity and equivariance. If the circuit depth $p \geq N/2$ is large enough, the picture reverses and the equivariant QAOA ansatz reliably reaches the ground state whereas the nonequivariant QAOA$'$ ansatz converges to an energy above the ground state on average.

2. Heisenberg model

Another model that has a continuous symmetry group is given by the Heisenberg model [64] with periodic boundary conditions on an even number N of spins captured by the Hamiltonian

$$H_{\text{Heis}} = \sum_{i=1}^{N} X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1}. \tag{78}$$

We can understand this Hamiltonian as the alignment of two neighboring spins. Quite logically, if we rotate all spins simultaneously, the *relative* alignment will not change. The symmetry group of the model is thus SU(2), represented by

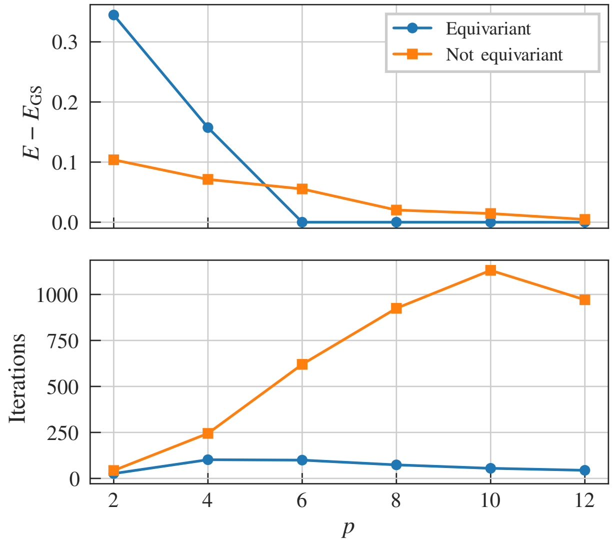


FIG. 9. Comparison of the equivariant QAOA ansatz with the nonequivariant QAOA$'$ ansatz for the TFIM on $N = 10$ spins with field $g = 1$. For every number of layers p, we perform 20 experiments with random initializations of the parameters and plot the difference between the mean achieved energy and the ground state energy (top) and the number of iterations necessary to reach it (bottom). We see that, for $p \geq N/2$, the equivariant ansatz performs better in both figures of merit.

$$U_V = V^{\otimes N} \quad \text{for } V \in \text{SU}(2). \tag{79}$$

As in the previous example, we initialize the quantum computer in a state that lies in the right symmetry sector, which in the case of the Heisenberg model is given by all states that have zero total spin, and we choose for the state vectors

$$|\psi_0\rangle = \bigotimes_{k=1}^{N/2} \frac{1}{\sqrt{2}}(|0,1\rangle - |1,0\rangle). \tag{80}$$

For the Heisenberg model, we start from the nonequivariant ansatz

$$|\psi(\boldsymbol{\theta})\rangle = \prod_{m=1}^{p}\prod_{i=1}^{N} e^{-i\alpha^{(m)}Y_i} e^{-iH_{\text{even}}(\boldsymbol{\beta}^{(m)})} e^{-iH_{\text{odd}}(\boldsymbol{\gamma}^{(m)})}|\psi_0\rangle , \tag{81}$$

where $H_{\text{even}}(\boldsymbol{\beta}_m)$ is an anisotropic Heisenberg Hamiltonian defined on the even lattice sites, i.e.,

$$H_{\text{even}}(\boldsymbol{\beta}) = \sum_{\substack{i=2\\ i \text{ even}}}^{N} \beta_x X_{i-1}X_i + \beta_y Y_{i-1}Y_i + \beta_z Z_{i-1}Z_i. \tag{82}$$

The Hamiltonian $H_{\mathrm{odd}}(\boldsymbol{\gamma})$ is analogously defined but acts on the odd sites. For the even and odd Hamiltonians, we have three parameters each; together with the α parameter controlling the Pauli-Y rotation, we have seven parameters per layer, yielding $7p$ parameters in total. We note that, since H_{even} and H_{odd} are both linear combinations of commuting operators, we can decompose the corresponding unitaries using two-qubit gates.

The above ansatz is not equivariant, which means that we need to apply the symmetrization procedure. The symmetrization procedure corresponds to a particular instance of a 2-design twirl

$$\begin{aligned}\mathcal{T}_U[X_1X_2] &= \int d\mu(V)\, V^{\otimes N} X_1X_2 (V^\dagger)^{\otimes N} \\ &= \int d\mu(V)\, V^{\otimes 2} X_1X_2 (V^\dagger)^{\otimes 2}. \qquad (83)\end{aligned}$$

These calculations can be straightforwardly performed using the Weingarten calculus (see, e.g., Ref. [70]). The important fact for us is that any outcome of a 2-design twirl will be a sum of identity and SWAP,

$$\begin{aligned}\mathcal{T}_U[X_1X_2] &= c_0\mathbb{I} + c_1\mathrm{SWAP} \\ &= c_0\mathbb{I} + \frac{c_1}{2}(\mathbb{I} + X_1X_2 + Y_1Y_2 + Z_1Z_2), \qquad (84)\end{aligned}$$

where we have used the expansion of SWAP into Pauli words. Note that identity terms only generate global phases, which means that the effective symmetrized generator associated with X_1X_2 can be taken to be

$$\mathcal{T}_U[X_1X_2] \sim X_1X_2 + Y_1Y_2 + Z_1Z_2. \qquad (85)$$

This generator gives rise to the particle-number-conserving *Givens rotations* [15]. The additional Pauli-Y rotation in turn is trivialized by the symmetrization procedure

$$\mathcal{T}_U[Y_1] = \int d\mu(V)\, VY_1V^\dagger = \frac{\mathrm{Tr}(Y_1)}{2}\mathbb{I} = 0, \qquad (86)$$

where we have exploited the formula associated with the first moment operator [71].

The same argument holds for the other terms, which yields the equivariant ansatz

$$|\psi(\boldsymbol{\theta})\rangle = \prod_{m=1}^{p}\prod_{i=1}^{N} e^{-i\beta^{(m)}H_{\mathrm{even}}}e^{-i\gamma^{(m)}H_{\mathrm{odd}}}\,|\psi_0\rangle\,, \qquad (87)$$

where we have chosen H_{even} and H_{odd} to be the isotropic variants given by $\boldsymbol{\beta} = \boldsymbol{\gamma} = (1,1,1)$. The equivariant ansatz we obtain with the symmetrization procedure has the same form as the *Hamiltonian variational ansatz* used in Refs. [67,69]. Note that the equivariant ansatz has $2p$ parameters in total compared to the $7p$ of the nonsymmetric ansatz.

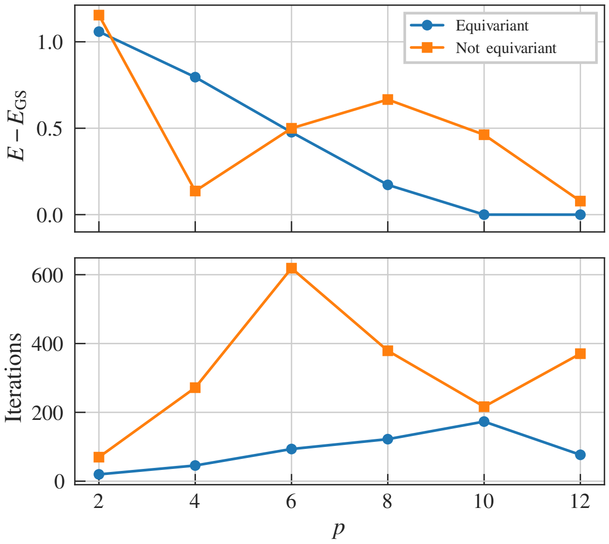


FIG. 10. Comparison of the equivariant and the nonequivariant ansätze for the Heisenberg model on $N = 10$ spins. For every number of layers p, we perform 20 experiments with random initializations of the parameters and plot the difference between the mean achieved energy and the ground state energy (top) the number of iterations necessary to reach it (bottom). We see that, for large enough p, the equivariant ansatz performs better in both figures of merit. For p small, however, the approximation to the ground state is worse.

We compare the performance of the two ansätze using the Heisenberg model with $N = 10$ spins and different values for the number of layers p. As in the TFIM numerics, we optimize using the L-BFGS optimizer until convergence is reached and average the outcomes over 20 random initializations of the circuit parameters to arrive at the data shown in Fig. 10.

We see that the equivariant ansatz reaches its minimum energy faster across all depths. However, only for a large enough depth p can it outperform the nonequivariant ansatz in terms of the energy expectation value. This nicely shows the trade-off between expressivity and symmetrization, as, for small p, the increased expressivity of the nonequivariant ansatz is at an advantage over the equivariance of the symmetrized ansatz. However, as soon as the equivariant ansatz becomes sufficiently expressive, the picture reverses.

3. Barren plateaus

In a further numerical experiment that uses the same setups as above—TFIM and the Heisenberg model—we analyze the influence of our symmetrization procedure on the barren plateaus phenomenon [72,73]. Symmetrization reduces the expressivity of the ansatz by reducing the number of free parameters and additionally alters the

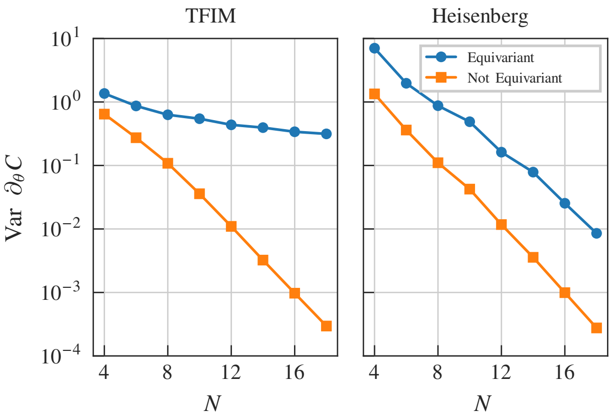


FIG. 11. Comparison of the barren plateaus phenomenon for the TFIM and the Heisenberg model considered in Figs. 9 and 10. The derivative is computed with respect to the rotation angle of the first qubit in the first layer using as the observable Z_1Z_2 with $p = 80$ for the TFIM and $p = 40$ for the Heisenberg model. The variance is calculated using 1000 randomly sampled parameters from a uniform distribution $[0, 2\pi]$.

dynamical Lie algebra associated with the ansatz generators, which is known to be intimately related to barren plateaus [74]. We therefore expect that the equivariant ansatz will have larger gradients in the applications studied here. In Fig. 11, we can indeed observe that the barren plateaus phenomenon is mitigated by the symmetrization procedure. For the transverse-field Ising model, the gradient decay of the equivariant ansatz is consistent with the polynomial scaling predicted by the reduced dimension of the dynamical Lie algebra found in Ref. [74], whereas the nonequivariant ansatz shows an exponential decay. In the case of the Heisenberg model, the equivariant ansatz shows an exponential decay like its nonequivariant counterpart, but the gradient magnitude is enhanced and we see signs of a slightly slower decay exponent.

4. Tic-tac-toe LTFIM

The last model we analyze is a variant of the *longitudinal-transverse-field Ising model* (LTFIM) defined on a two-dimensional lattice of nine sites. This model is constructed such that it has the same geometric symmetry as the tic-tac-toe example that we encountered above. We can write the Hamiltonian as

$$H_{\text{TTT}} = H_{ZZ} + \sum_{i=1}^{9} X_i + \sum_{i=1}^{9} Z_i, \tag{88}$$

where H_{ZZ} determines the interaction of the spins. As shown in Fig. 12, we distinguish three families of edges through their interaction strength: those in the contour of the lattice (triple lines in the figure), those along the diagonals (single lines), and those at the inside (double lines). So we choose H_{ZZ} to be

$$H_{ZZ} = H_{\text{cont}} + \tfrac{1}{2}H_{\text{inside}} + \tfrac{3}{2}H_{\text{diag}}, \tag{89}$$

where every term is a sum of the ZZ interactions for the given set of edges

$$H_A = \sum_{(i_1,i_2)\in A} Z_{i_1}Z_{i_2}. \tag{90}$$

This model has clear geometric symmetries that are the same as those we used in the tic-tac-toe learning task. In Fig. 12, we represent the qubits that are equivalent under symmetry transformation with the same colors. Note that we have three independent families of qubits and also of edges.

For this model, we use an ansatz that is composed of p repeated layers of entangling ZZ rotations for each edge in Fig. 12, and Pauli-X and Pauli-Z rotations on each qubit. The symmetrization procedure will enforce that the gate parameters of the gates are the same for equivalence classes of edges for the entangling gates and for equivalence classes of spins for the single-qubit gates. Thus, in the nonsymmetric case, we have $(16 + 9 + 9)p$ free parameters in total, while in the symmetric one we have a reduction to $(3 + 3 + 3)p$ free parameters. The invariant state vector $|+\rangle^{\otimes N}$ is used as the initial state.

We numerically test the performance of the two ansätze in a manner similar to that of previous models, as shown in Fig. 13. We can conclude that in this model, the equivariant ansatz performs better on average than the nonequivariant one, both in achieved energy and the number of needed iterations for the circuit depths tested. However, we see that there is an uptick in the energy achieved by the equivariant ansatz for larger layers that probably indicates that this advantage is not stable against increasing the depth of

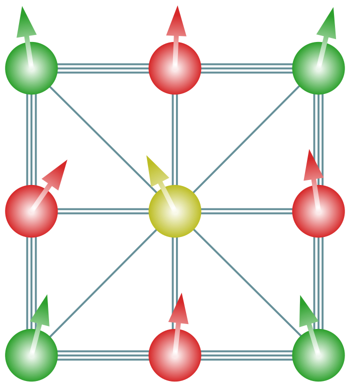


FIG. 12. Graph for the LTFIM lattice model. The different colors used for the vertices emphasize the fact that sites that share the same color also share the same parameter in the one-qubit gates, and similarly for the different thicknesses of edges used for the two-qubit gates.

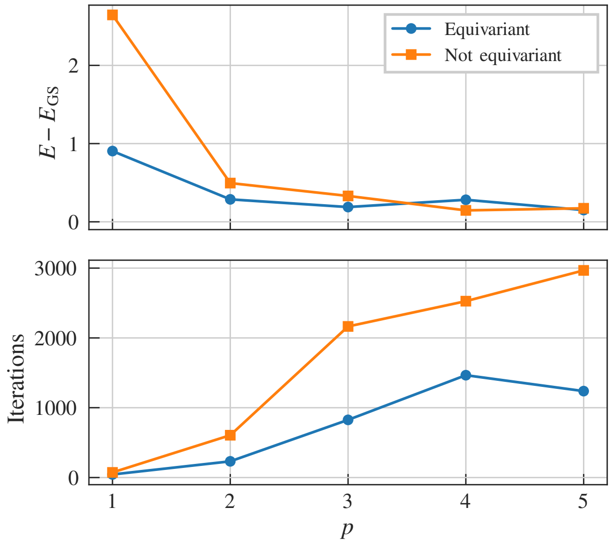


FIG. 13. Comparison of the equivariant and the nonequivariant ansätze for the TTT-LTFIM model. For every number of layers p, we perform ten experiments with random initializations of the parameters and plot the difference between the mean achieved energy and the ground state energy (top) and the number of iterations necessary to reach it (bottom). For small circuit depths, the equivariant ansatz performs better than the nonequivariant one. This advantage does not seem to persist to higher depths.

the circuit. This underscores the fact that symmetrization is usually expected to be helpful, but it is still a tool that needs to be treated with care as it is no panacea.

5. Discussion

In the examples shown above, we examined different figures of merit, such as the final energy, the required iterations for the optimizer, and barren plateaus, to see if the use of an equivariant ansatz can yield better performance. We verified that in some cases it actually does, especially in the number of iterations necessary to reach a solution. We attribute this to the reduction in the number of free parameters of the model, which simplifies the underlying optimization problem. However, the picture is less clear when looking at the achievable minimal energy. In our numerics, we observed the trade-off between the gain in equivariance of the ansatz on the one hand and the reduction in expressivity on the other hand. Especially at small circuit depths, it can be advantageous to have a nonequivariant ansatz that can explore larger portions of the Hilbert space, whereas at larger circuit depths the restriction of the expressivity to the relevant subspace of the Hilbert space can help.

This further motivates the use of carefully engineered symmetry breaking in conjunction with equivariant ansätze, especially at small circuit depths, as was explored in Ref. [24]. A theoretical underpinning to this reasoning was given in Ref. [75], where it was shown that an ansatz that preserves parity symmetry, such as QAOA with a shallow circuit, may be an obstacle for the preparation of the ground state of certain Hamiltonians. It is therefore essential to further clarify the conditions under which an equivariant ansatz can lead to better performance in ground-state preparation problems.

VI. SUMMARY AND OUTLOOK

In this work, we have laid the foundations for the construction of variational quantum learning models that make predictions invariant under a symmetry transformation of the data. We have shown that the embedding of the data into the Hilbert space of the quantum system plays a crucial role and that it has to be chosen suitably to induce a meaningful unitary representation of the data symmetry on the level of the Hilbert space. We have provided embeddings that allow doing this for the most important symmetries encountered in contemporary learning scenarios, namely the permutation-type symmetry, and also embeddings that induce a meaningful representation of the Lie group O(3) of orthogonal spatial transformations.

With the unitary representation of the symmetry at hand, we show how elementary results from representation theory can be used to construct equivariant gatesets from standard gatesets used for the construction of variational quantum learning models. Armed with these gatesets, the construction of invariant reuploading models becomes possible: alternating layers of equivariant data embeddings and equivariant trainable blocks applied to a symmetry-invariant initial state yield invariant predictions when evaluated on a symmetry-invariant observable. In this way, we give both a blueprint and tools for the construction of invariant variational quantum machine learning models. Moreover, using equivariant gatesets is a much needed building block that allows us to inform decisions about how to construct quantum learning models, which is a first step to a solution of the quantum model selection problem. To increase the applicability of these tools, we also outlined the pitfalls that one should avoid when using them.

The numerical experiments we have conducted on the tic-tac-toe toy example and the autonomous vehicles toy problem have confirmed our expectation that invariant learning models indeed have better generalization capabilities, as their expressivity is constrained to a set of output functions that include some knowledge about the underlying data. We have also ensured that we did not cherry-pick the results by comparing random invariant model architectures with their noninvariant counterparts where we have observed the same results.

As the existence of a unitary symmetry representation is sufficient for the construction of equivariant gatesets, these can also be applied to problems outside the realm of variational quantum machine learning. We paradigmatically explored this possibility by comparing equivariant with nonequivariant ansatz circuits for ground-state-type problems where conserved quantities of the Hamiltonian give rise to a symmetry. Our analysis on the transverse-field Ising model, the Heisenberg model, and a variant of the longitudinal transverse-field Ising model with geometric symmetry allow us to conclude that equivariant ansätze can be helpful in this application as well. They often allow us to reach a better energy estimate in fewer iterations and help to alleviate the problem of barren plateaus. Nevertheless, the picture is not as clear as in the learning scenario that we discussed in detail.

We have studied the question of equivariant quantum embeddings for the case of O(3). But there is no fundamental reason why other Lie groups that constitute a data symmetry should not also be amenable to equivariant quantum embeddings. While this is a somewhat exotic direction from the perspective of real-world learning tasks,

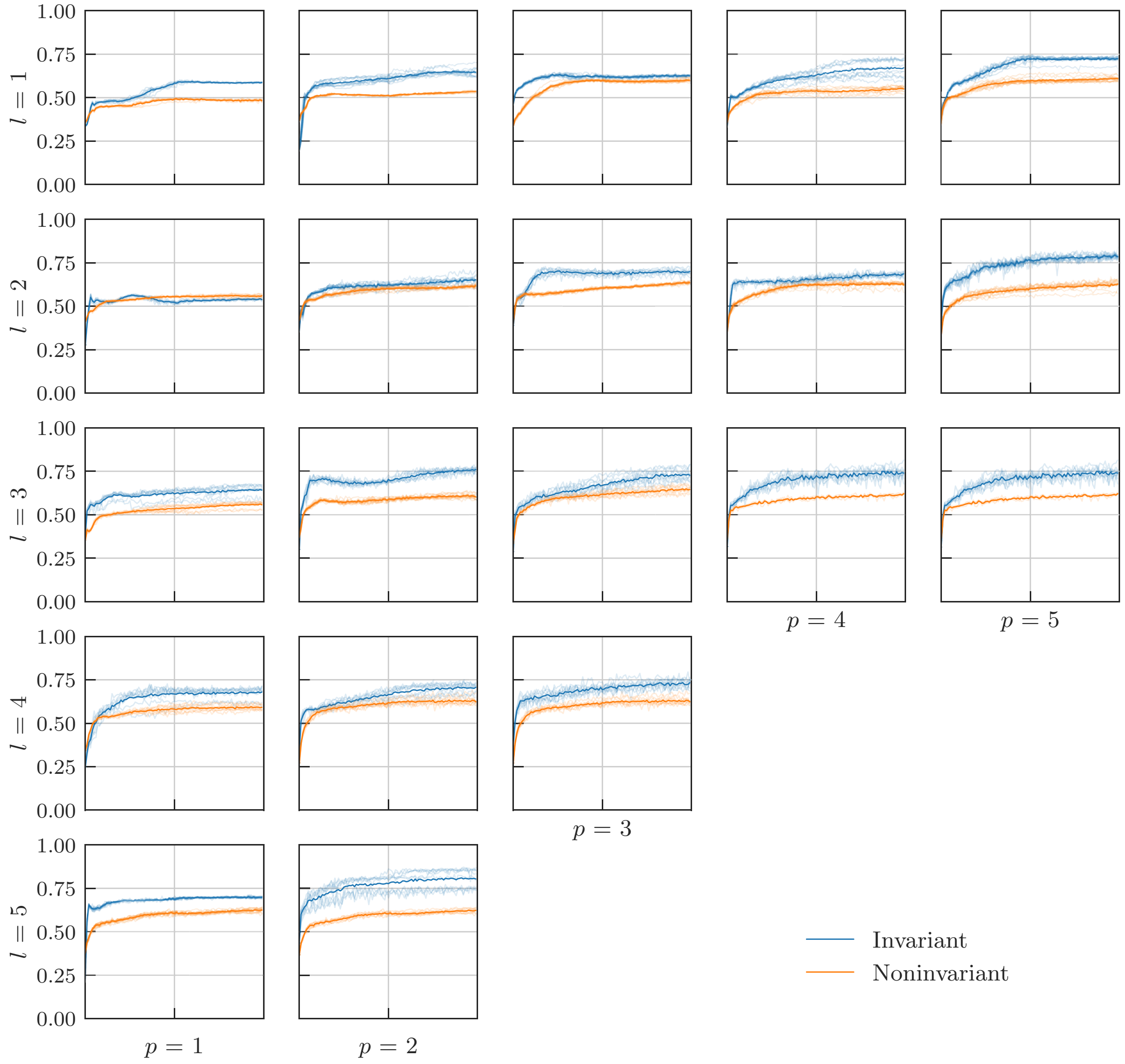


FIG. 14. Plots of the sweep over different values of l and p for the classification of tic-tac-toe games arranged in a grid. Shown on the y axis is the mean test accuracy for the invariant (blue) and noninvariant (orange) models, where the average over the ten runs is plotted in a darker color. Models are composed of l layers, each of which contains one data encoding followed by p times the standard layout “cemoid.” Optimizations are performed for 100 epochs with 30 steps each. At every step, the gradient is computed on 15 games with a total training set size of $15 \times 30 = 450$. When an epoch ends, the training set is shuffled and the accuracy is evaluated on a test set of 600 random games, which is fixed throughout a simulation. We can see that the invariant model clearly outperforms the noninvariant model. Not simulated are circuits for the values $(4, 4)$, $(4, 5)$, $(5, 3)$, $(5, 4)$, and $(5, 5)$ due to their growing computational demand. Further details can be found in the main text.

we expect that future research into the interplay between data embeddings in the quantum system's Lie algebra and unitary representations of the symmetry on the level of the Hilbert space will allow us to learn much more about the inner workings of variational quantum learning models.

Another interesting direction for future research is to prove rigorous results about the generalization capabilities of invariant quantum machine learning models, with the aim to find quantitative expressions of how much exploiting a particular symmetry helps to better solve the learning task at hand. One should also note that we have only run numerical experiments on maximally nine qubits for the learning problems, as our computational resources have been limited. In the future, it will be interesting to compare invariant variational quantum learning models to classical models on more realistic learning tasks.

In this work, we have only considered exact symmetries, both on the level of the problem and on the level of the implementation. However, both in nature and in learning problems, symmetries are often only approximate.

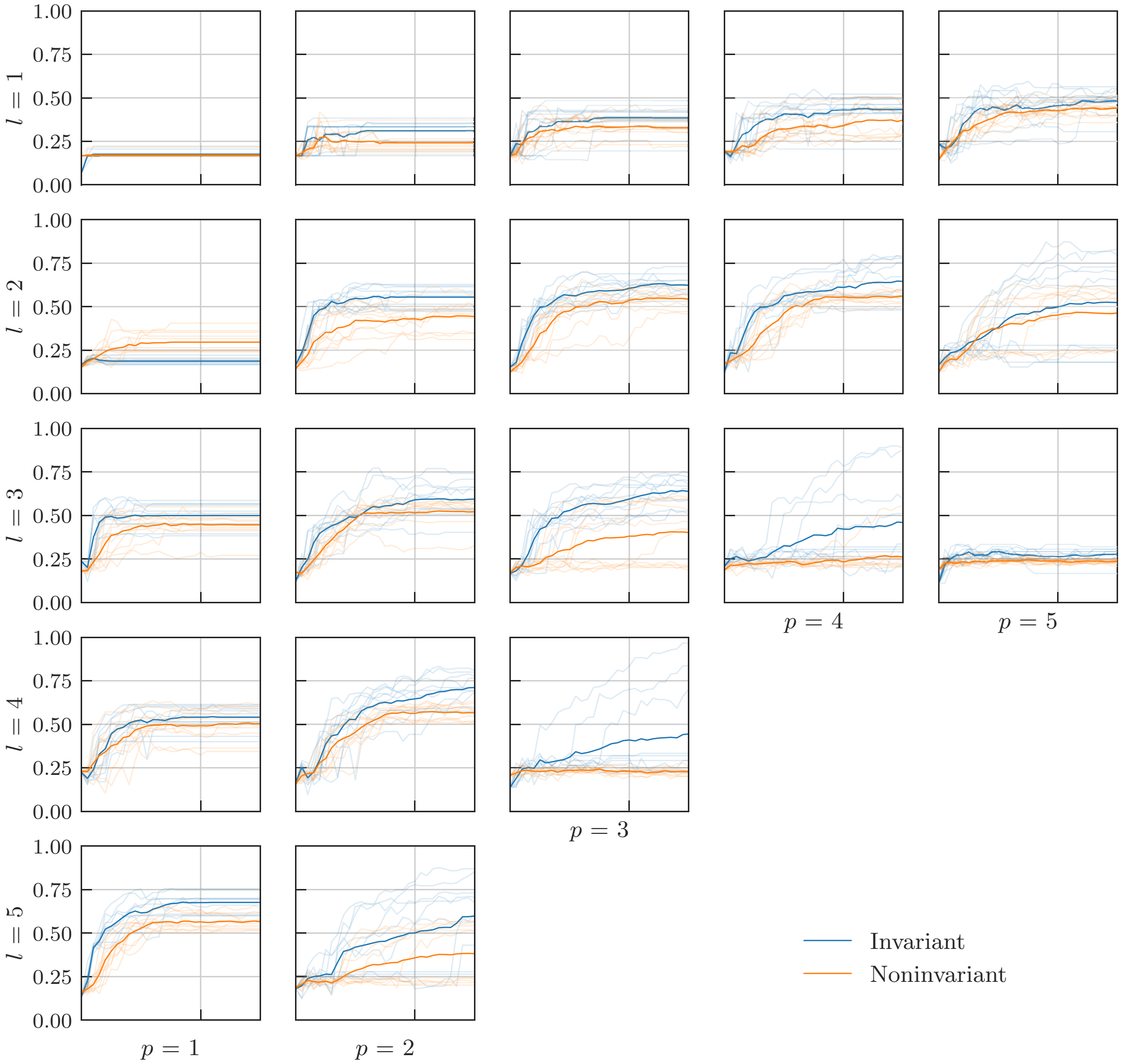


FIG. 15. Plots of the sweep over different values of l and p for the classification of autonomous vehicle scenarios arranged in a grid. Shown on the y axis is the mean test accuracy for the invariant (blue) and noninvariant (orange) models, where the average over the ten runs is plotted in a darker color. Models are composed of l layers, each of which contains one data encoding followed by p times the standard layout "cemoid." Optimizations are performed for 30 steps with no epochs on the same training set of 60 situations that are chosen randomly at the start of each simulation. The accuracies are evaluated after each step on a test set of 130 random but unique situations, which is fixed throughout a simulation. We can see that the invariant model clearly outperforms the noninvariant model. Additionally, it can be explicitly observed that, for high (l, p) values, the expressivity is empirically too high as accuracies drop sharply for both models. Not simulated are circuits for the values $(4, 4), (4, 5), (5, 3), (5, 4)$, and $(5, 5)$ due to their growing computational complexity. Further details can be found in the main text.

We believe that it will be a fruitful research direction to analyze the capabilities of approximately invariant learning models in these cases. Finally, it is our hope that this work stimulates further research efforts aimed at exploiting symmetry in variational quantum algorithms in the context of quantum machine learning and beyond.

The code for implementations and data of the numerical experiments conducted in this work will be made available upon reasonable request.

ACKNOWLEDGMENTS

The authors would like to thank Hakop Pashayan, Regina Kirschner, and Jordi Daura for insightful discussions and Matthias Caro, Simon Marshall, Sergi Masot, Franz Schreiber, Adrián Pérez-Salinas, and Andrea Skolik for useful comments on an earlier version of this manuscript. F.A. acknowledges the support of the Alexander von Humboldt Foundation. We thank the BMBF (Hybrid), the BMWK (PlanQK, EniQmA), the QuantERA (HQCC), the Munich Quantum Valley (K8), and the Einstein Foundation (Einstein Research Unit on Quantum Devices) for their support. We also thank the Porsche Digital GmbH.

J.J.M. conceived and supervised the project. J.J.M., E.G.-F., and F.A. developed the theory for equivariant embeddings and gate symmetrization. M.M. and F.A. conducted the quantum machine learning experiments based on ideas by F.A., M.M., and A.W. A.A.M. conceived and conducted the ground-state search experiments. A.W. and J.E. supported research and development. All authors contributed to the writing of the manuscript under the lead of J.J.M.

Note added.—During the finalization of this manuscript, Ref. [76] was published, which has conceptual overlap with our work, especially in the definition of invariant and equivariant models. However, Larocca *et al.* [76] considered a different setting, where learning is performed on quantum states rather than classical data, and focused on binary classification. Our work discusses how and when symmetries on the level of the quantum system arise when dealing with classical data symmetries and how we can then exploit these to construct invariant learning models. As our techniques can also be readily applied to the setting of Ref. [76], we give a way of constructing equivariant architectures and therefore answer a question which was left as a future direction in their work.

APPENDIX: ADDITIONAL FIGURES

We provide additional details on the numerical experiments conducted in the main text in Figs. 14 and 15.

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, and B. Burkett, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[2] Y. Wu, *et al.*, Strong Quantum Computational Advantage Using a Superconducting Quantum Processor, Phys. Rev. Lett. **127**, 180501 (2021).

[3] D. Hangleiter and J. Eisert, Computational advantage of quantum random sampling (2022), ArXiv:2206.04079.

[4] J. Preskill, Quantum computing in the NISQ era and beyond, Quantum **2**, 79 (2018).

[5] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[6] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nat. Rev. Phys. **3**, 625 (2021).

[7] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Rev. Mod. Phys. **94**, 015004 (2022).

[8] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. **1**, 541 (1989).

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, in *Advances in Neural Information Processing Systems*, Vol. 30 (Curran Associates, Inc., 2017).

[10] M. M. Bronstein, J. Bruna, T. Cohen, and P. Veličković, Geometric deep learning: Grids, groups, graphs, geodesics, and gauges (2021), ArXiv:2104.13478.

[11] J.-G. Liu, Y.-H. Zhang, Y. Wan, and L. Wang, Variational quantum eigensolver with fewer qubits, Phys. Rev. Res. **1**, 023025 (2019).

[12] B. R. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm, npj Quant. Inf. **6**, 1 (2020).

[13] K. Setia, R. Chen, J. E. Rice, A. Mezzacapo, M. Pistoia, and J. D. Whitfield, Reducing qubit requirements for quantum simulations using molecular point group symmetries, J. Chem. Th. Comp. **16**, 6091 (2020).

[14] G.-L. R. Anselmetti, D. Wierichs, C. Gogolin, and R. M. Parrish, Local, expressive, quantum-optnumber-preserving VQE ansätze for fermionic systems, New J. Phys. **23**, 113010 (2021).

[15] J. M. Arrazola, O. Di Matteo, N. Quesada, S. Jahangiri, A. Delgado, and N. Killoran, Universal quantum circuits for quantum chemistry, Quantum **6**, 742 (2022).

[16] N. Vogt, S. Zanker, J.-M. Reiner, M. Marthaler, T. Eckl, and A. Marusczyk, Preparing ground states with a broken symmetry with variational quantum algorithms, Quantum Sci. Technol. **6**, 035003 (2021).

[17] F. Zhang, N. Gomes, N. F. Berthusen, P. P. Orth, C.-Z. Wang, K.-M. Ho, and Y.-X. Yao, Shallow-circuit variational quantum eigensolver based on symmetry-inspired Hilbert space partitioning for quantum chemical calculations, Phys. Rev. Res. **3**, 013039 (2021).

[18] G. S. Barron, B. R. T. Gard, O. J. Altman, N. J. Mayhall, E. Barnes, and S. E. Economou, Preserving Symmetries for Variational Quantum Eigensolvers in the Presence of Noise, Phys. Rev. Appl. **16**, 034003 (2021).

[19] K. Seki, T. Shirakawa, and S. Yunoki, Symmetry-adapted variational quantum eigensolver, Phys. Rev. A **101**, 052340 (2020).

[20] K. Seki and S. Yunoki, Spatial, spin, and charge symmetry projections for a Fermi-Hubbard model on a quantum computer, Phys. Rev. A **105**, 032419 (2022).

[21] Y. Herasymenko and T. E. O'Brien, A diagrammatic approach to variational quantum ansatz construction, Quantum **5**, 596 (2021).

[22] R. Selvarajan, M. Sajjan, and S. Kais, Variational quantum circuits to prepare low energy symmetry states, Symmetry **14**, 457 (2022).

[23] C. Lyu, X. Xu, M.-H. Yung, and A. Bayat, Symmetry enhanced variational quantum eigensolver (2022), ArXiv:2203.02444.

[24] C.-Y. Park, Efficient ground state preparation in variational quantum eigensolver with symmetry breaking layers (2021), ArXiv:2106.02509.

[25] A. D. Bookatz, E. Farhi, and L. Zhou, Error suppression in Hamiltonian-based quantum computation using energy penalties, Phys. Rev. A **92**, 022317 (2015).

[26] I. G. Ryabinkin and S. N. Genin, Symmetry adaptation in quantum chemistry calculations on a quantum computer (2018), ArXiv:1812.09812.

[27] K. Kuroiwa and Y. O. Nakagawa, Penalty methods for a variational quantum eigensolver, Phys. Rev. Res. **3**, 013197 (2021).

[28] P. Mernyei, K. Meichanetzidis, and İ. İ. Ceylan, Equivariant quantum graph circuits (2021), ArXiv:2112.05261.

[29] G. Verdon, T. McCourt, E. Luzhnica, V. Singh, S. Leichenauer, and J. Hidary, Quantum graph neural networks (2019), ArXiv:1909.12264.

[30] H. Zheng, Z. Li, J. Liu, S. Strelchuk, and R. Kondor, Speeding up learning quantum states through group equivariant convolutional quantum ansätze (2021), ArXiv:2112.07611.

[31] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, Covariant quantum kernels for data with group structure (2021), ArXiv:2105.03406.

[32] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[33] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Parameterized quantum circuits as machine learning models, Quantum Sci. Technol. **4**, 043001 (2019).

[34] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Phys. Rev. A **101**, 032308 (2020).

[35] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, Data re-uploading for a universal quantum classifier, Quantum **4**, 226 (2020).

[36] M. Schuld, R. Sweke, and J. J. Meyer, Effect of data encoding on the expressive power of variational quantum-machine-learning models, Phys. Rev. A **103**, 032430 (2021).

[37] S. Jerbi, L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods (2021), ArXiv:2110.13162.

[38] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Comm. **5**, 4213 (2014).

[39] We choose *equivariant* over the synonymous *covariant* as this is the terminology used in geometric deep learning.

[40] This new group is the *semidirect product* $S_d \rtimes \mathbb{Z}_2^d$ that captures the intuition that every element $g \in S_d \rtimes \mathbb{Z}_2^d$ can be written as the product $g = f \circ \sigma$ of one permutation $\sigma \in S_d$ and a number of sign flips $f \in \mathbb{Z}_2^d$.

[41] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, Phys. Rev. A **70**, 052328 (2004).

[42] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum-enhanced feature spaces, Nature **567**, 209 (2019).

[43] G. B. Folland, *A Course in Abstract Harmonic Analysis* (Chapman and Hall/CRC, Boca Raton, 2016), 0th ed.

[44] J. Helsen, Quantum information in the real world: Diagnosing and correcting errors in practical quantum devices (2021), PhD thesis, http://resolver.tudelft.nl/uuid:312b719d-32bc-4219-82bb-8e6febc2abcc.

[45] M. S. Rudolph, S. Sim, A. Raza, M. Stechly, J. R. McClean, E. R. Anschuetz, L. Serrano, and A. Perdomo-Ortiz, ORQVIZ: Visualizing high-dimensional landscapes in variational quantum algorithms (2021), ArXiv:2111.04695.

[46] X. You and X. Wu, in *Proceedings of the 38th International Conference on Machine Learning* (PMLR, 2021), p. 12144.

[47] E. R. Anschuetz and B. T. Kiani, Beyond barren plateaus: Quantum variational algorithms are swamped with traps (2022), ArXiv:2205.05786.

[48] B. T. Kiani, S. Lloyd, and R. Maity, Learning unitaries by gradient descent (2020), ArXiv:2001.11897.

[49] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks (2021), ArXiv:2109.11676.

[50] M. Oszmaniec and Z. Zimborás, Universal Extensions of Restricted Classes of Quantum Operations, Phys. Rev. Lett. **119**, 220502 (2017).

[51] I. Marvian, Restrictions on realizable unitary operations imposed by symmetry and locality, Nat. Phys. **18**, 283 (2022).

[52] X. Wang, D. Burgarth, and S. Schirmer, Subspace controllability of spin-$\frac{1}{2}$ chains with symmetries, Phys. Rev. A **94**, 052319 (2016).

[53] T.-C. Yen, R. A. Lang, and A. F. Izmaylov, Exact and approximate symmetry projectors for the electronic structure problem on a quantum computer, J. Chem. Phys. **151**, 164111 (2019).

[54] B. Elesedy and S. Zaidi, in *Proceedings of the 38th International Conference on Machine Learning* (PMLR, 2021), p. 2959.

[55] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, Generalization in quantum machine learning from few training data, Nat. Commun. **13**, 4919 (2022).

[56] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, M. S. Alam, S. Ahmed, J. M. Arrazola, C. Blank, A. Delgado, S. Jahangiri, K. McKiernan, J. J. Meyer, Z. Niu, A. Szva, and N. Killoran, PennyLane: Automatic differentiation of hybrid quantum-classical computations, (2020), ArXiv:1811.04968.

[57] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, in *Advances in Neural Information Processing Systems 32*, edited by H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett (Curran Associates, Inc., 2019), p. 8024.

[58] T. Menzel, G. Bagschik, and M. Maurer, Scenarios for development, test and validation of automated vehicles (2018), ArXiv:1801.08598.

[59] S. Geyer, M. Baltzer, B. Franz, S. Hakuli, M. Kauer, M. Kienle, S. Kwee-Meier, T. Weigerber, K. Bengler, R. Bruder, F. Flemisch, and H. Winner, Concept and development of a unified ontology for generating test and use-case catalogues for assisted and automated vehicle guidance, Intell. Trans. Syst., IET **8**, 183 (2014).

[60] M. Steimle, T. Menzel, and M. Maurer, Toward a consistent taxonomy for scenario-based development and test approaches for automated vehicles: A proposal for a structuring framework, a basic vocabulary, and its application, IEEE Access **9**, 147828 (2021).

[61] A. Sonka, F. Krauns, R. Henze, F. Küçúkay, R. Katz, and U. Lages, in *2017 IEEE Intelligent Vehicles Symposium (IV)* (2017), p. 97.

[62] J. Xie, A. R. Hilal, and D. Kulić, Driving maneuver classification: A comparison of feature extraction methods, IEEE Sen. J. **18**, 4777 (2018).

[63] A. Schwab, (2019-05-10), Eine Methode zur Auswahl kritischer Fahrszenarien für automatisierte Fahrzeuge anhand einer objektiven Charakterisierung des Fahrverhaltens, https://mediatum.ub.tum.de/doc/1518844/1518844.pdf.

[64] F. Franchini, *An Introduction to Integrable Techniques for One-Dimensional Quantum Systems* (Springer International Publishing, Cham, 2017).

[65] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), ArXiv:1411.4028.

[66] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes (2019), ArXiv:1906.08948.

[67] W. W. Ho and T. H. Hsieh, Efficient variational simulation of non-trivial quantum states, SciPost Phys. **6**, 029 (2019).

[68] D. Wierichs, C. Gogolin, and M. Kastoryano, Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer, Phys. Rev. Res. **2**, 043246 (2020).

[69] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring Entanglement and Optimization within the Hamiltonian Variational Ansatz, PRX Quantum **1**, 020319 (2020).

[70] Z. Puchała and J. Miszczak, Symbolic integration with respect to the Haar measure on the unitary groups, Bull. Pol. Acad. Sci. Tech. Sci. **65**, 21 (2017).

[71] M. Kliesch and I. Roth, Theory of Quantum System Certification, PRX Quantum **2**, 010201 (2021).

[72] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, Connecting Ansatz Expressibility to Gradient Magnitudes and Barren Plateaus, PRX Quantum **3**, 010313 (2022).

[73] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).

[74] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control (2021), ArXiv:2105.14377.

[75] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, Obstacles to Variational Quantum Optimization from Symmetry Protection, Phys. Rev. Lett. **125**, 260505 (2020).

[76] M. Larocca, F. Sauvage, F. M. Sbahi, G. Verdon, P. J. Coles, and M. Cerezo, Group-invariant quantum machine learning (2022), ArXiv:2205.02261.