

Lattice Surgery with a Twist: Simplifying Clifford Gates of Surface Codes

Daniel Litinski and Felix von Oppen

Dahlem Center for Complex Quantum Systems and Fachbereich Physik, Freie Universität Berlin, Arnimallee 14, 14195 Berlin, Germany

We present a planar surface-code-based scheme for fault-tolerant quantum computation which eliminates the time overhead of single-qubit Clifford gates, and implements long-range multi-target CNOT gates with a time overhead that scales only logarithmically with the control-target separation. This is done by replacing hardware operations for single-qubit Clifford gates with a classical tracking protocol. Inter-qubit communication is added via a modified lattice surgery protocol that employs twist defects of the surface code. The long-range multi-target CNOT gates facilitate magic state distillation, which renders our scheme fault-tolerant and universal.

1 Introduction

The performance of quantum computers is limited by the coherence times of the underlying physical qubits. Quantum error correction [1] offers the possibility to enhance the qubits' survival times by encoding quantum information using logical qubits consisting of many physical qubits. Topological quantum error-correcting codes [2, 3] are of particular interest, as they only require the measurement of spatially local operators – a feature that is compatible with the local operations accessible in two-dimensional solid-state qubit architectures, such as superconducting qubits [4], spin qubits [5], or Majorana-based qubits [6].

Quantum error-correcting codes typically operate in cycles. In each code cycle, mutually commuting operators called stabilizers [7] are measured to reveal the error syndrome, which is used to determine and correct errors. Surface codes [8, 9] are topological codes that feature a high error threshold [10, 11], and only require the measurement of four-qubit stabilizer operators for the readout of the error syndrome. The low-weight stabilizers are an advantage over other codes such as color codes [12, 13], which require the measurement of six-qubit operators. This facilitates syndrome readout in many physical architectures such as superconducting qubits, where the measurement of higher-weight sta-

bilizers requires more potentially faulty controlled-not (CNOT) gates.

The main drawback of surface codes in comparison to color codes is the absence of transversal single-qubit Clifford gates, i.e., the gates that are products of the Hadamard gate H and the phase gate S . While the transversal Clifford gates of color codes provide them with fast logical H and S gates, defect-based proposals for surface codes [14] implement the H gate via a multi-step measurement protocol, and the S gate via a distilled ancilla qubit. In order to lower the overhead of single-qubit Clifford gates, surface code qubits can be encoded using twist defects [15], which are essentially Majoranas that can be braided via code deformation [16]. It was pointed out that braiding of twists can also be implemented via a classical tracking protocol [17], in accordance with the Gottesman-Knill theorem [18].

In this work, we present a scheme that implements this tracking protocol for planar surface codes, as opposed to twist-based encodings. We refer to this protocol as *edge tracking*. In our scheme, Clifford completeness is achieved via a modified lattice surgery [19]

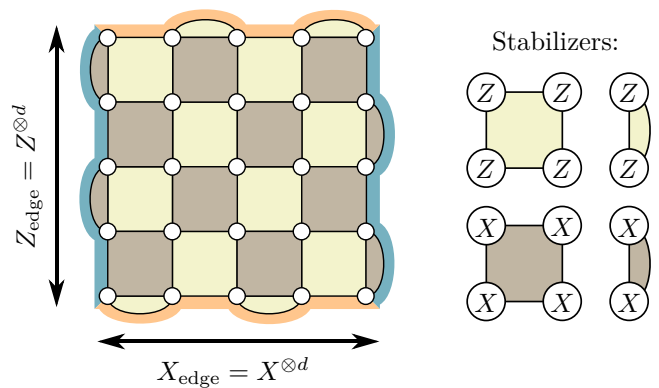


Figure 1: An example of a surface code qubit with code distance $d = 5$. Physical qubits are located on the vertices, and the faces define the two- and four-qubit Z type (bright) and X type (dark) stabilizer operators. X strings along the X edge (orange) are logical X_L operators, whereas Z strings along Z edges (blue) are Z_L operators.

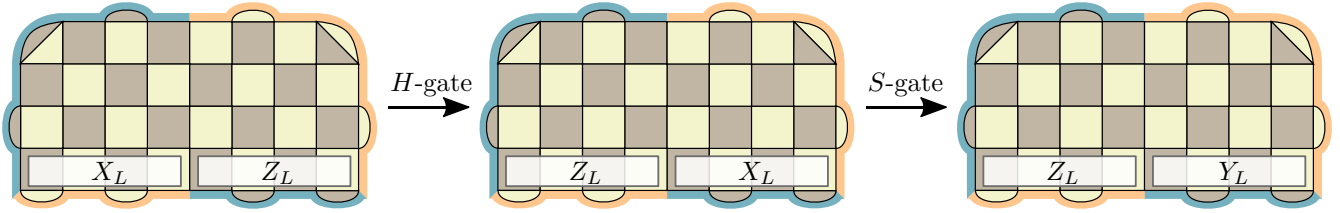


Figure 2: An example of edge tracking with a wide surface code qubit. Starting from the default encoding $X_{\text{edge}} = X_L$ and $Z_{\text{edge}} = Z_L$, an H gate changes it to $X_{\text{edge}} = Z_L$ and $Z_{\text{edge}} = X_L$, and a subsequent S gate modifies it to $X_{\text{edge}} = Z_L$ and $Z_{\text{edge}} = Y_L$.

protocol. Twist defects are no longer used to encode quantum information, but reappear in lattice-surgery protocols involving the logical Y_L operator, so that we refer to the protocol as *twist-based lattice surgery*. Our scheme provides long-range multi-target CNOT gates – i.e., CNOTs with one control and arbitrarily many targets – between any set of edge-tracked surface code qubits. These gates are particularly useful for magic state distillation [20], which completes the universal gate set by fault-tolerantly implementing the T gate (or $\pi/8$ gate). Our scheme not only eliminates the need for hardware operations for single-qubit Clifford gates, but also conceptually simplifies the twist-defect-based approach to surface-code quantum computing. Even though our scheme features twist defects and dislocation lines, the only concepts necessary to understand our scheme are the encoding of logical qubits and the measurement of two-qubit parity operators. We discuss the implementation of the single-qubit Clifford gates, CNOT gates, and T gates in Secs. 2, 3 and 4, respectively. In a concluding section, we discuss our scheme in the context of possible hardware implementations and in comparison to alternative topological codes.

2 Edge Tracking

The basic framework of our scheme are physical qubits arranged on a 2D square lattice which allow for the measurement of local stabilizer operators. Examples of possible physical realizations include superconducting qubits emulating stabilizer measurements using ancilla qubits and CNOT gates [14], or Majorana-based qubits using direct measurements of the stabilizers via Majorana fermion parity measurements [21]. A single surface code qubit can be defined using the checkered square shown in Fig. 1, where physical qubits are located at the vertices. We refer to the Pauli operators of the physical qubits as X , Y , and Z . The faces define the $X^{\otimes n}$ - and $Z^{\otimes n}$ -stabilizers of the code, where n is the number of qubits that are part of the face. The figure shows an example of a code with code distance $d = 5$, but this construction can be generalized to arbitrary

code distances.

Surface code qubits have two distinct types of boundaries, usually referred to as rough and smooth edges. Here, we call them X and Z edges in analogy to the logical Pauli operators X_L and Z_L that they encode. Surface code qubits can be easily initialized in the logical $+1$ -eigenstates $|0_L\rangle$ and $|+L\rangle$ of Z_L and X_L by initializing all physical qubits in the corresponding physical states $|0\rangle$ and $|+\rangle$, measuring all stabilizers, and correcting the errors. Conversely, they can be read out in the X_L and Z_L basis by measuring all physical qubits in the X or Z basis, and performing classical error correction.

We define the operator X_{edge} (Z_{edge}) as the string of X operators (Z operators) on all physical qubits along an X edge (Z edge). In the default encoding, $X_{\text{edge}} = X_L$ and $Z_{\text{edge}} = Z_L$. The edge tracking procedure that we now introduce essentially modifies which logical operators are encoded by X_{edge} and Z_{edge} . Logical single-qubit Clifford gates map the logical Pauli operators X_L , Y_L , and Z_L onto other Pauli operators. In particular, an H gate maps $X_L \rightarrow Z_L$, $Y_L \rightarrow -Y_L$, and $Z_L \rightarrow X_L$. An S gate maps $X_L \rightarrow Y_L$, $Y_L \rightarrow -X_L$, and $Z_L \rightarrow Z_L$. Thus, we can replace single-qubit Clifford gates by a classical tracking procedure. This is essentially the content of the Gottesman-Knill theorem [18], which states that Clifford gates can be simulated efficiently on a classical computer. For now, we only consider tracking of single-qubit Clifford gates H and S , whereas CNOT gates are performed explicitly.

In order to combine this tracking scheme with lattice surgery, it will be convenient to use the wide qubits shown in Fig. 2 instead of the square qubits that were previously introduced. These qubits have an X and Z edge on the same side, such that the logical operators X_L , Y_L and Z_L can all be accessed by lattice surgery from the same side of the qubit. Compared to square qubits with the same code distance, this comes at the price of a larger number of physical qubits for each logical qubit. The figure also shows an example of edge tracking. The default encoding is $X_{\text{edge}} = X_L$ and $Z_{\text{edge}} = Z_L$. An H gate changes the encoding to $X_{\text{edge}} = Z_L$ and

$Z_{\text{edge}} = X_L$. A subsequent S gate modifies it to $X_{\text{edge}} = Z_L$ and $Z_{\text{edge}} = Y_L$.

3 Lattice surgery with a twist

Edge tracking requires a suitable CNOT gate protocol in order to be useful for universal quantum computation. This is provided by twist-based lattice surgery. It essentially implements the circuit identity shown in Fig. 3 for edge-tracked qubits. Here, a CNOT between a control and target qubit corresponds to three measurements: a $Z \otimes Z$ parity measurement between the control and an ancilla initialized in the X eigenstate $|+\rangle$, a subsequent $X \otimes X$ parity measurement between ancilla and target, and a final Z basis readout of the ancilla qubit. In order to use this protocol for *logical* CNOTs, measurements of logical two-qubit parity operators are required, e.g., operators such as $Z_L \otimes Z_L$, which are nonlocal operators involving $2d$ physical qubits.

3.1 Nearest-neighbor CNOT

Let us first discuss standard lattice surgery between two neighboring wide qubits in the default encoding. Consider the CNOT protocol in Fig. 4. Lattice surgery [19] is a protocol for fault-tolerant logical parity measurements which only requires the measurement of local stabilizer operators. After initializing an ancilla qubit in the $|+\rangle$ state, lattice surgery between the Z edges of the control and ancilla qubit in step (2) measures their $Z_L \otimes Z_L$ parity. This is done by modifying the stabilizers along the boundaries. The boundary X stabilizers are merged to form four-qubit stabilizers (orange), and new Z stabilizers (blue) are introduced. While the stabilizers still mutually commute, this procedure increases the total number of stabilizers by one. In other words, the number of degrees of freedom is reduced by one, and one bit of information is measured during this proto-

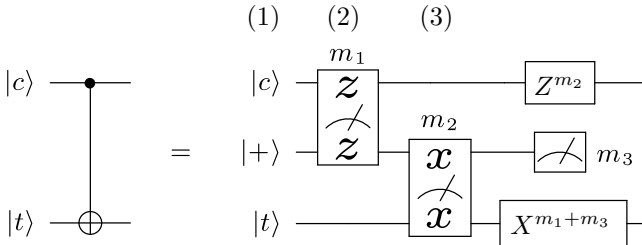


Figure 3: A CNOT between a control $|c\rangle$ and a target $|t\rangle$ is equivalent to a $Z \otimes Z$ parity measurement between $|c\rangle$ and an ancilla in the $|+\rangle$ state, followed by an $X \otimes X$ parity measurement between ancilla and $|t\rangle$, and finally a Z basis measurement of the ancilla. The measurement outcomes determine a Pauli correction.

col. The measurement outcome of the orange stabilizers is trivial, as they are products of previously known boundary stabilizers. The outcome of the blue stabilizers, on the other hand, is nontrivial. They contain each boundary qubit exactly once. Therefore, their product is precisely the operator $Z_{\text{edge}}^{(\text{control})} \otimes Z_{\text{edge}}^{(\text{ancilla})}$, which corresponds to the $Z_L \otimes Z_L$ parity in the default encoding. Thus, lattice surgery implements a fault-tolerant parity measurement between logical qubits. Similarly, in the following lattice surgery step (3), the blue stabilizers are trivial, and the product of orange stabilizers is

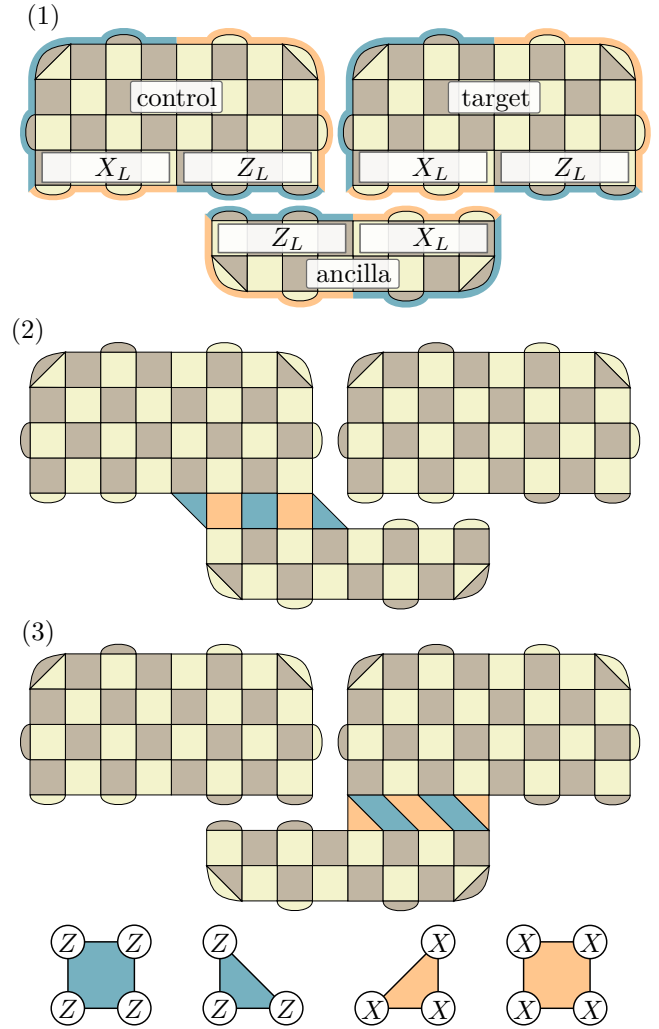


Figure 4: CNOT by lattice surgery corresponding to the gate circuit in Fig. 3. (1) All qubits are in the default encoding $X_{\text{edge}} = X_L$ and $Z_{\text{edge}} = Z_L$, and the ancilla is initialized in the $|+\rangle$ state. (2) To measure the $Z_L \otimes Z_L$ parity between control and target, the two-qubit boundary stabilizers are merged (orange), and new Z type stabilizers (blue) are introduced, whose product is precisely the parity. (3) Similarly, the $X_L \otimes X_L$ parity between ancilla and target is measured by the product of new X type stabilizers (orange).

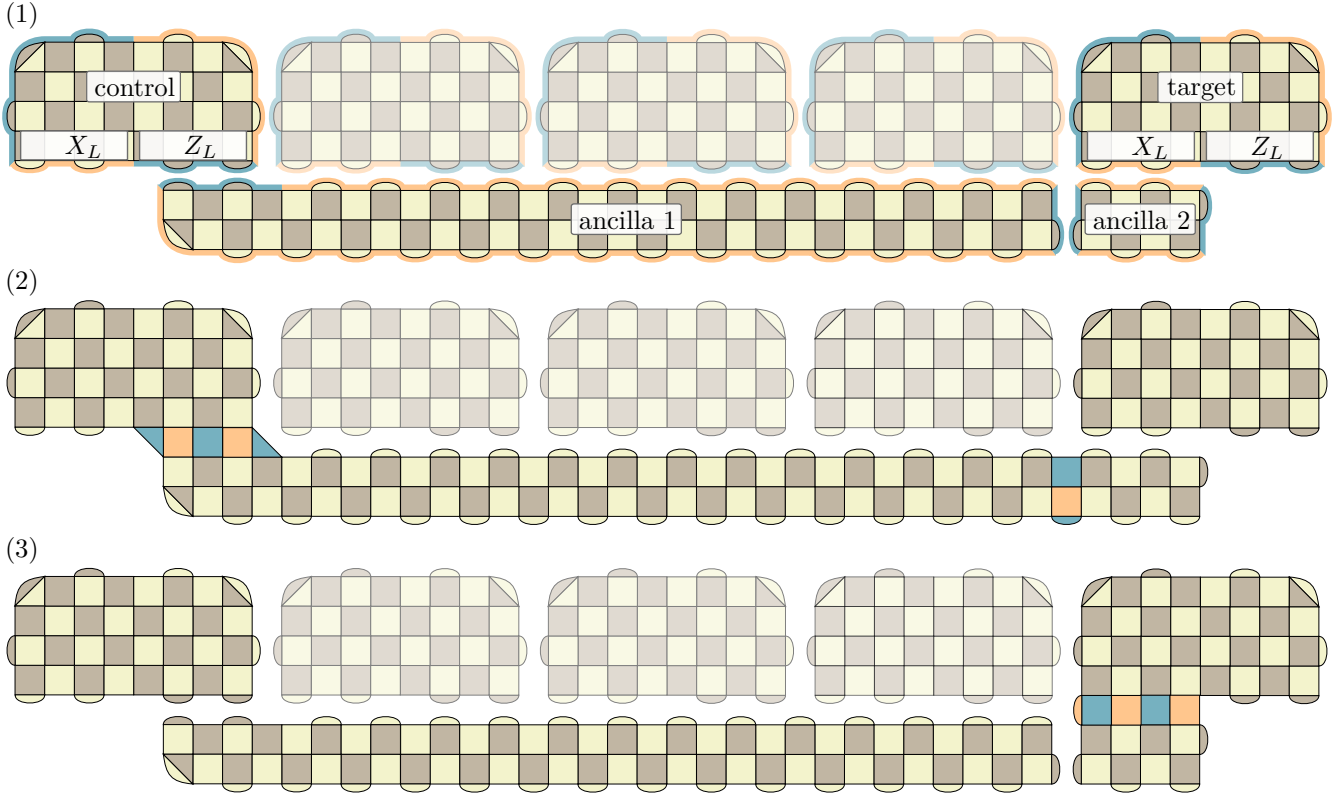


Figure 5: Long-range CNOT between two wide qubits in the default encoding that are separated by three other qubits. After initializing two ancillas in the $|+\rangle$ state (1), lattice surgery (2) simultaneously measures the $Z_L \otimes Z_L$ parities between control and ancilla 1, and ancilla 1 and ancilla 2. This also yields the $Z_L \otimes Z_L$ parity between control and ancilla 2, such that ancilla 2 can be used for an $X_L \otimes X_L$ parity measurement (3) with the target qubit. At the end of the CNOT protocol, ancilla 1 is read out in the X basis with outcome m , leading to a Z^m correction on the control.

$X_{\text{edge}}^{(\text{ancilla})} \otimes X_{\text{edge}}^{(\text{target})}$. A Z_L basis measurement of the ancilla qubit completes the gate circuit in Fig. 3. The subsequent Pauli corrections are Clifford gates and can be handled by edge tracking.

3.2 Long-range CNOT

A similar protocol can be used to perform CNOTs between logical qubits that are not nearest neighbors, but separated by some distance. For this, we use lattice surgery to measure the $Z_L \otimes Z_L$ parities between the control qubits and multiple ancilla qubits simultaneously [19, 22, 23]. In the protocol in Fig. 5, two ancilla qubits are initialized in the $|+\rangle$ state, one long ancilla that spans the entire distance between the control and target, and another that is adjacent to the X edge of the target. In step (2), lattice surgery simultaneously measures the $Z_L \otimes Z_L$ parities between control and long ancilla, and between both ancillas. This effectively measures the $Z_L \otimes Z_L$ parity between control and ancilla 2 as the product of both measurements. Thus, ancilla 2 can be used as the ancilla of the CNOT protocol of Fig. 3. An $X_L \otimes X_L$ parity measurement between an-

cilla 2 and the target qubit, and a subsequent Z basis readout of ancilla 2 complete the CNOT protocol. Since ancilla 1 is still entangled with the control qubit, it cannot be discarded right away, but needs to be measured in the X basis with outcome $m \in \{0, 1\}$, which leads to a subsequent Z^m Pauli correction on the control qubit.

Vertical X error strings connecting the (orange) X edges of the long ancilla qubit can introduce errors to the CNOT protocol. While the number of possible error strings increases linearly with the control-target separation s , the probability of error strings decreases exponentially with the width of the ancilla. Therefore, the width needs to increase with $\mathcal{O}(\log s)$ in order to maintain the CNOT gate fidelity, implying a space overhead of $\mathcal{O}(s \log s)$ for the long-range CNOT. There are two factors that contribute to the time overhead of the protocol: decoding and syndrome readout errors. While decoding can be done with a runtime that scales with $\mathcal{O}(\log s)$ [24], the correction of stabilizer measurement errors is handled by recording multiple rounds of syndrome extraction for one code cycle [25]. This effectively introduces a third dimension to the code. The

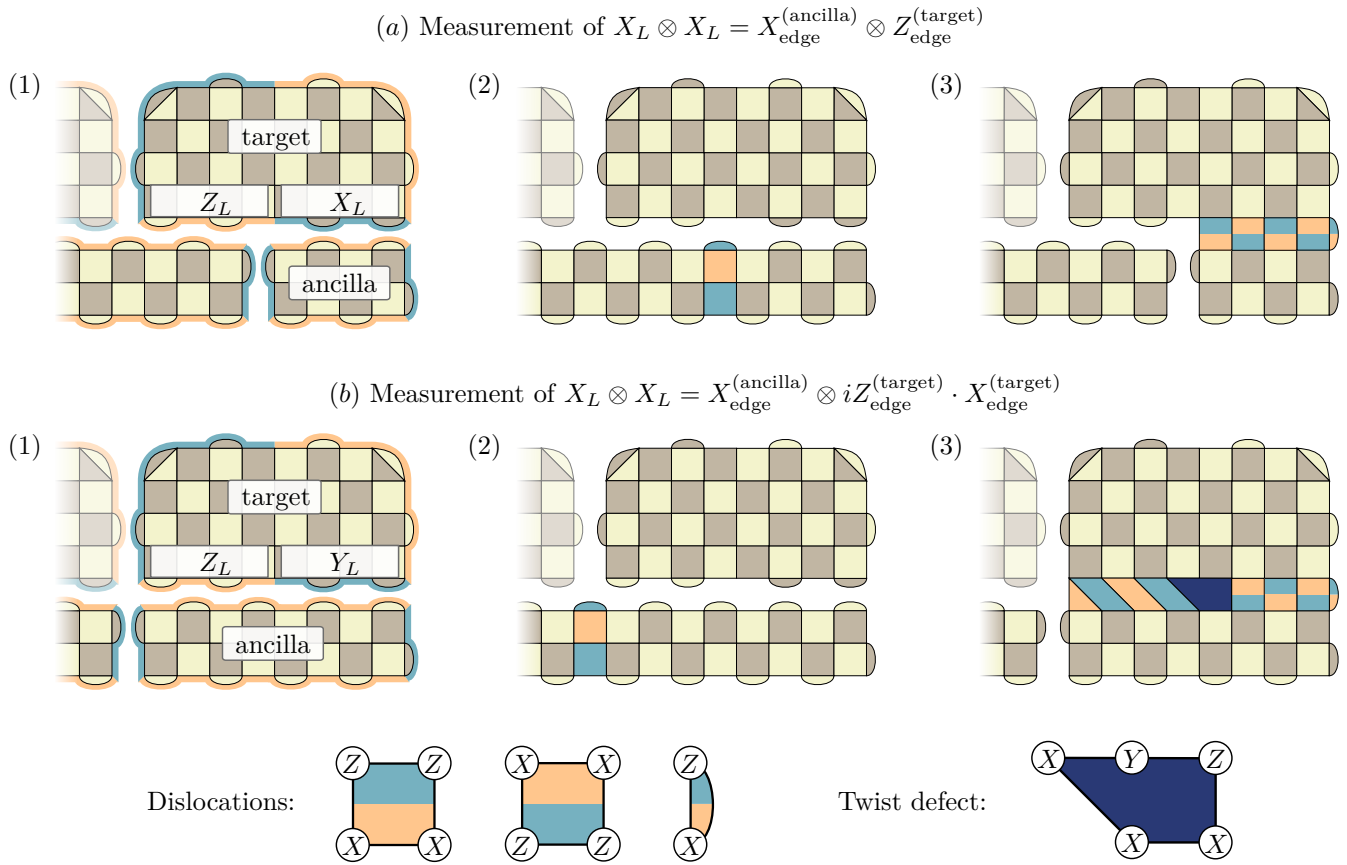


Figure 6: $X_L \otimes X_L$ parity measurements between an ancilla and an edge-tracked target qubit. In (a), edge tracking has changed the encoding of the target to $X_{\text{edge}} = Z_L$ and $Z_{\text{edge}} = X_L$. The stabilizer configuration that measures the $X_L \otimes X_L$ parity corresponds to a dislocation line. In (b), the encoding of the target qubit is $X_{\text{edge}} = Z_L$ and $Z_{\text{edge}} = Y_L$. Here, the $X_L \otimes X_L$ parity is measured by a stabilizer configuration that corresponds to a dislocation line that is terminated by a twist defect.

number of recorded measurement rounds for each code cycle depends on the measurement fidelity. With higher measurement fidelity, fewer measurement rounds are required to reach the same logical CNOT gate fidelity. As with the width of the long ancilla, errors in the time dimension are suppressed exponentially with the number of measurement rounds, i.e., with the code distance in time, but the number of possible error strings increases linearly with s . This implies that the number of measurement rounds needs to increase with $\mathcal{O}(\log s)$. Thus, the total time overhead is still just $\mathcal{O}(\log s)$, which is essentially constant for finite-size systems.

Note that in our figures (such as Fig. 5), the widths of the ancilla qubits, and therefore their code distances, are chosen to be smaller than the code distances of the wide qubits. This may be a valid choice for some computations, since the ancillas only need to survive for the duration of the CNOT, as opposed to data qubits that may need to survive for the entire computation. In practice, however, we expect that the space reserved for ancilla qubits will be in use for various CNOT gates for essentially the entire duration of the quantum compu-

tion. Therefore, for most applications, the code distances of the ancilla qubits and the data qubits should be chosen to be equal, and the logarithmic space overhead scaling with the control-target separation can be ignored. In this case, all logical qubits are protected against error strings of length $(d - 1)/2$ during each code cycle. There is still a logarithmic space overhead scaling, since the necessary code distance to reach a certain target error probability at the end of a quantum computation involving n logical qubits scales with $\mathcal{O}(\log n)$.

3.3 CNOT between edge-tracked qubits

The previously discussed standard lattice surgery protocols can be used to measure $Z_{\text{edge}} \otimes Z_{\text{edge}}$ and $X_{\text{edge}} \otimes X_{\text{edge}}$. However, CNOTs between edge-tracked qubits may require additional parity measurements. This is where dislocations and twist defects come into play.

In Fig. 6, we explore the two additional situations that may occur for $X_L \otimes X_L$ parity measurements be-

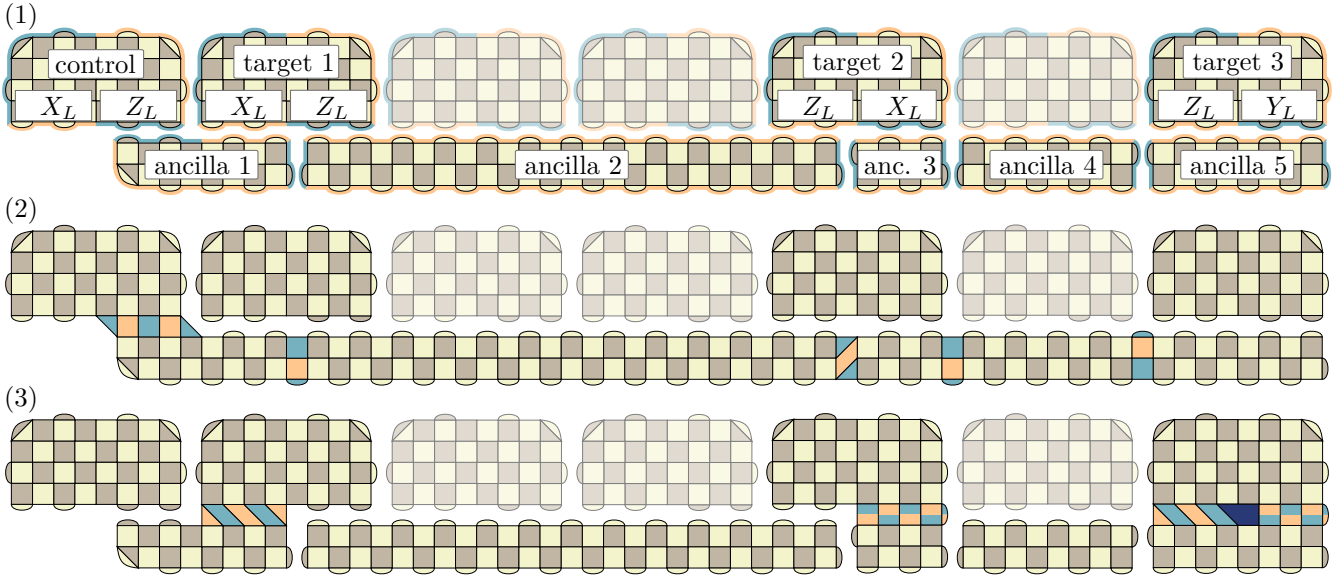


Figure 7: Long-range multi-target CNOTs with edge-tracked qubits. The control, ancillas, and target 1 are in the default encoding $Z_{\text{edge}} = Z_L$ and $X_{\text{edge}} = X_L$, whereas target 2 and target 3 have been modified by edge tracking, such that $X_L \otimes X_L$ parity measurements require lattice surgery between different edge types. Five ancilla qubits are initialized in the $|+\rangle$ state (1) and their $Z_L \otimes Z_L$ parities with the control qubit are measured simultaneously (2). Ancillas 2 and 4 merely provide long-range communication and are not used for CNOTs, but are instead read out in the X basis. Subsequent $X_L \otimes X_L$ parity measurements (3) use the previously discussed lattice surgery protocols for edge-tracked qubits.

tween an ancilla and an edge-tracked target qubit during a CNOT protocol. In the first situation (a), the X_L operator is defined by the target's Z edge as a consequence of edge tracking. Thus, lattice surgery needs to measure the operator $X_{\text{edge}}^{(\text{ancilla})} \otimes Z_{\text{edge}}^{(\text{target})}$. For this, the boundary stabilizers are merged, and new stabilizers are introduced. One can check that all stabilizers commute, and that the product of the nontrivial stabilizers indeed yields $X_L \otimes X_L$.

The remaining possibility is that, as a consequence of edge tracking, none of the edges of the target define its X_L . In (b), the target qubit is in the encoding where $X_{\text{edge}} = Z_L$ and $Z_{\text{edge}} = Y_L$. Since $X_L = iZ_L Y_L$, and therefore $X_L = iX_{\text{edge}} Z_{\text{edge}}$, lattice surgery now needs to measure $X_{\text{edge}}^{(\text{ancilla})} \otimes iX_{\text{edge}}^{(\text{target})} \cdot Z_{\text{edge}}^{(\text{target})}$. Similar to the previous cases, stabilizers along the boundary in (b3) are merged yielding the trivial stabilizers. The product of the newly introduced nontrivial stabilizers is again the $X_L \otimes X_L$ parity. Note that the center qubit of the blue five-qubit operator contributes to the stabilizer measurement in the Y basis, since it is part of both the X and the Z edge.

The three different lattice surgeries in panel (3) of Fig. 5, and panels (a3) and (b3) of Fig. 6 can also be interpreted as protocols to measure $X_L \otimes X_L$, $Z_L \otimes X_L$ and $Y_L \otimes X_L$ between a wide qubit and a square qubit in the default encoding. The protocol involving Y_L is what we refer to as twist-based lattice surgery, since the

five-qubit operator corresponds to a twist defect.

Such a parity measurement can also be used to measure the product $iX_{\text{edge}} \cdot Z_{\text{edge}}$ of a qubit, e.g., to read out the qubit in the Y_L basis in the default encoding. For this, an ancilla can be initialized in the $|0\rangle$ state, such that a $Y_L^{(\text{qubit})} \otimes Z_L^{(\text{ancilla})}$ parity measurement between qubit and ancilla is equivalent to a Y_L measurement of the qubit.

This covers all the necessary lattice surgery protocols for CNOTs between edge-tracked qubits. The $Z_L \otimes Z_L$ parity measurements between ancilla qubits and edge-tracked control qubits are analogous to the $X_L \otimes X_L$ parity measurements in Fig. 6. The concrete implementation of the required stabilizer measurements depends on the given architecture. While Majorana-based implementations allow for direct measurements of the necessary operators, non-topological setups such as superconducting qubits require the use of measurement qubits. In the latter case, the stabilizer measurement protocol requires special care in order to avoid correlated errors that lower the effective code distance, as we show in Appendix A.

3.4 Connection to twist defects

The stabilizer configurations in these modified lattice surgery protocols feature dislocations and twist defects. The mixed stabilizers in (a3) correspond to a dislocation in the surface code. The stabilizer configuration in (b3)

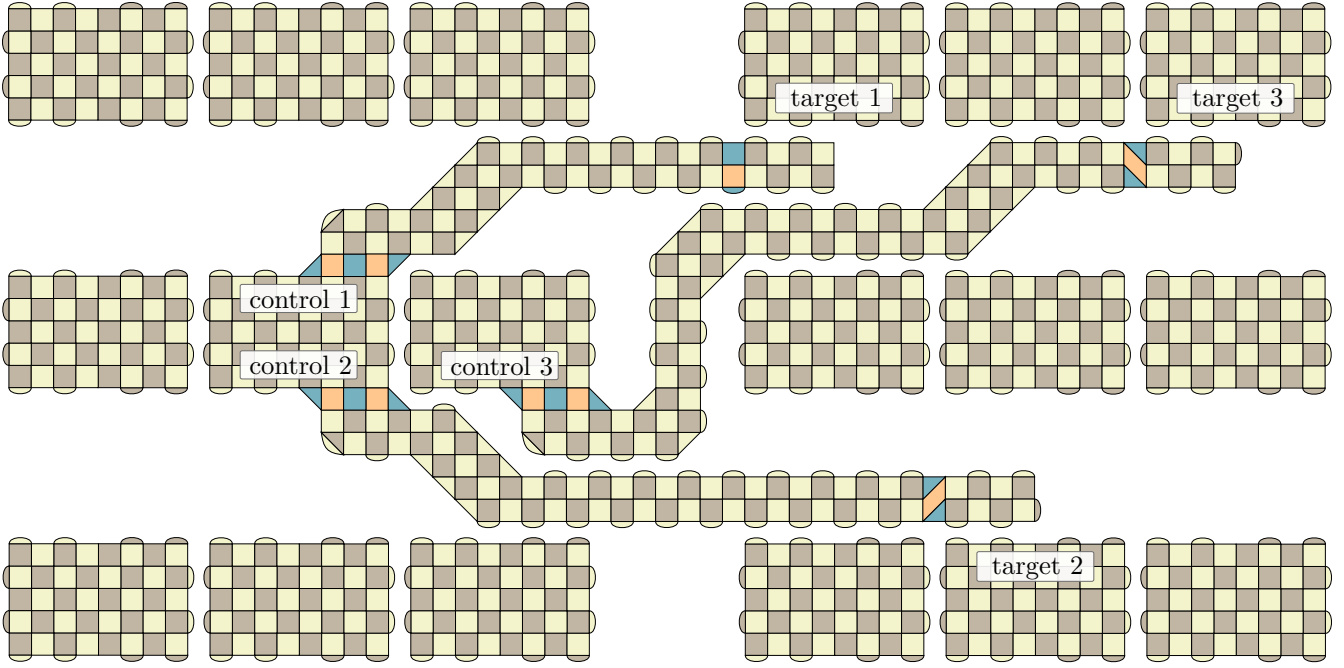


Figure 8: Example of a two-dimensional arrangement of surface code qubits, where qubits are grouped in blocks of six. The long ancilla qubits can be used for three simultaneous long-range CNOT gates.

corresponds to a dislocation line between the X edge of the ancilla and the Z edge of the target which is terminated by a five-qubit twist defect [15, 16].

Twist-based lattice surgery can also be interpreted in a Majorana fermion picture. It was pointed out that the corners of square surface code qubits (as in Fig. 1) correspond to twist defects [16]. Similarly, the ends of the X and Z edges of wide qubits can be replaced by twist defects – i.e., Majorana fermions – such that the logical operators X_L , Z_L , and Y_L are two-Majorana fermion parity operators. Lattice surgery then effectively implements a four-Majorana fermion parity measurement [16]. In Fig. 6 (b3), these four Majorana fermions are in the bottom left and right corners of the target, and in the top left and right corners of the ancilla. The twist defect corresponds to the remaining Majorana fermion residing between the X and Z edge of the target qubit, which is not part of the parity measurement.

3.5 Long-range multi-target CNOT

The simultaneous $Z_L \otimes Z_L$ parity measurements of long-range CNOTs can be used for multi-target CNOTs, i.e., for multiple CNOTs with the same control, but different target qubits. An example is shown in Fig. 7, where five ancillas are used to perform three CNOTs with three edge-tracked targets simultaneously. Step (2) shows the simultaneous measurement of $Z_L \otimes Z_L$ parities of six

neighboring qubits, which correspond to one control and five ancilla qubit. This protocol effectively measures the $Z_L \otimes Z_L$ parities of all pairs of qubits, and in particular of the control and each ancilla qubit. Thus, each of the five ancilla qubits can be used for $X_L \otimes X_L$ parity measurements with target qubits. While ancillas 1, 3, and 5 are used for CNOTs with targets 1, 2, and 3, ancillas 2 and 4 merely bridge distances between distant qubits.

Thus, by simultaneously initializing multiple ancillas, lattice surgery provides long-range multi-target CNOTs with edge-tracked qubits with the same time overhead as single CNOTs. At the end of the protocol, ancillas that are used for $X_L \otimes X_L$ parity measurements with target qubits are read out in the Z basis, whereas ancillas used to bridge long distances are read out in the

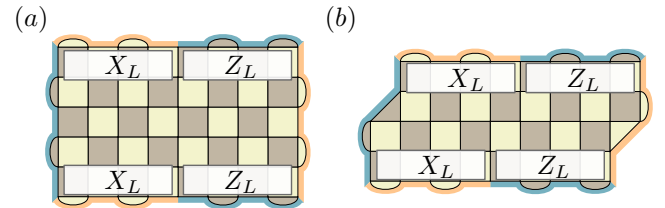


Figure 9: Double-sided qubits encode two logical qubits using (a) $2d^2 + d - 1$ or (b) $2d^2 - d$ physical qubits. The left and right edges correspond to the logical operators $Z_L \otimes Z_L$ and $X_L \otimes X_L$ of both encoded qubits, respectively.

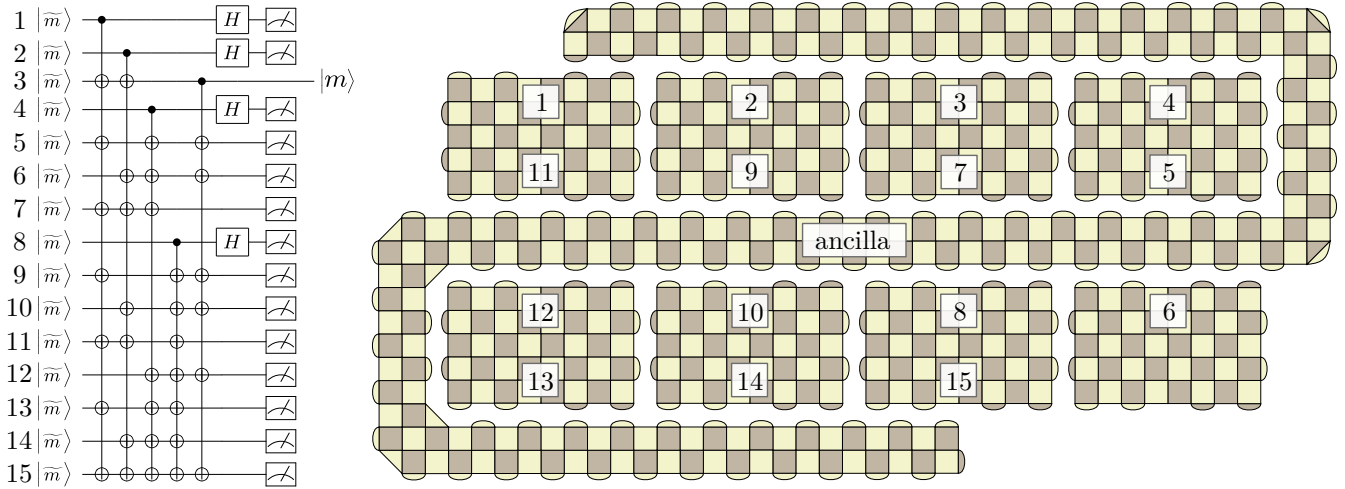


Figure 10: Example of the 15-to-1 magic state distillation protocol using long-range multi-target CNOTs via lattice surgery. By appropriately partitioning the long ancilla qubit, it can be used for each of the five multi-target CNOTs of the protocol.

X basis. Multi-target CNOTs are particularly useful for logical T gates, as magic state distillation schemes typically consist of many multi-target CNOTs. These complete the universal gate set of our scheme, as we discuss in the following section.

4 2D arrangement of logical qubits

So far, we have considered logical qubits arranged on a line. The lattice-surgery-based CNOT gates can also provide long-range connectivity in two dimensions. For this, it will be convenient to use the space of wide qubits to encode *two* logical qubits instead of just one. The double-sided qubits shown in Fig. 9 reduce the space overhead from $\sim 2d^2$ physical qubits for each logical (wide) qubit back to $\sim d^2$ physical qubits, similar to the square qubits in Fig. 1. The downside of double-sided qubits is that state initialization and readout is more complicated, as the two encoded qubits cannot be measured independently. However, one can use lattice surgery to initialize and read out in any Pauli basis. For instance, a qubit can be initialized in the $|0\rangle$ state by initializing a standard ancilla encoding a single qubit in the $|0\rangle$ state and performing lattice surgery via the Z edges of both qubits. Readout is done the same way, using the appropriate edge of the qubit. Should one require fast initialization and readout, it is still possible to use wide qubits instead of double-sided qubits.

An example of a 2D arrangement of double-sided qubits is shown in Fig. 8, where they form blocks of six logical qubits. The space between blocks is used for ancilla qubits for long-range CNOT gates. The separation between blocks not only sets the maximum width of the ancilla qubits, but also influences the number

of multi-target CNOTs that can be performed simultaneously. The larger the separation, the more ancilla qubits can fit between the qubit blocks. The example in Fig. 8 shows three simultaneous CNOT gates, where the space between qubit blocks allows for two parallel “lanes” of CNOT ancillas. Thus, a larger separation between qubit blocks increases the connectivity, but also the space overhead.

4.1 Example: Magic state distillation

Having discussed the implementation of the logical Clifford gates in our scheme, the remaining gate for universal quantum computation is the logical T gate. One possibility to implement the logical T gates and logical Clifford gates is magic state distillation [20]. The aim of this scheme is to generate an encoded magic state $|m\rangle = (|0\rangle + e^{i\pi/4}|1\rangle)/\sqrt{2}$, which corresponds to a $|+\rangle$ -state on which a T gate has been performed. A CNOT gate between $|m\rangle$ and a target qubit, followed by the measurement of $|m\rangle$ corresponds to a logical T gate on the target qubit, up to a Clifford correction.

However, it is only possible to prepare *physical* magic states, which are moreover faulty states $|\tilde{m}\rangle$, i.e., generated using an imprecise physical T gate. These physical states can be converted into *logical* faulty magic states $|\tilde{m}\rangle$ via state injection [19]. Magic state distillation protocols take many faulty magic states and convert them to fewer, but more precise magic states. These protocols typically consist of many multi-target CNOT gates.

One example of a magic state distillation protocol is shown in Fig. 10 for the example of 15-to-1 conversion [20], which converts 15 faulty magic states into one better magic state. It consists of 34 CNOT gates

	color code	wide surface code	double-sided surface code
space overhead	+ low ($\approx \frac{3}{4}d^2$ or $\frac{1}{2}d^2$)	- high ($\approx 2d^2$)	\sim moderate ($\approx d^2$)
initialization & readout	+ fast X, Y, Z	\sim fast X, Z ; slow Y	- slow X, Y, Z
stabilizer weight	- high (six or eight)	+ low (four)	+ low (four)

Table 1: Comparison between color-code-based [13, 23] and surface-code-based schemes.

grouped into five multi-target CNOTs. The figure also shows an arrangement of qubits that can be used to implement the protocol. By appropriately partitioning the long ancilla qubit, each of the five multi-target CNOTs can be performed using the protocol in Fig. 7. We provide the detailed stabilizer configurations for this 15-to-1 conversion in Appendix B. The space overhead of the 15-to-1 conversion depends on the code distances of the magic states and the width of the ancilla. The time overhead is mostly determined by the five multi-target CNOTs, which require two code cycles (including repetitions accounting for stabilizer measurement errors) for their parity measurements by lattice surgery.

5 Conclusion

We have demonstrated that edge tracking can be used to eliminate the time overhead of logical single-qubit Clifford gates in surface codes, as should be expected considering the Gottesman-Knill theorem. Twist-based lattice surgery provides long-range multi-target CNOTs with a time overhead that only scales with $\mathcal{O}(\log s)$ of the control-target separation s , and a space overhead that scales with $\mathcal{O}(s \log s)$. Compared to color code qubits, the surface code qubits used in our scheme require more physical qubits ($\sim d^2$) for each logical qubit with code distance d , but – with the exception of twist defects – only require the measurement of weight-four stabilizers. Our scheme can provide full 2D connectivity between logical qubits, where the degree of connectivity is governed by the separation of qubit blocks, and therefore by the space overhead. Together with magic state distillation, our scheme allows for fault-tolerant universal quantum computation.

One may be wondering whether there is still any ad-

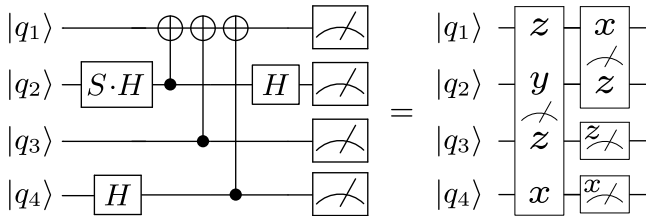


Figure 11: Example of a Clifford circuit that is reduced to Pauli product measurements.

vantage offered by the transversal single-qubit Clifford gates of color codes and the color-code-based lattice-surgery scheme presented in Ref. [23]. A comparison of these codes is shown in Tab. 1. While color codes require the measurement of higher-weight stabilizers, they offer fast qubit readout in all Pauli bases, and a lower space overhead of $\sim \frac{3}{4}d^2$ physical qubits per logical qubit for 6.6.6 color codes, or even $\sim \frac{1}{2}d^2$ for 4.8.8 color codes. So if the measurement of higher-weight stabilizers is not substantially more difficult in a given physical implementation, as might be the case for Majorana-based qubits, it is advantageous to use the color-code-based scheme. In other implementations, such as superconducting qubits, the difficulty of higher-weight stabilizer measurements shifts the preference towards surface-code-based architectures.

An important point is that the Gottesman-Knill theorem allows for the classical tracking of all Clifford gates, including CNOT gates. As CNOT gates map $X \otimes \mathbb{1} \rightarrow X \otimes X$ and $\mathbb{1} \otimes Z \rightarrow Z \otimes Z$, tracking of CNOTs does not preserve the locality of the logical operators, in

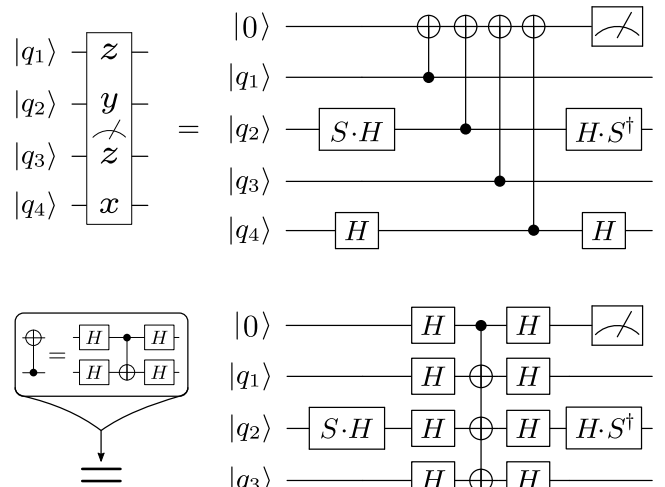


Figure 12: Circuit identity for the measurement of the Pauli product operator $Z \otimes Y \otimes Z \otimes X$ using an ancilla and a multi-target CNOT gate. The circuit identity exploits the fact that the roles of control and target can be reversed by the application of Hadamard gates before and after a CNOT gate. Any product of Pauli operators can be measured this way.

— semiconductor — topological superconductor
— superconductor — dot-mediated tunnel coupling

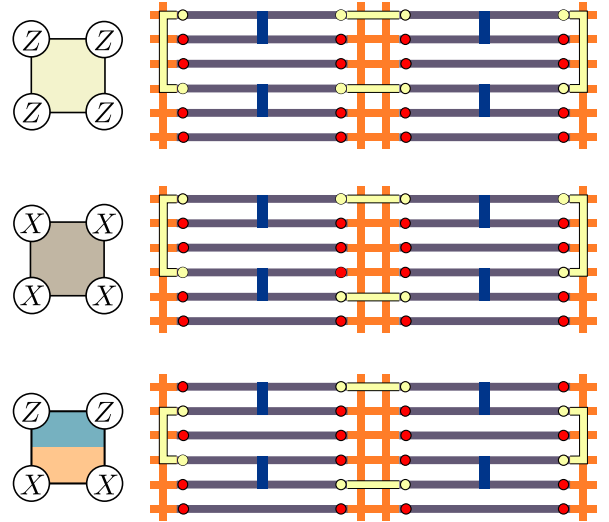
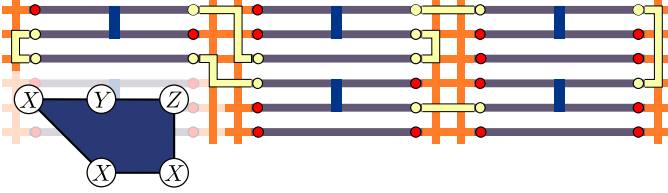
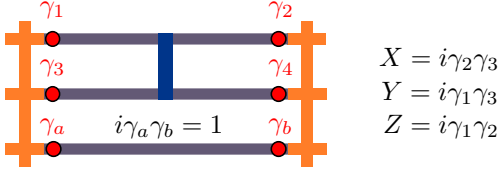


Figure 13: Tunnel coupling configurations for the measurement of various check operators using a square network of Majorana-based tetron qubits, as introduced in Ref. [21].

contrast to single-qubit Clifford gates. By tracking *all* Clifford gates, any layer of Clifford gates followed by n single-qubit measurements can always be compressed to n measurements of *nonlocal* products of Pauli operators without any preceding gate operations (see Fig. 11 for an example). With distilled magic states as a resource, any non-Clifford gate corresponds to trackable Clifford gates and a measurement of the magic state. In this case, Pauli product measurements are the only hardware operations that need to be performed explicitly. The fault-tolerant measurement of any nonlocal Pauli product can be implemented using an ancilla qubit and a multi-target CNOT gate on edge-tracked qubits. An example of such a protocol is shown in Fig. 12 for the measurement of the Pauli product $Z_L \otimes Y_L \otimes Z_L \otimes X_L$. Thus, any quantum computation can be performed using only two types of hardware operations: distillation of resource states and Pauli product measurements via multi-target CNOT gates on edge-tracked qubits.

A crucial problem of quantum information theory is the optimization of quantum circuits in order to minimize the space-time overhead of any quantum computation. However, any circuit optimization depends on the constraints set by the quantum computer hardware and the code used for error correction. Based on the existing schemes for surface-code and color-code quantum computation, the following minimal assumptions concerning the underlying hardware and the logical operations accessible by the code appear reasonable: (i) The underlying hardware can measure local products of physical Pauli operators. (ii) The quantum error-correcting code allows for the measurement of nonlocal products of logical Pauli operators. (iii) Resource states can be

generated for the implementation of logical non-Clifford gates. Based on these constraints, an important circuit optimization problem is to find heuristics that minimize the number of required resource states and the number of layers of Pauli product measurements, as these are the only operations that cannot be relegated to a classical computer.

Open questions related to our surface-code scheme include the efficient decoding of wide, long and double-sided qubits, estimations of their survival times, and implementations of our scheme in a concrete physical system. Our scheme may also be adapted to surface-code quantum computing with twist-based triangle codes [26], in order to avoid the reorientation of triangles, and to further reduce the space overhead of surface codes. We hope that our lattice-surgery-based approach can contribute to ongoing efforts to realize a surface-code quantum computer.

Acknowledgments

We thank Benjamin J. Brown, Jens Eisert, Markus S. Kesselring, and Fernando Pastawski for insightful discussions. This work has been supported by the Deutsche Forschungsgemeinschaft (Bonn) within the network CRC TR 183.

A Stabilizer measurements in concrete implementations

Our twist-based surgery scheme requires the measurement of certain operators that are products of Pauli op-

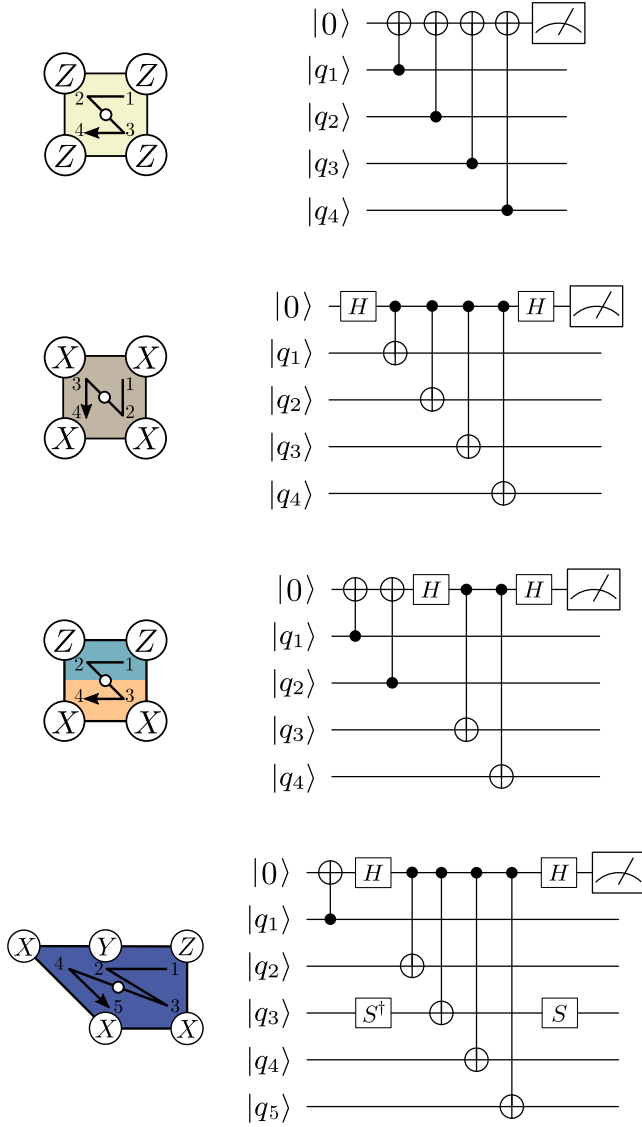


Figure 14: Circuits for the stabilizer readout using an ancillary measurement qubit placed in the center of the stabilizer.

erators on up to 5 qubits. How these operators are measured in practice depends on the concrete hardware used for quantum computing. In this appendix, we show how these measurements could in principle be implemented with Majorana-based qubits, and with non-topological qubits such as superconducting qubits that require the use of ancillary measurement qubits for stabilizer readout.

A.1 Majorana-based qubits

The primary operation of Majorana-based qubits is the measurement of local products of Majorana operators, which correspond to local products of Pauli operators. Thus, they can be straightforwardly used to measure

the stabilizers in our twist-based surgery scheme. In Fig. 13, we show how this can be done in a network of tetron qubits introduced in Ref. [21]. In a nutshell, these are qubits that are encoded in the doubly degenerate ground-state space of four Majorana zero modes $\gamma_1 \dots \gamma_4$ that are localized at the ends of two topological superconducting nanowires which are put into a fixed parity sector ($\gamma_1 \gamma_2 \gamma_3 \gamma_4 = -1$) by a non-topological superconductor bridging the two wires. The Majorana operators are self-conjugate $\gamma = \gamma^\dagger$ and mutually anticommute $\{\gamma_i, \gamma_j\} = 2\delta_{i,j}$. Therefore, the Pauli operators of each tetron qubit can be chosen as $Z = i\gamma_1 \gamma_2$ and $X = i\gamma_2 \gamma_3$.

In a square lattice of tetrons, each tetron qubit is connected to a network of semiconductors. Local products of Majorana operators are measured by opening tunnel couplings between tetrons and the semiconductor network, such that the tunnel couplings form closed paths. The semiconducting wire segments between tetrons form quantum dots whose energy levels are shifted by virtual processes that tunnel electrons around the closed path. Since these processes involve each Majorana operator along the path exactly once, spectroscopy on any of the dots can be used to measure the product of the Majorana operators along the path. In Fig. 13, we show tunnel coupling configurations that can be used to measure X and Z stabilizers, dislocation operators, and twist operators. For the twist operator, in particular, additional Majoranas γ_a and γ_b in a fixed parity sector $i\gamma_a \gamma_b = 1$ are used as so-called coherent links in order to form the closed path. More details on operator measurements in tetron networks are found in Refs. [21, 23].

A.2 Non-topological qubits

For non-topological qubits such as superconducting qubits, Pauli products cannot be measured directly, but are usually read out using ancilla qubits (measurements qubits) that are located in the center of each stabilizer operator, such as in the scheme of Ref. [14]. These measurement qubits are entangled via two-qubit gates with each data qubit that is part of the stabilizer. Afterwards, they are read out to yield the corresponding Pauli product. The readout can be done using the circuits shown in Fig. 14. Depending on the elementary operations accessible in a given hardware, a different (but equivalent) circuit may be used, but in any case the readout of each stabilizer requires up to 5 two-qubit gates which need to be performed in succession.

One practical problem of this approach to stabilizer measurements is that due to the use of two-qubit gates, single errors on measurement qubits can spread to multiple data qubits. This can lead to correlated errors

which are referred to as hook errors [25]. In particular, Z errors on measurement qubits of Z stabilizers can lead to correlated Z errors on the surrounding data qubits. Similarly, X errors on X measurement qubits lead to correlated X errors. Since three errors are equivalent to just one error (up to a multiplication with a stabilizer), the worst case is the case of one error on a measurement qubit leading to two errors on data qubits. Since these errors will occur on the first two (or last two) qubits that were part of entangling two-qubit gates, the order of the two-qubit gates in the circuits of Fig. 14 is important.

The aim is to avoid these correlated errors from lowering the effective code distance. That is, we need to find an ordering of the two-qubit gates, such that a logical operator of a distance d qubit can only be formed by no fewer than d errors. For square surface code patches (as in Fig. 1), this is done by orienting the ordering of CNOT gates for X stabilizers in an N shape (or in a ∇ shape), and for Z stabilizers in a Z shape (or Σ shape), as was shown in Ref. [27]. This guarantees that correlated Z errors only form in the horizontal direction, while logical Z strings are all oriented vertically. Similarly, correlated X errors form horizontally, which does not contribute towards a vertical logical X string.

However, in our scheme, we use the double-sided qubits of Fig. 9, which have Z and X operators in both the horizontal and vertical direction. Thus, it is not sufficient to assign one orientation to the Z stabilizers and the other to X stabilizers. In fact, the left half of the double-sided qubit looks like the square qubit in Fig. 1, i.e., Z stabilizers should be oriented in a Z shape, and X stabilizers in an N shape. In contrast, the right half of the double-sided qubit looks like a rotated square qubit, i.e., Z stabilizer should be oriented in an N shape, and X stabilizers in a Z shape. In the crossover region in the center, both logical X and Z operators are vertical strings, such that both Z and X stabilizers should be oriented in a Z shape. This motivates the first condition for a valid ordering shown in Fig. 15a. Z stabilizers in the blue region should be oriented in a Z shape, and in an N shape outside of the blue region. X stabilizers in the red region should be oriented in a Z shape, and in an N shape outside of the red region. One can verify that with this choice of orientations, no logical operator string can be formed with fewer than d physical errors.

The only remaining problem is the scheduling of the two-qubit gates. Since the largest check operator is the 5-qubit twist defect, each two-qubit gate needs to be assigned to one of 5 time steps. This implies two other conditions on the ordering of the two-qubit gates. Since each data qubit can only be part of one two-qubit gate in a given time step, the four time steps assigned to the two-qubit gates that a given data qubit is part of

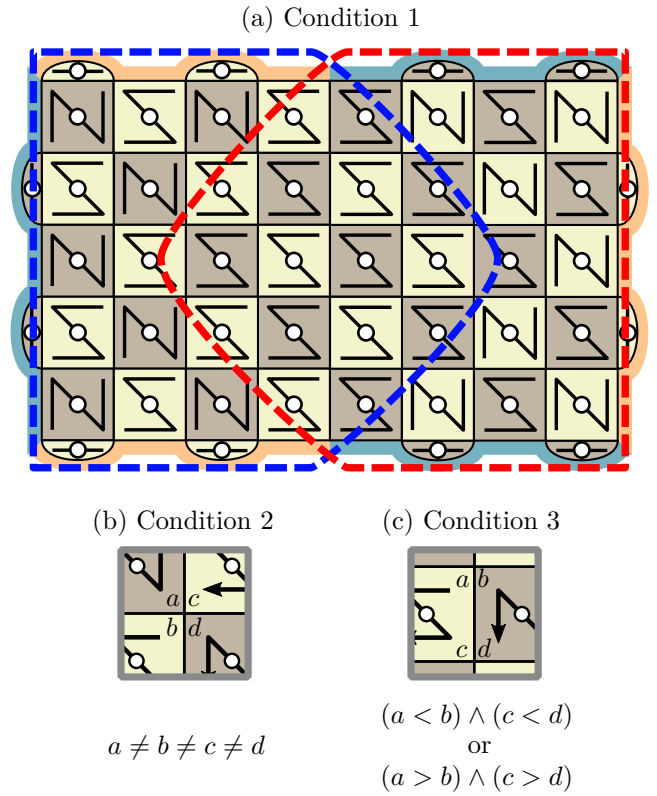


Figure 15: Three conditions for a valid ordering of two-qubit gates during stabilizer readout. (a) In the blue region, the Z stabilizers are oriented in a Z shape, and outside of this region in an N shape. In the red region, the X stabilizers are oriented in a Z shape, and outside of this region in an N shape. This ensures that hook errors do not lower the effective code distance. (b) The time steps a , b , c , and d , that are assigned to the (up to) four two-qubit gates of one data qubits need to be all different. (c) For all edges between neighboring stabilizers, the condition shown in the figure needs to be fulfilled to ensure that the sequence of two-qubit measurements reproduces the desired stabilizer measurements.

need to be all different, which is the second condition in Fig. 15b.

Finally, one needs to ensure that the sequence of two-qubit measurements reproduces the desired stabilizer measurements. For this, consider the action of the CNOT gates of the readout circuit for a Z stabilizer in the Heisenberg picture (or see Appendix B of Ref. [14]). The aim of the readout circuit is to map the Z operator of the measurement qubit onto the operator $Z^{\otimes 5}$ on the measurement qubit and all four data qubits. Since a CNOT maps $\mathbb{1} \otimes Z$ onto $Z \otimes Z$ (where the first qubit is the control and the second is the target), the circuit in Fig. 14 achieves exactly that. However, notice that two CNOT gates of the readout circuit of a neighboring X stabilizer also act on the Z operators of the data qubits. These may map the operator onto an operator

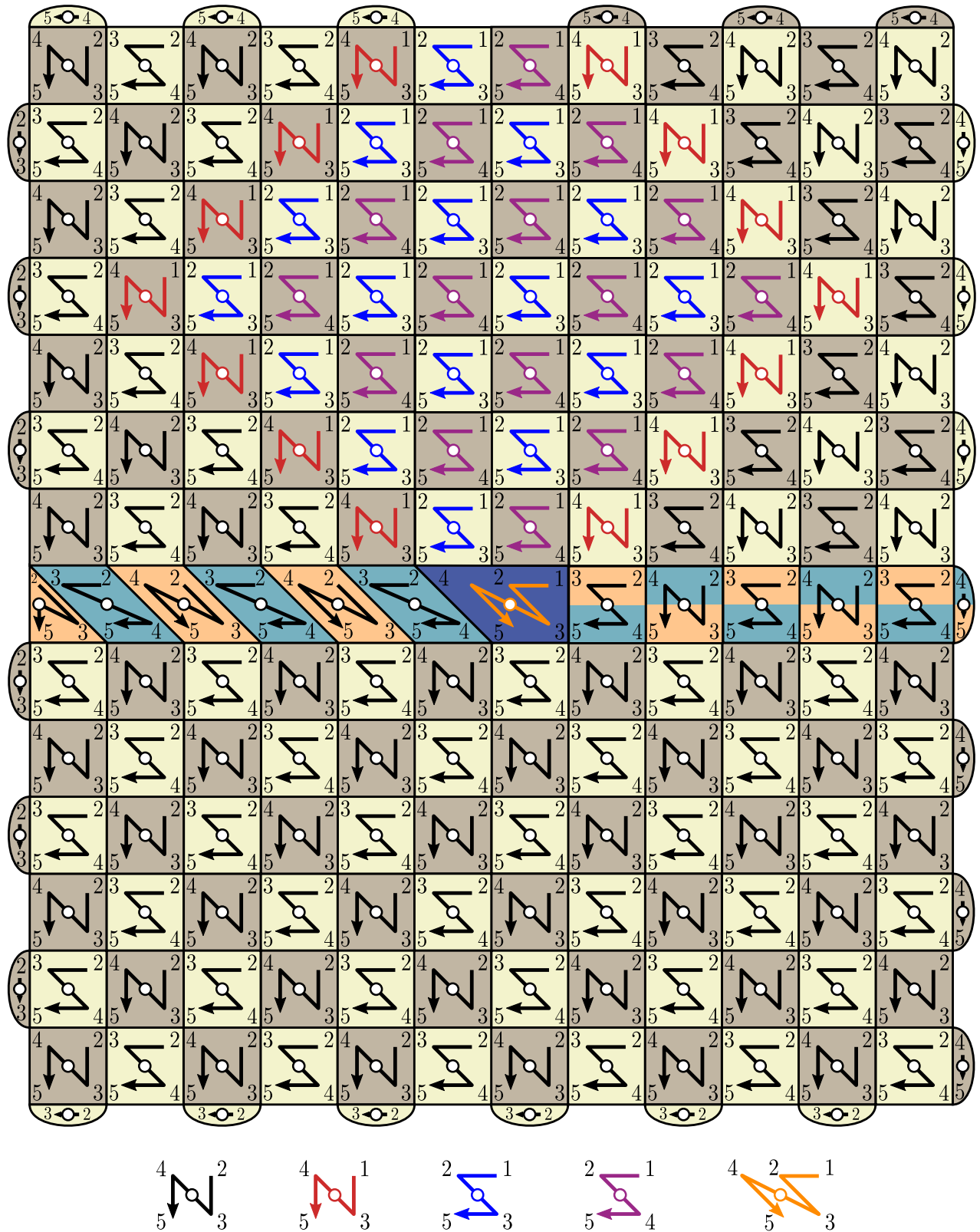


Figure 16: Possible ordering of two-qubit gates that fulfills the three conditions shown in Fig. 15. This stabilizer configuration is the most general, as it involves bulk stabilizers, a dislocation line, and a twist defect. It corresponds to a twist-based lattice surgery between a double-sided qubit and a standard rectangular qubit.

that involves the Z operator of the wrong measurement qubit, i.e., the measurement qubit of the neighboring stabilizer. This needs to be avoided, as this neighboring measurement qubit is measured in X , which anti-commutes with the aforementioned operator, leading to random measurement outcomes. For concreteness, let us refer to the time steps assigned to the two CNOTs of the Z stabilizer in question as a and c , and to the CNOTs of the X stabilizer as b and d , as in Fig. 15c. There are only two choices of time steps that map the Z operators of the measurement qubit to the correct $Z^{\otimes 5}$ operator. The first possibility is that $a > b$ and $c > d$, such that the two CNOTs of the neighboring X stabilizer have already been performed, which precludes them from acting on the Z operator of the data qubit. The second possibility is that $a < b$ and $c < d$, such that the mapping is performed twice: The first CNOT maps $\mathbb{1} \otimes Z$ to $Z \otimes Z$, and the second CNOT maps $Z \otimes Z$ back to $\mathbb{1} \otimes Z$. This is the third condition shown in Fig. 15c. It needs to hold for all edges between neighboring stabilizers.

It is possible to find an ordering of two-qubit gates in 5 time steps (due to the 5-qubit twist operators) that fulfills all three conditions. Such a possibility is shown in Fig. 16. The figure shows the most generic situation which involves the bulk stabilizers of standard and double-sided qubits, as well as a dislocation line and a twist defect.

B Magic state distillation protocol

Here, we explicitly show the lattice surgery protocols for the multi-target CNOTs part of the 15-to-1 magic state distillation scheme in Fig. 10. Figures 17 and 18 show the five multi-target CNOTs of the distillation protocol, where the control and target qubits are highlighted in blue and orange, respectively. Note that the default encodings of the X and Z edges of qubits 5, 9 and 11 are inverted in this protocol. The figures only show the $Z_L \otimes Z_L$ parity measurements. The subsequent $X_L \otimes X_L$ parity measurements are done via lattices surgeries between the highlighted orange edges and the adjacent ancilla qubits.

References

- [1] J. Preskill, *Reliable quantum computers*, *Proc. Roy. Soc. Lond. A* **454**, 385 (1998).
- [2] A. Kitaev, *Fault-tolerant quantum computation by anyons*, *Ann. Phys.* **303**, 2 (2003).
- [3] B. M. Terhal, *Quantum error correction for quantum memories*, *Rev. Mod. Phys.* **87**, 307 (2015).
- [4] M. H. Devoret and R. J. Schoelkopf, *Superconducting circuits for quantum information: An outlook*, *Science* **339**, 1169 (2013).
- [5] D. Loss and D. P. DiVincenzo, *Quantum computation with quantum dots*, *Phys. Rev. A* **57**, 120 (1998).
- [6] R. M. Lutchyn, E. P. A. M. Bakkers, L. P. Kouwenhoven, P. Krogstrup, C. M. Marcus, and Y. Oreg, *Realizing Majorana zero modes in superconductor-semiconductor heterostructures*, [arXiv:1707.04899](https://arxiv.org/abs/1707.04899) (2017).
- [7] D. Gottesman, *Stabilizer codes and quantum error correction*, *Ph.D. thesis*, California Institute of Technology (1997).
- [8] S. B. Bravyi and A. Y. Kitaev, *Quantum codes on a lattice with boundary*, [arXiv:quant-ph/9811052](https://arxiv.org/abs/quant-ph/9811052) (1998).
- [9] E. T. Campbell, B. M. Terhal, and C. Vuillot, *Roads towards fault-tolerant universal quantum computation*, *Nature* **549**, 172 (2017).
- [10] D. S. Wang, A. G. Fowler, A. M. Stephens, and L. C. L. Hollenberg, *Threshold error rates for the toric and planar codes*, *Quantum Info. Comput.* **10**, 456 (2010).
- [11] R. S. Andrist, H. G. Katzgraber, H. Bombin, and M. A. Martin-Delgado, *Error tolerance of topological codes with independent bit-flip and measurement errors*, *Phys. Rev. A* **94**, 012318 (2016).
- [12] H. Bombin and M. A. Martin-Delgado, *Topological quantum distillation*, *Phys. Rev. Lett.* **97**, 180501 (2006).
- [13] A. J. Landahl, J. T. Anderson, and P. R. Rice, *Fault-tolerant quantum computing with color codes*, [arXiv:1108.5738](https://arxiv.org/abs/1108.5738) (2011).
- [14] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, *Surface codes: Towards practical large-scale quantum computation*, *Phys. Rev. A* **86**, 032324 (2012).
- [15] H. Bombin, *Topological order with a twist: Ising anyons from an abelian model*, *Phys. Rev. Lett.* **105**, 030403 (2010).
- [16] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, *Poking holes and cutting corners to achieve Clifford gates with the surface code*, *Phys. Rev. X* **7**, 021029 (2017).
- [17] M. B. Hastings and A. Geller, *Reduced space-time and time costs using dislocation codes and arbitrary ancillas*, *Quantum Info. Comput.* **15**, 962 (2015).
- [18] D. Gottesman, *The Heisenberg representation of quantum computers*, *Proc. XXII Int. Coll. Group. Th. Meth. Phys.* **1**, 32 (1999).
- [19] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, *Surface code quantum computing by lattice surgery*, *New J. Phys.* **14**, 123011 (2012).

- [20] S. Bravyi and A. Kitaev, *Universal quantum computation with ideal Clifford gates and noisy ancillas*, *Phys. Rev. A* **71**, 022316 (2005).
- [21] T. Karzig, C. Knapp, R. M. Lutchyn, P. Bonderson, M. B. Hastings, C. Nayak, J. Alicea, K. Flensberg, S. Plugge, Y. Oreg, C. M. Marcus, and M. H. Freedman, *Scalable designs for quasiparticle-poisoning-protected topological quantum computation with Majorana zero modes*, *Phys. Rev. B* **95**, 235305 (2017).
- [22] D. Litinski, M. S. Kesselring, J. Eisert, and F. von Oppen, *Combining topological hardware and topological software: Color-code quantum computing with topological superconductor networks*, *Phys. Rev. X* **7**, 031048 (2017).
- [23] D. Litinski and F. von Oppen, *Braiding by Majorana tracking and long-range CNOT gates with color codes*, *Phys. Rev. B* **96**, 205413 (2017).
- [24] G. Duclos-Cianci and D. Poulin, *Fast decoders for topological quantum codes*, *Phys. Rev. Lett.* **104**, 050504 (2010).
- [25] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, *Topological quantum memory*, *Journal of Mathematical Physics* **43**, 4452 (2002).
- [26] T. J. Yoder and I. H. Kim, *The surface code with a twist*, *Quantum* **1**, 2 (2017).
- [27] Y. Tomita and K. M. Svore, *Low-distance surface codes under realistic quantum noise*, *Phys. Rev. A* **90**, 062320 (2014).

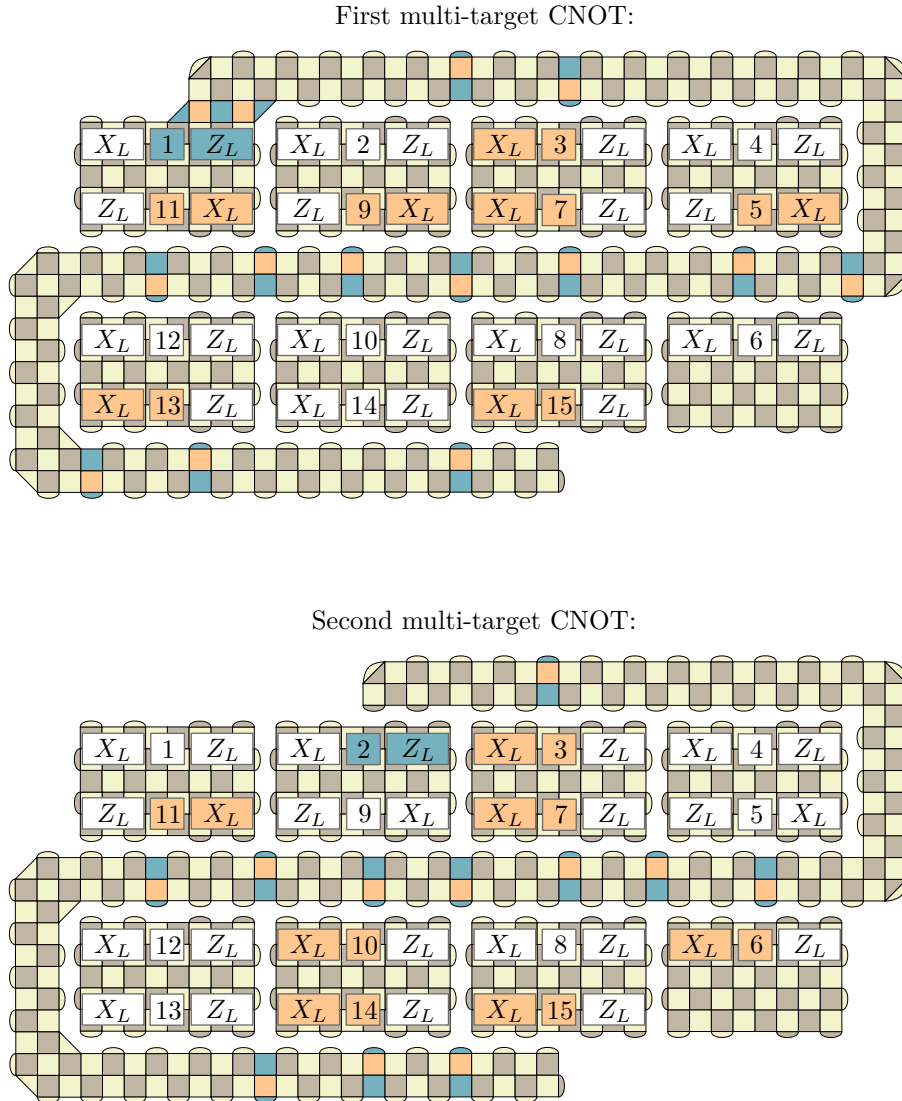


Figure 17: First and second multi-target CNOT of the distillation protocol in Fig. 10.

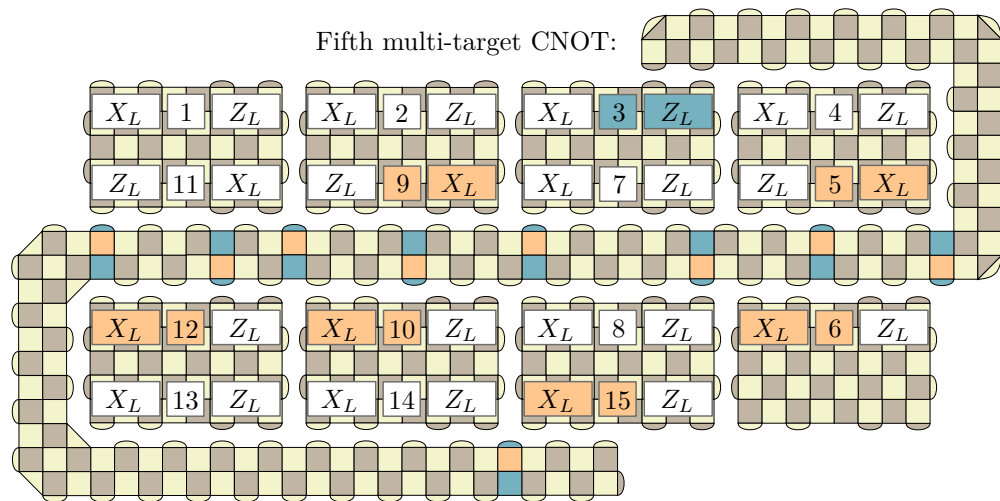
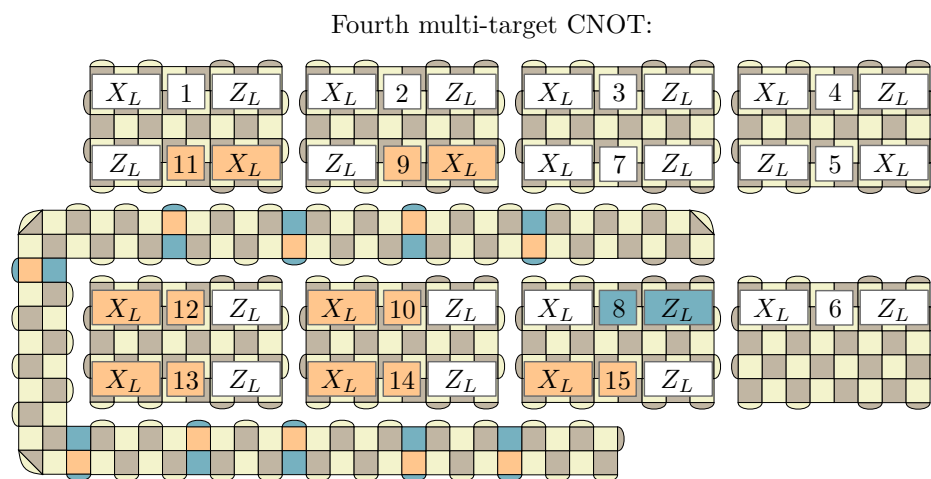
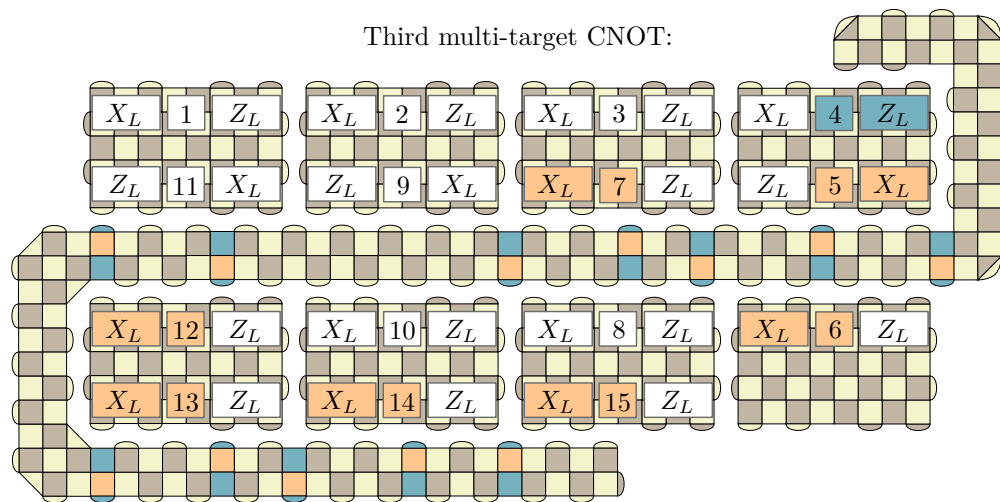


Figure 18: Third, fourth and fifth multi-target CNOT of the distillation protocol in Fig. 10.