Circuit-level protocol and analysis for twist-based lattice surgery

Christopher Chamberland [1,2,*] and Earl T. Campbell [3,†]
[1]*AWS Center for Quantum Computing, Pasadena, California 91125, USA*
[2]*IQIM, California Institute of Technology, Pasadena, California 91125, USA*
[3]*AWS Center for Quantum Computing, Cambridge CB1 2GA, United Kingdom*

(Received 28 January 2022; accepted 13 April 2022; published 2 May 2022)

Lattice surgery is a measurement-based technique for performing fault-tolerant quantum computation in two dimensions. When using the surface code, the most general lattice surgery operations require lattice irregularities called twist defects. However, implementing twist-based lattice surgery may require additional resources, such as extra device connectivity, and could lower the threshold and overall performance for the surface code. Here we provide an explicit twist-based lattice surgery protocol and its requisite connectivity layout. We also provide new stabilizer measurement circuits for measuring twist defects which are compatible with our chosen gate scheduling. We undertake the first circuit-level error correction simulations during twist-based lattice surgery using a biased depolarizing noise model. Our results indicate a slight decrease in the threshold for timelike logical failures compared to lattice surgery protocols with no twist defects in the bulk. However, comfortably below threshold (i.e., with CNOT infidelities below 5×10^{-3}), the performance degradation is mild and in fact preferable over proposed alternative twist-free schemes. Lastly, we provide an efficient scheme for measuring Y operators along boundaries of surface codes, which bypasses certain steps that were required in previous schemes.

DOI: 10.1103/PhysRevResearch.4.023090

I. INTRODUCTION

The surface code is a quantum error correcting code with a high threshold and which can be realized with two-dimensional hardware [1,2]. Other codes have been observed to have comparable thresholds for toy noise models [3–8], but when considering realistic circuit-level noise their performance (relative to the surface code) drops appreciably. The surface code's high threshold can be partially attributed to the simplicity of its stabilizers and the circuits used to measure them. Realizing the surface code requires measurements of pure X-type and Z-type stabilizers involving no more than four qubits. These measurements can be performed using a depth-four sequence of two-qubit gates; the minimum we can hope for. Furthermore, choosing the right gate schedule can make the stabilizer measurement circuits immune to "hook" errors that effectively reduce the code distance. The moral here is that careful, circuit-level analysis of the microscopic details is crucial for a full understanding of the best route to fault-tolerant quantum computing.

Logical operations in two-dimensional surface code architectures can be performed by code deformation where the code stabilizers change over time, such as through braiding [2,9] or lattice surgery [10–12], with the latter being more resource efficient [13,14]. The basic primitive in lattice surgery is a logical multiqubit Pauli measurement, which when combined with logical ancilla states can be used to fault tolerantly perform logical Clifford operations in addition to logical non-Clifford gates by teleportation. When measuring logical Paulis composed of X and Z operators, the code is deformed with several surface code patches merged into a large patch, though locally the code stabilizers are unchanged. However, more generally, measurements involving Pauli Y operators are needed. Such measurements can be realized with twist-based lattice surgery [11], though this requires some weight-five measurements in addition to long-range multiqubit gates. These proposed twist-based approaches were presented abstractly, ignoring circuit-level implementation details and any related performance impacts.

Our work gives the first proposal for a circuit-level implementation of twist-based lattice surgery and numerically benchmarks the performance. In twist-based lattice surgery, we need to measure some longer range stabilizers that we call elongated rectangles and twist defects. To perform these measurements without increasing the circuit-depth (which would significantly degrade performance), we assume the hardware connectivity illustrated in Fig. 1. We say two qubits are connected when a two-qubit gate can be performed between these qubits. Standard surface code implementations have each qubit connected to four other qubits. In this work, we assume a minimal extension where some qubits are connected to eight other qubits. Flux-tunable transmon qubits [15–18] offer some flexibility in the connectivity degree constraints between qubits. As such, such architectures may be able to support the degree-eight connectivity requirements considered in our work. Cat-qubit based architectures, such as those considered in Ref. [19] appear to be constrained by cross-talk noise when

*cchmber@amazon.com
†earltcampbell@gmail.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.

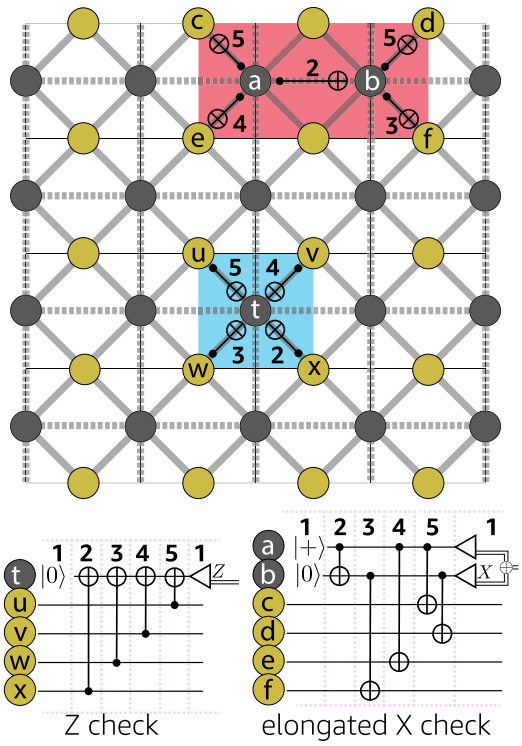

FIG. 1. Connectivity diagram between data qubits (yellow) and ancilla qubits (grey) with two example gate schedules. Thick, solid grey lines show the connectivity needed to perform a standard weight-four check as shown here for a blue $Z^{\otimes 4}$ stabilizer. To perform a check along an elongated rectangle (such as the red check shown in the figure) additional connectivity can be used which we illustrate as dashed grey lines. For the elongated $X^{\otimes 4}$ check, the two grey ancilla qubits are prepared in a GHZ state which is used to perform the desired stabilizer measurement. Examples of gate schedulings and circuits used to measure stabilizers are shown (numbers indicate the time step for each two-qubit gate) and further circuit details are given in Appendix A. The first and last time steps are used for state preparation and measurement and so are both labeled as time step 1. In subsequent figures, we will sometimes just provide numerical labels when the type of gate is clear from the context.

the connectivity degree exceeds six. However, more work in this direction, along with Kerr-cat qubit architectures [20,21] might be able to support degree-eight connectivity constraints. Yoder and Kim proposed a twist-based scheme with lower weight measurements [22], but they required a substantial change to the underlying surface code [22] and did not use a homogeneous (translationally invariant) hardware. We leave open the question of whether twist-based lattice surgery can be realized using less connectivity within a homogeneous two-dimensional architecture.

To benchmark twist-based lattice surgery, we perform Monte Carlo simulations with a minimum-weight perfect matching (MWPM) decoding protocol (following the method of Ref. [23]) of $Y \otimes Y$ and $Z \otimes Z$ measurements. We perform our simulations for biased noise and rectangular surface codes since this enables us to perform larger simulations and these noise models are of recent interest [19,21,24–27]. We observe that the measurement of a logical $Y \otimes Y$ operator has a higher logical failure rate relative to a comparable $Z \otimes Z$ measurement of the same area. We attribute such differences to the presence of twist defects and elongated checks. The logical failure probability of these measurements is exponentially suppressed in the number of stabilizer measurement rounds used. Therefore, we can ask how many more rounds (and therefore time cost) is needed for a $Y \otimes Y$ measurement to achieve the same logical fidelity as a $Z \otimes Z$ measurement. We find, well below threshold ($p_{\mathrm{CNOT}} \leqslant 1\%$), the use of twists incurs only a small multiplicative time cost (less than $1.2\times$ for one of the studied noise models when $p_{\mathrm{CNOT}} = 10^{-3}$) since the twists are a small fraction of all possible fault locations. However, the twist defects are the weakest point and thus fail with a higher probability compared to other fault locations. Consequently, we find these multiplicative time costs increase as we approach the threshold.

We remark that for depolarizing noise models, or noise models with a smaller bias between phase-flip and bit-flip errors, the routing space area used to perform lattice surgery could be larger than what we obtain using our layout (as is the case in Ref. [14]). Furthermore, since Z and Y Pauli errors (in addition to measurement errors) can result in timelike failures during $Y \otimes Y$ and $Z \otimes Z$ measurements, an unbiased noise model would also increase timelike failure rates. As such, we expect that logical failure rates of $Y \otimes Y$ and $Z \otimes Z$ measurements performed via lattice surgery to increase in settings with a smaller noise bias relative to the results obtained in our work.

In previous work [23], we proposed an alternative twist-free lattice surgery scheme that replicated the computational power of Pauli Y measurements without the need for twist defects. However, this alternative strategy came with its own $\sim 2\times$ time cost. Therefore, a key conclusion of this work is that twist-based approaches outperform twist-free alternatives when operating comfortably below threshold, and under our connectivity assumptions along with the biased noise model used for the simulations. Twist-free lattice surgery may still prove useful in connectivity limited hardware or very close to threshold. Hardware afflicted by a different noise model than considered in this work might also benefit from twist-free lattice surgery.

II. LATTICE SURGERY WITH TWIST DEFECTS IN THE BULK

In this section, we describe our protocol for performing twist-based lattice surgery. In Sec. II A, we provide a gate scheduling that can be used to measure an arbitrary number of Pauli Y operators using twist defects in the bulk. Fault-tolerant circuits for measuring stabilizers at twist defects are provided in Appendix A. In Sec. II B, we will then compare the performance of a lattice surgery protocol for measuring $P = Y \otimes Y$ to a $P = Z \otimes Z$ protocol that does not require twist defects in the bulk. The Pauli $Z \otimes Z$ is chosen for comparison since the measurement of both $Y \otimes Y$ and $Z \otimes Z$ require domain walls between the logical patches and the routing space. In what follows, the surface code encoding a logical qubit will be referred to as a logical patch. The extra space required to perform lattice surgery will be referred to as the routing space.

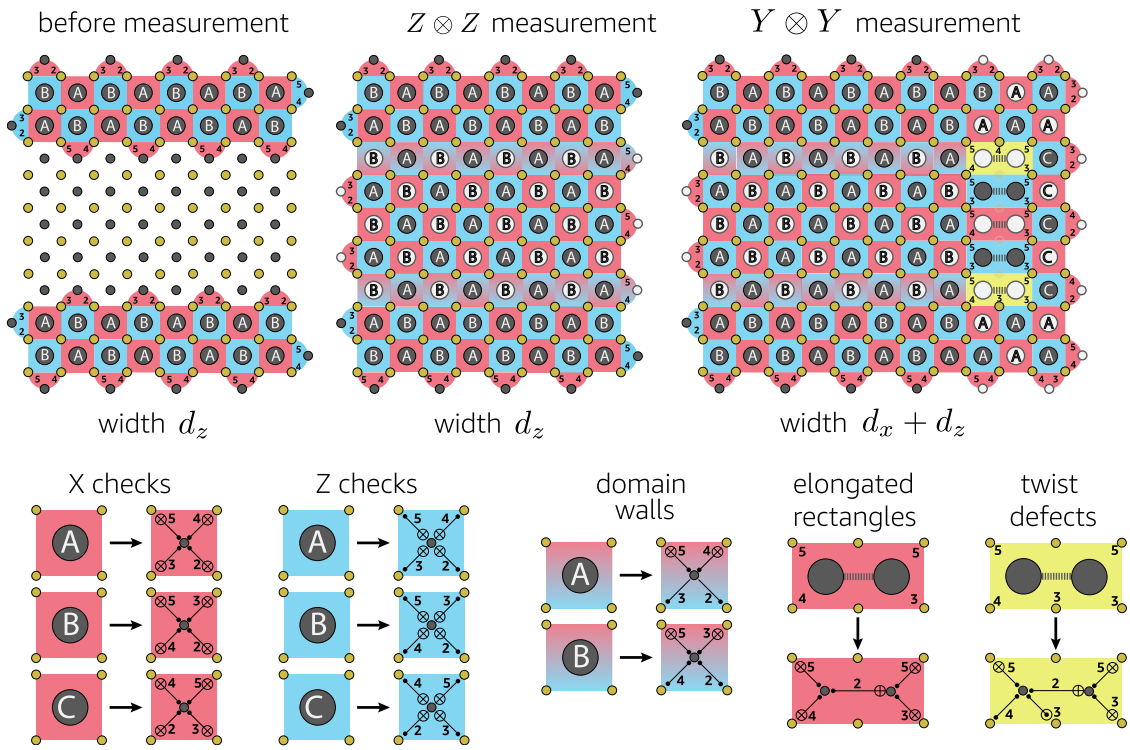

FIG. 2. Illustration of a $Z \otimes Z$ and $Y \otimes Y$ measurement performed by lattice surgery. The left most figure shows two $d_x = 3$ and $d_z = 9$ logical patches prior to the measurements. The qubits (yellow and grey vertices) in between the two patches are part of the routing space. The middle figure shows the measurement of $Z \otimes Z$ by gauge fixing, i.e., measuring the appropriate stabilizers in the routing space such that a $Z \otimes Z$ measurement is performed along the logical Z boundaries of the surface code patches. Red plaquettes correspond to X-type stabilizers, blue plaquettes to Z-type stabilizers. Gradient color plaquettes correspond to domain walls that measure mixed X- and Z-type stabilizers. The labels A, B and C describe particular gate schedulings. Stabilizers with white vertices encode the parity of the logical $Z \otimes Z$ measurement. The rightmost figure illustrates a logical $Y \otimes Y$ measurement implemented by lattice surgery. As explained in Appendix C, the $Y \otimes Y$ measurement can be done directly without first extending the logical patches so that logical Y operators can be expressed along their respective horizontal boundaries. When performing the gauge fixing step, weight-five stabilizers illustrated by yellow plaquettes (referred to as twist defects) must be measured, in addition to elongated X and Z-type stabilizers. The measurements of both twist defects and elongated stabilizers require the use two ancilla qubits prepared in a GHZ state with the assumed hardware connectivity constraints described in Sec. I. More details of the stabilizer measurement circuits for elongated X and Z stabilizers and twist defects can be obtained from Fig. 7(a). Lastly, we remark that in addition to the A and B gate schedules, the $Y \otimes Y$ also requires the use of the C schedule.

A. Gate scheduling for twist-based lattice surgery

Recall that in a lattice surgery protocol, multiqubit Pauli measurements between logical patches are performed by measuring a set of operators in the routing space between the logical patches. Such measurements (which correspond to gauge fixing [12]) merge the logical patches into one large surface code patch, and a subset of the stabilizers of the merged patch encode the parity of the logical multiqubit Pauli measurement. A simple example for a $Z \otimes Z$ and $Y \otimes Y$ measurement is provided in Fig. 2.

Let $P = P_1 \otimes P_2 \otimes \cdots \otimes P_k$ be a multiqubit Pauli operator that is to be measured using a lattice surgery protocol. If $P_j \in \{X, Z\}$ for all $j \in \{1, \ldots, k\}$, then the standard surface code gate scheduling can be used to measure all the stabilizers during the lattice surgery protocol [28]. For instance, see the scheduling used for a logical $Z \otimes Z$ measurement performed by lattice surgery in Fig. 2. With such a scheduling, all two-qubit gates can be applied in four time steps thus minimizing the number of idling qubit locations. One of the important features of the standard surface code gate scheduling circuits is that a weight-two error arising from a single fault will be perpendicular to the relevant logical operator thus preserving the surface code distance (i.e., the code will be guaranteed to correct any error arising from at most $(d-1)/2$ faults). By relevant logical operator, we mean that if the weight-two data-qubit error arising from a single fault is of the form $E_X = X_{q_1} \otimes X_{q_2}$ and the minimum-weight representatives of X_L form vertical strings, then E_X will have support along horizontal strings. Similarly, E_Z will have support along vertical strings since the minimum-weight representatives of Z_L form horizontal strings.

Now suppose there exists at least one $j \in \{1, \ldots, k\}$ such that $P_j = Y$. In Fig. 2(d) of Ref [14], Litinski proposed measuring the Y operators as follows. First, logical patches are extended using qubits in the routing space such that the logical Y_L operator of the logical patch can be expressed along a horizontal boundary. The logical Y operator is then given by

$$Y = Z_{q_1} \otimes Z_{q_2} \otimes \cdots \otimes Z_{q_{d_z-1}} \otimes Y_{q_{d_z}} \otimes X_{q_{d_z+1}} \otimes \cdots \otimes X_{q_{d_z+d_x}}, \tag{1}$$

where d_x and d_z are the minimum weights of the logical X and Z operators of the logical patch. The Y operator can then be measured via gauge fixing by performing Z- and X-type stabilizer measurements between the extended logical patch and qubits in the routing space. The gauge-fixing step requires the measurement of mixed $X \otimes X \otimes Z \otimes Z$ stabilizers which we refer to as domain walls. In addition, it requires the measurement of a weight-five operator operator containing a physical Y term. Such weight-five operators are referred to as twist defects. An important feature of our scheme is that when measuring multiqubit Pauli's involving Y operators, such measurements can be done *directly*, without first extending the logical patches. Using an extension step doubles the implementation time and uses additional routing space, thus making our direct approach more efficient. Further, we show in Appendix C that our direct approach for measuring Pauli Y operators is fault-tolerant using the gauge fixing formalism of Ref. [12].

The presence of twist defects reduces the set of gate scheduling solutions that can be used for a twist-based lattice surgery protocol with minimal circuit depth. A valid gate scheduling for the measurement of $P = Y \otimes Y$ via lattice surgery is shown in Fig. 2. The chosen gate scheduling ensures that all two-qubit gates can be applied in four time steps. Three different gate schedules are used, which are labeled A, B, and C. The C schedule is only used in the routing space to the right of the twist defects. Note that the A, B, and C schedules must follow a set of rules for how they can be tilled on the lattice in order to avoid scheduling conflicts or invalid schedules for performing the stabilizer measurements.[1] For instance, a stabilizer measured using an A schedule cannot be adjacent to a stabilizer measured using a C schedule in the horizontal direction. If they were, qubits shared by both stabilizers would interact with two-qubit gates in the same time steps. On the other hand, C and A schedules can be adjacent to each other in the vertical direction (see the legend at the bottom of Fig. 2).

Unlike the standard surface code gate scheduling where all X-type stabilizers use the A schedule and Z-type stabilizers use the B schedule (or vice versa), the gate scheduling for measuring $Y \otimes Y$ includes some regions where the A schedule is applied uniformly. Consequently, there will be single failures resulting in weight-two data qubit errors which are parallel to the logical X_L or Z_L operators of the merged patch. For a given quantum algorithm, the d_x and d_z distances of logical patches are chosen such that a single logical data qubit error is very unlikely during the course of the entire computation. It is important to note that the merged lattices of a twist-based protocol with distances d'_x and d'_z for measuring Y operators will have $d'_x > d_x$ and $d'_z > d_z$. In other words, the distances of the merged patches will be larger than the distances required for the logical patches prior to the merged. Therefore, even though our gate scheduling has the property that some weight-two errors arising from single failures will be parallel to X_L or Z_L logical operators, the increased effective distances during lattice surgery protocols will ensure that d_x and d_z distances of the unmerged logical patches remain unchanged when performing twist-based lattice surgery, which we observe numerically. There are, however, some subtleties when performing twist-based lattice surgery that affect the minimum required value of d_x (see Appendix B for details).

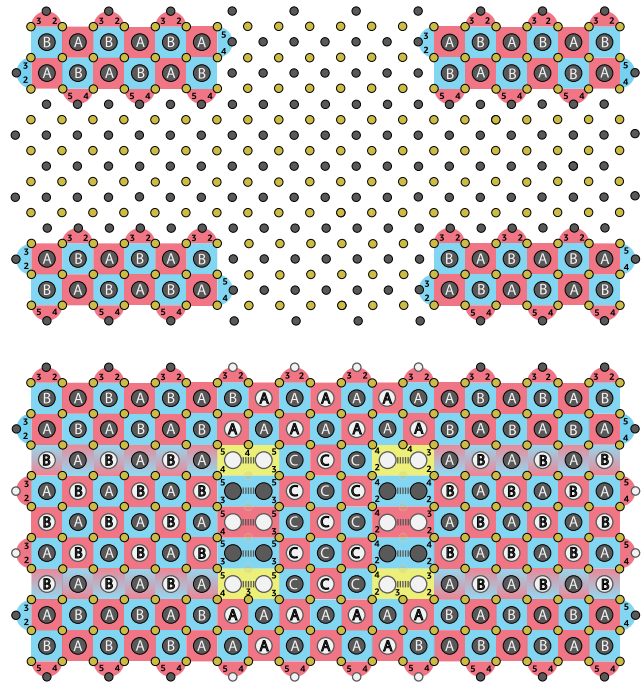

FIG. 3. Lattice and valid gate scheduling for measuring the multiqubit Pauli operator $P = Y \otimes Y \otimes Y \otimes Y$ via lattice surgery. The A, B, and C gate schedules are described in Fig. 2. The chosen gate scheduling imposes the need for different ancilla states used to measure twist defects on the left strip compared to twist defects on the right strip. More details are provided in Appendix A.

We now provide a generalization of the gate scheduling shown in Fig. 2 used for twist-based lattice surgery. In Fig. 3, we illustrate the gate scheduling required to measure the logical Pauli operator $P = Y \otimes Y \otimes Y \otimes Y$. The inclusion of a second pair of Y operators has the feature that all stabilizers of the merged patch in the routing space between the four twist defects are measured using the C schedule. As was the case in Fig. 2, the presence of twist defects requires the measurement of elongated stabilizers along vertical strips located between the twist defects in the routing space. In order to respect the nearest-neighbor connectivity constraints imposed by many quantum hardware architectures, additional ancilla qubits need to be used to measure the elongated stabilizers and weight-five operators. However, an important feature of the circuit in Fig. 3 is that the gate scheduling imposes the need for different syndrome measurement circuits for the twist defects on the left strip compared to those on the right strip. More details are provided in Appendix A with the circuits shown in Fig. 7.

We remark that the gate scheduling shown in Fig. 3 can be extended to measure arbitrary multiqubit Pauli operators

[1]A valid schedule is one where all stabilizers are guaranteed to be measured as $+1$ in the absence of faults. Further, if an error anticommutes with a stabilizer g, then g is guaranteed to be measured as -1.

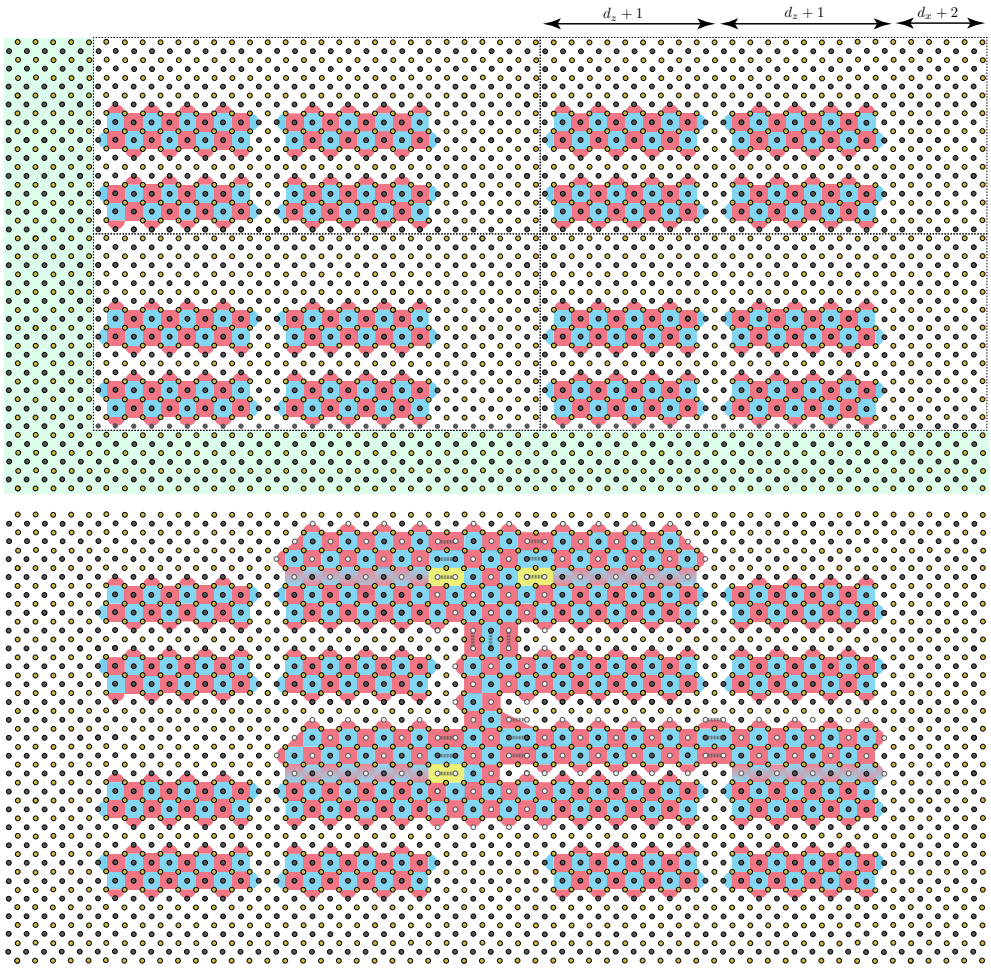

FIG. 4. Example layout of 16 logical patches with routing overhead used for performing arbitrary multiqubit Pauli measurements via lattice surgery. In the top figure, we illustrate the logical patches prior to performing the Pauli measurements. We illustrate four unit cells, with each unit cell (shown with a dashed line) containing four surface code patches and some routing space. Some additional edge padding (shown in green) is also need for routing space. In the bottom figure, we illustrate the merged patch that is obtained after measuring a logical $Y \otimes Y \otimes X \otimes Y \otimes X \otimes Z$ Pauli operator. The gate scheduling presented in Fig. 3 can be used for performing the three Y measurements.

such as the one shown in Fig. 4. In particular, Fig. 4 provides the minimal routing space requirements to perform arbitrary multiqubit Pauli measurements for logical patches placed on a two-dimensional grid with qubits afflicted by a biased noise model. We assess the routing overhead by considering an individual *unit cell* of four logical patches shown in Fig. 4. Each unit cell uses $8d_xd_z$ physical qubits (which includes yellow and grey vertices) for the four logical patches and contains a total of $2(3d_x+2)(2d_z+d_x+4)$ physical qubits. The factor $3d_x+2$ is the height of the unit cell. The factor $2d_z+d_x+4$ is the width of the unit cell. Note with the width factor needs $2(d_z+1)$ to account for the two patches and d_x+2 for the routing space between patches. It suffices to use a distance proportional to d_x (rather than d_z) because when merging our lattice surgery patches, the effective distance is maintained. This leads to a multiplicative routing overhead of

$$O_{(d_z,d_x)} = \frac{2(3d_x+2)(2d_z+d_x+4)}{8d_xd_z} \approx \frac{3}{2}. \tag{2}$$

where the approximation holds for large asymmetric codes $d_z \gg d_x \gg 4$. Equation (2) corresponds to the asymptotic routing overhead since there is extra routing space required shown as green edge padding in Fig. 4. However, the cost of edge padding (as a fraction of total costs) vanishes with the number of logical qubits. The routing overhead can be reduced further if not every logical qubit is ready to participate in lattice surgery (e.g., see the core-cache model of Ref. [23]).

B. Performance comparison between twist-based and twist-free lattice surgery

Suppose we wish to measure a multiqubit Pauli operator P using lattice surgery. The parity of the multiqubit Pauli measurement is given by a product of stabilizers in the routing space. For instance, for the $P = Y \otimes Y \otimes Y \otimes Y$ measurement in Fig. 3, the parity of the measurement is given by the product of all X-type stabilizers and twist defects in the routing space, in addition to domain wall stabilizers with the X component of the operator incident to the routing space. In what follows, we will refer to timelike failures as a collection of failure mechanisms which can result in the wrong parity outcome of a multiqubit Pauli measurement implemented via lattice

surgery. A logical timelike failure occurs when the incorrect parity of a multiqubit Pauli is obtained after performing error correction over the full syndrome history of a lattice surgery protocol. Similarly, we will refer to spacelike failures as failure mechanisms which can result in logical Pauli errors on the data qubits encoding the logical information. Lastly, we note that some failure mechanisms can result in both timelike and spacelike failures. More details can be found in Fig. 8 of Ref. [23].

Due to timelike failures in addition to spacelike failures resulting in data qubit errors which anticommute with P, the measurement of stabilizers in the routing space needs to be repeated d_m times where d_m is determined from the noise model and size of the computation. In particular, in a Pauli-based computation requiring the measurement of multiqubit Pauli operators $\{P_1, \ldots, P_\mu\}$, in Ref. [23] it was shown that d_m can be chosen to satisfy $\mu L a(bp)^{(d_m+1)/2} \leqslant \delta$ where we require each multiqubit Pauli measurement to fail with probability no greater than δ. Here L is the worst case area of the routing space used for lattice surgery and $\{a, b\}$ are constants and p quantifies the failure probability of all physical operations on the qubits.

The goal of this section is to quantify the relative performance of twist-based lattice surgery to a lattice surgery protocol free of twist defects. Such a comparison will allow us to quantify the extra algorithm runtime costs associated with twist defects. In particular, the weight-five stabilizers and elongated stabilizers have more fault locations compared to the weight-four stabilizers of the surface code and will thus make timelike failures more likely when measuring multiqubit Pauli operators containing Y terms. In performing the comparison, we use the following biased circuit-level depolarizing noise model:

(1) Each single-qubit gate location is followed by a Pauli Z error with probability $\frac{p}{3}$ and Pauli X and Y errors each with probability $\frac{p}{3\eta}$.

(2) Each two-qubit gate is followed by a $\{Z \otimes I, I \otimes Z, Z \otimes Z\}$ error with probability $p/15$ each, and a $\{X \otimes I,\ I \otimes X,\ X \otimes X,\ Z \otimes X,\ Y \otimes I,\ Y \otimes X, I \otimes Y, Y \otimes Z, X \otimes Z, Z \otimes Y, X \otimes Y, Y \otimes Y\}$ each with probability $\frac{p}{15\eta}$.

(3) With probability $\frac{2p}{3\eta}$, the preparation of the $|0\rangle$ state is replaced by $|1\rangle = X|0\rangle$. Similarly, with probability $\frac{2p}{3}$, the preparation of the $|+\rangle$ state is replaced by $|-\rangle = Z|+\rangle$.

(4) With probability $\frac{2p}{3\eta}$, a single-qubit Z basis measurement outcome is flipped. With probability $\frac{2p\alpha}{3}$, a single-qubit X-basis measurement outcome is flipped.

(5) Lastly, each idle gate location is followed by a Pauli Z with probability $\frac{p}{3}$, and a $\{X, Y\}$ error each with probability $\frac{p}{3\eta}$.

Such a biased noise model, where phase-flip errors occur with higher probabilities than bit-flip errors, is motivated from recent advances in dissipative cat-qubit and Kerr-cat qubit architectures such as those found in Refs. [19–21,29–32]. In performing our simulations, we consider two regimes for X-basis measurement errors. The first regime is for $\alpha = 1$ and the second is for $\alpha = 10$. All simulations are performed using $\eta = 100$, so that Z errors are one hundred times more likely than X and Y errors. The value of η depends a lot on the implementation of a particular hardware architecture. However, $\eta = 100$ is achievable with Kerr-cat qubit architectures or dissipative cat-qubit architectures with modest mean photon numbers. In what follows, we use a MWPM algorithm [33] on graphs with weighted edges to correct errors arising on the logical patches and in the routing space during a lattice surgery protocol. In particular, we use the decoding algorithm described in Sec. III of Ref. [23].

For the twist-free simulation of lattice surgery, we consider a $P_1 = Z \otimes Z$ measurement due to the presence of domain walls at the boundaries of the logical patches and routing space. To gain some intuition on the expected scaling of the timelike logical failure rate (i.e., failures which result in the wrong interpretation of the parity of the P_1 measurement), suppose that the stabilizers of the merged patch for measuring $Z \otimes Z$ via lattice surgery are measured d_m times. Suppose further that starting in the first syndrome measurement round of the merged patch, a string of $(d_m + 1)/2$ consecutive measurement errors occur on one of the stabilizers in the routing space encoding the parity of the $Z \otimes Z$ measurement. Since the series of measurement errors begin in the first round, the inferred parity of the P_1 will be incorrect. Further, the minimum-weight path connecting the highlighted vertices[2] will not pass through the parity vertices in the first syndrome round thus resulting in a logical timelike failure (see Ref. [23] for more details). Additionally, the number of fault locations where timelike failures can occur is proportional to the area of the qubits in the routing space. As such, we propose the following ansatz for the timelike logical failure rate of a $Z \otimes Z$ measurement:

$$P_L^{ZZ}(p) = a_1 r_s d_z (b_1 p)^{(d_m+1)/2}, \tag{3}$$

where $d_z r_s$ is the area of the routing space between the two domain walls (see Fig. 2), and $\{a_1, b_1\}$ are constants.

For the measurement of the operator $P_2 = Y \otimes Y$, we expect a similar scaling to the one provided in Eq. (3). However, as can be seen in Figs. 2 and 3, measuring logical Y operators using twist defects results in narrow vertical strips of length r_s which contain elongated stabilizers. Given the extra fault locations of both twist defects and elongated stabilizers, we consider the following ansatz for timelike logical failures:

$$P_L^{YY}(p) = P_L^{ZZ}(p) + a_2 r_s (b_2 p)^{(d_m+1)/2}. \tag{4}$$

In Eq. (4), $P_L^{ZZ}(p)$ is given by Eq. (3) with $r_s d_z$ replaced by the area of the routing space used for the $Y \otimes Y$ measurement. In other words, the additional "cost" of performing twist-based lattice surgery is an additive error contribution proportional to the height r_s containing twist defects and elongated stabilizers. For a multiqubit Pauli measurement $P = P_1 \otimes P_2 \otimes \cdots \otimes P_k$ containing $t \leqslant k$ Y terms, we expect the timelike failure probability to scale as

$$P_L^{P}(p) = P_L^{\tilde{P}}(p) + t a_2 r_s (b_2 p)^{(d_m+1)/2}, \tag{5}$$

[2]A vertex $v_j^{(k)}$ of the matching graph used to perform MWPM encodes the parity of the measurement outcome of a given stabilizer g_j in round k. The vertex $v_j^{(k)}$ is highlighted if the measurement outcome of g_j changes between consecutive syndrome measurement rounds $k-1$ and k.

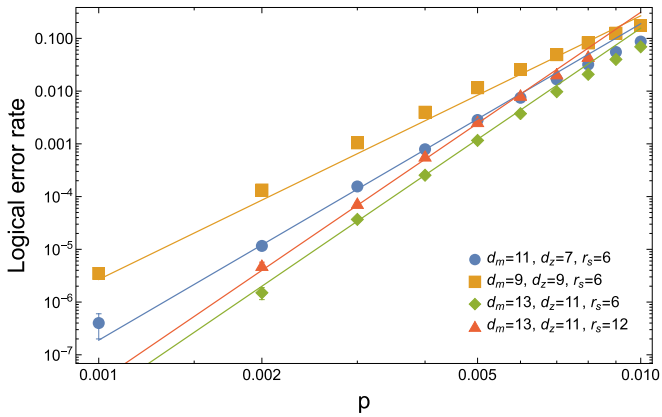

FIG. 5. Plot of the timelike logical failure rates of a $Y \otimes Y$ measurement implemented by lattice surgery with twist defects for various values of d_z, d_m and r_s with $\alpha = 10$. The results are obtained by preforming 10^7 Monte Carlo simulations using the noise model described in Sec. II B. The solid lines correspond to the best fit polynomial of Eq. (6). Our fits closely match the data for physical error rates $p \lesssim 7\times10^{-3}$. Larger values of p require the inclusion of higher order contributions to the polynomials in Eqs. (3) and (4).

for some constants $\{a_2, b_2\}$ where $\tilde{P} = Z_1 \otimes Z_2 \otimes \cdots \otimes Z_k$. In Eq. (5), we assume that the merged surface code patch for the P measurement has the same area as the merged surface code patch for the $\tilde{P}$ measurement.

Performing full circuit-level simulations using the biased noise model and decoding algorithm described above, we obtained the following best fit polynomials describing the timelike logical failure rates using the ansatzes of Eqs. (3) and (4):

$$P^{YY}_{L;\alpha=10}(p) = 0.0365 r_s(d_x + d_z)(49.44p)^{(d_m+1)/2} + 0.0514 r_s(88.145p)^{(d_m+1)/2}, \quad (6)$$

and

$$P^{YY}_{L;\alpha=1}(p) = 0.0162 r_s(d_x + d_z)(22.43p)^{(d_m+1)/2} + 0.0280 r_s(36.345p)^{(d_m+1)/2}. \quad (7)$$

The first term in Eqs. (6) and (7) can be used to extract the best fit polynomials for $Z \otimes Z$ lattice surgery with a routing space area given by $r_s(d_x + d_z)$. In Fig. 5, we show how the polynomial in Eq. (6) fits the data obtained from our Monte Carlo simulations for various values of d_z, d_m and r_s (with fixed $d_x = 7$).

Now suppose that we wish to implement a quantum algorithm so that all lattice surgery operations fail with probability no more than δ. Hence we must have that $P^{(P)}_L \leqslant \delta$ which will fix the required value of d_m for each multiqubit Pauli measurement. Using the results of Eqs. (6) and (7), in Fig. 6, we plot the ratios $d^{(\alpha)}_m(Y)/d^{(\alpha)}_m(Z)$ for $\alpha = 10$ and $\alpha = 1$. Here $d^{(\alpha)}_m(Y)$ corresponds to the minimum value of d_m such that $P^{(YY)}_{L;\alpha}(p) \leqslant \delta$. Note that in computing $d^{(\alpha)}_m(Y)$, the extension of the logical patch to obtain a horizontal Y boundary is done simultaneously with the gauge fixing step as explained in Appendix C. As such, $d^{(\alpha)}_m(Y)$ counts all syndrome measurement rounds from the gauge fixing step until the round when the patches are split. Similarly, $d^{(\alpha)}_m(Z)$ corresponds to the minimum value of d_m such that $P^{(ZZ)}_{L;\alpha}(p) \leqslant \delta$ for a lattice surgery patch where the routing space has the same area as the one used for a $Y \otimes Y$ measurement. Hence, the ratio $d^{(\alpha)}_m(Y)/d^{(\alpha)}_m(Z)$ allows us to quantify the extra multiplicative time cost of measuring $Y \otimes Y$ via lattice surgery using twist defects compared to a lattice surgery protocol for measuring $Z \otimes Z$ which does not require the use of twist defects. A few remarks are in order.

First, the ratio $d^{(\alpha)}_m(Y)/d^{(\alpha)}_m(Z)$ is highly insensitive to the chosen values of d_x, r_s and d_z. As such, the plots in Fig. 6 were obtained by fixing $d_z = 31$ and $d_x = r_s = 7$. We chose the value $d_x = 7$ since for the biased noise model considered here, it was shown in Ref. [23] that $d_x = 7$ was sufficient to implement the Hubbard model [34,35] for lattice sizes $L \leqslant 32$.

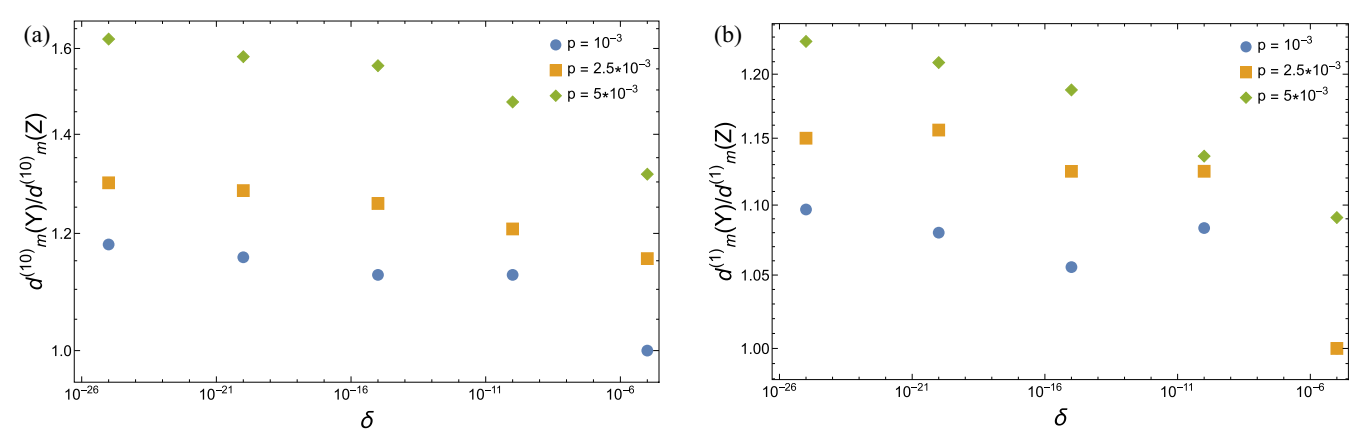

FIG. 6. (a) Ratio of $d^{(\alpha)}_m(Y)/d^{(\alpha)}_m(Z)$ for $\alpha = 10$ and various values of the target logical error rate δ such that both $P^{(YY)}_{L;\alpha}(p) \leqslant \delta$ and $P^{(ZZ)}_{L;\alpha}(p) \leqslant \delta$. The area of the routing space of the $Z \otimes Z$ measurement is taken to be identical to that of the $Y \otimes Y$ measurement. (b) Same as (a) but with $\alpha = 1$. Note that for $p = 10^{-3}$ and $p = 2.5\times10^{-3}$, $d^{(1)}_m(Y)/d^{(1)}_m(Z) = 1$ when $\delta = 10^{-5}$ which is why the data point for $p = 10^{-3}$ cannot be seen. All values were obtained by fixing $r_s = d_x = 7$ and $d_z = 31$. For large d_z, the results have very little sensitivity to the chosen values of d_x, d_z and r_s.

Second, comparing the results in Fig. 6(a) to those in Fig. 6(b), it can be seen that higher measurement failure rates increase the runtime for performing lattice surgery with twist defects. Further, for physical noise rates $p \approx 10^{-3}$, the ratio $d_m^{(\alpha)}(Y)/d_m^{(\alpha)}(Z) \approx 1.2$ when $\alpha = 10$. However, $d_m^{(\alpha)}(Y)/d_m^{(\alpha)}(Z)$ increases fairly quickly as p approaches the 10^{-2} regime which is near the threshold for logical time-like failures.

It can be seen in Fig. 6(a) that the ratio $d_m^{(1)}(Y)/d_m^{(1)}(Z)$ for $p = 10^{-3}$ increases when going from $\delta = 10^{-15}$ to $\delta = 10^{-10}$. The absolute values of $d_m^{(1)}(Y)$ and $d_m^{(1)}(Z)$ are larger when $\delta = 10^{-15}$. However, for both values of δ, we only require $d_m^{(1)}(Y) = d_m^{(1)}(Z) + 1$ to achieve the desired target logical failure rates. As such, the fraction $d_m^{(1)}(Y)/d_m^{(1)}(Z)$ is larger when the absolute values of $d_m^{(1)}(Y)$ and $d_m^{(1)}(Z)$ decrease.

Lastly, we remark that for multiqubit Pauli measurements requiring more than two Y terms, the ratios $d_m^{(\alpha)}(Y)/d_m^{(\alpha)}(Z)$ obtained in Fig. 6 would only increase in a meaningful way for very large values of t when using the result of Eq. (5) due to the exponential scaling of the second term. As such, for the noise rates considered in Fig. 6, there could be rare cases where a multiqubit Pauli measurement containing a very large number of Y terms would benefit from a twist-free approach such as the one considered in Ref. [23].

III. CONCLUSION

In Ref. [23], the twist-free scheme for measuring multiqubit Pauli operators containing Y terms results in an approximately $2\times$ slowdown in algorithm runtime due to the need to sequentially measure multiqubit Pauli operators containing pure X terms, followed by a multiqubit Pauli measurement containing pure Z terms. Our results thus suggest that, assuming a hardware architecture satisfying the connectivity constraints described in Fig. 1, twist-based lattice surgery will outperform the twist-free version for physical noise rates $p \lesssim 5\times10^{-3}$. However, for noise rates close to the threshold value $\sim 10^{-2}$, we expect twist-free based lattice surgery to achieve lower algorithm runtime relative to lattice surgery protocols using twist defects to measure Y operators. An open question is whether high threshold surface code implementations are possible using homogeneous architectures and less connectivity as was assumed in Fig. 1.

The above conclusions also depend on the noise model. For instance, comparing Figs. 6(a) and 6(b), it is clear that higher measurement failure rates results in longer $Y \otimes Y$ measurements compared to $Z \otimes Z$ measurements with an identical area. As such, for noise models where measurement and two-qubit gate failures are very high relative to other failure mechanisms, run times of twist-free lattice surgery protocols could potentially be smaller compared twist-based lattice surgery protocols.

APPENDIX A: STABILIZER MEASUREMENT CIRCUITS FOR TWIST DEFECTS IN THE BULK

In this Appendix, we describe the two types of circuits used to measure elongated checks and the weight-five stabilizers corresponding to twist defects on both the left and right vertical strips in the routing space of Fig. 3. We presented one such example in Fig. 1 and here give a more general procedure. Such circuits are shown in Figs. 7(a) and 7(b), where $|\overline{\psi}\rangle$ is the encoded state protecting the data during lattice surgery. P_1 and P_2 are Pauli operators with P_1P_2 corresponding to the stabilizer being measured.

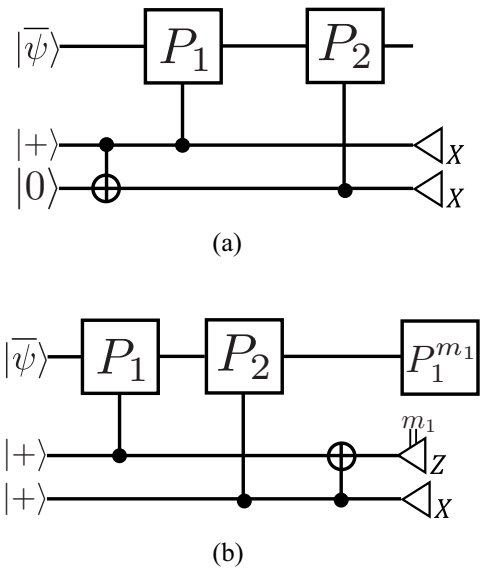

FIG. 7. (a) Circuit used to measure the weight-five stabilizers of the twist defects located on the left vertical strip of Fig. 3. A GHZ state is prepared, and the parity of the stabilizer measurement outcome is given by the product of the two X-basis measurement outcomes. (b) Second type of circuit used to measure the weight-five stabilizers of the twist defects located on the right vertical strip in Fig. 3. In this case, the X-basis measurement outcome gives the parity of the stabilizer measurement. A correction $P_1^{m_1}$ (or equivalently $P_2^{m_1}$) is applied based on the m_1 Z-basis measurement outcome. For both circuits, P_1P_2 is the stabilizer being measured.

Without loss of generality, assume the data qubits is in an eigenstate of P_1P_2 so that $P_1P_2|\overline{\psi}\rangle = \pm|\overline{\psi}\rangle$. For the circuit in Fig. 7(a), it is straightforward to show that the state prior to the measurements is given by $|\psi\rangle_f = (|0,0\rangle \pm |1,1\rangle)/\sqrt{2}|\overline{\psi}\rangle$. The ancilla state $(|0,0\rangle \pm |1,1\rangle)/\sqrt{2}$ is a eigenstate of $X \otimes X$ with eigenvalue ±1. As such, the product of the two X-basis measurement outcomes of the ancilla qubits gives the stabilizer measurement outcome.

For the circuit in Fig. 7(b), the state prior to the measurement is given by

$$|\psi\rangle_f = (|0,0\rangle + |0,1\rangle P_2 + |1,0\rangle P_1 + |1,1\rangle P_1P_2)|\overline{\psi}\rangle, \quad \text{(A1)}$$

which we can express as

$$\begin{aligned} |\psi\rangle_f &= |0\rangle(|0\rangle + |1\rangle P_1P_2)|\overline{\psi}\rangle + |1\rangle(|0\rangle P_1 + |1\rangle P_2)|\overline{\psi}\rangle \\ &= |0\rangle(|0\rangle + |1\rangle P_1P_2)|\overline{\psi}\rangle + P_1|1\rangle(|0\rangle + |1\rangle P_2)|\overline{\psi}\rangle \\ &= |0\rangle|\pm\rangle|\overline{\psi}\rangle + P_1|1\rangle|\pm\rangle|\overline{\psi}\rangle. \end{aligned} \quad \text{(A2)}$$

Therefore, measuring the second ancilla qubit in the X basis will reveal the desired ±1 eigenvalue. However, we also have a P_1 Pauli correction required depending on the value of the first ancilla qubit measured the Z basis.

Lastly, we point out that using flag based methods [7,36–48] to measure the elongated checks and weight-five

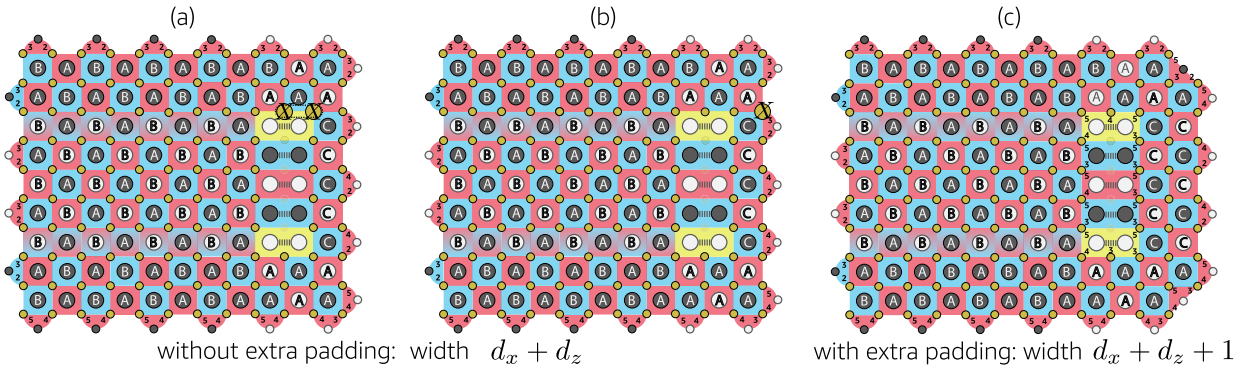

FIG. 8. In (a) and (b), we show two identical surface code patches obtained after performing a $Y \otimes Y$ measurement by lattice surgery. The width of both patches is $d_x + d_z$. (a) We consider a failure arising at the controlled-Y gate for measuring the top twist defect during the first syndrome measurement round of the merged patch. The failure results in an $X \otimes X$ error on two data qubits as shown in the figure. Since the measurement of the twist defect is random during the first round of the merge, and no changes are detected in subsequent rounds (assuming no other faults), only the vertex corresponding to the blue Z-type stabilizer which anticommutes with the error is highlighted. (b) A weight-one X error occurs on the data qubit shown in the figure resulting in the same syndrome measurement outcome as the one obtained in (a). Since both types of failures have the same syndrome measurement outcome and are parallel to a logical X operator of the surface code patch, a single failure on this lattice can lead to a logical fault. (c) Patch for measuring $Y \otimes Y$ with width $d_x + d_z + 1$. The patch uses extra padding to allow both leading order failures considered in this example to have distinct syndromes. Such a setting thus requires the logical patches to have width $d_x + 1$ prior to performing lattice surgery.

twist defects would require an extra two-qubit gate between the ancillas instead of a single CNOT. As such, an extra idling time would be required resulting in slower algorithm run times.

APPENDIX B: MINIMUM DISTANCE REQUIREMENT OF TWIST-BASED LATTICE SURGERY

In this Appendix, we show that twist-based lattice surgery protocols require extra padding to correct arbitrary X errors arising from $(d_x - 1)/2$ failures.

An example of a weight-one X error being indistinguishable from a weight-two X error (both arising from a single failure with probabilities proportional to p) for a lattice surgery protocol measuring $Y \otimes Y$ is shown in Fig. 8. The lattices in Figs. 8(a) and 8(b) have width $d_x + d_z$ where $d_x = 3$ and $d_z = 9$. The lattice in Fig. 8(c) has extra padding making its width $d_x + d_z + 1$. For the figure containing the lattices of width $d_x + d_z$, consider first the lattice in Fig. 8(a). A weight-two X data qubit error occurs during the first syndrome measurement round of the merged patch (i.e., the first syndrome measurement round when gauge fixing) due to a controlled-Y gate failure used for measuring the top twist defect. The error anticommutes with the weight-five yellow plaquette in addition to a blue weight-four Z-type stabilizer. However, the measurement outcome of the weight-five operator is random in the first round of the merge, and no change in measurement outcomes are observed in subsequent syndrome measurement rounds. As such, only the vertex associated with the weight-four Z-type stabilizer in the first round of the merge is highlighted. The observed syndrome measurement outcome is identical to the one that would be obtained if a weight-one X error, as shown in Fig. 8(b), occurred during the first round of the merge. Since both a weight-one and weight-two X errors result in the same syndrome measurement history, and $d_x = 3$, this example shows that a single failure occurring during the first syndrome measurement round when merging the surface code patches to measure $Y \otimes Y$ can lead to a logical failure. However, by increasing the d_x distance by one using extra padding, as shown in Fig. 8(c), such failure mechanisms can be distinguished.

More generally, suppose we perform lattice surgery using twist defects to measure Y operators and the initial logical patches prior to the merge have odd distance d_x. In such settings, fewer than $(d_x + 1)/2$ failures occurring during the first syndrome measurement round of the merged patch can result in a logical X error. To ensure that up to $(d_x + 1)/2$ failures never result in a logical failure, the logical patches can always be padded with an extra column of stabilizers thus increasing their d_x distance by one. Depending on the noise bias and size of the quantum algorithm, the extra padding might not be necessary. For instance, the simulations performed in Sec. II B with a noise bias of $\eta = 100$ used the odd distance $d_x = 7$ which is sufficient for implementing the Hubbard model with lattice sizes $L \leqslant 32$. Further, even though for odd distances d_x the logical X failure probability scales as $\binom{N}{(d_x-1)/2} p^{(d_x-1)/2}$, the combinatorial coefficient $\binom{N}{(d_x-1)/2}$ is very small relative to the combinatorial coefficient of higher order failure mechanisms since only $(d_x - 1)/2$ failures in the first round of the merged patch at very specialized regions near the twist defects can lead to a logical failure.

APPENDIX C: FAULT TOLERANCE PROOF

Given a rectangular logical patch, in Ref. [14] Litinski showed that the Pauli Y operator can be measured in several steps. We start with two separate surface code patches, the patch to be measured and a $|0_L\rangle$ ancilla. The next step is to extend the patch by moving the corner to express a minimum weight representative of the logical Y operator along a straight line. Afterward, a twist defect is used to measure $Z \otimes Y$ with

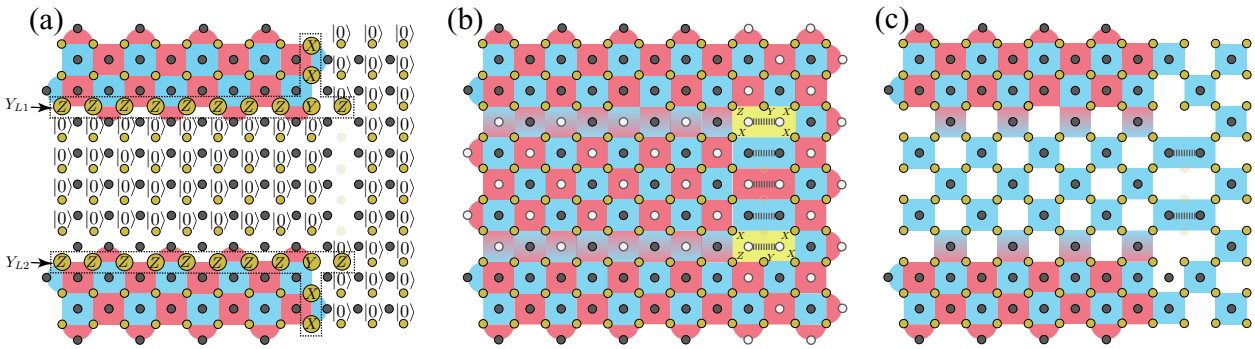

FIG. 9. Illustration showing an efficient protocol for measuring $Y \otimes Y$ fault tolerantly. In (a), we illustrate the stabilizers for the initial configuration of the logical patches including single qubit Z for $|0\rangle$ states. In (b), we show the stabilizers of the merged patched obtained via lattice surgery. In (c), we show the stabilizers of the subsystem code that (a) and (b) can be regarded as a gauge fixing of.

an ancilla prepared in a logical $|0\rangle$ state. The last step implements destructive measurements on qubits in the ancilla region in order to return to the original code space. Litinski argued that the first and second steps require d rounds of error correction, and all other steps require effectively zero time. We will refer to the second step as the *extension* step. If the extension step is required for a single Y logical measurement, it would seem natural to prefer this for $Y \otimes Y$ and more general logical measurements. However, the extension step effectively doubles the execution time for lattice surgery and also demands additional routing space.

Here we argue that the extension step can be skipped while retaining all the fault-tolerance properties of lattice surgery at the phenomenological level. We borrow the gauge-fixing ideas of Ref. [12] and apply them to the example of a $Y \otimes Y$ measurement as sketched in Fig. 9.

The initial state of Fig. 9(a) has some set of stabilizers $\mathcal{S}_{\text{old}}$ that includes all the stabilizer generators for the two logical patches. Note the data qubits in the routing space prepared in the $|0\rangle$ state. Unconventionally, but crucial to our proof, we regard the Z Pauli for each $|0\rangle$ qubit as being part of the stabilizer $\mathcal{S}_{\text{old}}$. This further allows us to choose an unusual representative for the Y_{L1} and Y_{L2} logical operators (each differs by a Z from $\mathcal{S}_{\text{old}}$ relative to the standard representatives for Y_{L1} and Y_{L2}).

The merged state of Fig. 9(b) illustrates the stabilizers $\mathcal{S}_{\text{new}}$ during the merge step (having skipped any extension step). Multiplying the stabilizers with white vertices gives our representative for $Y_{L1} \otimes Y_{L2}$ from the initial configuration and therefore these lattice surgery operations do measure the desired logical operator. If we had not included the single-qubit Z operators in $\mathcal{S}_{\text{old}}$, we would not have measured $Y_{L1} \otimes Y_{L2}$, so this is the key formal trick to prove fault tolerance.

Lastly, Ref. [12] prescribes that we consider the subsystem code with stabilizer $\tilde{\mathcal{S}} = \mathcal{S}_{\text{old}} \cap \mathcal{S}_{\text{new}}$ and gauge group $\tilde{\mathcal{G}} = \langle \mathcal{S}_{\text{old}}, \mathcal{S}_{\text{new}} \rangle$. The protocol is then (phenomenologically) fault tolerant provided that this sub-system code retains the distance of the initial code. We illustrate the local generators for $\tilde{\mathcal{S}}$ in Fig. 9(c). The logical operators for the subsystem code must commute with the stabilizer group and not be contained in the gauge group. Examining Fig. 9(c) one finds that the code distance is indeed maintained at d_z for pure Z errors and at least d_x for all other types of errors.

[1] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, J. Math. Phys. **43**, 4452 (2002).

[2] 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).

[3] T. Jochym-O'Connor and R. Laflamme, Using Concatenated Quantum Codes for Universal Fault-Tolerant Quantum Gates, Phys. Rev. Lett. **112**, 010505 (2014).

[4] T. J. Yoder, R. Takagi, and I. L. Chuang, Universal Fault-Tolerant Gates on Concatenated Stabilizer Codes, Phys. Rev. X **6**, 031039 (2016).

[5] C. Chamberland, T. Jochym-O'Connor, and R. Laflamme, Thresholds for Universal Concatenated Quantum Codes, Phys. Rev. Lett. **117**, 010501 (2016).

[6] C. Chamberland, T. Jochym-O'Connor, and R. Laflamme, Overhead analysis of universal concatenated quantum codes, Phys. Rev. A **95**, 022313 (2017).

[7] C. Chamberland, A. Kubica, T. J. Yoder, and G. Zhu, Triangular color codes on trivalent graphs with flag qubits, New J. Phys. **22**, 023019 (2020).

[8] M. E. Beverland, A. Kubica, and K. M. Svore, Cost of universality: A comparative study of the overhead of state distillation and code switching with color codes, PRX Quantum **2**, 020341 (2021).

[9] R. Raussendorf, J. Harrington, and K. Goyal, A fault-tolerant one-way quantum computer, Ann. Phys. **321**, 2242 (2006).

[10] C. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, Surface code quantum computing by lattice surgery, New J. Phys. **14**, 123011 (2012).

[11] D. Litinski and F. v. Oppen, Lattice surgery with a twist: Simplifying Clifford gates of surface codes, Quantum **2**, 62 (2018).

[12] C. Vuillot, L. Lao, B. Criger, C. G. Almudéver, K. Bertels, and B. M. Terhal, Code deformation and lattice surgery are gauge fixing, New J. Phys. **21**, 033028 (2019).

[13] A. G. Fowler and C. Gidney, Low overhead quantum computation using lattice surgery, arXiv:1808.06709.
[14] D. Litinski, A game of surface codes: Large-scale quantum computing with lattice surgery, Quantum **3**, 128 (2019).
[15] C. H. van der Wal, A. C. J. ter Haar, F. K. Wilhelm, R. N. Schouten, C. J. P. M. Harmans, T. P. Orlando, S. Lloyd, and J. E. Mooij, Quantum superposition of macroscopic persistent-current states, Science **290**, 773 (2000).
[16] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, C. Neill, P. O'Malley, P. Roushan, A. Vainsencher, J. Wenner, A. N. Korotkov *et al.*, Superconducting quantum circuits at the surface code threshold for fault tolerance, Nature (London) **508**, 500 (2014).
[17] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, I. C. Hoi, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, A. Vainsencher, J. Wenner *et al.*, State preservation by repetitive error detection in a superconducting quantum circuit, Nature (London) **519**, 66 (2015).
[18] M. D. Hutchings, J. B. Hertzberg, Y. Liu, N. T. Bronn, G. A. Keefe, M. Brink, J. M. Chow, and B. L. T. Plourde, Tunable Superconducting Qubits with Flux-Independent Coherence, Phys. Rev. Applied **8**, 044003 (2017).
[19] C. Chamberland, K. Noh, P. Arrangoiz-Arriola, E. T. Campbell, C. T. Hann, J. Iverson, H. Putterman, T. C. Bohdanowicz, S. T. Flammia, A. Keller, G. Refael, J. Preskill, L. Jiang, A. H. Safavi-Naeini, O. Painter, and F. G. S. L. Brandão, Building a fault-tolerant quantum computer using concatenated cat codes, PRX Quantum **3**, 010329 (2022).
[20] S. Puri, S. Boutin, and A. Blais, Engineering the quantum states of light in a Kerr-nonlinear resonator by two-photon driving, npj Quantum Inf. **3**, 18 (2017).
[21] A. S. Darmawan, B. J. Brown, A. L. Grimsmo, D. K. Tuckett, and S. Puri, Practical quantum error correction with the XZZX code and Kerr-cat qubits, PRX Quantum **2**, 030345 (2021).
[22] T. J. Yoder and I. H. Kim, The surface code with a twist, Quantum **1**, 2 (2017).
[23] C. Chamberland and E. T. Campbell, Universal quantum computing with twist-free and temporally encoded lattice surgery, PRX Quantum **3**, 010331 (2022).
[24] D. K. Tuckett, A. S. Darmawan, C. T. Chubb, S. Bravyi, S. D. Bartlett, and S. T. Flammia, Tailoring Surface Codes for Highly Biased Noise, Phys. Rev. X **9**, 041031 (2019).
[25] J. P. Bonilla Ataides, D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, The XZZX surface code, Nat. Commun. **12**, 2172 (2021).
[26] D. K. Tuckett, S. D. Bartlett, S. T. Flammia, and B. J. Brown, Fault-Tolerant Thresholds for the Surface Code in Excess of 5% Under Biased Noise, Phys. Rev. Lett. **124**, 130501 (2020).
[27] O. Higgott and N. P. Breuckmann, Subsystem Codes with High Thresholds by Gauge Fixing and Reduced Qubit Overhead, Phys. Rev. X **11**, 031039 (2021).
[28] Y. Tomita and K. M. Svore, Low-distance surface codes under realistic quantum noise, Phys. Rev. A **90**, 062320 (2014).
[29] J. Guillaud and M. Mirrahimi, Repetition Cat Qubits for Fault-Tolerant Quantum Computation, Phys. Rev. X **9**, 041053 (2019).
[30] J. Guillaud and M. Mirrahimi, Error rates and resource overheads of repetition cat qubits, Phys. Rev. A **103**, 042413 (2021).
[31] S. Singh, A. S. Darmawan, B. J. Brown, and S. Puri, High-Fidelity Magic-State Preparation with a Biased-Noise Architecture, arXiv:2109.02677.
[32] J. Claes, J. E. Bourassa, and S. Puri, Tailored cluster states with high threshold under biased noise, arXiv:2201.10566.
[33] J. Edmonds, Paths, trees, and flowers, Can. J. Math. **17**, 449 (1965).
[34] J. Hubbard and B. H. Flowers, Electron correlations in narrow energy bands, Proc. R. Soc. London Ser. A **276**, 238 (1963).
[35] E. T. Campbell, Early fault-tolerant simulations of the hubbard model, Quantum Sci. Technol. **7**, 015007 (2022).
[36] R. Chao and B. W. Reichardt, Quantum Error Correction with Only Two Extra Qubits, Phys. Rev. Lett. **121**, 050502 (2018).
[37] R. Chao and B. W. Reichardt, ault-tolerant quantum computation with few qubits, npj Quantum Inf. **4**, 42 (2018).
[38] C. Chamberland and M. E. Beverland, Flag fault-tolerant error correction with arbitrary distance codes, Quantum **2**, 53 (2018).
[39] T. Tansuwannont, C. Chamberland, and D. Leung, Flag fault-tolerant error correction, measurement, and quantum computation for cyclic Calderbank-Shor-Steane codes, Phys. Rev. A **101**, 012342 (2020).
[40] B. W. Reichardt, Fault-tolerant quantum error correction for Steane's seven-qubit color code with few or no extra qubits, Quantum Sci. Technol. **6**, 015007 (2021).
[41] C. Chamberland and A. W. Cross, Fault-tolerant magic state preparation with flag qubits, Quantum **3**, 143 (2019).
[42] C. Chamberland and K. Noh, Very low overhead fault-tolerant magic state preparation using redundant ancilla encoding and flag qubits, npj Quantum Inf. **6**, 91 (2020).
[43] C. Chamberland, G. Zhu, T. J. Yoder, J. B. Hertzberg, and A. W. Cross, Topological and Subsystem Codes on Low-Degree Graphs with Flag Qubits, Phys. Rev. X **10**, 011022 (2020).
[44] R. Chao and B. W. Reichardt, Flag fault-tolerant error correction for any stabilizer code, PRX Quantum **1**, 010302 (2020).
[45] T. Tansuwannont and D. Leung, Fault-tolerant quantum error correction using error weight parities, Phys. Rev. A **104**, 042410 (2021).
[46] P. Prabhu and B. W. Reichardt, Fault-tolerant syndrome extraction and cat state preparation with fewer qubits, arXiv:2108.02184.
[47] L. Postler, S. Heußen, I. Pogorelov, M. Rispler, T. Feldker, M. Meth, C. D. Marciniak, R. Stricker, M. Ringbauer, R. Blatt, P. Schindler, M. Müller, and T. Monz, Demonstration of fault-tolerant universal quantum gate operations, arXiv:2111.12654.
[48] T. Tansuwannont and D. Leung, Achieving fault tolerance on capped color codes with few ancillas, arXiv:2106.02649.