


## Error-correction and noise-decoherence thresholds for coherent errors in planar-graph surface codes

F. Venn<sup>1</sup> and B. Béri<sup>1,2</sup>

<sup>1</sup>*DAMTP, University of Cambridge, Wilberforce Road, Cambridge, CB3 0WA, United Kingdom*

<sup>2</sup>*T.C.M. Group, Cavendish Laboratory, University of Cambridge, J.J. Thomson Avenue, Cambridge, CB3 0HE, United Kingdom*

 (Received 29 June 2020; revised 9 November 2020; accepted 11 November 2020; published 23 December 2020)

We numerically study coherent errors in surface codes on planar graphs, focusing on noise of the form of  $Z$  or  $X$  rotations of individual qubits. We find that, similar to the case of incoherent bit and phase flips, a trade-off between resilience against coherent  $X$  and  $Z$  rotations can be made via the connectivity of the graph. However, our results indicate that, unlike in the incoherent case, the error-correction thresholds for the various graphs do not approach a universal bound. We also study the distribution of final states after error correction. We show that graphs fall into three distinct classes, each resulting in qualitatively distinct final-state distributions. In particular, we show that a graph class exists where the logical-level noise exhibits a decoherence threshold slightly above the error-correction threshold. In these classes, therefore, the logical level noise above the error-correction threshold can retain a significant amount of coherence even for large-distance codes. To perform our analysis, we develop a Majorana-fermion representation of planar-graph surface codes and describe the characterization of logical-state storage using fermion-linear-optics-based simulations. We thereby generalize the approach introduced for the square lattice by Bravyi *et al.* [*npj Quantum Inf.* **4**, 55 (2018)] to surface codes on general planar graphs.

DOI: [10.1103/PhysRevResearch.2.043412](https://doi.org/10.1103/PhysRevResearch.2.043412)

### I. INTRODUCTION

In recent years, significant progress has been made to improve the coherence times of qubits [1–3], including demonstrations of key ingredients for quantum error correction (QEC) [4–8]. To proceed further on the way to long-time stable qubits, topological codes such as the surface code [9–11] are considered promising candidates.

One of the major benefits of the surface code is its high tolerance to errors in physical qubits [11,12]. Error rates at the theoretically estimated fault-tolerance threshold have already been reached in experiments [1]. These thresholds are usually based on the assumption that the noise acts in the form of Pauli noise, an error model in which the action on the physical qubits is given by Pauli operators chosen from a probability distribution. In the uncorrelated case, the action on single qubits can be described by the channel

$$\mathcal{E}^p[\rho] = (1 - \epsilon)\rho + \epsilon_x X \rho X + \epsilon_y Y \rho Y + \epsilon_z Z \rho Z, \quad (1)$$

where  $\rho$  is the state of the qubit,  $X, Y, Z$  denote the Pauli operators, and  $\epsilon_x, \epsilon_y, \epsilon_z$  are suitably chosen probabilities ( $\epsilon = \sum_j \epsilon_j$ ). This channel is also referred to as an incoherent single-qubit error. Based on this error model, analytic results show that an error threshold exists under which, by increasing the number of qubits in the code, the error rate

for qubits encoded in the code (so-called logical qubits) can be made arbitrarily small [12]. The appeal of the incoherent error model is that all operations are from the Clifford group. This, together with the stabilizer code nature of the surface code, implies that the effect of such errors can be efficiently simulated classically according to the Gottesmann-Knill theorem [13]. This allowed the numerical establishment of high threshold rates [14,15], which gives reason for optimism that QEC and ultimately general quantum computation is achievable.

One of the limitations of the Pauli error model is that it does not include coherent noise, e.g., errors where each qubit undergoes a unitary rotation. These kinds of errors inevitably occur (e.g., due to qubit detuning) in quantum devices and therefore their interplay with QEC procedures needs to be understood. Mathematically, focusing on single-qubit errors, coherent errors correspond to the error channel

$$\mathcal{E}^c[\rho] = U \rho U^\dagger, \quad (2)$$

with  $U \in SU(2)$ .

Theoretical studies of coherent errors suggest that they act substantially different from incoherent errors [16–19]. In certain circumstances, they can build up quadratically faster than incoherent errors [20]. It has been shown that they affect average fidelities less than incoherent errors, but introduce higher diamond-norm error rates [21]. On the other hand, the logical-level diamond-norm error rate can scale with code distance as a more favorable power of the physical-qubit diamond-norm error for coherent rather than incoherent errors [22]. It was also found that even if physical qubits experience coherent errors, the logical-level noise, especially upon averaging over error syndromes, becomes increasingly incoherent with

---

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

increasing code distance [23–25]; however, quantifying this has some subtleties [26].

Simulations of QEC codes under coherent errors can give a useful picture of the resilience against this kind of noise. Direct simulations of the general coherent noise model are limited by the exponential scaling of Hilbert space dimension with the number of qubits. This may be partially sidestepped using tensor network descriptions of the surface code, using which, systems up to 153 qubits have been simulated [27]. The size of the system was not sufficient to establish a threshold but it provided evidence that using the so-called Pauli twirl to approximate coherent errors as incoherent noise on the level of physical qubits underestimates the logical error rate.

A key advance for understanding the effect of coherent errors was the recent development of an algorithm capable of simulating a subset of coherent errors with effort that scales polynomially with the system size [24]. The algorithm exploits a representation [28,29] of the surface code in terms of Majorana fermions and links this to coherent errors via the classically efficiently simulable [30] fermion linear optics (FLO) framework. By construction, the algorithm is limited to coherent errors acting as unitary rotations about one of the axes defined by the stabilizers, e.g.,  $U = \exp(i\eta Z)$ , and it was developed for surface codes defined on a square lattice.

Here we describe a general approach for representing surface codes with Majorana fermions on arbitrary planar graphs, including planar lattices, and show how the FLO-based algorithm can be adapted to this case. For incoherent errors, it was found [31,32] that by changing the lattice geometry, one can trade off resilience against phase flips for resilience against bit flips. By applying our method to various lattices, and relating  $Z$  rotations in one lattice to  $X$  rotations in its dual lattice, we show that a similar trade-off is present for coherent errors as well, but now for  $Z$  and  $X$  rotations instead of  $Z$  and  $X$  (that is, phase and bit) flips.

Furthermore, we study the distribution of states resulting from the application and correction of a coherent error and investigate whether, and if so in what sense, the logical-level noise decoheres, i.e., is approximable by a distribution of Pauli errors. We show that the answer depends on a graph classification that we establish. En route to our analysis of the logical-level coherence, we also describe a coherent decoder that takes advantage of the deterministic nature of coherent errors.

## II. SURFACE CODE ON GENERAL LATTICES

Stabilizer codes, and as such surface codes, are constructed by defining a set of independent, mutually commuting, products  $g_j$  of Pauli operators, in particular  $g_j^2 = I$  and  $g_j \neq -I$  [33,34]. The logical subspace is the subspace of the Hilbert space “stabilized” by the  $g_j$ :  $|\psi\rangle$  is in the codespace if  $g_j|\psi\rangle = |\psi\rangle$  for all  $j$ . The condition  $g_j \neq -I$  is required for the logical subspace to be nontrivial. To perform error corrections with such a code, first each of the stabilizers is measured. The tuple  $s$  of outcomes that are obtained is referred to as syndrome. From this syndrome, the decoder of the code computes a Pauli correction operation  $C_s$  that brings the code back to a state in which all stabilizers measure +1, i.e., the logical subspace.

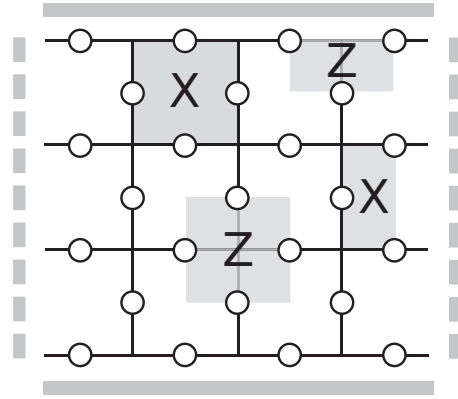


FIG. 1. A small surface code on a square lattice. The white circles mark qubits. At each vertex of the lattice, a  $Z$  stabilizer is placed, acting on all adjacent qubits. On each plaquette, that is an area surrounded (or at the boundary partially surrounded) by links, an  $X$  stabilizer is placed, acting on all qubits that are on its boundary. Examples of  $X$  and  $Z$  stabilizers are indicated as grey boxes. The code patch has two rough (dashed bars) and two smooth boundaries (solid bars). For both types of boundaries, an appropriately truncated stabilizer is shown.

The surface code is a particular stabilizer code derived from the toric code [35]. It is usually defined on a patch of a square lattice with a qubit placed on each of the links as shown in Fig. 1. Each of the vertices are associated with a  $Z$  stabilizer,  $\prod_j Z_j$ , where the product is taken over the qubits adjacent to the vertex. Conversely, each plaquette, that is, a square surrounded by links, carries an  $X$  stabilizer,  $\prod_j X_j$ , where the product is over the qubits on the plaquette boundary. In the toric code, this pattern is placed on a torus or some other manifold without a boundary [35]. The surface code, in contrast, is a planar construction based on a patch with boundaries [9,11]. To obtain a finite-sized surface code, the pattern must be terminated. The choice of boundaries determines the number of encoded logical qubits. The most often used boundaries are so-called rough boundaries and smooth boundaries. The rough boundaries are made of qubits on which only one (instead of two)  $Z$  stabilizer acts. In terms of the lattice, they are stubs pointing out of the boundary of the patch, hence the name rough boundary. While the  $Z$  stabilizers are unchanged compared to their bulk form at such a boundary, the  $X$  stabilizers need to be modified due to the truncation of the plaquettes. Smooth boundaries are boundaries without such stubs. Now the  $X$  stabilizers are unchanged compared to their bulk form but the  $Z$  stabilizers need to be modified. The boundary stabilizers are shown in Fig. 1. A patch with two rough and two smooth boundaries, in alternation, as shown in Fig. 1, encodes one logical qubit. The logical  $Z$  operator can be formed by a product of  $Z$  operators acting on each of the qubits on one of the rough boundaries. Similarly, the logical  $X$  operator can be constructed by a product of  $X$  operators acting on each qubit on one of the smooth boundaries.

To facilitate defining and describing surface codes on arbitrary planar graphs, we first reformulate the above discussion in terms of graphs. We illustrate our considerations in Fig. 2. For this construction, we start with a graph representing the  $Z$  stabilizers, together with external edges that mark the rough

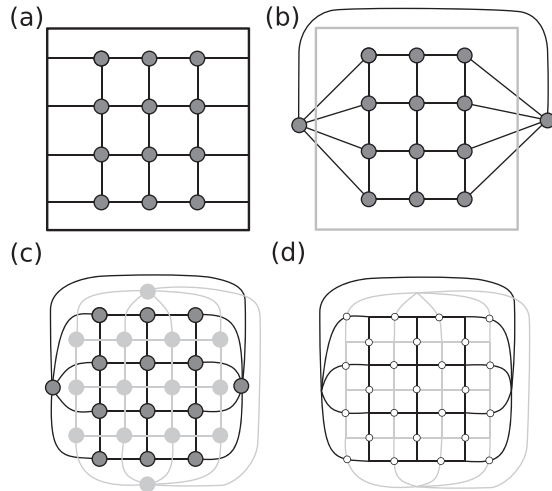


FIG. 2. (a) The embedded planar graph of the Z stabilizers for the setup equivalent to that in Fig. 1. Grey circles mark the nodes of the graph that represent Z stabilizers. (b) The same patch with added virtual stabilizers (see text). (c) The dual of the graph, forming the graph of X stabilizers. Light grey circles mark the nodes of the dual graph that represent X stabilizers. (d) The final code, made up of both the X- and Z-stabilizer graphs and the qubits. The qubits are shown as white circles. (To avoid clutter, the stabilizer nodes are not shown.)

boundaries [Fig. 2(a)]. To distinguish the two rough boundaries from each other and to keep track of them we add two connected virtual nodes, each connecting to all edges that belong to one of the rough boundaries as shown in Fig. 2(b). Using the virtual nodes, we can bring the external edges, which form the rough boundaries, to the conventional notion of graphs. This enables us to formulate the next steps in terms of standard graph operations. The virtual nodes themselves will not be translated into stabilizers of the code; instead, they can be used to define the logical operators, as we shall later explain.

From that graph, we can obtain the graph of X stabilizers by building the dual. The dual of a graph embedded in a surface is constructed by placing a node inside all faces that are formed by the edges of that graph and connecting two nodes if the faces they were placed on share a common edge. This is illustrated in Fig. 2(c). The graph that is obtained is the graph of X stabilizers. It also contains two virtual nodes, defined as the nodes that are connected by the edge that is crossing the edge connecting the virtual Z stabilizers. They are also not translated into stabilizers. Finally, the qubits are placed on the intersections of edges from the X- and Z-stabilizer graphs, where the intersection of the edges connecting virtual nodes is left out, resulting in a code patch shown in Fig. 2(d). Each stabilizer acts on all qubits it is directly connected to. The stabilizers thus defined are guaranteed to commute because a face always shares two edges with each of the vertices on its corner, therefore, the resulting node in the X-stabilizer graph will share two qubits with the node from the Z-stabilizer graph. Since the overlap is on an even number of qubits, the corresponding operators commute.

Logical operators, i.e., Pauli products that commute with all stabilizers but are independent of them, can be obtained

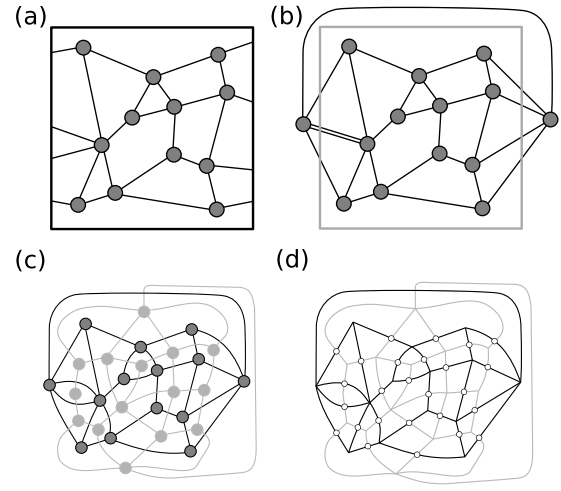


FIG. 3. (a) An example of a general planar graph with two rough boundaries (left and right), defining the Z stabilizers (dark grey circles) of the surface code patch. (b) The same graph with added virtual stabilizers to keep track of the code boundaries (see text). (c) In light grey, the dual graph, defining the X stabilizers (light grey circles). (d) The qubits (white circles) placed on the resulting surface code patch.

from the virtual nodes. Constructing an operator  $X_L$  by building a product of X operators over all qubits that are connected to one of the virtual nodes in the X-stabilizer graph produces an operator that commutes with all stabilizers for the same reasons the stabilizers commute with all other stabilizer, i.e., it overlaps on an even number of qubits with any of the Z stabilizers and trivially commutes with all X stabilizers. The same holds for the operator  $X'_L$  that can be obtained by choosing the other virtual node from the X-stabilizer graph, as well as for  $Z_L$  and  $Z'_L$ , the operators obtained from placing Z operators on the qubits attached to the virtual nodes of the Z-stabilizer graph.

This gives a total of four operators (one for each virtual node), however, only two of them are independent:  $X'_L$  can be obtained from  $X_L$  by multiplication with all X stabilizers and  $Z'_L$  from  $Z_L$  by multiplication with all Z stabilizers. Hence, we need to study only  $X_L$  and  $Z_L$ . While  $X_L$  and  $Z_L$  commute with all stabilizers, they anticommute with each other: We did not include the qubit in the intersection of the edges between the virtual nodes, hence  $Z_L$  and  $X_L$  overlap only on a single qubit. Therefore, they form a pair of logical X and Z operators. We choose  $Z_L$  to be the logical Z operator and  $X_L$  the logical X operator.

Applied to the square lattice, this construction recovers our previous discussion, as can be seen by comparing Figs. 1 and 2. However, it provides a framework for describing arbitrary planar graphs; an example is shown in Fig. 3 with the four panels describing the steps analogous to those in Fig. 2.

A surface code defined on a general planar graph shares many properties with the standard square-lattice surface code. Both are Calderbank-Shor-Steane [5] codes, which means that all stabilizers are either formed only by X or only by Z operations, a property we shall exploit to analyze the action of the error. Also, they can both be decoded by a minimum weight perfect matching (MWPM) algorithm [12,36,37]. Neverthe-

less, there are subtle differences, e.g., in the average stabilizer weight or the connectivity of the graph. In particular, the connectivity has been shown to influence whether the code, under incoherent errors, is more resilient against bit or phase flips [31,32].

### III. ERROR MODELS

In these planar graph surface codes, we shall study error channels,

$$\mathcal{E} = \bigotimes_{j=1}^N \mathcal{E}_j, \quad (3)$$

where  $N$  is the number of qubits and  $\mathcal{E}_j$  is a single-qubit error acting on qubit  $j$ . Our primary focus is the study of coherent errors of the form

$$\mathcal{E}_j[\rho] = \exp(iZ_j\eta_j)\rho \exp(-iZ_j\eta_j), \quad (4)$$

where  $\eta_j$  is a real parameter. The consideration of mere  $Z$  rotations is linked to the FLO simulability of the system [24]. We note, however, that due to the duality relation between the  $X$ - and  $Z$ -stabilizer graphs, we can also study  $X$  rotations by exchanging the graphs for their duals.

To compare the effects of coherent errors to the incoherent case, we shall also study Eq. (3) with Eq. (4) replaced by its Pauli twirl:

$$\mathcal{E}_j[\rho] = \rho \cos^2 \eta_j + Z_j \rho Z_j \sin^2 \eta_j. \quad (5)$$

We shall be interested in studying what refinements of the finding of Ref. [27], that the Pauli twirl underestimates the coherent error, may arise in more general lattices and in assessing how potential trade-offs between resilience against  $X$  and  $Z$  errors compare in the coherent and incoherent [31] cases.

### IV. QUANTUM ERROR-CORRECTION AND ITS CHARACTERIZATION

To study the logical errors that arise, we apply  $\mathcal{E}$  to the code followed by error correction  $\mathcal{R} = \sum_s \mathcal{R}_s$  based on the MWPM decoder. Here  $\mathcal{R}_s$  is the quantum operation of measuring syndrome  $s$  followed by the application of the corresponding Pauli correction. (In Sec. VII, we provide a more detailed description of the procedure.)

Inspired by Ref. [24], we shall investigate the logical error rate  $p_L$  and the properties of the distribution of states after error correction. A key difference between the square-lattice case [24] and codes on general planar graphs is related to whether the weight of all  $Z$  stabilizers (of  $Z_L$ ) is even (odd). (The weight of a Pauli operator is the number of qubits on which it acts nontrivially.) In Ref. [24], an alignment of the square lattice is chosen where all  $Z$  stabilizers have even weight and  $Z_L$  has odd weight. (This does not hold in the conventional orientation shown in Fig. 1 due to the boundary stabilizers.) As explained in Ref. [24], a key consequence of this is that  $\mathcal{R}_s \circ \mathcal{E}$  acts as a unitary channel on the logical qubit; the state  $\rho_s$  arising after  $\mathcal{R}_s \circ \mathcal{E}$  is

$$\rho_s = \frac{1}{P_s} \mathcal{R}_s \circ \mathcal{E}[\rho] = \exp(i\theta_s Z_L) \rho \exp(-i\theta_s Z_L), \quad (6)$$

where neither  $\theta_s$  nor the probability  $P_s$  of syndrome  $s$  depend on the initial logical state  $\rho$ . Hence, both  $p_L$  and the properties of the final state  $\rho_s$  can be studied via a statistical analysis of  $\theta_s$ .

For more general graphs, apart from some special cases, this property is absent and  $P_s$  depends on the initial logical state  $|\psi_L\rangle$  of the code; the error correction process thus reveals information about  $|\psi_L\rangle$ . In studying the logical error rate, we eliminate this  $|\psi_L\rangle$  dependence by defining  $p_L$  as the diamond-norm distance between the actions on the logical subspace of the identity and the average logical channel [24,38]  $\Lambda_L$  defined by

$$\Lambda_L[\rho] = \sum_s P_s \rho_s = \sum_s \mathcal{R}_s \circ \mathcal{E}[\rho] = \mathcal{R} \circ \mathcal{E}[\rho]. \quad (7)$$

We are also interested in the properties of final states  $\rho_s$ . To mitigate the  $|\psi_L\rangle$  dependence of  $P_s$  in this case, we adopt a statistical approach based on averaging with respect to a uniform distribution of  $|\psi_L\rangle$  across the Bloch sphere. With  $|\psi_L\rangle$  thus chosen randomly, the syndrome probability

$$P_s = \text{Tr}(\mathcal{R}_s \circ \mathcal{E}[\rho]) \equiv P(s|\rho) \quad (8)$$

must be viewed as the probability of  $s$  conditioned on the initial state being  $\rho = |\psi_L\rangle\langle\psi_L|$ . We shall be interested in the Bloch-sphere-averaged distance between  $\rho_s$  and  $\rho$ . For a suitable (semi)metric  $\delta^2(\rho_s, \rho)$  on the space of logical states, this is

$$\langle \delta^2(\rho_s, \rho) \rangle_\Omega = \int_\Omega d\rho P(\rho|s) \delta^2(\rho_s, \rho), \quad (9)$$

where  $\Omega$  is the Bloch sphere and the conditional probability  $P(\rho|s)$  enters because we are after the Bloch-sphere average given that the syndrome outcome is  $s$ . The combined Bloch-sphere and syndrome average is

$$\sum_s P(s) \langle \delta^2(\rho_s, \rho) \rangle_\Omega = \sum_s \int_\Omega d\rho P(s, \rho) \delta^2(\rho_s, \rho), \quad (10)$$

where  $P(s) = \int_\Omega d\rho P(s, \rho)$  is a marginal of the joint syndrome-Bloch-sphere distribution  $P(s, \rho)$ . For computational convenience, for  $\delta^2$  we shall use the square of the trace-norm distance,

$$\delta(\rho_s, \rho) = \sqrt{1 - |\langle \psi_L | \psi_s \rangle|^2}, \quad (11)$$

between  $\rho$  and  $\rho_s = |\psi_s\rangle\langle\psi_s|$ . That is, we consider the average infidelity conditioned on measuring syndrome  $s$ . Equation (10) thus gives the average infidelity to the identity of the average logical channel. For a discussion of the relation between the average infidelity and the diamond-norm distance, see Refs. [21,25,26,39,40] and for the relation between the worst-case and average infidelities see Ref. [41].

### V. MAJORANA GRAPH

To study the model introduced above, we represent the surface code on a planar graph in terms of a corresponding Majorana fermion graph. Our approach is based on that of Refs. [24,28,29]; it proceeds by representing physical qubits in terms of Majorana fermions and a local constraint. In this way, the eigenstates of the surface code are described in



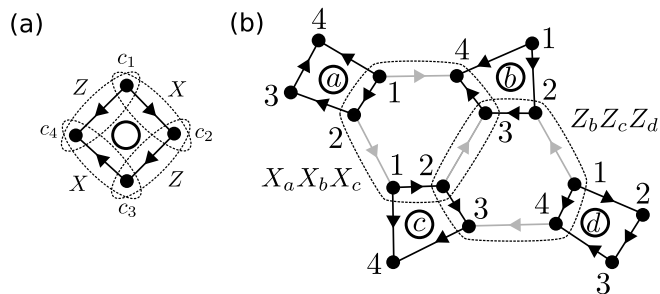


FIG. 4. (a) Representation of a qubit in the  $C4$  encoding with four Majorana fermions  $c_1, c_2, c_3, c_4$ . The Pauli operators  $X$  and  $Z$  can each be formed by two equivalent dimers (pairs) of Majorana fermions. The order of the operators is indicated by arrows, e.g.,  $X = ic_1c_2$  is represented by an arrow from  $c_1$  to  $c_2$ . (b) Majorana graph for four qubits. A stabilizer with support on different qubits can be represented in terms of the Pauli dimers (black) of the qubits, but also using link dimers (grey) between the qubits. Link dimer orientations must be chosen to ensure equivalence to Pauli dimerization of stabilizers; this results in a Majorana graph where edges (Pauli and link dimers) have Kasteleyn orientation.

terms of free-fermion eigenstates of a quadratic commuting-dimer Majorana Hamiltonian projected to the physical, qubit, Hilbert space.

To obtain this representation, each qubit  $j$  is encoded in four Majorana fermions  $c_{j1}, c_{j2}, c_{j3}, c_{j4}$ . This is referred to as  $C4$  encoding. The Majorana fermions  $c_{jk}$  satisfy

$$c_{jk}^\dagger = c_{jk}, \quad \{c_{ik}, c_{jl}\} = 2\delta_{ij}\delta_{kl}, \quad (12)$$

where  $\{\dots\}$  denotes the anticommutator and  $\delta_{ij}$  is the Kronecker delta. A conventional fermion, satisfying  $\{d_m, d_n^\dagger\} = \delta_{mn}$ , is built out of a pair of Majorana fermions via  $d_m = c_{m1} + ic_{m2}$ . The four Majorana fermions for qubit  $j$  thus correspond to two conventional fermions, hence a four-dimensional Hilbert space. To arrive at a two-dimensional Hilbert space encoding a qubit, we introduce the stabilizer  $S_j = -c_{j1}c_{j2}c_{j3}c_{j4}$  and work in the subspace satisfying  $S_j = 1$  for all qubits  $j$ . We shall refer to the  $S_j$  as qubit stabilizers.

In the  $C4$  encoding, the Pauli operators on a qubit are given by Majorana bilinears,  $X_j = ic_{j1}c_{j2}$ ,  $Y_j = ic_{j1}c_{j3}$ ,  $Z_j = ic_{j2}c_{j3}$ . We shall call these bilinear Pauli dimers. They satisfy the commutation relations for Pauli operators and commute with the qubit stabilizer  $S_j$ . Since  $C4$ -encoded states are stabilized by  $S_j$ , there is for each Pauli dimer an equivalent Pauli dimer:  $X_j = ic_{j2}c_{j3}S_j$ ,  $Y_j = ic_{j2}c_{j4}S_j$ ,  $Z_j = ic_{j4}c_{j1}S_j$ .

It is beneficial to represent  $C4$ -encoded qubits in terms of a Majorana graph, as shown in Fig. 4. In this graph, nodes represent Majorana fermions and the edges between them represent bilinears  $ic_{ik}c_{jl}$ . The edges have an orientation indicated by arrows reflecting the operator order: for a bilinear  $ic_{ik}c_{jl}$ , the arrow points from fermion  $c_{ik}$  to fermion  $c_{jl}$ . The Pauli dimers we shall use for a single qubit  $j$  are those for  $X_j$  and  $Z_j$ . The graph for a single qubit is shown in Fig. 4(a).

The  $Z$  and  $X$  stabilizers of the surface code involve products of Pauli operators from different qubits. Such products translate into products of Majorana operators which we have the freedom to reorder, provided we keep track of the signs. As illustrated in Fig. 4(b), this freedom can be used to change

from Pauli dimers to link dimers, i.e., on the links between the qubits. In the example of Fig. 4(b), the stabilizer  $X_aX_bX_c$  is rearranged:

$$\begin{aligned} X_aX_bX_c &= (ic_{a1}c_{a2})(ic_{b3}c_{b4})(ic_{c1}c_{c2}) \\ &= (ic_{a1}c_{b4})(ic_{c2}c_{b3})(ic_{a2}c_{c1}). \end{aligned} \quad (13)$$

Note that the rearrangement is performed in such a way that the dimer  $ic_{c2}c_{b3}$  is also a part of the rearrangement of the stabilizer  $Z_bZ_cZ_d$ .

More generally, consider a stabilizer  $g$  involving a product of  $n$  Pauli operators, for which we can pick dimer representations such that the Pauli dimers can be connected by additional edges such that the joint set of added edges and Pauli dimers forms the boundary of a face in the Majorana graph. Then we can represent  $g$  in terms of a clockwise product of Majorana fermions along the boundary of that face. Considering this operator order, each of the Pauli dimers in  $g$  has the form  $is_kc_{\alpha_k}c_{\beta_k}$ , where  $\alpha_k$  and  $\beta_k$  enumerate the double (i.e., qubit and Majorana) indices along the boundary of the face and  $s_k = -1$  if this operator order is opposite to that of the original Pauli dimer ( $s_k = 1$  otherwise). That is,

$$g = (is_1c_{\alpha_1}c_{\beta_1})(is_2c_{\alpha_2}c_{\beta_2}) \dots (is_nc_{\alpha_n}c_{\beta_n}). \quad (14)$$

We now rearrange the product such that it is over link dimers. A simple rebracketing is sufficient for this for all but the first and the  $n$ th Pauli dimer:

$$\begin{aligned} g &= c_{\alpha_1}(is_1c_{\beta_1}c_{\alpha_2})(is_2c_{\beta_2}c_{\alpha_3}) \dots \\ &\dots (is_{n-1}c_{\beta_{n-1}}c_{\alpha_n})(is_nc_{\beta_n}). \end{aligned} \quad (15)$$

To form the link dimer  $ic_{\beta_n}c_{\alpha_1}$ , however,  $c_{\alpha_1}$  has to be commuted through an odd number of Majorana fermions yielding

$$\begin{aligned} g &= -(is_1c_{\beta_1}c_{\alpha_2})(is_2c_{\beta_2}c_{\alpha_3}) \dots \\ &\dots (is_{n-1}c_{\beta_{n-1}}c_{\alpha_n})(is_nc_{\beta_n}c_{\alpha_1}). \end{aligned} \quad (16)$$

Provided we absorb this additional sign in the orientation of one of the link dimers, we now find that the stabilizer is expressed as a product over these. Since all but one of these have a corresponding Pauli dimer with the same orientation, the total number of clockwise-oriented edges (i.e., Pauli and link dimers) around the face of the Majorana graph is odd. By the same logic, the same holds for each of the faces that represent a stabilizer in the graph. The Majorana graph thus has an orientation in which all faces have an odd number of clockwise oriented edges: a so-called Kasteleyn orientation.

The free-fermion state underlying the description of the surface code emerges from the observation that the entire set of stabilizer generators can be rearranged in the way described above and thereby be described in terms of mutually commuting link dimers. (These dimers, however, do not commute with the qubit stabilizers  $S_j$ , highlighting the fact that the surface code eigenstates are not free-fermion states, but projections thereof.) For the square lattice, this is shown in Refs. [24,28,29]. In the following, we shall describe an algorithm with which one can construct Majorana graphs for surface codes on arbitrary planar graphs. We shall illustrate our algorithm using the  $Z$ -stabilizer graph in Fig. 3(a) and the corresponding qubit graph in Fig. 3(d).

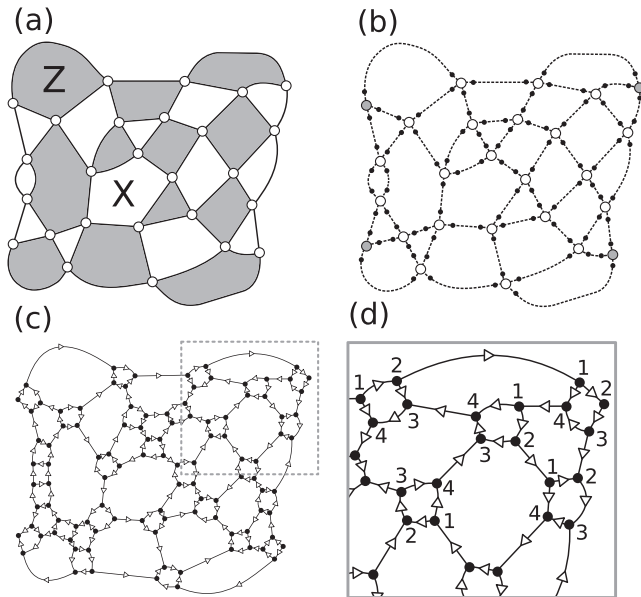


FIG. 5. (a) The construction of graph  $\mathcal{G}$  with one face per stabilizer for the surface code in Fig. 3. (b) The edges of  $\mathcal{G}$  can be used as Majorana dimers for the  $C4$  encoding of the surface code. This results in four Majorana fermions per qubit except for the four qubits in the corners (grey). (c) A Kasteleyn orientation of the dimers. (d) Zoom from panel (c) showing the  $C4$  encoding of the individual qubits (cf. Fig. 4).

We start the construction of the Majorana graph  $\mathcal{M}$  with the qubit graph  $\mathcal{Q}$ , the graph containing both qubits and stabilizers as vertices and connecting each stabilizer to the qubits they are acting on. The construction of  $\mathcal{Q}$  starting from any initial planar graph of  $Z$  stabilizers is described in Sec. II [an example is shown in Fig. 3(d)]. To construct  $\mathcal{M}$ , we will construct one intermediate graph  $\mathcal{G}$  by taking all qubits from  $\mathcal{Q}$  and connecting them if this can be done without crossing any edge in  $\mathcal{Q}$ . Figure 5(a) shows graph  $\mathcal{G}$  obtained in this way from  $\mathcal{Q}$  in Fig. 3(d). Graph  $\mathcal{G}$  has the property that it contains a face for every stabilizer generator and the operator it represents acts on the qubits on the boundary of that face. Additionally, it has the property that every qubit, apart from those at the corners, has four edges connected to it. Qubits at the corners are different because these miss the link to the qubit that was not inserted because it would have been at the intersection of edges connecting virtual stabilizers.

To obtain the Majorana graph  $\mathcal{M}$ , we place two Majorana fermions on each of the edges in  $\mathcal{G}$  and associate each of the fermions to one of the qubits at the edge’s ends. Using this method, we associate with each qubit the same number of Majorana fermions as the qubit has edges connected to it. Since in graph  $\mathcal{G}$  each of the vertices, except the four on the corners of the graph, is connected with four edges, all qubits, except the qubits in the corner, have four Majorana fermions associated to them. For the example system we are considering, this stage of the construction is shown in Fig. 5(b). To be able to encode all qubits in the  $C4$  encoding, we add one additional Majorana fermion to each of the qubits at the corners of the code, such that these additional fermions are in none of the faces of graph  $\mathcal{G}$ . This gives the complete set of vertices for graph  $\mathcal{M}$ .

To construct the edges of  $\mathcal{M}$ , we first add an edge between two of its vertices if they were on the same edge in  $\mathcal{G}$ ; these edges will form the link dimers. We also add the edges associated to the  $C4$  encoding of the qubits by adding the edges forming a face around each qubit; these edges will form the Pauli dimers. This completes the construction of the (thus far unoriented) edges of  $\mathcal{M}$ .

Next, we have to associate some more structure to  $\mathcal{M}$ : We must assign (i) a Kasteleyn orientation [cf. Fig. 4 and under Eq. (16)] and (ii) a placement of Pauli dimers (i.e., a numbering  $c_{j1}, \dots, c_{j4}$  of Majorana fermions) around each qubit such that the clockwise Majorana product around each face of  $\mathcal{G}$  encodes the correct stabilizer. That such a structure exists can be seen as follows. First, we consider the Pauli dimers. In the bulk of the code, the stabilizers surrounding a qubit alternate between  $X$  and  $Z$ , since qubits are placed on the intersection of an edge of the  $X$ -stabilizer graph and an edge of the  $Z$ -stabilizer graph. Similarly, the dimer representation of the Pauli operators in the  $C4$  encoding alternates between  $X$  and  $Z$  dimers [cf. Fig. 4(a)]. Therefore, we can always arrange Pauli dimers such that each of them is adjacent to the stabilizer face to which it contributes. (On the edge of the code, there are less than four surrounding stabilizers, however, the existing adjacent stabilizers already specify the placement of  $X$  and  $Z$  dimers). In a convention where  $X$  dimers are oriented clockwise [Fig. 4(a)], there are two possible orientations for each qubit: We can choose which of the  $Z$  Pauli dimers is oriented anticlockwise. We can pick any of the two. This defines the Pauli dimer part of the Majorana graph, including the orientation of the faces around each of the qubits.

To complete the structuring of  $\mathcal{M}$ , we must orient the link dimers such that, globally, a Kasteleyn orientation is obtained. To this end, we can use that each of the so-far unoriented faces (edges) in  $\mathcal{M}$  is associated to a face (edge) in the graph  $\mathcal{G}$ . Then, for any face in  $\mathcal{G}$ , we count the number  $n$  of clockwise-oriented Pauli dimers surrounding the corresponding face in  $\mathcal{M}$ . If  $n$  is odd, we have to orient the edges of this face of  $\mathcal{G}$  such that an even number of edges are clockwise (and vice versa for  $n$  even). In this way, it is sufficient to find an orientation of  $\mathcal{G}$  such that each face has the parity of clockwise oriented edges as determined by  $n$  before. To produce this orientation for  $\mathcal{G}$ , we can proceed similarly to the first steps of the Fisher-Kasteleyn-Temperley (FKT) algorithm [42]. By orienting  $\mathcal{M}$ ’s link dimers according to the orientation obtained for  $\mathcal{G}$ , we have obtained a Kasteleyn orientation of  $\mathcal{M}$ . The resulting graph and orientations for our example are shown in Figs. 5(c) and 5(d).

The logical state of the surface code patch is defined by the state of the qubit encoded in the four unpaired corner Majorana fermions [24]. This becomes clear when we consider the Majorana encoding of the logical operators. Following steps analogous to Eq. (16), the Majorana encoding of either of the logical operators requires a new link dimer connecting two of the four corner fermions (as shown in Fig. 6); the new link dimer has an orientation such that the resulting new face (which corresponds to a virtual stabilizer for  $\mathcal{G}$ ) has an odd number of edges pointing clockwise. In the initial state in which all stabilizers measure  $+1$ , all equivalent realizations of a logical operator must have the same expectation value. Therefore, we have to add such faces for both realizations

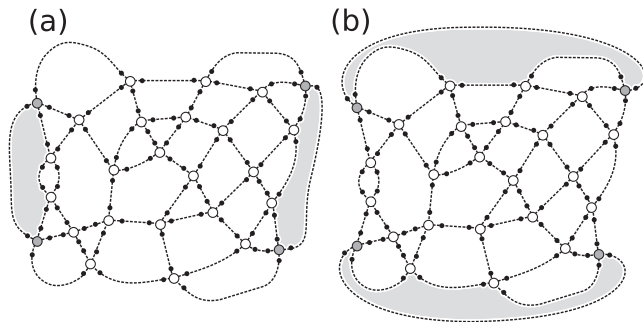


FIG. 6. Majorana encoding of logical operators. A logical operator of the code translates to an additional face in  $\mathcal{M}$  bounded by a link dimer formed by two of the four initially unpaired corner Majorana fermions. (a) shows  $\mathcal{M}$  with the two link dimers that form both realizations of  $Z_L$  (each dimer corresponds to one of the two virtual  $Z$  stabilizers of Sec. II). The additional faces of  $\mathcal{M}$  for including these dimers are shown in grey. (b) Link dimers for  $X_L$ , showing again the two additional faces.

(corresponding to both of the virtual stabilizers) of the logical operator. By stabilizing the state that is encoded in the Majorana graph with a logical operator, we fix the logical state of the code to be in a  $+1$  eigenvalue of that logical operator. We thereby fix the code to be in either the  $|0_L\rangle$  state by choosing to stabilize with  $Z_L$  or in the  $|+L\rangle$  state by using  $X_L$ . To initialize the code in the  $|Y_L\rangle$  state, we can pair up fermions from diagonally opposite ends of the code patch.

## VI. FLO SIMULATION

In the following, we describe how to use the methods introduced in Ref. [24] to sample from the distribution of syndromes and how to compute, given a syndrome  $s$ , the overlaps

$$\langle \pm_L | C_s \exp i\eta \mathbf{Z} | +_L \rangle, \langle \pm_L | C_s \exp i\eta \mathbf{Z} | Y_L \rangle, \quad (17)$$

where  $C_s$  is the Pauli correction for syndrome  $s$ ,  $\eta = (\eta_1, \eta_2, \dots, \eta_N)$ ,  $\mathbf{Z} = (Z_1, Z_2, \dots, Z_N)$ , and  $\eta \mathbf{Z}$  is the scalar product between the two. From these overlaps, the quantities characterizing the error correction process [cf. Sec. IV] can be extracted using Monte Carlo simulation, as shown in Secs. VII and IX.

To perform these operations, we use the framework of FLO. Within this framework we have access to the following operations:

- (1) Initializing a dimer in the  $+1$  eigenstate.
- (2) Applying the unitary operation  $R = \exp(\eta c_j c_j)$  with an arbitrary real  $\eta$ .
- (3) Projectively measuring a dimer operator, with or without postselection.

These are operations that maintain the property of a state to be a fermionic Gaussian state, which can be exploited to simulate their actions efficiently [30].

The limitation of the FLO algorithm is that it cannot treat quartic products of Majorana operators such as those in the qubit stabilizers  $S_j$ . To bypass this problem, each qubit is projectively measured in the  $|\pm\rangle$  basis; we shall see that this allows working with objects involving Majorana bilinears. Although this makes it impossible to evaluate the  $Z$  stabilizers,

such evaluation is not needed: Since we apply only  $Z$  rotations, we know that none of the  $Z$  stabilizers could have been flipped.

To sample from the syndrome distribution, we sample from the eigenvalues  $m_a$  of single-qubit Pauli operators  $X_a$ ; the eigenvalues of  $X$  stabilizers can be computed from  $m_a$  classically. The probability for measuring  $m_a$  requires performing three steps on each qubit  $a$ : First, switch to fermions and project into the  $C4$  encoding using  $(1 + S_a)/2$ , then apply the coherent error  $U_a = \exp i\eta_a Z_a$ , and, finally, apply the projector  $(1 + m_a X_a)/2$ . Since  $Z_a$  commutes with the qubit stabilizer  $S_a$ , we can perform the rotation first; a further reordering of the Majorana fermions gives

$$P_a(m_a) = \frac{1}{2}(1 + m_a X_a) \frac{1}{2}(1 + m_a S_a X_a) \exp(i\eta_a Z_a), \quad (18)$$

in terms of which the joint probability of measuring  $\mathbf{m} = (m_1, m_2, \dots, m_n)$  is the expectation value of  $\prod_{a=1}^N P_a(m_a)$  with respect to a Gaussian state. Note that computing this joint distribution requires only rotations and measurements with postselection, both involving dimers only, allowing a computation using the FLO operations introduced above. However, this by itself does not offer a route to efficiently sample from the exponentially many outcomes  $\mathbf{m}$ . Reference [24] showed how one may do this qubit by qubit, thus breaking the sampling down to a repeated sampling from just two states. The success of this approach hinges on choosing a correct order in which to measure the qubits: The order must be such that graph  $\mathcal{G}$  stays connected when removing, after every measurement, the qubit that was measured. Such an ordering can be obtained for our graphs by performing a breadth-first search through the graph; the obtained order can be used in reverse.

From the sampled  $\mathbf{m}$ , we compute the syndrome  $s$  classically using suitable products of  $m_a$  for the corresponding  $X$  stabilizers. From  $s$ , the decoder produces the correction  $C_s$ . Since, by construction, no  $Z$  stabilizer is flipped, the correction contains only operators that correct  $X$  stabilizers:  $C_s$  contains only  $Z$  operators. Using this, we define  $\eta_s$  according to

$$\exp i\eta_s \mathbf{Z} = C_s \exp i\eta \mathbf{Z}, \quad (19)$$

that is, we absorb the correction operations into the parameters of the coherent rotations.

The quantity we aim to compute is  $\langle +_L | \exp i\eta_s \mathbf{Z} | +_L \rangle$ . We can expand  $|+_L\rangle$  in the computational basis; it is given by the sum over the set  $\mathcal{L}$  of all computational basis states that satisfy all  $Z$  stabilizers:

$$|+_L\rangle = |\mathcal{L}|^{-1/2} \sum_{x \in \mathcal{L}} |x\rangle. \quad (20)$$

Furthermore,

$$\begin{aligned} \langle +_L | \exp i\eta_s \mathbf{Z} | +_L \rangle &= |\mathcal{L}|^{-1} \sum_{x \in \mathcal{L}} \sum_{y \in \mathcal{L}} \langle y | \exp i\eta_s \mathbf{Z} | x \rangle \\ &= |\mathcal{L}|^{-1} \sum_{y \in \{0,1\}^N} \sum_{x \in \mathcal{L}} \langle y | \exp i\eta_s \mathbf{Z} | x \rangle \\ &= 2^{N/2} |\mathcal{L}|^{-1} \sum_{x \in \mathcal{L}} \langle +^{\otimes N} | \exp i\eta_s \mathbf{Z} | x \rangle \\ &= 2^{N/2} |\mathcal{L}|^{-1/2} \langle +^{\otimes N} | \exp i\eta_s | +_L \rangle, \end{aligned} \quad (21)$$

where in the second line we used that  $\exp i\eta_s \mathbf{Z}$  is diagonal in the computational basis to replace  $\sum_{y \in \mathcal{L}}$  by a summation over all computational basis states, and in the third line we used that  $2^{N/2} |+\rangle^{\otimes N} = \sum_{y \in \{0,1\}^N} |y\rangle$ . Hence,

$$|\langle +_L | \exp i\eta_s \mathbf{Z} | +_L \rangle|^2 = M^{-2} |\langle +^{\otimes N} | \exp i\eta_s \mathbf{Z} | +_L \rangle|^2, \quad (22)$$

with  $M = 2^{-N/2} |\mathcal{L}|^{1/2}$ . Equation (22) is a constant  $M^{-2}$  times the probability to measure the outcome  $|+\rangle$  for all qubits. Hence it can be computed using the FLO algorithm, this time without sampling, to find the probability for the outcome  $\mathbf{m} = (1, 1, \dots)$ . To eliminate the factor  $M$  we can build the ratio

$$\begin{aligned} q_s &= \frac{|\langle -_L | C_s \exp i\eta \mathbf{Z} | +_L \rangle|^2}{|\langle +_L | C_s \exp i\eta \mathbf{Z} | +_L \rangle|^2} \\ &= \frac{|\langle +^{\otimes N} | Z_L \exp i\eta_s \mathbf{Z} | +_L \rangle|^2}{|\langle +^{\otimes N} | \exp i\eta_s \mathbf{Z} | +_L \rangle|^2}. \end{aligned} \quad (23)$$

For the simulation, the operator  $Z_L$  can be absorbed into  $\eta$  the same way we absorbed  $C_s$ .

In a similar fashion, we can compute the ratio

$$r_s = \frac{|\langle -_L | C_s \exp i\eta \mathbf{Z} | Y_L \rangle|^2}{|\langle +_L | C_s \exp i\eta \mathbf{Z} | Y_L \rangle|^2}, \quad (24)$$

i.e., the same expectation values but starting with the  $|Y_L\rangle$  state. This can be done by initializing the simulation in a different state such that the logical state is given by  $|Y_L\rangle$ .

## VII. AVERAGE LOGICAL CHANNEL

In the following, we explain how to obtain the full action of the average logical channel from the observables  $q_s, r_s$  that are accessible via the FLO simulation.

To this end, we first need a description of the recovery procedure  $\mathcal{R}$ . The recovery scheme consists of two steps. First, all stabilizers are measured; this projects the state into one syndrome  $s$ . This projection is performed by the projector  $\Pi_s$ . Next, depending on the syndrome  $s$ , the decoder chooses a correction operation  $C_s$ . The combined recovery is given by

$$\mathcal{R}[\rho] = \sum_s C_s \Pi_s \rho \Pi_s C_s^\dagger. \quad (25)$$

Since  $C_s$  maps between the space in which the stabilizers have syndrome  $s$  and the logical subspace, we can represent  $\Pi_s = C_s \Pi_0 C_s^\dagger$ , where  $\Pi_0$  denotes the projection into the logical subspace. Using this relation and the fact that the corrections  $C_s$  are Pauli operators and thus satisfy  $C_s = C_s^\dagger$ , the operation  $\mathcal{R}$  can be expressed as

$$\mathcal{R}[\rho] = \sum_s \Pi_0 C_s \rho C_s \Pi_0. \quad (26)$$

The correction  $\mathcal{R}$  together with the error  $\mathcal{E}$  is

$$\mathcal{R} \circ \mathcal{E}[\rho] = \sum_s \Pi_0 C_s \exp(i\eta \mathbf{Z}) \rho \exp(-i\eta \mathbf{Z}) C_s \Pi_0. \quad (27)$$

This operation maps any state of the logical subspace back to the logical subspace. Within that subspace, it is the average logical channel

$$\Lambda_L[\rho] = \mathcal{R} \circ \mathcal{E}[\Pi_0 \rho \Pi_0], \quad (28)$$

where we introduced  $\Pi_0$  to remind that we view  $\Lambda_L[\rho]$  as a quantum channel on the logical subspace.

Since we have an algorithm to sample from the distribution of syndromes, we study the action corresponding to an individual syndrome  $s$ ,

$$\mathcal{R}_s \circ \mathcal{E}[\Pi_0 \rho \Pi_0] = D_s \rho D_s^\dagger, \quad (29)$$

where we introduced  $D_s = \Pi_0 C_s \exp(i\eta \mathbf{Z}) \Pi_0$ . We consider the action of  $D_s$  in the logical subspace.  $D_s$  commutes with the logical  $Z_L$  operator, therefore,  $D_s$  is diagonal in the  $Z_L$  basis and hence can be represented as

$$D_s = \text{diag}(a_s, b_s) = a_s \pi_0 + b_s \pi_1, \quad (30)$$

with  $a_s, b_s \in \mathbb{C}$  and  $\pi_i$  the projector on the logical states  $|i_L\rangle \langle i_L|$ ,  $i \in 0, 1$ . In terms of  $a_s$  and  $b_s$ , we have

$$\begin{aligned} \mathcal{R}_s \circ \mathcal{E}[\Pi_0 \rho \Pi_0] &= |a_s|^2 \pi_0 \rho \pi_0 + |b_s|^2 \pi_1 \rho \pi_1 \\ &\quad + a_s \bar{b}_s \pi_1 \rho \pi_0 + \bar{a}_s b_s \pi_0 \rho \pi_1, \end{aligned} \quad (31)$$

where the bar indicates complex conjugation. The action of  $\Lambda_L$  follows from  $\sum_s \mathcal{R}_s$ . We find

$$\Lambda_L[\rho] = \alpha \pi_0 \rho \pi_0 + \beta \pi_1 \rho \pi_1 + \gamma \pi_0 \rho \pi_1 + \bar{\gamma} \pi_1 \rho \pi_0, \quad (32)$$

with

$$\alpha = \sum_s |a_s|^2, \quad \beta = \sum_s |b_s|^2, \quad \gamma = \sum_s \gamma_s, \quad \gamma_s = a_s \bar{b}_s. \quad (33)$$

Since  $\Lambda_L$  is trace preserving, we have  $\alpha = 1$  and  $\beta = 1$ . Thus, the entire action of  $\Lambda_L$  is encoded in the single complex parameter  $\gamma$ . Furthermore,

$$(\Lambda_L - \mathbb{1})[\rho] = (\gamma - 1) \pi_0 \rho \pi_1 + (\bar{\gamma} - 1) \pi_1 \rho \pi_0, \quad (34)$$

from which, by the proportionality of Eq. (34) to the action of a unitary channel minus the identity, we read off [21,24,43] the diamond-norm [35] distance:

$$p_L = \|\Lambda_L - \mathbb{1}\|_\diamond = |\gamma - 1|. \quad (35)$$

We wish to estimate  $\gamma$  using Monte Carlo simulation. Using the FLO approach, we are able to sample from the distribution of syndromes starting from the initial state  $|+_L\rangle$ . The syndrome probability is

$$P_s^+ = \|D_s |+_L\rangle\|^2. \quad (36)$$

To estimate  $\gamma$ , we seek a quantity  $c_s$  accessible from the simulation such that

$$P_s^+ c_s = \gamma_s. \quad (37)$$

In this way, the Monte Carlo average  $\sum_s P_s^+ c_s = \gamma$ .

Using the simulation algorithm introduced in Sec. VI, we have access to

$$q_s = \frac{|\langle +_L | Z_L D_s | +_L \rangle|^2}{|\langle +_L | D_s | +_L \rangle|^2} \quad \text{and} \quad r_s = \frac{|\langle +_L | Z_L D_s | Y_L \rangle|^2}{|\langle +_L | D_s | Y_L \rangle|^2}. \quad (38)$$



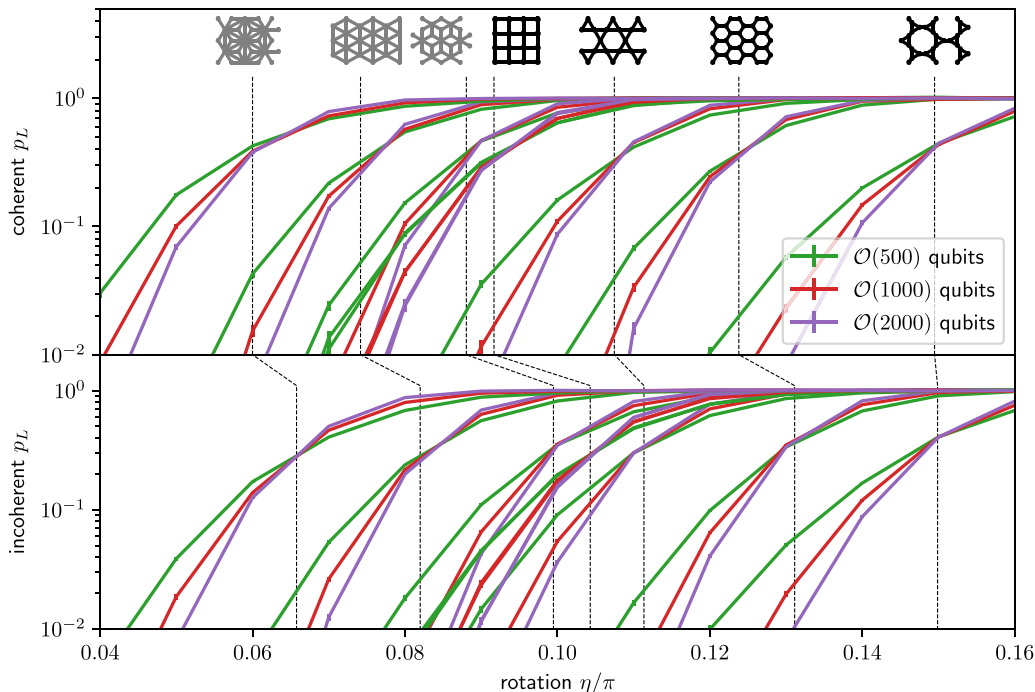


FIG. 7. The logical error rate  $p_L$  for the coherent and Pauli-twirled error models versus the angle  $\eta$  for three different code sizes of each lattice. (The argument of  $\mathcal{O}$  indicates the approximate number  $N$  of qubits; the concrete value of  $N$  depends on the lattice.) The dashed lines indicate the thresholds obtained by fitting a finite-size scaling ansatz [15]. Above the threshold, we sketch the graphs of the  $X$  stabilizers. They are, from left to right, dual of trihex, dual of hexagonal, dual of kagome, square, kagome, hexagonal, and trihex.

By expanding both expressions in  $a_s$  and  $b_s$  and reordering, we find the relations

$$q_s = \frac{P_s^+ - \text{Re } \gamma_s}{P_s^+ + \text{Re } \gamma_s}, \quad r_s = \frac{P_s^+ - \text{Im } \gamma_s}{P_s^+ + \text{Im } \gamma_s}, \quad (39)$$

which imply

$$\text{Re } \gamma_s = \frac{1 - q_s}{1 + q_s} P_s^+, \quad \text{Im } \gamma_s = \frac{1 - r_s}{1 + r_s} P_s^+. \quad (40)$$

Conveniently, both expressions match the form of Eq. (37). Therefore, we can approximate  $\gamma$  using a Monte Carlo approximation of the sum

$$\gamma = \sum_s P_s^+ c_s = \sum_s P_s^+ \left( \frac{1 - q_s}{1 + q_s} + \frac{1 - r_s}{1 + r_s} i \right). \quad (41)$$

### VIII. THRESHOLD

Simulations of surface codes on various lattices have shown that the thresholds of the codes depend significantly on the connectivity of the lattice [31,32]. In the following, we study lattices with different connectivity under coherent and incoherent errors.

In the choice of lattices, we follow Ref. [31] and perform simulations for the square, kagome, hexagonal, (3, 12<sup>2</sup>) (triangle-hexagonal, also referred to as trihex) lattices, and their duals. For the square lattice, we study codes with distances 25 (625 qubits), 37 (1369 qubits), 49 (2401 qubits), and for the other lattices we study system sizes with a comparable number of qubits.

For each surface code, we perform two simulations: one for coherent errors [i.e., with  $\mathcal{E}$  using Eqs. (3) and (4)] and one with incoherent errors using the Pauli twirl [i.e., with  $\mathcal{E}$  using Eqs. (3) and (5)]. For simplicity, we apply the same error to all qubits,  $\eta_j = \eta$ . For both error models and all lattices, we first simulate an initial overview spanning from  $\eta = 0.4\pi$  to  $\eta = 1.6\pi$  in  $0.1\pi$  steps with 10 000 Monte Carlo samples for the coherent error and 40 000 Monte Carlo samples for the incoherent error. The results are shown in Fig. 7. We then estimate the thresholds  $\eta_{\text{th}}$  by first estimating their position from this overview, and then performing a simulation in  $0.01\pi$  steps around the estimated position and fit a finite-size scaling ansatz [15]. Our threshold estimates are shown in Figs. 7 and 8. We find that for these lattices the coherent thresholds are consistently lower than the incoherent ones (or are at best comparable to them as for the trihex lattice). Our results also indicate that for  $\eta \lesssim \eta_{\text{th}}$ , the logical error rate decreases with code distance slower for coherent than for incoherent errors.

The trade-off between resilience against bit and phase flips that is obtainable in the incoherent (twirled) case is reflected by the thresholds approaching [31] the bound

$$R \leq 1 - h(p_x) - h(p_z) \quad (42)$$

for zero asymptotic encoding rate  $R \rightarrow 0$ . (For our case of a single encoded qubit,  $R = 1/N$ .) Here  $h$  is the binary Shannon entropy and  $p_x$  and  $p_z$  are the probabilities of  $X$  and  $Z$  flips on individual physical qubits [33]. In our case, the thresholds  $p_{z,\text{th}}$  are parametrized by  $\eta$ , i.e.,  $p_{z,\text{th}} = \sin^2 \eta_{\text{th}}$  and  $p_{x,\text{th}}$  is obtained by lattice duality. In Fig. 8, we visualize this bound.

The results show that the trade-off between resilience against bit and phase flips translates, for coherent errors, to

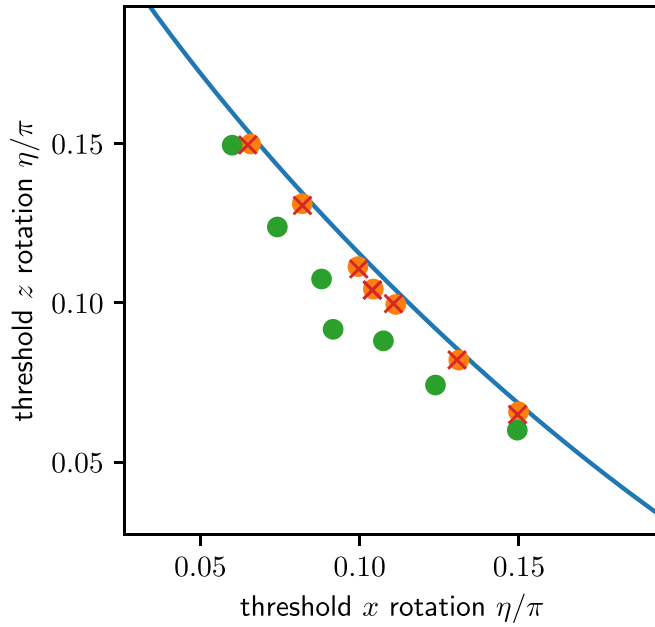


FIG. 8. The thresholds for  $X$  rotations versus the thresholds for  $Z$  rotations for both the coherent (green) and Pauli-twirled (orange) error models, for the lattices indicated in Fig. 7. (The thresholds obtained in Ref. [31] are shown in red.) The blue line indicates the bound Eq. (42).

a trade-off between resilience against  $X$  and  $Z$  rotations. In Ref. [31], it is argued that the trade-off for incoherent errors is present because it is easier for the MWPM decoder to match up syndromes in a sparse graph. It is reasonable to assume that a similar effect is also causing the trade-off in the coherent case. However, considering Fig. 8, unlike for the incoherent thresholds, there does not appear to be a universal curve delineating this trade-off for coherent thresholds.

To provide further evidence for the absence of such a universal curve, we construct a lattice that is self-dual and therefore can be directly compared to the square lattice, i.e., its thresholds for  $X$  and  $Z$  errors are equal by design. We call this lattice the doubly odd lattice; it has faces with three and five vertices (see Fig. 9 inset). We compare the results for this lattice, both for coherent and incoherent errors, to the square-lattice case in Fig. 9. The results show that the coherent threshold for the doubly odd lattice is significantly higher than that for the square lattice. For incoherent errors, however, the thresholds for the two lattices are very close, consistent with the observation of Ref. [31] that most surface codes with a MWPM decoder perform very close to the bound Eq. (42).

### IX. FINAL-STATE DISTRIBUTION

To get further insight into the properties of the states after error correction, we study the action of the error and correction process conditioned on the individual syndromes. The final states have certain properties that are dependent on properties of the stabilizer group. It turns out that the parity of the stabilizers is of central importance. We shall generalize the property [24] that for codes which have only even-weight  $Z$  stabilizers together with an odd-weight logical  $Z_L$  operator,

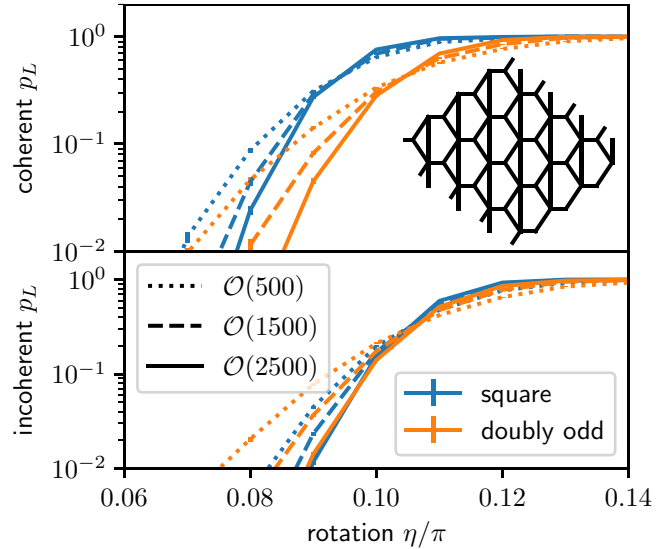


FIG. 9. A comparison between the logical error rate  $p_L$  for the doubly odd and square lattice. (A small patch of the  $Z$ -stabilizer graph for the doubly odd lattice shown as inset.) The upper graph shows  $p_L$  for coherent errors, the lower shows  $p_L$  for incoherent errors. The simulations for the square lattice are the same used for Fig. 7. The doubly odd lattice is simulated for system sizes of 437, 1365, and 2805 qubits.

a coherent  $Z$  error followed by a correction acts as a unitary operation. We will assess the different symmetries that are present in the lattices we considered above and determine the consequences for the final-state distributions. Note that we are now investigating properties of the error-correction process based on properties of the  $Z$ -stabilizer graph, while the argument that it is easier to correct errors in sparse graphs was based on the  $X$ -stabilizer graph.

We start by considering an individual syndrome  $s$  that is corrected with the operator  $C_s$  that is made up only of  $Z$  operators. In the following, we denote a string of  $Z$  operators by  $Z(b)$ , where  $b$  is a bit string of length  $N$  whose value is one (zero) for qubits on which  $Z(b)$  acts nontrivially (trivially). In particular,  $Z(0) = I$ . Expanding both the error and the correction using this representation yields

$$C_s = Z(h_s), \quad \exp(i\eta Z) = \sum_{g \in \mathcal{B}} c_g i^{\|g\|} Z(g), \quad (43)$$

where  $h_s$  is the bit string encoding the correction operation,  $c_g$  are real coefficients,  $\|g\|$  denotes the Hamming weight of bit-string  $g$ , and  $\mathcal{B}$  is the set of all length- $N$  bit strings. Therefore [24],

$$D_s = \sum_{g \in \mathcal{B}} c_g i^{\|g\|} \Pi_0 Z(h_s \oplus g) \Pi_0, \quad (44)$$

where  $\oplus$  is bit-wise addition modulo 2.

As discussed in Sec. VII,  $D_s$  is diagonal in the logical space of the code. Therefore, we can represent it as

$$D_s = k_s \Pi_0 + l_s \Pi_0 Z_L, \quad (45)$$

with suitable coefficients  $k_s$  and  $l_s$ . When expressing  $k_s$  and  $l_s$  as a sum over contributions from Eq. (44),  $k_s$  is formed by terms in which  $Z(h_s \oplus g)$  acts trivially on the logical space, i.e., when  $Z(h_s \oplus g)$  is within the  $Z$ -stabilizer group. Conversely,  $l_s$  is formed by those terms in which  $Z(h_s \oplus g)$  acts as the logical  $Z_L$  operator, i.e., those corresponding to  $Z_L$  times a stabilizer. Hence,

$$\begin{aligned}
 k_s &= \sum_{g \in \mathcal{B} \text{ with } g \oplus h_s \in \mathcal{A}} c_g i^{\|g\|}, \\
 l_s &= \sum_{g \in \mathcal{B} \text{ with } g \oplus h_s \in \mathcal{A} \oplus l} c_g i^{\|g\|},
 \end{aligned}
 \tag{46}$$

where  $\mathcal{A}$  is the set of all bit strings corresponding to operators in the  $Z$ -stabilizer group,  $l$  is the bit string that encodes  $Z_L$ , and  $\mathcal{A} \oplus l$  denotes the set  $\{a \oplus l | a \in \mathcal{A}\}$ . Since  $g$  runs over all bit strings, we can convert Eq. (46) into sums over  $\mathcal{A}$ ,

$$\begin{aligned}
 k_s &= \sum_{a \in \mathcal{A}} c_{a \oplus h_s} i^{\|a \oplus h_s\|}, \\
 l_s &= \sum_{a \in \mathcal{A}} c_{a \oplus h_s \oplus l} i^{\|a \oplus h_s \oplus l\|}.
 \end{aligned}
 \tag{47}$$

We next study the complex phase of  $k_s$  and  $l_s$ . In general,  $\mathcal{A}$  contains both even- and odd-weight bit strings and we cannot make a definite statement. If, however, all bit strings in  $\mathcal{A}$  have even weight, the exponent of  $i$  has the same parity for all the terms in each sum in Eq. (47). This constrains  $k_s$  and  $l_s$  to be either real or imaginary, with their relative phase set by  $l$ . Therefore, we can distinguish between two families: codes in which all  $Z$  stabilizers are of even weight and codes that do not satisfy this condition.

In the case that not all  $Z$  stabilizers are of even weight, we can make no further statements about the distribution. In fact, checking the distribution that is obtained for a code based on a  $Z$ -stabilizer graph defined on a hexagonal lattice, i.e., a system in which the majority of the  $Z$  stabilizers are of weight 3, we find [Fig. 10(a)] that at least with some probability all parts of the Bloch sphere of final states are reached.

For those lattices for which every  $Z$  stabilizer has even weight, we can identify the relative complex phase between  $l_s$  and  $k_s$ . For this phase, the parity of  $h_s$  is irrelevant since, whether it is odd or even, it affects both  $l_s$  and  $k_s$  the same way. However, the weight of  $Z_L$  affects only  $l_s$ . Hence, in the family in which all  $Z$  stabilizers are of even weight, we have two subfamilies, those with even-weight  $Z_L$  and those with odd-weight  $Z_L$ .

The case that  $Z_L$  is of odd weight is already explored in Ref. [24]: we have  $\text{Im } k_s = 0$  and  $\text{Re } l_s = 0$  and

$$D_s = \sqrt{P_s} U_s, \quad P_s = |k_s|^2 + |l_s|^2, \quad U_s = \frac{k_s}{\sqrt{P_s}} + \frac{l_s Z}{\sqrt{P_s}}, \tag{48}$$

where  $U_s$  is a unitary operator. That is, the state  $\rho_s$  satisfies Eq. (6). In consequence, the logical state is constrained to a circle on the Bloch sphere that is parallel to the  $XY$  plane [Fig. 10(b)]. The distribution of states that can be obtained starting from the  $|+\rangle_L$  state is shown in Fig. 10(c).

If  $Z_L$  is of even weight, we have  $\text{Im } k_s = 0$  and  $\text{Im } l_s = 0$ . The operation  $D_s$  in that case is given by the, unusual, real

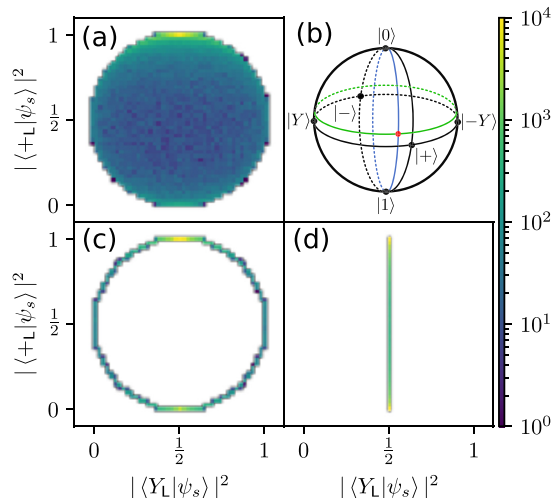


FIG. 10. Panels (a), (c), (d): The projection of 100 000 sampled states  $|\psi_s\rangle$  resulting after error and correction operation with initial state  $|+\rangle_L$ . Distribution for a surface code on the (a) hexagonal lattice, (c) square lattice, (d) kagome lattice. The size of the system is for all examples chosen such that the number of qubits is around 500. (b) The Bloch sphere and an example of an initial state (red dot). The green (blue) circle marks the possible final states in lattices with even-weight  $Z$  stabilizers and odd-weight (even-weight)  $Z_L$ .

combination of the identity and  $Z_L$ . If  $|k_s| > |l_s|$ ,

$$D_s = (|k_s| - |l_s|) + 2|l_s| \begin{cases} \pi_0 & \text{if } l_s k_s > 0 \\ \pi_1 & \text{else,} \end{cases} \tag{49}$$

and if  $|l_s| > |k_s|$ ,

$$D_s = Z_L(|l_s| - |k_s|) + 2Z_L|k_s| \begin{cases} \pi_0 & \text{if } l_s k_s > 0 \\ \pi_1 & \text{else.} \end{cases} \tag{50}$$

Equations (49) and (50) show that syndrome measurements reveal information about the  $Z_L$  polarization of the initial logical state  $|\psi_L\rangle$ . They also imply that the final state after the action of  $D_s$  lies on the circle spanning  $|0\rangle_L$ ,  $|1\rangle_L$ , and  $|\psi_L\rangle$ . This is also illustrated in Fig. 10(b), and a numerical example is shown in Fig. 10(d).

This leaves us with three classes of lattices to build  $Z$ -stabilizer graphs, and we have examples for each:

- (1) containing odd-weight  $Z$  stabilizers: trihex, dual of trihex, hexagonal, doubly odd, and dual of kagome,
- (2) all  $Z$  stabilizers even weight, odd-weight  $Z_L$ : square, dual of hexagonal,
- (3) all  $Z$  stabilizers even weight, even-weight  $Z_L$ : kagome.

To compare the final-state distributions for the different lattices, we proceed as explained in Sec. IV. We have  $|\psi_s\rangle = D_s |\psi_L\rangle / \sqrt{P(s|\rho)}$  with  $P(s|\rho) = \langle \psi_L | D_s^\dagger D_s | \psi_L \rangle$  (recall,  $\rho = |\psi_L\rangle \langle \psi_L|$ ). Hence,

$$\delta^2(\rho_s, \rho) = 1 - \frac{|\langle \psi_L | D_s | \psi_L \rangle|^2}{\langle \psi_L | D_s^\dagger D_s | \psi_L \rangle}. \tag{51}$$

To prepare for the Bloch-sphere average, we parametrize

$$|\psi_L\rangle = |\theta, \phi\rangle = \cos(\theta/2)|0\rangle_L + \sin(\theta/2)e^{i\phi}|1\rangle_L. \tag{52}$$

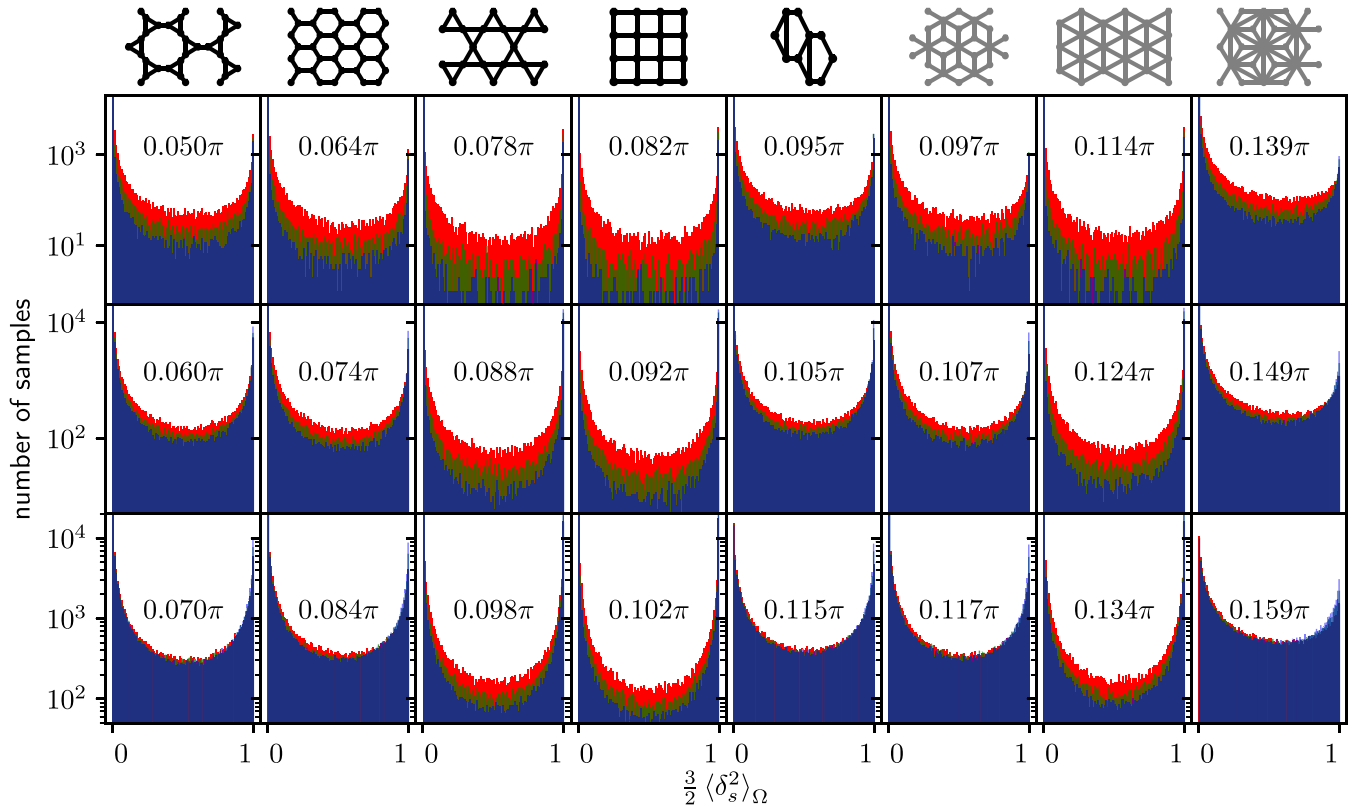


FIG. 11. Distributions of 100 000 samples of  $\langle \delta_s^2 \rangle_\Omega$  for the seven lattices in Fig. 7, and the doubly odd lattice (on the right of the square lattice). We give a sketch of the corresponding Z-stabilizer graph above the distributions. (These are the duals of the graphs in Fig. 7.) The values of the coherent noise parameter  $\eta$  are shown above the distributions; they are chosen as  $\eta \approx \eta_{th}$ , i.e., approximately at the threshold (middle row),  $\eta \approx \eta_{th} - 0.01\pi$  (top row), and  $\eta \approx \eta_{th} + 0.01\pi$  (bottom row). Samples are taken for three different systems sizes  $\mathcal{O}(500)$  qubits (red),  $\mathcal{O}(1500)$  qubits (green), and  $\mathcal{O}(2500)$  qubits (blue).

After some manipulation, we find

$$\delta^2(\rho_s, \rho) = \frac{|\langle -_L | D_s | +_L \rangle|^2 \sin^2 \theta}{P(s|\rho)}, \quad (53)$$

where

$$P(s|\rho) = \frac{1}{2} [ |a_s|^2 + |b_s|^2 + (|a_s|^2 - |b_s|^2) \cos \theta ] \quad (54)$$

due to Eq. (30). Using  $P(\rho|s) = P(s|\rho)P(\rho)/P(s)$ , the Bloch-sphere average is

$$\begin{aligned} \langle \delta_s^2 \rangle_\Omega &\equiv \int_\Omega d\rho P(\rho|s) \delta^2(\rho_s, \rho) \\ &= \frac{|\langle -_L | D_s | +_L \rangle|^2}{P(s)} \int_\Omega d\rho P(\rho) \sin^2 \theta \\ &= \frac{2}{3} \frac{|\langle -_L | D_s | +_L \rangle|^2}{P(s)} = \frac{1}{3} \left( 1 - \frac{\text{Re } \gamma_s}{P(s)} \right), \end{aligned} \quad (55)$$

where we used that for  $P(\rho)$  uniformly distributed over the Bloch sphere,  $\int_\Omega d\rho P(\rho) \sin^2 \theta = 2/3$ . Note that

$$P(s) = \int_\Omega d\rho P(s|\rho)P(\rho) = \frac{|a_s|^2 + |b_s|^2}{2} = P_s^+, \quad (56)$$

the probability in Eq. (36). Hence, Eq. (55) can be entirely expressed in terms of quantities that can be extracted from the FLO-based simulation.

We sample from  $\langle \delta_s^2 \rangle_\Omega$  for three different values of the error parameter  $\eta$  for each lattice: approximately the threshold value ( $\eta \approx \eta_{th}$ ) and  $\eta \approx \eta_{th} \pm 0.01\pi$ . The results are shown in Fig. 11. For all simulations, we observe that  $\langle \delta_s^2 \rangle_\Omega$  has sharp peaks around  $2/3$  and  $0$ ; these values correspond to  $|\psi_s\rangle = Z_L |\psi_L\rangle$  and  $|\psi_s\rangle = |\psi_L\rangle$ , respectively. This shows that the coherent Z rotations for the codes at the distances we study can be well approximated by a distribution of Pauli errors. However, in contrast to the effect of an incoherent error, each state  $\rho_s$  after the operation  $\mathcal{R}_s \circ \mathcal{E}$  is still a pure state; we get a mixed state only if the information of the syndrome outcome  $s$  is deleted (i.e., only for the output of  $\Lambda_L$ ).

The distributions show patterns characteristic of the lattice geometry. In particular, while below the threshold all lattices have a distribution that is increasingly concentrated on  $0$  and  $2/3$  with increasing the code distance, slightly above the threshold ( $\eta \approx \eta_{th} + 0.01\pi$ ) this increasing concentration can be observed only for codes containing only even-weight stabilizers.

## X. NOISE DECOHERENCE THRESHOLDS AND THE COHERENT DECODER

The code-distance dependence of the final-state distributions in Fig. 11 suggests that, at least for certain lattices, a second threshold  $\eta_{th}^c$  might exist such that increasing the code distance makes the logical-level noise increasingly Pauli-like



only for  $\eta < \eta_{\text{th}}^c$ . To study the existence of such a decoherence threshold, we first invoke a notion [25] of coherence for the operation  $\mathcal{R}_s \circ \mathcal{E}$  on the logical subspace. For a logical state  $\rho$ , we have (with proportionality factor  $|k_s|^2 + |l_s|^2$ )

$$\mathcal{R}_s \circ \mathcal{E}[\rho] \propto (1 - \epsilon_s^P)\rho + \epsilon_s^P Z_L \rho Z_L + \epsilon_s^c Z_L \rho + \bar{\epsilon}_s^c \rho Z_L, \tag{57}$$

where  $\epsilon_s^P = |l_s|^2 / (|k_s|^2 + |l_s|^2)$  and  $|\epsilon_s^c| = \sqrt{\epsilon_s^P(1 - \epsilon_s^P)}$ . The coherent part can be defined [25] as the non-Pauli contribution to Eq. (57), quantified by  $\epsilon_s^c$ . The coherent part is much smaller than the Pauli part if  $|\epsilon_s^c| \ll 1$ , i.e., if  $\epsilon_s^P$  is either close to zero or one. (A good Pauli approximation thus requires action that is either nearly pure identity or nearly pure  $Z_L$ .) This precisely corresponds to  $\langle \delta_s^2 \rangle_\Omega$  being 0 or 2/3 because  $\langle \delta_s^2 \rangle_\Omega = 2\epsilon_s^P/3$ .

We next introduce a linearized proxy for  $|\epsilon_s^c|$ :

$$\langle \delta_s^2 \rangle_\Omega^c = \frac{2}{3} \min[\epsilon_s^P, (1 - \epsilon_s^P)]. \tag{58}$$

The less  $|\epsilon_s^c|$  is, the less is  $\langle \delta_s^2 \rangle_\Omega^c$ , and vice versa. The distribution for  $\langle \delta_s^2 \rangle_\Omega^c$  is obtained from the  $\langle \delta_s^2 \rangle_\Omega$  distribution by mirroring, around 1/3, the part above 1/3 into the lower values.

The quantity  $\langle \delta_s^2 \rangle_\Omega^c$  can also be interpreted as the average infidelity obtained by choosing between the Pauli correction  $C_s$  and its alternative,  $C'_s = C_s Z_L$  such that it minimizes  $\langle \delta_s^2 \rangle_\Omega$ . With the alternative correction we have

$$\langle \delta_s^2 \rangle_\Omega = \frac{|\langle -_L | Z_L D_s | +_L \rangle|^2}{P(s)} = \frac{1}{3} \left( 1 + \frac{\text{Re } \gamma_s}{P(s)} \right), \tag{59}$$

from where we get the expression

$$\begin{aligned} \langle \delta_s^2 \rangle_\Omega^c &\equiv \min \left( \frac{1}{3} \left( 1 - \frac{\text{Re } \gamma_s}{P(s)} \right), \frac{1}{3} \left( 1 + \frac{\text{Re } \gamma_s}{P(s)} \right) \right) \\ &= \frac{1}{3} \left( 1 - \left| \frac{\text{Re } \gamma_s}{P(s)} \right| \right). \end{aligned} \tag{60}$$

The latter interpretation is reminiscent of the optimal decoder [12,15,44–47]: this calculates, given an error model, whether the syndrome  $s$  is more likely to require correction with  $C_s$  or  $C_s Z_L$ . For the optimal decoder for incoherent errors, this calculation holds for any initial logical state  $\rho$  and requires no choice of error measure. Here, we optimize the Bloch-sphere-average and target a concrete error measure. [A  $\rho$ -independent variant optimizing  $\theta_s$  is, however, possible for lattices corresponding to Eq. (6).] Our approach is tailored for coherent errors: it takes advantage of the deterministic nature of these in an essential manner. Hence, we can refer to  $\langle \delta_s^2 \rangle_\Omega^c$  as the average infidelity for the coherent decoder.

To assess whether a decoherence threshold  $\eta_{\text{th}}^c$  exists, we study

$$\Delta = \sum_s P(s) \langle \delta_s^2 \rangle_\Omega^c, \tag{61}$$

which we simulate using Monte Carlo sampling. The results obtained for the lattices and sizes introduced in Sec. VIII are shown in Fig. 12. In these averages, we can readily observe qualitative effects related to the three  $Z$ -stabilizer graph classes introduced in Sec. IX: the three classes can be distinguished by the plateau value of  $\Delta$  attained when the rotation parameter  $\eta$  is sufficiently beyond  $\eta_{\text{th}}$ . (A different

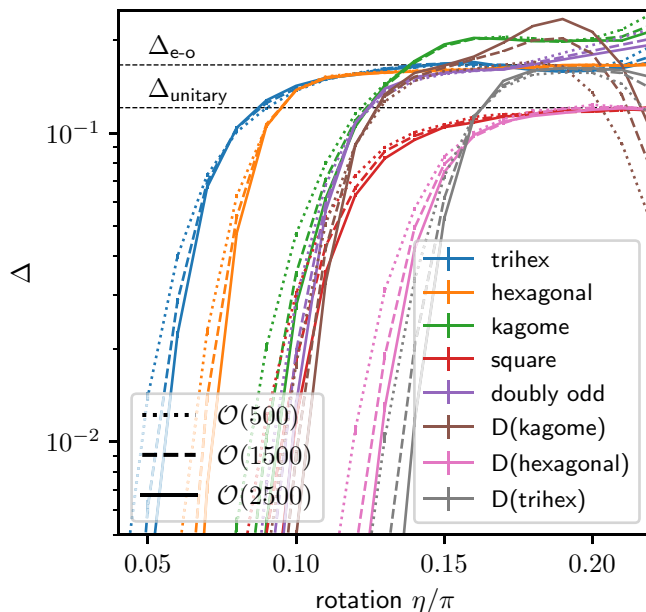


FIG. 12. The syndrome average  $\Delta$  [cf. Eq. (61)] for a decoder that chooses a Pauli correction optimizing  $\langle \delta_s^2 \rangle_\Omega$  for each  $s$ . Different colors correspond to different  $Z$ -stabilizer lattices. (D denotes the dual of a lattice.) The value of  $\Delta$  for strong rotations depends on the graph class; the estimate Eq. (62) [Eq. (63)] for lattices with even- and odd-weight stabilizers (even-weight stabilizer and odd-weight logical operators) is indicated by  $\Delta_{e-o}$  ( $\Delta_{\text{unitary}}$ ). The system sizes are the same as those considered in Fig. 7.

behavior sets in upon approaching the  $S$ -gate limit  $\eta \approx \pi/4$ ; we henceforth focus on  $\eta$  sufficiently below this value.) For  $\eta$  in this regime, we find that the codes with even-weight  $Z$  stabilizers and odd-weight  $Z_L$  have the smallest  $\Delta$ , followed by those with odd-weight  $Z$  stabilizers; those with even-weight  $Z$  stabilizers and even-weight  $Z_L$  have the largest  $\Delta$ .

The plateau value of  $\Delta$  for lattices with odd-weight  $Z$  stabilizers can be estimated by assuming that the ensemble resulting from the action of the coherent decoder on the initial state  $|+_L\rangle$  corresponds to states uniformly distributed over the Bloch hemisphere closest to  $|+_L\rangle$ . This results in

$$\Delta_{e-o} = \frac{2}{3} \frac{1}{2\pi} \int d\theta \sin \theta \int_{-\pi/2}^{\pi/2} d\phi |\langle \theta, \phi | -_L \rangle|^2 = \frac{1}{6}. \tag{62}$$

We can estimate the plateau of  $\Delta$  for lattices with even-weight  $Z$  stabilizers and odd-weight  $Z_L$  similarly. Since in these lattices the accessible states, starting from  $|+_L\rangle$ , are the part of the equator spanning  $|Y_L\rangle$  over  $|+_L\rangle$  to  $|-Y_L\rangle$ , the average is given by

$$\Delta_{\text{unitary}} = \frac{2}{3} \frac{1}{\pi} \int_{-\pi/2}^{\pi/2} d\phi \left| \left\langle \theta = \frac{\pi}{2}, \phi \middle| -_L \right\rangle \right|^2 = \frac{\pi - 2}{3\pi}. \tag{63}$$

Both of these values are indicated in Fig. 12 and fit well to the simulation. In the case of even-weight  $Z$  stabilizers and even-weight  $Z_L$ , an estimate based on the final states evenly distributed among the accessible states would give the same value as Eq. (63), but this is not the distribution we numerically observe. Instead, we find that the final states are increasingly concentrated around  $|0_L\rangle$  and  $|1_L\rangle$  upon increas-

ing  $\eta$ . This explains the significantly higher plateau value in comparison to the lattices with even-weight  $Z$  stabilizers and odd-weight  $Z_L$ .

We now investigate the existence of a decoherence threshold  $\eta_{\text{th}}^c$ . To this end, we study the code-distance dependence of  $\Delta$ . As already suggested by the final-state distributions in Fig. 11, we find qualitatively different behavior for graphs that include odd-weight  $Z$  stabilizers and for those with even-weight  $Z$  stabilizers. In the former case, using a similar fitting procedure as in Sec. VIII, we observe a decoherence threshold at  $\eta_{\text{th}}^c \gtrsim \eta_{\text{th}} + 0.01\pi$ : the value of  $\Delta$  decreases with code distance only for  $\eta < \eta_{\text{th}}^c$  but it increases for  $\eta > \eta_{\text{th}}^c$ . For graphs with even-weight  $Z$  stabilizers, we find that if a threshold exists, it is at a much higher value of  $\eta$ , however, we could not clearly establish threshold behavior.

These observations highlight that the sense in which increasing the code distance decoheres logical level noise depends on the graph class for  $\eta \gtrsim \eta_{\text{th}}$ . While for graphs with even-weight  $Z$  stabilizers our findings are consistent with the final-state distribution being increasingly well approximated by that resulting from a distribution of Pauli errors, for the complementary graph class,  $\eta_{\text{th}}^c \gtrsim \eta_{\text{th}}$  implies that the action of  $\mathcal{R}_s \circ \mathcal{E}$  on the logical subspace can retain significant coherence for  $\eta \gtrsim \eta_{\text{th}}$ ; the impact of the coherent part of the logical error is suppressed only upon averaging  $\rho_s$  over syndromes (i.e., only on the average-logical-channel level).

## XI. CONCLUSION

We described how the C4 encoding of qubits can be used to obtain a Majorana-fermion representation of surface codes on arbitrary planar graphs, and we characterized logical-state storage under coherent  $Z$  rotations (or coherent  $X$  rotations) using FLO-based simulations. These methods generalize the approach introduced for the square lattice by Ref. [24].

We studied surface codes on lattices with varying connectivity and estimated the average-logical-channel threshold values  $\eta_{\text{th}}$  of the rotation parameter. Comparing  $\eta_{\text{th}}$  to the thresholds  $\eta_{\text{th}}^p$  for the Pauli twirl of the physical-qubit coherent error, we found that while  $\eta_{\text{th}}$  and  $\eta_{\text{th}}^p$  are similar, the inequality  $\eta_{\text{th}} \leq \eta_{\text{th}}^p$  holds for all considered systems. We also found that, analogously to the case of incoherent Pauli noise, there is a trade-off between resilience against coherent  $X$  and  $Z$  rotations, depending on the graph connectivity. However, while for Pauli noise the thresholds against bit and phase flips approach a universal bound, Eq. (42), this is not the case for coherent errors. To demonstrate this, we have identified the doubly odd lattice that is self-dual (hence has the same threshold for  $Z$  and  $X$  rotations) just as the square lattice, but has higher  $\eta_{\text{th}}$  than the square lattice.

We also studied the properties of final states corresponding to individual syndrome measurements followed by recovery. These properties were found to follow a categorization of codes into three classes: those whose  $Z$  stabilizers include

odd-weight operators, those with only even-weight  $Z$  stabilizers and even-weight logical  $Z$  operator  $Z_L$ , and those with only even-weight  $Z$  stabilizers and odd-weight  $Z_L$ . The three classes correspond to three distinct patterns of accessible final states, as shown in Fig. 10(b).

The square lattice studied in Ref. [24] corresponds to the third class; it is only in this class that per-syndrome error and recovery  $\mathcal{R}_s \circ \mathcal{E}$  corresponds to a unitary  $Z_L$  rotation of logical states, with state-independent syndrome probability and rotation angle. In all other cases, the syndrome probability depends on the initial state. To assess the average case (in the sense of this dependence), we studied the distribution of the average infidelity conditioned on syndrome  $s$  [Fig. 11], and introduced a measure of coherence [Eq. (58) and Fig. 12] and the related coherent decoder. While for  $\eta < \eta_{\text{th}}$ , upon increasing the code distance the distributions are increasingly well approximated by those resulting from a distribution of Pauli errors, codes that include odd-weight  $Z$  stabilizers were found to display a decoherence threshold  $\eta_{\text{th}}^c \approx \eta_{\text{th}} + 0.01\pi$  above which increasing code distance increases the coherence of the logical-level noise. The sense in which logical-level noise decoheres for  $\eta \gtrsim \eta_{\text{th}}$  therefore depends on the graph class. In particular, for graphs with odd-weight  $Z$  stabilizers, the action of  $\mathcal{R}_s \circ \mathcal{E}$  on the logical subspace can retain significant coherence so the decoherence of the logical-level noise holds only upon averaging  $\rho_s$  over syndromes, i.e., only on the average-logical-channel level.

That correcting coherent errors is possible in all graph classes is an encouraging result. It shows that a unitary action for  $\mathcal{R}_s \circ \mathcal{E}$  in the logical subspace, as for the square-lattice case of Ref. [24], is not a key requirement. There are, however, several directions in which further studies are needed. For example, while we focused on the case where the same coherent error is applied to each qubit [i.e.,  $\eta_j = \eta$  in terms of Eq. (4)], our approach is not limited to this setting. A natural generalization of our study would therefore be to investigate the effects of random  $\eta_j$ ; this would provide another step toward a more realistic description of the errors one expects in physical devices. Furthermore, it remains an open question how systems behave under errors that go beyond the uniaxial rotations along one of the directions specified by the stabilizers (i.e.,  $X$  or  $Z$  rotations), including more general forms of coherent rotations or error models with the probabilistic occurrence of different coherent components such that the overall error is inequivalent to Pauli noise.

## ACKNOWLEDGMENTS

We thank S. Brierley for bringing coherent errors, in particular Ref. [24], to our attention, and for a number of encouraging conversations. We acknowledge useful discussions with J. Bausch, A. Farjami, J. M. Martinis, J. K. Pachos, and S. Subramanian. This research was supported by the European Commission via the ERC Starting Grant No. 678795 TopInSy.

[1] R. Barends, J. Kelly, A. Megrant, A. Veitia, D. Sank, E. Jeffrey, T. C. White, J. Mutus, A. G. Fowler, B. Campbell *et al.*, Superconducting quantum circuits at the surface code threshold for fault tolerance, *Nature* **508**, 500 (2014).

[2] F. Yan, S. Gustavsson, A. Kamal, J. Birenbaum, A. P. Sears, D. Hover, T. J. Gudmundsen, D. Rosenberg, G. Samach, S. Weber *et al.*, The flux qubit revisited to enhance coherence and reproducibility, *Nat. Commun.* **7**, 12964 (2016).

- [3] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, Superconducting qubits: Current state of play, *Annu. Rev. Condens. Matter Phys.* **11**, 369 (2020).
- [4] A. R. Calderbank and P. W. Shor, Good quantum error-correcting codes exist, *Phys. Rev. A* **54**, 1098 (1996).
- [5] A. M. Steane, Error Correcting Codes in Quantum Theory, *Phys. Rev. Lett.* **77**, 793 (1996).
- [6] J. Kelly, R. Barends, A. G. Fowler, A. Megrant, E. Jeffrey, T. C. White, D. Sank, J. Y. Mutus, B. Campbell, Y. Chen *et al.*, State preservation by repetitive error detection in a superconducting quantum circuit, *Nature* **519**, 66 (2015).
- [7] M. Takita, A. D. Córcoles, E. Magesan, B. Abdo, M. Brink, A. Cross, J. M. Chow, and J. M. Gambetta, Demonstration of Weight-Four Parity Measurements in the Surface Code Architecture, *Phys. Rev. Lett.* **117**, 210505 (2016).
- [8] B. M. Terhal, Quantum error correction for quantum memories, *Rev. Mod. Phys.* **87**, 307 (2015).
- [9] 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).
- [10] A. Y. Kitaev, Fault-tolerant quantum computation by anyons, *Ann. Phys.* **303**, 2 (2003).
- [11] 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).
- [12] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Topological quantum memory, *J. Math. Phys.* **43**, 4452 (2002).
- [13] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, *Phys. Rev. A* **70**, 052328 (2004).
- [14] A. G. Fowler, A. M. Stephens, and P. Groszkowski, High-threshold universal quantum computation on the surface code, *Phys. Rev. A* **80**, 052312 (2009).
- [15] D. S. Wang, A. G. Fowler, and L. C. L. Hollenberg, Surface code quantum computing with error rates over 1%, *Phys. Rev. A* **83**, 020302(R) (2011).
- [16] P. Aliferis, D. Gottesman, and J. Preskill, Quantum accuracy threshold for concatenated distance-3 codes, *Quantum Inf. Comput.* **6**, 97 (2006).
- [17] C. Chamberland, J. Wallman, S. Beale, and R. Laflamme, Hard decoding algorithm for optimizing thresholds under general Markovian noise, *Phys. Rev. A* **95**, 042332 (2017).
- [18] M. Gutiérrez, C. Smith, L. Lulushi, S. Janardan, and K. R. Brown, Errors and pseudo-thresholds for incoherent and coherent noise, *Phys. Rev. A* **94**, 042338 (2016).
- [19] Z. Cai, X. Xu, and S. C. Benjamin, Mitigating coherent noise using Pauli conjugation, *npj Quantum Inf.* **6**, 17 (2020).
- [20] D. Gottesman, Maximally sensitive sets of states, [arXiv:1907.05950](https://arxiv.org/abs/1907.05950).
- [21] Y. R. Sanders, J. J. Wallman, and B. C. Sanders, Bounding quantum gate error rate based on reported average fidelity, *New J. Phys.* **18**, 012002 (2015).
- [22] E. Huang, A. C. Doherty, and S. Flammia, Performance of quantum error correction with coherent errors, *Phys. Rev. A* **99**, 022313 (2019).
- [23] D. Greenbaum and Z. Dutton, Modeling coherent errors in quantum error correction, *Quantum Sci. Technol.* **3**, 015007 (2018).
- [24] S. Bravyi, M. Englbrecht, R. König, and N. Peard, Correcting coherent errors with surface codes, *npj Quantum Inf.* **4**, 55 (2018).
- [25] S. J. Beale, J. J. Wallman, M. Gutiérrez, K. R. Brown, and R. Laflamme, Quantum Error Correction Decoheres Noise, *Phys. Rev. Lett.* **121**, 190501 (2018).
- [26] J. K. Iverson and J. Preskill, Coherence in logical quantum channels, *New J. Phys.* **22**, 073066 (2020).
- [27] A. S. Darmawan and D. Poulin, Tensor-Network Simulations of the Surface Code under Realistic Noise, *Phys. Rev. Lett.* **119**, 040502 (2017).
- [28] A. Kitaev, Anyons in an exactly solved model and beyond, *Ann. Phys.* **321**, 2 (2006).
- [29] X.-G. Wen, Quantum Orders in an Exact Soluble Model, *Phys. Rev. Lett.* **90**, 016803 (2003).
- [30] B. M. Terhal and D. P. DiVincenzo, Classical simulation of noninteracting-fermion quantum circuits, *Phys. Rev. A* **65**, 032325 (2002).
- [31] K. Fujii and Y. Tokunaga, Error and loss tolerances of surface codes with general lattice structures, *Phys. Rev. A* **86**, 020303(R) (2012).
- [32] B. Röthlisberger, J. R. Wootton, R. M. Heath, J. K. Pachos, and D. Loss, Incoherent dynamics in the toric code subject to disorder, *Phys. Rev. A* **85**, 022313 (2012).
- [33] D. Gottesman, Stabilizer codes and quantum error correction, [arXiv:quant-ph/9705052](https://arxiv.org/abs/quant-ph/9705052).
- [34] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
- [35] A. Y. Kitaev, Quantum computations: Algorithms and error correction, *Russ. Math. Surv.* **52**, 1191 (1997).
- [36] A. G. Fowler, A. C. Whiteside, and L. C. L. Hollenberg, Towards Practical Classical Processing for the Surface Code, *Phys. Rev. Lett.* **108**, 180501 (2012).
- [37] J. Edmonds, Paths, Trees, and Flowers, *Can. J. Math.* **17**, 449 (1965).
- [38] B. Rahn, A. C. Doherty, and H. Mabuchi, Exact performance of concatenated quantum codes, *Phys. Rev. A* **66**, 032304 (2002).
- [39] S. Beigi and R. König, Simplified instantaneous non-local quantum computation with applications to position-based cryptography, *New J. Phys.* **13**, 093036 (2011).
- [40] J. J. Wallman and S. T. Flammia, Randomized benchmarking with confidence, *New J. Phys.* **16**, 103032 (2014).
- [41] J. J. Wallman, Bounding experimental quantum error rates relative to fault-tolerant thresholds, [arXiv:1511.00727](https://arxiv.org/abs/1511.00727).
- [42] P. W. Kasteleyn, Graph theory and theoretical physics, in *Graph Theory and Crystal Physics*, edited by F. Harary (Academic-Press, New York, 1967), pp. 43–110.
- [43] N. Johnston, D. W. Kribs, and V. I. Paulsen, Computing stabilized norms for quantum operations via the theory of completely bounded maps, *Quantum Inf. Comput.* **9**, 16 (2009).
- [44] G. Duclos-Cianci and D. Poulin, Fast Decoders for Topological Quantum Codes, *Phys. Rev. Lett.* **104**, 050504 (2010).
- [45] J. R. Wootton and D. Loss, High Threshold Error Correction for the Surface Code, *Phys. Rev. Lett.* **109**, 160503 (2012).
- [46] A. Hutter, J. R. Wootton, and D. Loss, Efficient Markov chain Monte Carlo algorithm for the surface code, *Phys. Rev. A* **89**, 022326 (2014).
- [47] S. Bravyi, M. Suchara, and A. Vargo, Efficient algorithms for maximum likelihood decoding in the surface code, *Phys. Rev. A* **90**, 032326 (2014).