# Majorana loop stabilizer codes for error correction of fermionic quantum simulations

Zhang Jiang,[1,*] Jarrod McClean,[1] Ryan Babbush,[1] and Hartmut Neven[1]

[1] *Google LLC, Venice, CA 90291*

(Dated: December 21, 2018)

Locality-preserving fermion-to-qubit mappings are especially useful for simulating lattice fermion models (e.g., the Hubbard model) on a quantum computer. They avoid the overhead associated with non-local parity terms in mappings such as the Jordan-Wigner transformation. As a result, they often provide solutions with lower circuit depth and gate complexity. A major obstacle to achieving near-term quantum computation is quantum noises. Interestingly, locality-preserving mappings encode the fermionic state in the common +1 eigenstate of a set of stabilizers, akin to quantum error-correcting codes. Here, we discuss a couple of known locality-preserving mappings and their abilities to correct/detect single-qubit errors. We also introduce a locality-preserving map, whose stabilizers are products of Majorana operators on closed paths of the fermionic hopping graph. The code can correct all single-qubit errors on a 2-dimensional square lattice, while previous locality-preserving codes can only detect single-qubit errors on the same lattice. Our code also has the advantage of having lower-weight logical operators. We expect that error-mitigating schemes with low overhead to be useful to the success of near-term quantum algorithms such as the variational quantum eigensolver.



We are closer to realizing the potential of quantum computation [1–3] with the recent rapid advances in quantum computing devices such as ion traps [4–6] and superconducting qubits [7, 8]. While performing fault-tolerant quantum computation [9] is widely regarded as the ultimate goal, the substantial overheads prevent it from being implemented in the immediate future [10, 11]. In the meantime, error-mitigation schemes are likely to be an essential component for the success of near-term quantum algorithms [12–19].

Fermionic systems must be mapped to spin systems before they can be simulated on a digital quantum computer [20]. Mappings such as the Jordan-Wigner (JW) transformation introduce nonlocal parity terms when the spatial dimension is greater than one. These terms add considerable overhead to quantum simulations of local fermionic systems. Clever circuit compilation methods have been introduced to reduce the overhead of the parity terms on digital quantum computers with all-to-all connections [21, 22].

The non-locality resulting from parity becomes more prominent on near-term quantum devices, where only geometrically local two-qubit gates are available [23, 24]. To overcome this difficulty, the fermionic SWAP gate can be used to bring together fermonic modes encoded far apart in the JW transformation [25]. It was shown that fermionic SWAP network offers asymptotically optimal solution to simulating quantum chemistry problems in terms of circuit depth [26–28]. For other problems, such as the two-dimensional (2D) fermionic Fourier transformation, the fermionic SWAP network is not optimal; this transformation can be implemented with a quadratically shorter circuit depth by going between different bases [29].

Locality-preserving fermion-to-qubit mappings avoid the parity terms and can be extremely useful for simulating geometrically local fermionic systems on a quantum computer. Bravyi and Kitaev [25] discovered a way to map fermionic operators defined on a graph to qubit operators on its vertex-to-edge dual graph; a stabilizer operator (also called gauge operator in the literature) is introduced for each closed path in the graph. The fermionic state is encoded in the common +1 eigenspace of the stabilizers. This mapping is called Bravyi-Kitaev superfast (BKSF) transformation [30] or superfast encoding [31], referring to "superfast simulation of fermions on a graph." It is not to be confused with the widely known Bravyi-Kitaev (BK) transformation introduced in the same paper [25]; the difference is that the BK transformation does not preserve locality and requires no more qubits than the number of fermionic modes.

Locality-preserving maps are closely related to lattice gauge theories. Levin and Wen [32, 33] showed that particular fermionic lattice models can be described in terms of gauge fields in lattice spin models. Ball [34] extended the Levin-Wen result to general fermionic hopping Hamiltonians by introducing auxiliary Majorana modes; Verstraete and Cirac [35] discussed a similar map for fermions on $d$-dimensional square lattices, where $d-1$ auxiliary qubits are introduced for each fermionic mode. This approach is called the Ball-Verstraete-Cirac (BVC) transformation or the auxiliary fermion approach. Chen *et al.* introduced a map similar to the one by BKSF with examples from free fermions on square and honeycomb lattices to the Hubbard model [36]. Recently, Steudtner and Wehner [37] constructed a class of locality-preserving fermion-to-qubit mappings by concatenating the JW transformation with quantum codes. Some applications of locality-preserving mappings to quantum simulation were discussed in [30, 38].

Locality-preserving fermion-to-qubit mappings share many similarities with quantum error-correcting codes,

* qzj@google.com

where the fermionic states are encoded in the common +1 eigenspace of the stabilizer operators. Very recently, Setia *et al.* [31] discussed a generalized version of the BKSF transformation and their relations to error-correcting codes. We expect error-mitigating methods to be useful in near-term quantum algorithms, such as the variational quantum eigensolver (VQE) [39]. A crucial step in either VQE or phase estimation is to prepare the initial state, which is often taken to be the ground state of a meanfield approach such as Hartree-Fock. In chemistry, the difference between the meanfield energy and the exact energy is referred to as the "correlation energy", and despite determining essential elements of chemical bonding, it often only amounts to between 1-5% of the total energy [40, 41]. Physically, this is because a large fraction of the energy comes from core electrons that contribute very little directly to essential chemical bonding. However, because of the energy scale difference, even small errors in this part of the calculation can overwhelm any refinements a quantum computer can provide. Therefore, even a 1% error in the preparation of the mean-field state could render any correlations added by the rest of the circuit fruitless. One can drastically reduce the error rate in state preparation by using an error-detecting code and post selecting the measurement outcomes [14, 15, 18, 19]. Furthermore, error-detecting codes can also be used for quantum error suppression by creating an energy gap between the code subspace and the orthogonal subspace [42–44].

A no-go theorem was proven recently in [31] showing that the BKSF defined on graphs with vertex degree $d \leq 6$ cannot correct all single-qubit errors. We introduce a new type of locality-preserving mappings based on the BKSF, called Majorana loop stabilizer code (MLSC). The MLSC on a square lattice (vertex degree 4) can correct all single-qubit errors while the BKSF can only detect single-qubit errors. Moreover, the logical operators in MLSC have lower weights than those in BKSF, making it easier to implement.

In Sec. I, we review some properties of the Majorana fermions that are crucial to the rest of paper. In Sec. II, we review the BKSF and how it can be used to detect single-qubit errors. In Sec. III, we introduce the Majorana loop stabilizer code and discuss how to use them to correct single-qubit errors. In Sec. IV, we discuss how to encode a single Slater determinant with locality preserving mappings. In Sec. V, we conclude the paper. In App. A, we briefly review quantum error-correcting codes. In Apps. B and C, we discuss how to construct quantum error-correcting/detecting codes from the auxiliary fermion approaches.

## I. MAJORANA OPERATORS

In this section, we review some properties of the Majorana operators which will be used to construct the mappings in the later sections. The Majorana operators have the advantage of being Hermitian and unitary at the same time (self-inverse). They also allow for treating all the fermionic operators on an equal footing. The single-mode Majorana operators are defined as linear combinations of the fermionic ladder operators

$$f_{2k} = c_k^\dagger + c_k\,, \quad f_{2k+1} = i(c_k^\dagger - c_k)\,, \tag{1}$$

where $c_k^\dagger$ and $c_k$ are the fermionic creation and annihilation operators, respectively. They satisfy the simple relations

$$f_k^\dagger = f_k\,, \quad \{\, f_j,\, f_k \,\} = 2\delta_{j,k}\,, \tag{2}$$

where $\{\cdot\,,\,\cdot\}$ is the anticommutator and $\delta$ the Kronecker delta function.

The single-mode Majorana operators $f_0, f_1, \ldots, f_{2n-1}$, together with the phase factor $i$, generate a group $\mathcal{M}_{2n}$. An arbitrary element in $\mathcal{M}_{2n}$ takes the form

$$\varphi\, f_A\,, \qquad f_A = \prod_{k\in A} f_k\,, \tag{3}$$

where $\varphi \in \{\pm 1, \pm i\}$ is an overall phase factor. The set $A \subseteq \{0, 1, \ldots, 2n-1\}$ is the support of the Majorana operator. The order of the operators in the product $f_A$ matters, and we follow the convention that the index $k$ increases monotonically from left to right. The weight of a Majorana operator equals the number of fermionic modes that it acts nontrivially on, i.e., the weight of $f_A$ is $|A|$. Two Majorana operators $f_A$ and $f_B$ either commute or anticommute,

$$f_A f_B = (-1)^{|A|\cdot|B|+|A\cap B|}\, f_B f_A\,. \tag{4}$$

The commutation relationship is determined by the parity of the overlap $|A \cap B|$ when either $|A|$ or $|B|$ is even.

In terms of Majorana operators, the fermionic occupation operator reads

$$c_k^\dagger c_k = \frac{1}{2}\Big(\mathbb{1} + i f_{2k} f_{2k+1}\Big)\,, \tag{5}$$

where $\mathbb{1}$ is the identity operator. The fermionic hopping term takes the form

$$c_j^\dagger c_k + \text{h.c.} = \frac{i}{2}\Big(f_{2j} f_{2k+1} - f_{2j+1} f_{2k}\Big)\,. \tag{6}$$

We focus on problems where fermions can only hop between neighboring vertices of a undirected graph $G(V, E)$, with vertices $V$ and edges $E$. Consider the following set of quadratic Majorana operators

$$\eta_k = i f_{2k} f_{2k+1}\,, \qquad \text{for each vertex } k \in V, \tag{7a}$$

$$\xi_{jk} = i f_{2j} f_{2k}\,, \qquad \text{for each edge } (j,\, k) \in E. \tag{7b}$$

The occupation term (5) and the hopping term (6) can

be expressed as

$$c_k^\dagger c_k = \frac{1}{2}\big(\mathbb{1} + \eta_k\big)\,, \tag{8a}$$

$$c_j^\dagger c_k + \text{h.c.} = -\frac{i}{2}\big(\xi_{jk}\eta_k + \eta_j\xi_{jk}\big)\,. \tag{8b}$$

The operators in Eq. (7) generate a group $\mathcal{M}_{2n}^{\text{even}}$, consisting of Majorana operators of even weights. Any physical fermionic Hamiltonian can be written as a sum of elements in the group $\mathcal{M}_{2n}^{\text{even}}$.

The quadratic Majorana generators in Eq. (7) are both Hermitian and unitary. They satisfy the commutation relations

$$\eta_j\eta_k = \eta_k\eta_j\,, \tag{9a}$$

$$\eta_l\xi_{jk} = (-1)^{\delta_{jl}+\delta_{kl}}\xi_{jk}\eta_l\,, \tag{9b}$$

$$\xi_{lm}\xi_{jk} = (-1)^{\delta_{jl}+\delta_{kl}+\delta_{jm}+\delta_{km}}\xi_{jk}\xi_{lm}\,. \tag{9c}$$

To have a complete description of the algebra, we need to introduce the following condition on any closed path in the graph $G$,

$$(-i)^\ell\,\xi_{k_0k_1}\xi_{k_1k_2}\cdots\xi_{k_{\ell-1}k_0} = \mathbb{1}\,, \tag{10}$$

where $\ell$ is the length of the path. This relation implies that the generators in Eq. (9) are not independent. The total number of independent closed paths is $N_E - N_V + 1$, where $N_V$ ($N_E$) is the total number of vertices (edges) of the graph $G$. In the next section, we will see that the condition (10) is not automatically satisfied when the fermion operators are mapped to qubit operators.

## II. THE BRAVYI-KITAEV SUPERFAST TRANSFORMATION

In this section, we first review the fermion-to-qubit mapping introduced by Bravyi and Kitaev that conserves geometry locality [25]. This mapping was called Bravyi-Kitaev superfast (BKSF) transformation [30] or superfast encoding [31], referring to "superfast simulation of fermions on a graph." Concerning the Hubbard model, it was shown that the BKSF has an advantage over the JW transformation and the BK transformation in terms of total number of gates [45]. Here, we also show how to use the BKSF to detect (but not correct) all single-qubit errors on a 2D square lattice.

Bravyi and Kitaev considered fermionic hopping problem defined on a graphs $G$, where fermions can hop between neighboring vertices. One qubit is introduced for each edge of the graph $G$, and the fermionic state is encoded in a subspace of the qubits, see Fig. 1. The goal is to find qubit operators that share the same algebra as the fermionic operators, e.g., Eqs. (9) and (10). To achieve that, we first choose an arbitrary ordering of the incident edges for each vertex of $G$, e.g., $(l,k) < (j,k)$ means that the edge $(l,k)$ is placed before $(j,k)$ among

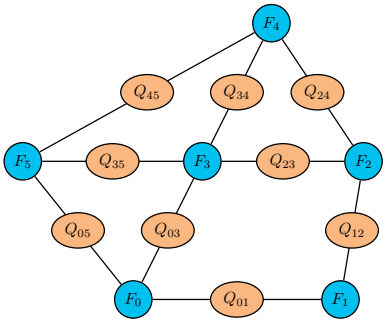


FIG. 1. BKSF on a graph: the vertices (blue circles) of the graph represent fermionic modes; fermions can hop between neighboring vertices. One qubit (orange ellipses) is introduced for each edge of the graph. The fermionic state is mapped to a subspace of the qubits.

all the incident edges of the vertex $k$. The quadratic Majorana operators in Eq. (9) can be mapped to the qubit operators by

$$\eta_k \mapsto \tilde{\eta}_k = \prod_{j:\,(j,k)\in E} Z_{jk}\,, \tag{11}$$

$$\xi_{jk} \mapsto \tilde{\xi}_{jk} = \epsilon_{jk}X_{jk}\prod_{\substack{l:\,(l,k)<(j,k);\\ (l,k)\in E}} Z_{lk}\prod_{\substack{m:\,(m,j)<(k,j);\\ (m,j)\in E}} Z_{mj}\,, \tag{12}$$

where the antisymmetric coefficients $\epsilon_{jk} = -\epsilon_{kj} = \pm 1$ can be chosen arbitrarily for each edge $(j,\,k)$. The qubit operators $\tilde{\eta}_k$ and $\tilde{\xi}_{jk}$ satisfy the same commutation relations as those of the fermionic operators in Eq. (9).

The qubit operator corresponding to the conserved quantity in Eq. (10) is

$$S_{k_0k_1\cdots k_{\ell-1}} = (-i)^\ell\,\tilde{\xi}_{k_0k_1}\tilde{\xi}_{k_1k_2}\cdots\tilde{\xi}_{k_{\ell-1}k_0}\,, \tag{13}$$

which remains unchanged under any cyclic permutation of the indices $k_0, k_1, \ldots, k_{\ell-1}$. These loop operators form an Abelian group, and the fermionic state is encoded in their common $+1$ eigenspace. We refer these operators as stabilizer operators. They are also called gauge operators in the literature, not to be confused with the noncommuting gauge operators in subsystem quantum error-correcting codes [46]. As a consequence of the commutation relations (9), the stabilizers commute with all the logical operators $\tilde{\eta}_k$ and $\tilde{\xi}_{jk}$. They also commute with each other, and their common $+1$ eigenspace define the code subspace $\mathcal{C}$,

$$|\,\psi\,\rangle \in \mathcal{C}\,,\ \text{if and only if}\ S_{k_0k_1\cdots k_{\ell-1}}|\,\psi\,\rangle = |\,\psi\,\rangle\,, \tag{14}$$

for all closed paths $(k_0, k_1, \ldots, k_{\ell-1}, k_0)$ in the graph $G$. Hence the restriction of the closed loop stabilizers to the code space act as the identity, as in the case of the Majorana operators.

The mapping (11) artificially introduces an extra con-

served quantity,

$$\prod_{k=0}^{N_V-1} \tilde{\eta}_k = \mathbb{1}\,, \tag{15}$$

which corresponds to the even-parity subspace of the original fermionic system. To simulate the odd-parity subspace, one can change the sign in the mapping (11) for a particular value of $k$, say $k=0$,

$$\tilde{\eta}_k \mapsto (-1)^{\delta_{k,0}} \prod_{j:\,(j,k)\in E} Z_{jk}\,. \tag{16}$$

The stabilizer operators in the BKSF can be used for quantum error detection/correction if implemented properly. This ideas were also discussed very recently in [38], where a no-go theorem was proved: the BKSF cannot correct all single-qubit errors if $d \leq 6$, where $d$ is the vertex degree of the graph $G$. Here, we show that BKSF can detect all single-qubit errors on a 2D square lattice by properly choosing the orders of the incident edges.

Consider the BKSF encoding of spinless fermionic modes on a 2D lattice, see Fig. 2; the ordering of the edges associated with a vertex is indicated by the numbers next to it. We use the convention that $\epsilon_{jk} = 1$ if the

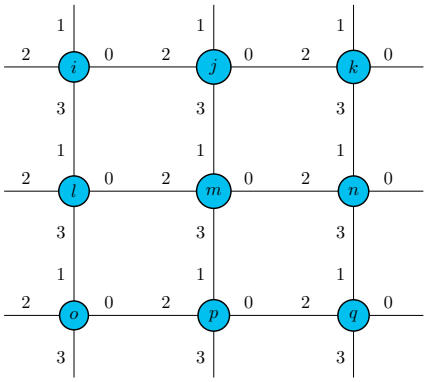


FIG. 2. BKSF on a 2D lattice with a specific ordering of the incident edges. This code can detect all single-qubit errors.

site $k$ is to the right or below the site $j$; under this convention, we have $\epsilon_{k_0k_1}\epsilon_{k_1k_2}\cdots\epsilon_{k_{\ell-1}k_0} = 1$ for any closed path. The closed paths containing the smallest number of edges are plaquettes. The stabilizer operator corresponding to the plaquette $(m,n,q,p,m)$ is the product of the corresponding edge operators, see Fig 3,

$$\begin{aligned} S_{mnqp} &= \tilde{\xi}_{mn}\,\tilde{\xi}_{nq}\,\tilde{\xi}_{qp}\,\tilde{\xi}_{pm} \\ &= X_{mn}X_{nq}Z_{nq}X_{qp}Z_{qp}X_{pm}Z_{jm}Z_{lm} \\ &= -X_{mn}Y_{nq}Y_{qp}X_{pm}Z_{jm}Z_{lm}\,, \end{aligned} \tag{17}$$

where we used the condition $\epsilon_{mn}\epsilon_{nq}\epsilon_{qp}\epsilon_{pm} = +1$. These plaquette operators generate the stabilizer group,

$$\mathcal{S} = \left\{ S_{k_0k_1k_2k_3} \mid \text{for all plaquettes} \right\}. \tag{18}$$

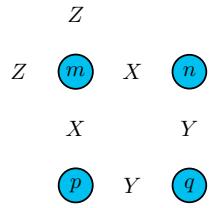


FIG. 3. Plaquette stabilizer for the BKSF code given the vertex and edge ordering in Fig 2, up to an overall minus sign. From the chosen ordering, the above figure specifies the stabilizer that results from any generating plaquette in the graph. In the case of the infinite or periodic lattice, it is uniform. In the case of a finite lattice, the structure stays roughly the same, however the dangling $Z$ operators are removed in the case there is no edge they connect to in the graph, and the adjacent edges may change from $X$ to $Y$.

The syndrome of a Pauli error can be described by the set of plaquette stabilizers with which it anticommutes. The single-qubit error $Z_{lm}$ anticommutes with two stabilizers, $S_{ijml}$ and $S_{lmpo}$. In general, a Pauli-$Z$ error anticommutes with all the stabilizers involving the edge it acting on. The single-qubit error $X_{lm}$ anticommutes with only two stabilizers, $S_{ijml}$ and $S_{mnqp}$. The same holds true for the single-qubit error $X_{jm}$; therefore, these two errors have the same syndrome and can only be detected but not corrected. The Pauli error $Y_{lm}$ anticommutes with $S_{lmpo}$ and $S_{mnqp}$, which has the same syndrome as $Z_{mp}$. In App. F, we list the syndromes of all the single qubit errors for the BKSF on an infinite lattice with the specific edge ordering in Fig. 2. These results show that all single-qubit errors can be detected, but some of which share the same syndromes and cannot be corrected. One has to be careful for open-boundary conditions; for example, the Pauli-$Y$ errors at the bottom of an open lattice commute with all stabilizers and cannot be even detected.

## III. MAJORANA LOOP STABILIZER CODES

A nature question to ask is whether the BKSF on a square lattice can correct all single-qubit errors. The no-go thereom in [31] implies that the BKSF on a 2D (or even 3D) square lattice cannot correct all single-qubit errors. An example was also given in [31] to get around the no-go theorem lattice by introducing extra ancillary vertices and edges, which increase the degree of the original vertices from 4 to 8 on 2D square lattices. Here, we introduce a locality-preserving map that correct all single-qubit errors on 2D square lattices, without introducing any additional vertices or edges. Compared to the BKSF, logical operations can also be implemented with lower-weight Pauli operators with our code.

Similar to the BKSF, the logical operators in our code satisfy the commutation relations of the quadratic Majorana operators in Eq. (9). The stabilizers (13) of the code also correspond to the products of Majorana opera-

tors on a loop. Therefore, we call such a code Majorana loop stabilizer code (MLSC); the BKSF is a special case of the MLSC. The logical operators are allowed to be non-uniform with respect to vertices as well as to involve Pauli-$Y$ operators, which allows the MLSC to evade the no-go theorem in [31]. We discuss the code using the 2D square lattice as an example, see Fig. 4; each circle therein represents a vertex (fermionic mode), and its color indicates the type of the vertex. There is one stabilizer operator associated with each plaquette (4 vertices), and its color indicates the type of the stabilizer.

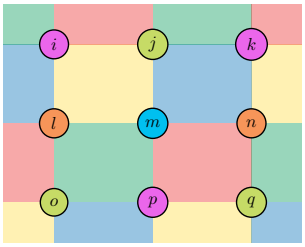


FIG. 4. The structure of the code: each circle represents a vertex in the 2D lattice; its color denotes one of the four type of vertex. Similarly, the color of a plaquette indicates the type of the corresponding stabilizer.

The logical operators and the stabilizer related to a red plaquette, e.g., the one consisting of the vertices $m$, $n$, $p$, and $q$, are plotted in Fig. 5. The Pauli operators $X$, $Y$, and $Z$ therein act on the qubits on the corresponding edges. Each quadratic Majorana operator in Eq. (7) is mapped to a product of single-qubit Pauli operators linked by a line. The vertex operator $\eta_m$ is mapped to a product of three Pauli-$Z$ operators linked by a curved dashed line around the site $m$, and similar for the other vertex operators. The edge operator $\xi_{mn}$ is mapped to the Pauli operator $Z\otimes X\otimes Y\otimes Z$ on the upper dotted line, where the only Pauli-$X$ operator acts on the edge $(m,n)$. The same rule applies to the remaining edges operators $\xi$, where the Pauli-$X$ operator always acts on the corresponding edge of $\xi$. These operators satisfy the commutation relations (9), which we checked numerically. The stabilizer associated to the plaquette is plotted in Fig. 6a, up to an overall minus sign.

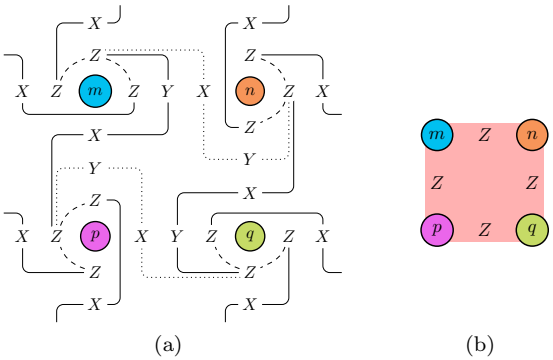


FIG. 5. (a) The logical operators associated to the red plaquette in Fig. 4. Similar to the BKSF, there is one physical qubit situated on each edge of the lattice, and $X$, $Y$, and $Z$ denote the corresponding Pauli operators. Each vertex operator $\eta$ is mapped to a product of three Pauli-$Z$ operators linked by a curved dashed line. Each edge operator $\xi$ is mapped to a product of single-qubit Pauli operators linked by a solid/dotted line, where the unique Pauli-$X$ operator acts on the corresponding edge of $\xi$. The two rightmost (leftmost) Pauli-$X$ operators are shared by the vertices $n$ and $q$ ($m$ and $p$) and the their right (left) neighbors, and similarly for the two top (bottom) Pauli-$X$ operators. This plaquette has a rotational symmetry of order 4 (unchanged by a rotation of angle $\pi/2$). (b) The stabilizer operator $S_{mnqp}$ associated to the plaquette up to a minus sign.

$$S_{mnqp} = \tilde{\xi}_{mn}\,\tilde{\xi}_{nq}\,\tilde{\xi}_{qp}\,\tilde{\xi}_{pm} = -Z_{mn}Z_{nq}Z_{qp}Z_{pm}\,, \qquad (19)$$

where we use the same definition (13) and convention as in the BKSF example; $\epsilon_{jk}=1$ if the site $k$ is to the right or below the site $j$. We refer terms such as $\xi_{mn}$, $\xi_{mn}\eta_m$, $\xi_{mn}\eta_n$, or $\xi_{mn}\eta_m\eta_n$ as generalized edge operators. The weight of any generalized edge operator in the red plaquette is greater or equal to three. This is a necessary condition for correcting all single-qubit errors. The BKSF violates this condition on a 2D lattice no matter how the incident edges are ordered.

The entire code can be constructed by putting together the four types of vertices illustrated in Fig. 5a, where each Pauli-$X$ operator is shared by two neighboring vertices. For clarity, we plot the logical operators in the green plaquette consisting of the vertices $l$, $m$, $o$, and $p$ in Fig. 6; the logical operators and the stabilizer of the yellow plaquette consisting of the vertices $i$, $j$, $l$, and $m$ are plotted in Fig. 7; the logical operators of the blue plaquette consisting of the vertices $j$, $k$, $m$, and $n$ are plotted in Fig. 8. Any generalized edge operator in the green, yellow, or blue plaquette also has weight $w\geq 3$.

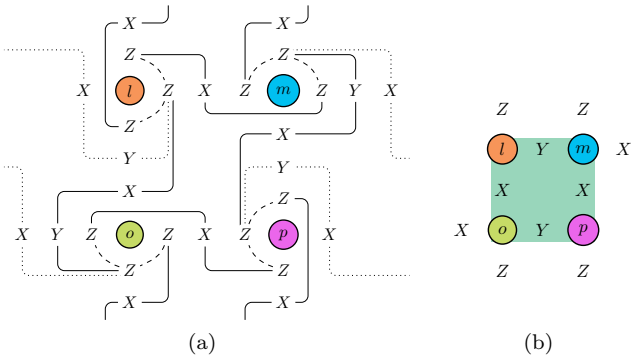


FIG. 6. (a) The logical operators of the green plaquette consisting of the vertices $l$, $m$, $o$, and $p$. This plaquette has a rotational symmetry of order 2 (unchanged by a rotation of an angle $\pi$). (b) The stabilizer $S_{lmpo}$ of the plaquette up to a minus sign.

We now go a step further to argue that any logical operator $M\notin\mathcal{S}$ must have weight $w\geq 3$. Such an operator can be specified as the product (up to an overall

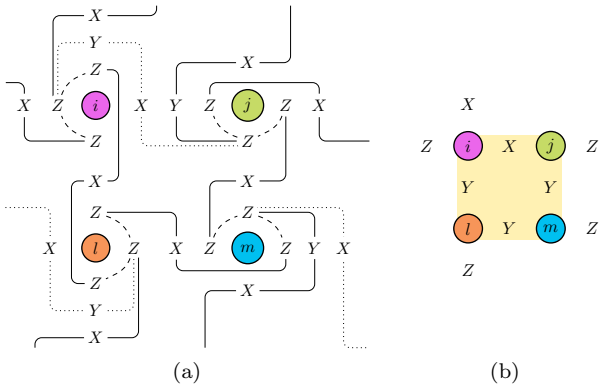


FIG. 7. (a) The logical operators of the yellow plaquette consisting of the vertices $i$, $j$, $l$, $m$. (b) The corresponding stabilizer $S_{ijml}$ up to a minus sign.

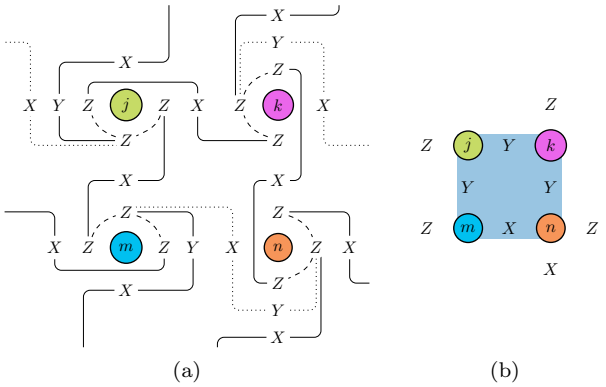


FIG. 8. (a) The logical operators of the blue plaquette consisting of vertices $j$, $k$, $m$, and $n$. These logical operators are identical to those in Fig. 7a by a $\pi$ rotation. (b) The corresponding stabilizer $S_{jknm}$ up to a minus sign, which is identical to the one in Fig. 7b by a $\pi$ rotation.

sign)

$$M = \prod_{(j,k)\in E_M} \tilde{\xi}_{jk} \prod_{l\in V_M} \tilde{\eta}_l \,, \tag{20}$$

where $E_M$ and $V_M$ are the sets of edges and vertices that determines $M$. Without loss of generality, we assume that for any vertex $j \in V_M$ there must be an edge $(j,k) \in E_M$ for some vertex $k$; otherwise, the weight of $M$ can be reduced by removing $j$ from $V_M$. We refer a vertex $j$ as a single-paired vertex if exactly one of its incident edges is in $E_M$; at least one Pauli-$Z$ operator on the other three incident edges of $j$ contributes to $M$ no matter whether $j \in V_M$ or not. If there are more than one single-paired vertex in $V_M$, the weight of $M$ must be greater or equal to three; this is because there must be some Pauli-$X$ or $Y$ components in $M$ to make it anticommute with the vertex operators on the single-paired vertices, besides at least two Pauli-$Z$ operators associated with the single-paired vertices. When $V_M$ contain two edges, the only case with less-than-two single-paired vertices is two parallel edges sitting next to each other. One can check that the weights of such logical operators are always greater or equal to three in Figs. 5-8. The specific choice of the vertex layout in Fig. 4 is important to preventing the opposite from happening.

To make the discussion of the syndromes of the single-qubit errors easier, we label the stabilizers in Fig. 9. The

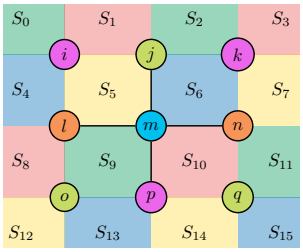

FIG. 9. The stabilizers on the plaquettes for syndrome detection.

syndrome of a single-qubit error can be described by the stabilizer operators that it anticommutes with. Here, we list the syndromes of the single-qubit errors on all the incident edges of the vertex $m$ in Fig. 9,

$$X_{mp} : S_{10},\, S_{13}, \qquad Z_{mp} : S_9,\, S_{14}\,, \tag{21}$$

$$X_{lm} : S_4,\, S_5, S_6,\, S_9, \qquad Z_{lm} : S_5,\, S_9\,, \tag{22}$$

$$X_{mn} : S_5,\, S_{10}, \qquad Z_{mn} : S_6,\, S_9\,, \tag{23}$$

$$X_{jm} : S_2,\, S_5,\, S_6,\, S_9, \qquad Z_{jm} : S_5,\, S_6\,. \tag{24}$$

These single-qubit errors (Pauli-$Y$ errors not listed) have different syndromes. We have checked that all the single-qubit errors have different syndromes on a $8 \times 8$ lattice with periodic boundary conditions (128 qubits).

To simulate lattice fermion problems, we often need to implement the hopping term $c_j^\dagger c_k + c_k^\dagger c_j$ and the occupation term $c_k^\dagger c_k$, see Eqs. (8a) and (8b). With the 2D BKSF in Fig. 2, both the vertical and horizontal hopping terms are mapped to qubit operators of weight 6. With our code, the hopping terms are mapped to qubit operators of weight no more than 4 (sometimes 3); the occupation terms can also be implemented with lower-weight qubit operators. Being able to implement logical operations with low-weight Pauli operators while correcting all single-qubit errors is a desired feature for near-term quantum error-mitigating schemes. In Fig. 10, we summarize the differences between our code and the BKSF on a 2D lattice.

## IV. ENCODING A SLATER DETERMINANT

In this section, we discuss how to encode a single Slater determinant with MLSC in circuit depth $O(1)$. For simplicity, we only consider Slater determinants in the computational basis; arbitrary Slater determinants can be

| | distance | occupation | hopping |
|---|---|---|---|
| BKSF | 2 | 4 | 6 |
| Our code | 3 | 3 | 4 (3) |

FIG. 10. A comparison of our code to BKSF on a 2D square lattice in terms of the code distance, weights of the occupation terms, and weights of the hopping terms.. The code here exhibits improved distance and weight properties.

prepared by applying single-particle basis transformations to these determinants. The encoded Slater determinant is a common eigenstate of the stabilizers and the vertex operators,

$$S_\ell |\,\psi_{\rm det}\,\rangle = |\,\psi_{\rm det}\,\rangle, \text{ for all } \ell \in L\,, \tag{25}$$

$$\tilde{\eta}_k |\,\psi_{\rm det}\,\rangle = z_k |\,\psi_{\rm det}\,\rangle, \text{ for all } k \in V\,, \tag{26}$$

where $L$ consists of a complete set of independent closed paths in $G$, and $z_k = \pm 1$ indicates whether the $k$th fermionic site is unoccupied or occupied. We start from a product state in the computational basis

$$Z_{jk} |\,\psi_{\rm prod}\,\rangle = z_{jk} |\,\psi_{\rm prod}\,\rangle\,, \tag{27}$$

where $Z_{jk}$ is the Pauli-$Z$ operator on the edge $(j,k)$.

We first consider the case of BKSF. The eigenvalues $z_{jk} = z_{kj}$ are chosen such that

$$\prod_{j:\,(j,k)\in E} z_{jk} = z_k\,. \tag{28}$$

There are some freedoms in assigning the values $z_{jk}$; one for each independent closed paths in $G$. The constraints in Eq. 28 can be converted to a system of linear equations over $GF(2)$ by mapping $z_{jk} \pm 1$ to the binary values 0 and 1. When a Hamiltonian path $(0, 1, \ldots, \ell-1)$ of the graph $G$ is known, we can assign the values of $z_{jk}$ in the Hamiltonian path according to

$$z_{0,1} = z_0\,, \quad z_{k,k+1} = z_k\, z_{k-1,k}\,, \tag{29}$$

where $k = 1, 2, \ldots, \ell-2$. We then assign the value $+1$ to all of the remaining $z_{jk}$, and the condition (28) is satisfied. More generally, we can assign values to $z_{jk}$ with the following four steps: 1. find a spanning tree $T$ of the graph $G$, which takes linear time in $|V|$, 2. assign the values in $T$ starting from the leaves towards the root; if $j$ is a parent node of $k$, assign $z_{jk}$ to be the product of $z_k$ and $z_{kl}$ for all the children $l$ of $k$, 3. set the value of the remaining edges that are not in $T$ to $+1$. This algorithm requires that the total parity of the vertex operators to be even as each Pauli-$Z$ on the edges of the spanning tree appears in two vertex operators. In App. D, we also present an algorithm to assign values to $z_{jk}$ without using a Hamiltonian path or a spanning tree.

With the MLSC in Sec. III, not all the incident edges participate in the vertex operator $\tilde{\eta}_k$, and the condition (28) is modified to

$$\prod_{j:\,(j,k)\in E_k} z_{jk} = z_k\,, \quad \text{for all } k \in V\,, \tag{30}$$

where $E_k$ is the set of edges that are involved in $\tilde{\eta}_k$. We use the notion $E'$ to denote the set of edges $(j,k)$ such that $(j,k) \in E_j$ and $(j,k) \in E_k$. One can assign values to $z_{jk}$ with the following four steps: 1. check the total parity of the vertex operators for each connected part of the graph $(V, E')$, 2. if the parity is odd, find an edge $(j,k)$ in the connected part such that $(j,k)$ is involved in $\tilde{\eta}_j$ but not $\tilde{\eta}_k$, set the value of $z_{jk}$ to $-1$, 3. find a spanning tree of the connected part and use the algorithm for BKSF to assign the values to the edges in the spanning tree, 4. after going through all the connected parts, set the values of the remaining edges to $+1$.

The resulting state $|\,\psi_{\rm prod}\,\rangle$ is an eigenstate of all the vertex operators with the prescribed eigenvalues,

$$\tilde{\eta}_k |\,\psi_{\rm prod}\,\rangle = z_k\,, \quad \text{for all } k \in V\,. \tag{31}$$

We then measure the values of the stabilizer operators $S_\ell$ and record the measurement results as $s_\ell$. Since the stabilizers commute with the operators $\tilde{\eta}_j$, the post-measurement state satisfies

$$S_\ell |\,\psi_{\rm post}\,\rangle = s_\ell |\,\psi_{\rm post}\,\rangle\,, \quad \text{for all } \ell \in L\,, \tag{32}$$

$$\tilde{\eta}_j |\,\psi_{\rm post}\,\rangle = z_j |\,\psi_{\rm post}\,\rangle\,, \quad \text{for all } j \in V\,. \tag{33}$$

Generally, some of the measurement results of the stabilizers will be $-1$. One simple way to resolve this problem is to reassign the values $\epsilon_{jk}$ in Eq. (12), and the signs of the stabilizers change accordingly. This procedure corresponds to encoding the fermionic state in a different subspace of the stabilizers; the signs of logical operators change accordingly. The reassignment is straightforward when there is a known Hamiltonian path in $G$. We keep $\epsilon_{jk}$ unchanged for all the edges $(j,k)$ in the Hamiltonian path. Any remaining edge $\epsilon_{lm}$ along with a section in the Hamiltonian path form an independent closed path. We flip the value of $\epsilon_{lm}$ if the measurement outcome of the corresponding stabilizer is $-1$. More generally, the reassignment of $\epsilon_{jk}$ can be achieved with three steps: 1. find a spanning tree $T$ of the graph $G$, 2. assign the original value $\epsilon_{jk}$ to all of the edges $(j,k)$ in the spanning tree $T$, 3. flip the sign of any remaining term if the measurement outcome of the stabilizer is $-1$.

## V. CONCLUSION

While hardware that supports full fledged fault-tolerant schemes is still unavailable, error mitigation schemes are likely to be an essential part of quantum algorithms in the NISQ era [47]. A merit of these near-term quantum error-mitigating schemes is to be flexible to the

hardware and the problem being implemented. We consider error correction/detection in quantum simulation of many-body fermonic systems with local interactions, e.g., the Hubbard model. The locality of the fermion Hamiltonian is often lost when they are mapped to qubits using conventional methods, such as the Jordan-Wigner transformation. Locality-preserving fermion-to-qubit maps prevent this by encoding the original fermions into a subspace of the qubit operators corresponding to the common +1 eigenstates of a set of stabilizer operators. This makes sure that parity is conserved when fermions transit around closed paths.

Locality-preserving mappings also allow for quantum error correction/detection, and one of the leading candidates is the superfast encoding presented by Bravyi and Kitaev [25]. Indeed, we show that the Bravyi and Kitaev encoding can detect all single-qubit errors on a 2D square lattice when the orders of the incident edges are chosen properly.

Very recently, it was proved that the Bravyi-Kitaev superfast encoding cannot correct all single-qubit errors on graphs with vertex degree less or equal to six [31]. We get around this no-go theorem by introducing a mapping that is more general than the original Bravyi-Kitaev superfast encoding. We call it a Majorana loop stabilizer code, meaning that the stabilizers are constructed from the qubit correspondences of products of Majorana fermion operators on closed paths. We show that these codes can correct all single-qubit errors on 2D square lattices with vertex degree four. The logical operators in our code can also be implemented with Pauli-operators with lower weights compared to the Bravyi-Kitaev superfast encoding. We also discuss how to encode a single Slater determinant with the Majorana loop stabilizer code in circuit depth $O(1)$. Future work will include generalizing the code to other lattices and higher dimensions, designing fast gate sequences to implement the logical operators, simplifying syndrome measurements, and making the code more robust to errors.

## ACKNOWLEDGMENTS

The authors would like to acknowledge the enlightening and helpful discussions with Dave Bacon, Eleanor Rieffel, Sergio Boixo, Vadim Smelyanskiy, and Kevin Sung.

## Appendix A: A brief review of quantum error-correcting codes

Quantum devices are vulnerable to decoherence due to interactions with environments. It was once believed that decoherence sets an upper limit on time and size for quantum computation [48]. Such a limitation was later overcome, at least in theory, with the invention of quantum error-correcting codes [49–51]. It was shown that quantum computation can be made robust against errors when the error rate is smaller than a constant threshold [52, 53]. The surface code approach is a natural choice for fault-tolerant quantum computation, which requires only nearest-neighbor couplings and modest gate fidelity [9, 54].

The error threshold of fault-tolerant protocols are derived by bounding an operator norm such as the diamond norm [55], which translates to a pretty stringent requirement when the amplitudes of coherent errors add together. For example, if one gate has over-rotation $\theta$, then a sequence of $N$ gates could produce an over-rotation of $N\theta$ and an error probability proportional to $N^2$. In order to get this worst case behavior, however, the errors have to add together in a coherent way. This is unlikely to happen in practice, therefore, coherent errors may be no worse for fault tolerance than incoherent ones. Indeed, a Pauli- or Clifford-twirling may be used to convert any noise channel into a simple mixture of Pauli errors or depolarizing noise [56, 57].

A quantum $[[n, k, d]]$ code specifies a $2^k$-dimensional subspace of $n$ physical qubits, with which $k$ logical qubits can be encoded. The distance $d$ of a quantum error-correcting code is the minimum weight of a Pauli error by which an element of the code space can be transformed into an orthogonal element of the code space, e.g., a logical error happens. The weight of a Pauli error is the number of qubits on which a non-identity Pauli transformation acts on. An $[[n, k, d]]$-quantum code can detect up to weight $d-1$ errors, and can correct errors of weight up to $t$ satisfying $2t < d$.

A large class of quantum error-correcting codes can be described by the stabilizer formalism. Consider the Pauli group $\mathcal{P}_n$ of an $n$ qubit system, and let $\mathcal{S}$ be its subgroup generated by $r$ commuting, independent generators (the stabilizers). The common +1 eigensubspace of all the elements in $\mathcal{S}$ defines the code subspace of dimension $2^k$ where $k = n - r$. Let $\mathcal{C}(\mathcal{S})$ be the centralizer of $\mathcal{S}$, the set of elements in $\mathcal{P}_n$ that commute with all elements of $\mathcal{S}$. The logical operators can be specified by using the elements in $\mathcal{C}(\mathcal{S})$ that satisfy the commutations of Pauli operators.

## Appendix B: The auxiliary fermion approach

In this appendix, we take a different approach to construct locality-preserving fermion-to-qubit mappings that can also detect/correct single-qubit errors. This approach is based on the auxiliary fermion method introduced by Ball [34], and by Verstraete and Cirac [35]. We show that the map itself is not particularly good for detecting/correcting errors. In the next appendix, we improve this situation by combining two unit cells as a block.

Consider fermions hopping between neighboring sites on a 2D lattice; one auxiliary fermionic mode is introduced for each data fermionic mode. Each vertical hopping term is accompanied with a quadratic Majorana

operator on the two corresponding auxiliary modes. We then map the fermions to qubits using the Jordan-Wigner transformation with row-major order, snake shape, and data modes interleaved with the corresponding auxiliary modes. The parity terms from the quadratic Majorana operator cancel the ones from the vertical hopping terms, and this makes the resulting Hamiltonian local. To reproduce the original physics, the state of the auxiliary subsystem is restricted to the common +1 eigenspace of the quadratic Majorana operators.

We use $f_{2k}$ and $f_{2k+1}$ ($\gamma_{2k}$ and $\gamma_{2k+1}$) to denote the two Majorana operators associated to the $k$th data (auxiliary) mode. Using the JW transformation, the Pauli operators of the data qubits can be expressed as

$$X_k = P_0^{k-1} f_{2k}\,,\quad Y_k = P_0^{k-1} f_{2k+1}\,,\quad Z_k = i f_{2k} f_{2k+1}\,, \tag{B1}$$

where the parity operator is the product of all the Majorana operators in a particular interval,

$$P_j^k = \prod_{m=j}^{k} \big(i f_{2m} f_{2m+1}\big)\big(i\gamma_{2m}\gamma_{2m+1}\big)\,, \tag{B2}$$

The parity operator anticommutes with any single-mode Majorana operator in its support. Similarly, the Pauli operators of the auxiliary qubits take the form

$$\widetilde{X}_k = P_0^{k-1} Z_k \gamma_{2k}\,,\quad \widetilde{Y}_k = P_0^{k-1} Z_k \gamma_{2k+1}\,,\quad \widetilde{Z}_k = i\gamma_{2k}\gamma_{2k+1}\,. \tag{B3}$$

The horizontal hopping term can be implemented with local qubit operators,

$$i f_{2j+1} f_{2j+2} = i P_0^{j-1} Y_j P_0^j X_{j+1} = -X_j \widetilde{Z}_j X_{j+1}\,. \tag{B4}$$

The vertical fermionic hopping term takes the form

$$i f_{2j+1} f_{2k} = i P_0^{j-1} Y_j P_0^{k-1} X_k = i Y_j P_j^{k-1} X_k\,, \tag{B5}$$

where $j$ and $k$ are neighboring qubits in the same column. To get rid of the non-local parity operator in Eq. (B5), we introduce the gauge operator of the auxiliary modes,

$$\begin{aligned} i\gamma_{2j+1}\gamma_{2k} &= i\big(P_0^{j-1} Z_j \widetilde{Y}_j\big)\big(P_0^{k-1} Z_k \widetilde{X}_k\big) \\ &= i Z_j \widetilde{Y}_j P_j^{k-1} Z_k \widetilde{X}_k\,, \end{aligned} \tag{B6}$$

where $j$ and $k$ are neighboring sites in the same column. The hopping term can be implemented locally by attaching the gauge operator to it,

$$(i f_{2j+1} f_{2k})(i\gamma_{2j+1}\gamma_{2k}) = X_j \widetilde{Y}_j Y_k \widetilde{X}_k\,. \tag{B7}$$

To retain the same physics, we restrict the state to be a common +1 eigenstate of the gauge operators,

$$\langle\,\psi\,|i\gamma_{2j+1}\gamma_{2k}|\,\psi\,\rangle = 1\,. \tag{B8}$$

This constrain can be made local by combining two neighboring gauge operators together,

$$\begin{aligned} &(i\gamma_{2j+1}\gamma_{2k})(i\gamma_{2j+3}\gamma_{2k-2}) \\ &\quad = (i Z_j \widetilde{Y}_j P_j^{k-1} Z_k \widetilde{X}_k)(i Z_{j+1} \widetilde{Y}_{j+1} P_{j+1}^{k-2} Z_{k-1} \widetilde{X}_{k-1}) \\ &\quad = -\widetilde{X}_j Z_{j+1} \widetilde{Y}_{j+1} Z_k \widetilde{X}_k \widetilde{Y}_{k-1}\,. \end{aligned} \tag{B9}$$

The introduction of the stabilizer operators allows for correcting/detecting certain kinds of errors. The syndrome of a Pauli error can be described by the set of gauge operators that anticommute with it. Any Majorana operator anticommutes with a gauge operator if and only if it contains exactly one of the two single-mode Majorana operators in the gauge operator. The code cannot detect Pauli-$Z$ errors on the data qubits as they are logical operations and commute with all gauge operators. It can detect any single-qubit Pauli-$X$ or -$Y$ error on the data qubits and to correct any single-qubit error on the auxiliary qubits. The original auxiliary fermion approach is not especially efficient in correcting errors occurred on the data qubits.

## Appendix C: Combining two unit cells in BVC

In the last example, the Pauli operators of the physical qubits are expressed in terms of the Majorana fermion operators of the data modes and the auxiliary modes. Here, we use the same gauge operators defined in Eq. (B6), but we encode four data Majorana modes and four auxiliary Majorana modes into a block of 4 physical qubits. This allows for correcting 10 out of 12 single-qubit errors in the block and detecting the rest two.

For simplicity of notation, we omit the block indices in the operators, e.g., $f_{2k+2}$ is denoted as $f_2$. The Pauli operators on the $k$th block are constructed from the Majorana operators,

$$X_0 = iP f_0 f_1 \gamma_1\,,\quad Y_0 = iP f_2 f_3 \gamma_0\,, \tag{C1}$$

$$X_1 = iP f_1 f_2 \gamma_3\,,\quad Y_1 = iP f_0 f_3 \gamma_2\,, \tag{C2}$$

$$X_2 = iP f_2 \gamma_1 \gamma_2\,,\quad Y_2 = iP f_0 \gamma_0 \gamma_3\,, \tag{C3}$$

$$X_3 = iP f_3 \gamma_1 \gamma_3\,,\quad Y_3 = iP f_1 \gamma_0 \gamma_2\,, \tag{C4}$$

where $P$ is a shorthand of the parity operator $P_0^{k-1}$, i.e., the product of all the preceding Majorana operators. These operators commute with any single-mode Majorana operator on the first $k-1$ blocks and the corresponding Pauli operators. These operators satisfy the commutation relations of the Pauli matrices. Inversely, the Majorana operators take the form

$$f_0 = P X_0 Y_1 Y_2 Z_3\,,\qquad \gamma_0 = P Y_0 Z_1 Y_2 Y_3\,, \tag{C5}$$

$$f_1 = -P X_0 X_1 Z_2 Y_3\,,\qquad \gamma_1 = -P X_0 Z_1 X_2 X_3\,, \tag{C6}$$

$$f_2 = P Y_0 X_1 X_2 Z_3\,,\qquad \gamma_2 = -P Z_0 Y_1 X_2 Y_3\,, \tag{C7}$$

$$f_3 = -P Y_0 Y_1 Z_2 X_3\,,\qquad \gamma_3 = -P Z_0 X_1 Y_2 X_3\,. \tag{C8}$$

The logical fermionic operators within the cell are

$$if_0f_3 = -Z_0X_2Y_3\,, \qquad if_1f_2 = -Z_0Y_2X_3\,, \tag{C9}$$
$$if_0f_1 = -Z_1X_2X_3\,, \qquad if_2f_3 = Z_1Y_2Y_3 \tag{C10}$$

which correspond to the horizontal hopping terms and the occupation terms. The inter-block horizontal hopping terms can be constructed by using the inverse relations (C5-C8); the weight of the corresponding logical operators is seven. All the terms containing one data Majorana operator and one auxiliary Majorana operator from the same block are three local, e.g.,

$$if_0\gamma_1 = X_1Z_2Y_3\,, \quad if_1\gamma_1 = Y_1Y_2Z_3\,. \tag{C11}$$

Vertical hopping terms can be implemented by a 6-local Pauli operator corresponding to the product of two pairs of such operators. The stabilizer can be either 6 local or 14 local depending whether they are intra-block or inter-blocks.

There are 4 auxiliary Majorana operators $\gamma_{0,1,2,3}$ associated to a block, each of which form a gauge operator with another auxiliary Majorana operator from the block above or below. To analyze the syndromes of the single-qubit errors, we first look at the parity term $P_0^{k-1}$. The parity term anticommutes with gauge operators that share one common single-mode Majorana operator with it. This set includes two kinds of gauge operators: 1. those consist of one auxiliary Majorana operator to the left (right) of the $k$th block and an auxiliary Majorana operator below it, 2. those consist of one auxiliary Majorana operator to the right (left) of the $k$th block, including the $k$th block, and an auxiliary Majorana operator above it, One can determine the block $k$, based on the syndromes of $P$, if a Pauli-$X$ or $Y$ error has occurred. Any single-qubit Pauli-$X$ or $Y$ error contains a unique combination of the auxiliary Majorana operators in the block $k$, see Eqs. (C1)-(C4). Therefore, their syndromes are different.

The Pauli-$Z$ errors do no involve the parity term. For example, the Pauli operator $Z_0$ and $Z_1$ contain the auxiliary Majorana operators $\gamma_0\gamma_1$ and $\gamma_2\gamma_3$ in the block, these syndromes allows one to identify the block where the error has happened. Both of the Pauli errors $Z_2$ and $Z_3$ contain the same product $\gamma_0\gamma_1\gamma_2\gamma_3$ and have the same syndrome. Therefore, the code can correct 10 out of the 12 single-qubit errors in the block and detect the remaining two single-qubit errors $Z_2$ and $Z_3$.

## Appendix D: Algorithms to set initial qubit values

In this appendix, we give an algorithm to assign initial values to the qubits for preparation of a Slater determinant. This algorithm requires neither a Hamiltonian path nor a spanning tree.

**Algorithm 1** Assign values to $z_{jk}$

**Input:**
$G$ ▷ a connected undirected graph
$V$ ▷ the set of vertices of $G$
$E$ ▷ the set of edges of $G$
$z_k \in \{\pm 1\}$ ▷ for each vertex $k \in V$

**procedure** RmLeg($k$) ▷ remove the leg with the degree-one vertex $k \in V$, i.e., Deg($k$) = 1
  find $j \in V$ such that $(j,k) \in E$
  $z_{jk} \leftarrow z_k$
  $V \leftarrow V\backslash\{k\}$
  $E \leftarrow E\backslash\{(j,k)\}$
  **if** Deg($j$) = 0 **then** $V \leftarrow V\backslash\{j\}$
  **else if** Deg($j$) = 1 **then** RmLeg($j$)
  **end if**
**end procedure**
**while** there exist degree-one vertex $k \in V$ **do**
  RmLeg($k$)
**end while**
**while** $E \neq \emptyset$ **do**
  randomly pick an edge $(j,k) \in E$
  $z_{jk} \leftarrow +1$
  $E \leftarrow E\backslash\{(j,k)\}$
  **if** Deg($k$) = 1 **then** RmLeg ($k$)
  **end if**
  **if** Deg($j$) = 1 **then** RmLeg($j$)
  **end if**
**end while**

## Appendix E: Code on single open plaquette

### 1. BKSF

Here we work out the simple example of the BKSF stabilizer in a $2\times2$ graph of vertices, labeled $i$, $j$, $k$, $l$. In this case, there are 4 vertices, and 4 edges representing the physical qubits. There is one stabilizer following the general relation $N_E - N_V + 1$ for independent closed paths on the graph. That reduces the total logical space to 3 qubits, representing a parity conserving subspace. A number of different stabilizers can result from different edge orderings, however the most convenient for our purposes is one that follows a cycle such that the stabilizer is

$$\begin{aligned} S_{ijkl} &= \tilde{\xi}_{ij}\tilde{\xi}_{jk}\tilde{\xi}_{kl}\tilde{\xi}_{li} \\ &= (X_{ij}Z_{jk})(X_{jk}Z_{kl})(X_{kl}Z_{li})(X_{li}Z_{ij}) \\ &= (X_{ij}Z_{ij})(Z_{jk}X_{jk})(Z_{kl}X_{kl})(Z_{li}X_{li}) \\ &= -Y_{ij}Y_{jk}Y_{kl}Y_{li} \end{aligned} \tag{E1}$$

From this, we see that any individual $X$ or $Z$ error on a physical qubit of this code anti-commutes with this stabilizer. As a result, it's possible to detect and post-select on any single qubit $X$ or $Z$ error, and more generally any odd number of $X$ or $Z$ errors. However it is easy to see

that the code fails to detect a $Y$ error on any single qubit and by extension we cannot correct arbitrary single qubit errors with this encoding.

### Appendix F: Code on BKSF infinite lattice

Here we consider the properties of a BKSF encoding of a square regular lattice of fermions that extends in each direction. For this code, the physical qubits on an edge always participate in two generating plaquettes, and each edge is treated identically to all others from symmetry.

The following 4 stabilizers involve vertex $m$, and can be read directly from Fig 3 as

$$\begin{aligned} S_{ijml} &= -X_{ij}Y_{jm}Y_{ml}X_{li}Z_{ii'}Z_{ii''} \\ S_{jknm} &= -X_{jk}Y_{kn}Y_{nm}X_{jm}Z_{ij}Z_{jj''} \\ S_{lmpo} &= -X_{lm}Y_{mp}Y_{op}X_{ol}Z_{ll'}Z_{li} \\ S_{mnqp} &= -X_{mn}Y_{nq}Y_{qp}X_{pm}Z_{mj}Z_{ml} \end{aligned} \tag{F1}$$

where we have labeled the off figure indices by $i'$ for to the left of $i$ and $i''$ for above $i$. Taking these 4 plaquettes constitute the complete set of stabilizers that act on the physical edge qubits $E_{mj}$ and $E_{ml}$. We characterize the syndromes of each of these errors in table 11.

To determine the possibility of error correction, one wants to see that for any $\mathcal{E}_a^\dagger\mathcal{E}_b$ where $\mathcal{E}_p \in \{X_{mj}, Y_{mj}, Z_{mj}, X_{ml}, Y_{ml}, Z_{ml}\}$ there is a stabilizer operator that anti-commmutes with it. This is sufficient for a recovery operation to exist for that set of errors. From table 11, we see this is not the case for all errors. Hence the BKSF transformation on an infinite square lattice with the given ordering is capable of detecting single qubit errors, but not correcting them in general.

### Appendix G: Code on BKSF torus

Here we look at a patch of the BKSF code under periodic boundary conditions so that it forms a torus. For a size lattice that is $3 \times 3$ or larger, by symmetry it is again acceptable to look at the representative stabilizers that involve the vertex $m$ in the figure pictured in the main text. From the infinite lattice, we can get the corresponding stabilizers from appropriate assignment of the indices with periodic boundary conditions corresponding to a 2D-torus. This yields the stabilizers involving vertex $m$ as

$$\begin{aligned} S_{ijml} &= -X_{ij}Y_{jm}Y_{ml}X_{li}Z_{ik}Z_{io} \\ S_{jknm} &= -X_{jk}Y_{kn}Y_{nm}X_{jm}Z_{ij}Z_{jp} \\ S_{lmpo} &= -X_{lm}Y_{mp}Y_{op}X_{ol}Z_{ln}Z_{li} \\ S_{mnqp} &= -X_{mn}Y_{nq}Y_{qp}X_{pm}Z_{mj}Z_{ml} \end{aligned} \tag{G1}$$

which from inspection we see are qualitatively the same as the case for the infinite lattice. That is, we expect the same commutation relations for the stabilizers and the representative edges $E_{ml}$ and $E_{mj}$.

| | $S_{ijml}$ | $S_{jknm}$ | $S_{lmpo}$ | $S_{mnqp}$ |
|---|---|---|---|---|
| $X_{mj}$ | $-1$ | $1$ | $1$ | $-1$ |
| $Y_{mj}$ | $1$ | $-1$ | $1$ | $-1$ |
| $Z_{mj}$ | $-1$ | $-1$ | $1$ | $1$ |
| $X_{ml}$ | $-1$ | $1$ | $1$ | $-1$ |
| $Y_{ml}$ | $1$ | $1$ | $-1$ | $-1$ |
| $Z_{ml}$ | $-1$ | $1$ | $-1$ | $1$ |
| $X_{mj}Y_{mj}$ | $-1$ | $-1$ | $1$ | $1$ |
| $X_{mj}Z_{mj}$ | $1$ | $-1$ | $1$ | $-1$ |
| $X_{ml}Y_{ml}$ | $-1$ | $1$ | $-1$ | $1$ |
| $X_{ml}Z_{ml}$ | $1$ | $1$ | $-1$ | $-1$ |
| $X_{mj}X_{ml}$ | $1$ | $1$ | $1$ | $1$ |
| $X_{mj}Y_{ml}$ | $-1$ | $1$ | $-1$ | $1$ |
| $X_{mj}Z_{ml}$ | $1$ | $1$ | $-1$ | $-1$ |
| $X_{ml}Y_{mj}$ | $-1$ | $-1$ | $1$ | $1$ |
| $Z_{ml}Y_{mj}$ | $-1$ | $-1$ | $-1$ | $-1$ |
| $X_{ml}Z_{mj}$ | $1$ | $-1$ | $1$ | $-1$ |
| $Z_{mj}Y_{ml}$ | $-1$ | $-1$ | $-1$ | $-1$ |
| $Z_{mj}Z_{ml}$ | $1$ | $-1$ | $-1$ | $1$ |

FIG. 11. Syndromes of different errors on the infinite square BKSF lattice. The convention here currently uses $-1$ to imply anti-commutation with the error and 1 to imply commutation. The first block looks at syndromes of single errors on the edges for the purpose of detection and the second block examines the related quantity $\mathcal{E}_a^\dagger\mathcal{E}_b$ related to the possibility of correcting those errors.

### Appendix H: Code on BKSF open section

Considering now a finite square lattice with open boundary conditions with the standard ordering in this work, we may again work from Fig 3. In this case, all the plaquettes except those on the left or upper boundaries of the graph are the same, where the dangling $Z$ operators are truncated.

For interior edges more than 1 plaquette from a boundary, the same detection and correction results from the infinite case apply. However, at the boundary, there are now generally edges that are covered by only 1 or 2 plaquettes instead of 3, as in the interior cases. As a result, even error detection at the edges can become problematic. For example, in the bottom left plaquette of a square lattice with open boundary conditions, one is incapable of detecting a single $X$ error on that edge. This means that we expect there are a number of undetectable errors in this case as well, especially at boundaries. This may still allow the mitigation of some errors, but is less powerful than the periodic case for this choice of ordering. For a finite graph, it may be possible to devise a

custom or non-uniform ordering that restores the power to detect errors, however we do not investigate that here.


[1] R. P. Feynman, "Simulating physics with computers," International Journal of Theoretical Physics **21**, 467 (1982).

[2] S. Lloyd, "Universal Quantum Simulators," Science **273**, 1073 (1996).

[3] I. Georgescu, S. Ashhab, and F. Nori, "Quantum simulation," Reviews of Modern Physics **86**, 153 (2014).

[4] J. I. Cirac and P. Zoller, "Quantum Computations with Cold Trapped Ions," Physical Review Letters **74**, 4091 (1995).

[5] D. Kielpinski, C. Monroe, and D. J. Wineland, "Architecture for a large-scale ion-trap quantum computer," Nature **417**, 709 (2002).

[6] H. Haeffner, C. F. Roos, and R. Blatt, "Quantum computing with trapped ions," Physics Reports **469**, 155 (2008).

[7] M. H. Devoret, A. Wallraff, and J. M. Martinis, "Superconducting Qubits: A Short Review," arXiv:0411174 (2004).

[8] G. Wendin, "Quantum information processing with superconducting circuits: a review," Reports on Progress in Physics. Physical Society (Great Britain) **80**, 106001 (2017).

[9] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," Physical Review A **86**, 032324 (2012).

[10] S. J. Devitt, A. M. Stephens, W. J. Munro, and K. Nemoto, "Requirements for fault-tolerant factoring on an atom-optics quantum computer," Nature Communications **4**, 2524 (2013).

[11] J. O' Gorman and E. T. Campbell, "Quantum computation with realistic magic-state factories," Physical Review A **95**, 032338 (2017).

[12] D. Gottesman, "Quantum fault tolerance in small experiments," arXiv:1610.03507 (2016).

[13] J. Roffe, D. Headley, N. Chancellor, D. Horsman, and V. Kendon, "Protecting quantum memories using coherent parity check codes," Quantum Science and Technology **3**, 035010 (2018).

[14] J. R. McClean, M. E. Kimchi-Schwartz, J. Carter, and W. A. de Jong, "Hybrid quantum-classical hierarchy for mitigation of decoherence and determination of excited states," Phys. Rev. A **95**, 042308 (2017).

[15] J. I. Colless, V. V. Ramasesh, D. Dahlen, M. S. Blok, M. E. Kimchi-Schwartz, J. R. McClean, J. Carter, W. A. de Jong, and I. Siddiqi, "Computation of molecular spectra on a quantum processor with an error-resilient algorithm," Phys. Rev. X **8**, 011021 (2018).

[16] K. Temme, S. Bravyi, and J. M. Gambetta, "Error Mitigation for Short-Depth Quantum Circuits," Physical Review Letters **119**, 180509 (2017).

[17] R. Harper and S. Flammia, "Fault tolerance in the IBM Q Experience," arXiv:1806.02359 (2018).

[18] S. McArdle, X. Yuan, and S. Benjamin, "Error mitigated digital quantum simulation," arXiv:1807.02467 (2018).

[19] X. Bonet-Monroig, R. Sagastizabal, M. Singh, and T. E. O'Brien, "Low-cost error mitigation by symmetry verification," arXiv:1807.10050 (2018).

[20] G. Ortiz, J. E. Gubernatis, E. Knill, and R. Laflamme, "Quantum algorithms for fermionic simulations," Physical Review A **64**, 022319 (2001).

[21] M. B. Hastings, D. Wecker, B. Bauer, and M. Troyer, "Improving Quantum Algorithms for Quantum Chemistry," Quantum Info. Comput. **15**, 1 (2015).

[22] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, "Solving strongly correlated electron models on a quantum computer," Physical Review A **92**, 062318 (2015).

[23] A. A. Houck, H. E. Türeci, and J. Koch, "On-chip quantum simulation with superconducting circuits," Nature Physics **8**, 292 (2012).

[24] R. Barends, A. Shabani, L. Lamata, J. Kelly, A. Mezzacapo, U. L. Heras, R. Babbush, A. G. Fowler, B. Campbell, Y. Chen, Z. Chen, B. Chiaro, A. Dunsworth, E. Jeffrey, E. Lucero, A. Megrant, J. Y. Mutus, M. Neeley, C. Neill, P. J. J. O'Malley, C. Quintana, P. Roushan, D. Sank, A. Vainsencher, J. Wenner, T. C. White, E. Solano, H. Neven, and J. M. Martinis, "Digitized adiabatic quantum computing with a superconducting circuit," Nature **534**, 222 (2016).

[25] S. B. Bravyi and A. Y. Kitaev, "Fermionic Quantum Computation," Annals of Physics **298**, 210 (2002).

[26] R. Babbush, N. Wiebe, J. McClean, J. McClain, H. Neven, and G. K.-L. Chan, "Low-Depth Quantum Simulation of Materials," Physical Review X **8**, 011044 (2018).

[27] I. D. Kivlichan, J. McClean, N. Wiebe, C. Gidney, A. Aspuru-Guzik, G. K.-L. Chan, and R. Babbush, "Quantum Simulation of Electronic Structure with Linear Depth and Connectivity," Physical Review Letters **120**, 110501 (2018).

[28] M. Motta, E. Ye, J. R. McClean, Z. Li, A. J. Minnich, R. Babbush, and G. K.-L. Chan, "Low Rank Representations for Quantum Simulation of Electronic Structure," arXiv:1808.02625 (2018).

[29] Z. Jiang, K. J. Sung, K. Kechedzhi, V. N. Smelyanskiy, and S. Boixo, "Quantum Algorithms to Simulate Many-Body Physics of Correlated Fermions," Physical Review Applied **9**, 044036 (2018).

[30] J. D. Whitfield, V. Havlíček, and M. Troyer, "Local spin operators for fermion simulations," Phys. Rev. A **94**, 030301 (2016).

[31] K. Setia, S. Bravyi, A. Mezzacapo, and J. D. Whitfield, "Superfast encodings for fermionic quantum simulation," arXiv:1810.05274 (2018).

[32] M. Levin and X.-G. Wen, "Fermions, strings, and gauge fields in lattice spin models," Physical Review B **67**, 245316 (2003).

[33] X.-G. Wen, "Quantum order from string-net condensations and the origin of light and massless fermions," Physical Review D **68**, 065003 (2003).

[34] R. C. Ball, "Fermions without fermion fields," Phys. Rev. Lett. **95**, 176407 (2005).

[35] F. Verstraete and J. I. Cirac, "Mapping local Hamiltonians of fermions to local Hamiltonians of spins," Journal

of Statistical Mechanics: Theory and Experiment **2005**, P09012 (2005).
[36] Y.-A. Chen, A. Kapustin, and D. Radičević, "Exact bosonization in two spatial dimensions and a new class of lattice gauge theories," Annals of Physics **393**, 234?253 (2018).
[37] M. Steudtner and S. Wehner, "Quantum codes for quantum simulation of Fermions on a square lattice of qubits," arXiv:1810.02681 (2018).
[38] K. Setia and J. D. Whitfield, "Bravyi-Kitaev Superfast simulation of electronic structure on a quantum computer," The Journal of Chemical Physics **148**, 164104 (2018).
[39] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," New Journal of Physics **18**, 023023 (2016).
[40] T. Helgaker, P. Jorgensen, and J. Olsen, *Molecular Electronic Structure Theory* (Wiley, 2002).
[41] B. L. Hammond, W. A. Lester, and P. J. Reynolds, *Monte Carlo methods in ab initio quantum chemistry*, Vol. 1 (World Scientific, 1994).
[42] K. C. Young, M. Sarovar, and R. Blume-Kohout, "Error Suppression and Error Correction in Adiabatic Quantum Computation: Techniques and Challenges," Physical Review X **3**, 041013 (2013).
[43] A. D. Bookatz, E. Farhi, and L. Zhou, "Error suppression in Hamiltonian-based quantum computation using energy penalties," Physical Review A **92**, 022317 (2015).
[44] Z. Jiang and E. G. Rieffel, "Non-commuting two-local Hamiltonians for quantum error suppression," Quantum Information Processing **16**, 89 (2017).
[45] V. Havlíček, M. Troyer, and J. D. Whitfield, "Operator locality in the quantum simulation of fermionic models," Physical Review A **95**, 032332 (2017).
[46] D. Bacon, "Operator quantum error-correcting subsystems for self-correcting quantum memories," Physical Review A **73**, 012340 (2006).
[47] J. Preskill, "Quantum computing in the NISQ era and beyond," arXiv:1801.00862 (2018).
[48] W. G. Unruh, "Maintaining coherence in quantum computers," Physical Review A **51**, 992 (1995).
[49] P. W. Shor, "Scheme for reducing decoherence in quantum computer memory," Physical Review A **52**, R2493 (1995).
[50] A. M. Steane, "Error Correcting Codes in Quantum Theory," Physical Review Letters **77**, 793 (1996).
[51] A. R. Calderbank and P. W. Shor, "Good quantum error-correcting codes exist," Physical Review A **54**, 1098 (1996).
[52] A. Y. Kitaev, "Quantum computations: algorithms and error correction," Russian Mathematical Surveys **52**, 1191 (1997).
[53] D. Aharonov and M. Ben-Or, "Fault tolerant quantum computation with constant error," in *STOC '97 Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997) pp. 176–188.
[54] 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, A. N. Cleland, and J. M. Martinis, "Superconducting quantum circuits at the surface code threshold for fault tolerance," Nature **508**, 500 (2014).
[55] P. Aliferis, D. Gottesman, and J. Preskill, "Quantum Accuracy Threshold for Concatenated Distance-3 Codes," Quantum Info. Comput. **6**, 97 (2006).
[56] J. J. Wallman and J. Emerson, "Noise tailoring for scalable quantum computation via randomized compiling," Physical Review A **94**, 052325 (2016).
[57] Y. Li and S. C. Benjamin, "Efficient Variational Quantum Simulator Incorporating Active Error Minimization," Physical Review X **7**, 021050 (2017).