

Fault-tolerant structures for measurement-based quantum computation on a network

Yves van Montfort,¹ Sébastien de Bone,^{1,2} and David Elkouss^{1,3,*}

¹*QuTech, Delft University of Technology, Lorentzweg 1, 2628 CJ Delft, The Netherlands*

²*QuSoft, CWI, Science Park 123, 1098 XG Amsterdam, The Netherlands*

³*Networked Quantum Devices Unit, Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan*

(Dated: April 18, 2025)

In this work, we introduce a method to construct fault-tolerant *measurement-based quantum computation* (MBQC) architectures and numerically estimate their performance over various types of networks. A possible application of such a paradigm is distributed quantum computation, where separate computing nodes work together on a fault-tolerant computation through entanglement. We gauge error thresholds of the architectures with an efficient stabilizer simulator to investigate the resilience against both circuit-level and network noise. We show that, for both monolithic (*i.e.*, non-distributed) and distributed implementations, an architecture based on the diamond lattice may outperform the conventional cubic lattice. Moreover, the high *erasure* thresholds of non-cubic lattices may be exploited further in a distributed context, as their performance may be boosted through *entanglement distillation* by trading in entanglement success rates against erasure errors during the error-decoding process. These results highlight the significance of lattice geometry in the design of fault-tolerant measurement-based quantum computing on a network, emphasizing the potential for constructing robust and scalable distributed quantum computers.

I. INTRODUCTION

Large-scale quantum computation with low error rates requires handling noise in a correct and efficient manner—one would like to *fault-tolerantly* transmit, store, and process quantum information with quantum computing hardware. Quantum error-correction encompasses methods to achieve fault tolerance from faulty hardware. Topological error-correction codes, including surface codes, are a promising avenue to achieve this goal, as these codes have high error thresholds against local errors, and only require nearest-neighbor interactions between qubits in a two-dimensional layout.

Surface codes achieve fault-tolerance by repeatedly combining measurement outcomes of consecutive rounds of stabilizer measurements [1]. Geometrically, the addition of a time dimension to a two-dimensional decoding (*i.e.*, *syndrome*) graph creates a three-dimensional structure, that may be interpreted as a noisy quantum channel where logical information is propagated in the direction of time. For example, the conventional planar or toric surface code may be constructed from qubits on a square lattice, but its corresponding decoding graphs are three-dimensional cubic lattices.

Foliation [2] is a method to transform any surface code to a three-dimensional *cluster state* that forms the resource for fault-tolerant *measurement-based quantum computation* (MBQC), with the same geometry as the corresponding logical quantum channel of the surface code [3, 4]. Although foliated codes can be interpreted as having replaced time with another spatial dimension, the resulting cluster states can be initialized and consumed in arbitrary directions over time, lifting the rigid

duality of time and space of surface codes. A concrete example is so-called *interleaving* in fusion-based quantum computation [5], which bears many resemblances to the measurement-based architectures considered here.

However, foliation does not exhaust all possible fault-tolerant cluster states—*i.e.*, there are *non-foliated* three-dimensional cluster states that cannot be constructed from a two-dimensional surface code. We study these types of lattices because they may produce higher fault-tolerant error thresholds than conventional surface codes, at least when assuming simple combinations of independent and identically distributed (i.i.d.) single-qubit and measurement errors [6]. In practice though, non-foliated lattices will not necessarily produce higher error thresholds, because faulty operations during cluster state preparation and measurement typically introduce highly coupled and non-identically distributed errors. Both the complexity of the error decoder (through the syndrome graph) and the complexity of the error model (through the quantum circuit) eventually determine the error threshold.

Because state-of-the-art hardware is currently not capable of realizing large cluster states that can achieve sufficiently low logical error rates, some research has focused its attention to modular implementations of fault-tolerant MBQC states [7–9]. The entire cluster state is prepared from resource states that are generated by small, separate devices, and entangling operations between devices create the entanglement to encode the entire cluster state. Several physical systems are suitable for modular MBQC architectures via optical interfaces [10].

In this paper, we explore fault-tolerant cluster states under more realistic noise models compared to previous investigations [6]. This brings their realization in MBQC closer to reality. We numerically estimate fault-tolerant thresholds of previously proposed cluster states [6] for

* david.elkouss@oist.jp

both *monolithic* (*i.e.*, non-distributed) and distributed implementations. We use an efficient stabilizer simulator and the Union-Find decoder [11] to evaluate the performance of our architectures. In our distributed noise models, we consider circuit-level noise during state preparation, measurement noise of single-qubit measurements, and *network noise* between nodes introduced by qubits prepared in the Greenberger-Horne-Zeilinger (GHZ) basis.

We investigate the first step towards including entanglement distillation for one of the distributed cluster states considered and find that including distillation is particularly effective in the context of measurement-based fault tolerance. This is due to the high *erasure*-type error thresholds of non-foliated lattices that may shield against probabilistic entanglement generation resulting from distillation.

II. CLUSTER STATE CONSTRUCTION

In this section, we briefly summarize the mathematical description of cluster states used throughout this paper along the same lines as Fujii [1]. We assume that the reader is already familiar with definitions of the Pauli group, the Clifford group and the stabilizer group, which may be found in detail in Nielsen and Chuang [12].

In Sec. II A, we introduce relevant concepts of a \mathbb{Z}_2 chain complex. In Sec. II B, we define a fault-tolerant three-dimensional cluster state using such a chain complex. In Sec. II C, we describe the error correction process for three-dimensional cluster states.

A. \mathbb{Z}_2 chain complex

A \mathbb{Z}_2 chain complex starts with a definition of *vector spaces* C_i , each constructed over the field \mathbb{Z}_2 and indexed with the *dimension* $i \in \{0, 1, \dots, D\}$. These vector spaces form a sequence $C_D \rightarrow C_{D-1} \rightarrow \dots \rightarrow C_0$, where subsequent pairs are connected by homomorphisms $\partial_i : C_i \rightarrow C_{i-1}$ called *boundary operators*.

In this work, we construct chain complexes over a three-dimensional set $\mathcal{S} = (Q, F, E, V)$ of cells (*i.e.* volumes) $Q = \{\mathbf{q}_k\}$, faces $F = \{\mathbf{f}_k\}$, edges $E = \{\mathbf{e}_k\}$ and vertices $V = \{\mathbf{v}_k\}$. Cells, faces, edges and vertices form the basis elements of the vector spaces C_3 , C_2 , C_1 and C_0 , respectively:

$$\begin{array}{ccccccc} C_3 & \xrightarrow{\partial_3} & C_2 & \xrightarrow{\partial_2} & C_1 & \xrightarrow{\partial_1} & C_0 \\ \{\mathbf{q}_k\} & & \{\mathbf{f}_k\} & & \{\mathbf{e}_k\} & & \{\mathbf{v}_k\} \end{array}$$

An element of C_i is called an *i-chain* and notated as \mathbf{c}_i . Such a chain can be considered a linear combination of basis elements of C_i with \mathbb{Z}_2 coefficients. For example, a

1-chain $\mathbf{c}_1 \in C_1$ is a combination of edges:

$$\mathbf{c}_1 = \sum_k z_k \mathbf{e}_k \equiv [z_0 \ z_1 \ \dots]^T \quad \text{where } z_k \in \mathbb{Z}_2. \quad (1)$$

Here, we take the vector representation with respect to the basis $E = \{\mathbf{e}_k\}$. Analogous definitions hold for the remaining vector spaces and their basis elements.

Boundary operators $\partial_i : C_i \rightarrow C_{i-1}$ are linear operators:

$$\partial_i(\mathbf{c}_i + \mathbf{c}'_i) = \partial_i \mathbf{c}_i + \partial_i \mathbf{c}'_i. \quad (2)$$

For chain complexes defined on \mathcal{S} , boundary operators have an intuitive geometric interpretation. A cell \mathbf{q}_k is mapped to the faces $\{\mathbf{f}_j\}$ that enclose it, a face \mathbf{f}_k is mapped to the edges $\{\mathbf{e}_j\}$ that form its boundary and an edge \mathbf{e}_k is mapped to its endpoints $\{\mathbf{v}_j\}$, which will always be a pair of vertices on the graph (E, V) . By definition, two boundary maps applied in succession on a chain \mathbf{c}_i produce the zero map

$$\partial_{i-1} \partial_i = 0, \quad (3)$$

no matter the choice of \mathbf{c}_i . We equip vector spaces C_i with the standard inner product

$$\mathbf{c}_i \cdot \mathbf{c}'_i \equiv \mathbf{c}_i^T \mathbf{c}'_i, \quad (4)$$

that is, a dot product with addition modulo 2 over the pairwise multiplied coefficients. The inner product may be geometrically interpreted as a basis-independent parity measurement of the “overlap” of two *i*-chains. An *i*-chain \mathbf{c}_i with a zero boundary (*i.e.*, $\partial_i \mathbf{c}_i = 0$) is called an *i-cycle* (or simply, *cycle*). Note that *i*-cycles form a group under element-wise addition. A cycle \mathbf{c}_i is called *trivial* whenever there exists some chain $\mathbf{c}_{i+1} \in C_{i+1}$, such that $\mathbf{c}_i = \partial_{i+1} \mathbf{c}_{i+1}$. Cycles that cannot be formed in this way are called *non-trivial*. Trivial cycles form a normal subgroup of all cycles, such that the quotient groups

$$H_i \equiv \ker \partial_i / \text{Im } \partial_{i+1} \quad (5)$$

divide cycles into equivalence classes that are trivially related. The groups H_i are called *homology groups*, where two chains \mathbf{c}_i and \mathbf{c}'_i belong to the same class whenever $\mathbf{c}_i = \mathbf{c}'_i + \partial_{i+1} \mathbf{c}_{i+1}$.

The *dual complex* is another sequence $\overline{C}_D \rightarrow \overline{C}_{D-1} \rightarrow \dots \rightarrow \overline{C}_0$, where each \overline{C}_i shares the structure of C_{D-i} . Associated with the dual complex are the dual boundaries $\overline{\partial}_i : \overline{C}_i \rightarrow \overline{C}_{i-1}$. Chains in the dual complex are called *dual chains* or *cochains*, whereas those referring to the original complex are *primal* chains or, more succinctly, *chains*. For the three-dimensional lattice \mathcal{S} , vertices map to dual cells ($V \rightarrow \overline{Q}$), edges map to dual faces ($E \rightarrow \overline{F}$), faces map to dual edges ($F \rightarrow \overline{E}$), and cells map to dual vertices ($Q \rightarrow \overline{V}$), such that:

$$\begin{array}{ccccccc} \overline{C}_0 & \xleftarrow{\overline{\partial}_1} & \overline{C}_1 & \xleftarrow{\overline{\partial}_2} & \overline{C}_2 & \xleftarrow{\overline{\partial}_3} & \overline{C}_3 \\ \{\overline{\mathbf{v}}_k\} & & \{\overline{\mathbf{e}}_k\} & & \{\overline{\mathbf{f}}_k\} & & \{\overline{\mathbf{q}}_k\} \end{array}$$

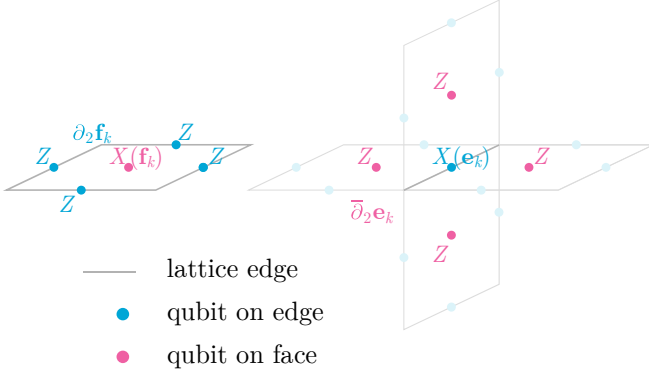


FIG. 1. A primal and dual stabilizer generator for a three-dimensional cluster state as defined in Sec. II B. The primal generator $X(\mathbf{f}_k)Z(\partial_2 \mathbf{f}_k)$ is associated with the highlighted face on the left of the figure. The dual generator $X(\mathbf{e}_k)Z(\partial_2 \mathbf{e}_k)$ is associated with the highlighted edge on the right of the figure. Stabilizer generators associated with the other faces and edges are not shown explicitly.

The dual boundaries on $\bar{\mathcal{S}} = (V, E, F, Q)$ behave similarly to their primal counterparts. With a slight abuse of notation, a dual cell $\bar{\mathbf{q}}_k = \mathbf{v}_k$ is mapped to the faces $\{\bar{\mathbf{f}}_j\}$ that enclose it, which correspond to the primal edges $\{\mathbf{e}_j\}$ incident to \mathbf{v}_k . A dual edge $\bar{\mathbf{e}}_k = \mathbf{f}_k$ is mapped to the endpoints $\{\bar{\mathbf{v}}_j\}$, which are the cells $\{\mathbf{q}_j\}$ adjacent to \mathbf{f}_k . Similar to cycles, *i-cocycles* are dual chains $\bar{\mathbf{c}}_i$ that have no coboundary. Dual boundaries share a similar concept as homology, called *cohomology*. Cohomology groups are formed in the same way as homology groups (Eq. 5), by replacing cycles $\ker \partial_i$ and trivial cycles $\text{Im } \partial_{i+1}$ by the dualized versions $\ker \bar{\partial}_i$ and $\text{Im } \bar{\partial}_{i+1}$, such that

$$\bar{H}_i \equiv \ker \bar{\partial}_i / \text{Im } \bar{\partial}_{i+1}. \quad (6)$$

B. Three-dimensional cluster states

We now make use of chain complexes $C_3 \rightarrow C_2 \rightarrow C_1 \rightarrow C_0$ to construct a fault-tolerant cluster state on a three-dimensional lattice \mathcal{S} , given the following recipe:

1. Place qubits on all basis elements of C_2 and \bar{C}_2 , *i.e.*, on all faces \mathbf{f}_k and edges \mathbf{e}_k of the lattice. Pauli operators on the qubits of an i -chain \mathbf{c}_i are denoted as $\sigma(\mathbf{c}_i) \equiv \prod_k \sigma^{z_k}$, where $\sigma \in \{X, Y, Z\}$. Qubits with $z_k = 0$ carry identity, and $z_k = 1$ carry σ .
2. For each face \mathbf{f}_k , define a *primal* stabilizer generator $g_k = X(\mathbf{f}_k)Z(\partial_2 \mathbf{f}_k)$. These operators generate stabilizers $X(\mathbf{c}_2)Z(\partial_2 \mathbf{c}_2)$ for all 2-chains $\mathbf{c}_2 \in C_2$.
3. For each edge \mathbf{e}_k , define a *dual* stabilizer $\bar{g}_k = X(\mathbf{e}_k)Z(\bar{\partial}_2 \mathbf{e}_k)$. These operators generate stabilizers $X(\bar{\mathbf{c}}_2)Z(\bar{\partial}_2 \bar{\mathbf{c}}_2)$ for all 2-cochains $\bar{\mathbf{c}}_2 \in \bar{C}_2$.

We depict an example of a primal and dual stabilizer generator associated with a single face \mathbf{f}_k and edge \mathbf{e}_k

in Fig. 1. Note that each stabilizer generator carries a Pauli- X operator on some face (edge) qubit and Pauli- Z operators on its direct neighbors on the (dual) boundary. This stabilizer composition is commonly associated with *cluster* or *graph* states [13]. Logical operators are derived from elements of the (co)homology groups of the underlying chain complex. For the logical identity channel, logical X -type operators correspond with elements of the homology group H_1 , and logical Z -type operators correspond with elements of the cohomology group \bar{H}_1 . The construction of a fault-tolerant channel from a cluster state that carries logical information is subtle: for details we refer the reader to Ref. [1]. We note that logical qubits are usually introduced by creating lattice boundaries—such as holes inside the bulk—by switching off stabilizers on these boundaries.

C. Error correction

For the cluster states described above, we can construct the primal (dual) *error syndrome* by measuring out the qubits on the faces (edges) in the Pauli- X basis. For each face \mathbf{f}_k (edge \mathbf{e}_k), this leads to a measurement outcome $\mu_k \in \{+1, -1\}$ ($\bar{\mu}_k \in \{+1, -1\}$). Error syndromes are constructed from measurement outcomes in the following way:

1. For each cell \mathbf{q}_k , we produce a *primal* error syndrome m_k as the product of measurement outcomes of the qubits that lie on its boundary: $m_k = \prod_{\mathbf{f}_j \in \partial_3 \mathbf{q}_k} \mu_j$. Note that m_k represents the measurement outcome of the stabilizer $s_k \equiv \prod_{\mathbf{f}_j \in \partial_3 \mathbf{q}_k} g_j = X(\partial_3 \mathbf{q}_k)Z(\partial_2 \partial_3 \mathbf{q}_k) = X(\partial_3 \mathbf{q}_k)$.
2. For each dual cell $\bar{\mathbf{q}}_k$ (*i.e.*, vertex \mathbf{v}_k), we produce a *dual* error syndrome $\bar{m}_k = \prod_{\mathbf{e}_j \in \bar{\partial}_3 \mathbf{v}_k} \bar{\mu}_j$. The syndrome represents the measurement outcome of the stabilizer $\bar{s}_k \equiv \prod_{\mathbf{e}_j \in \bar{\partial}_3 \mathbf{v}_k} \bar{g}_j = X(\bar{\partial}_3 \mathbf{v}_k)$.

Note that in the absence of errors, all error syndromes produce outcomes $m_k = +1$ and $\bar{m}_k = +1$. Since all cluster state qubits are measured out in the Pauli- X basis, we can restrict ourselves to probabilistic Pauli- Z errors—*i.e.*, phase-flips—on the qubits prior to syndrome measurement. These errors can appear *phenomenologically* as a result of i.i.d. sampling, or for example as a result of depolarizing or dephasing noise after a circuit or network operation. This model also includes measurement errors, because such an error is equivalent to a probabilistic Z gate before an X basis measurement. Measurement outcomes are then fully described by (anti)commutation relations of Pauli operators.

For notational convenience, let a particular set of primal error syndromes be described by a vector \mathbf{m} over the field \mathbb{Z}_2 , where the k th element m_k maps measurement outcomes $\{+1, -1\} \mapsto \{0, 1\}$. Given a set of Pauli- Z errors on dual edges (primal faces) described by a dual

chain $Z(\bar{\mathbf{c}}_1)$, the entire error syndrome takes the convenient form

$$\mathbf{m} = \bar{\partial}_1 \bar{\mathbf{c}}_1. \quad (7)$$

That is, the primal error syndrome corresponds to the *boundary* of all the Z -type errors on face qubits \mathbf{f}_k . Similarly, the dual syndrome $\bar{\mathbf{m}} = \partial_1 \mathbf{c}_1$ is the boundary of Pauli- Z errors described by the chain $Z(\mathbf{c}_1)$.

The decoding problem may now be stated as follows. Given a pair of primal and dual syndrome outcomes $\{\mathbf{m}, \bar{\mathbf{m}}\}$, we identify recovery chains $\bar{\mathbf{r}}_1$ and \mathbf{r}_1 such that $\bar{\partial}_1(\bar{\mathbf{r}}_1 + \bar{\mathbf{c}}_1) = 0$ and $\partial_1(\mathbf{r}_1 + \mathbf{c}_1) = 0$ —*i.e.*, the sum of recovery and error chains form *cycles* in the corresponding chain complex. We identify a *logical failure* whenever decoding introduces a logical X -type and/or Z -type error across the channel, which occurs whenever the sum of a recovery and error chain forms a *non-trivial* cycle.

We emphasize that we only evaluate fault tolerance of the cluster state as a pure *quantum memory*. That is, the noise thresholds that we determine exclusively assess the state’s ability to protect logical information and do not, *e.g.*, include the operations required to encode this logical information or operate on it.

III. METHODS

In the current work, we are interested in crystalline cluster states that are built from cellulations of flat three-dimensional space. There are various methods to find such structures. Two approaches that have previously been used are the splitting procedures on a known (foliated) structure, such as the cubic cluster state [6], and an algebraic approach based on combinatorial tiling theory [14]. We briefly discuss the former method below and show some of the lattices that were found through this method in Fig. 2. Although we have not used the latter method to construct new cluster states, we emphasize that the zoo of fault-tolerant cluster states merits further investigation under noise models considered here.

Below, we first discuss an extension of the concepts of the chain complex as discussed in Sec. II A. By adding indices that describe translational symmetry, we can use the chain complex to describe unit cells of a lattice that generate a full lattice. We discuss this method in Sec. III A. In Sec. III B, we describe the *cell-vertex splitting* operation of Nickerson and Bombín [6] in the context of this unit cell complex—this allows us to transform unit cells that describe three-dimensional lattices. In Sec. III C, we define a *face-edge splitting* operation that allows us to replace cluster state qubits with entangled states or Bell measurements. In Sec. III D, we introduce the noise models used for monolithic (*i.e.*, circuit-level) and network noise, and describe our method for generating entanglement in a distributed cluster state. In Sec. III E, we discuss how we use the stabilizer formalism to model and transfer Pauli errors in the circuits that we use to construct and measure cluster states, and

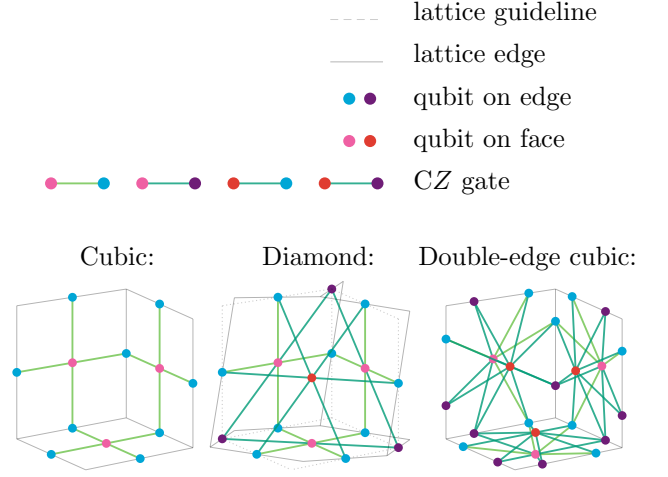


FIG. 2. Cluster states obtained through splitting. The cubic cluster state can be created by foliating the standard toric surface code. The diamond cluster state is obtained through two splits of the primal and dual vertex in the cubic unit cell. The complex is regular and self-dual, with each face connected to six edges. The double-edge cubic cluster state is obtained through multiple simple splits of primal and dual vertices. Each face is “double”-sided, supporting two different qubit that are connected to eight surrounding edges.

we elaborate on the numerical aspects of our model and simulations.

A. Unit cell complex

The *unit cell complex* is the set of basis elements (atoms) together with their boundary relations (bonds) in the crystal that forms a block with translation symmetry along the sides of the unit cell. We use the *Miller index* notation with square brackets $[abc]$ or (slightly unorthodox yet succinct) $[\mathbf{r}]$ for a translation $\mathbf{r} \equiv a\mathbf{x} + b\mathbf{y} + c\mathbf{z}$ along the *lattice vectors* \mathbf{x} , \mathbf{y} and \mathbf{z} that form the sides of the unit cell. Negative indices are denoted in the usual way as $[\bar{a}\bar{b}\bar{c}]$ or $[\bar{\mathbf{r}}]$ for a translation $\bar{\mathbf{r}} \equiv -a\mathbf{x} - b\mathbf{y} - c\mathbf{z}$.

The construction of the unit cell complex follows from the choice of lattice vectors. The subset of basis elements $\{\mathbf{b}_i\}$ in C_i that are equivalent under translations \mathbf{r} is mapped to a single *quotient* element \mathbf{q}_i , which serves as a basis vector for a new vector space Q_i over \mathbb{Z}_2 . Similarly, the boundary relation between elements $(\mathbf{b}_i)_n$ and $(\mathbf{b}_{i-1})_m$ is mapped to a relation to their quotient elements $(\mathbf{q}_i)_n$ and $(\mathbf{q}_{i-1})_m$ in the form of a *quotient* boundary map $\partial_i^{[\mathbf{r}]} : Q_i \mapsto Q_{i-1}$. Because some of the boundary relations are present between elements *across* two unit cells (such as a face with boundary edges from adjacent unit cells), quotient boundaries have a Miller index $[\mathbf{r}]$ that represents the translation \mathbf{r} required to jump to its neighboring unit cell. Intracellular boundaries are encoded by $\partial_i^{[0]}$, whilst intercellular boundaries are encoded by $\partial_i^{[abc]}$ for a non-zero translation $\mathbf{r} \equiv a\mathbf{x} + b\mathbf{y} + c\mathbf{z}$. A

more detailed description of the quotient boundaries in the unit cell complex can be found in App. A.

Given a unit cell complex, an *embedding* is a map that takes each Q_i and the quotient maps $\partial_i^{[r]}$ to a crystalline chain complex $C_3 \rightarrow C_2 \rightarrow C_1 \rightarrow C_0$, with lattice dimensions given by the embedding. For a periodic lattice of $N \equiv N_{\mathbf{x}} \times N_{\mathbf{y}} \times N_{\mathbf{z}}$ unit cells along the \mathbf{x} , \mathbf{y} , and \mathbf{z} lattice directions, respectively, vector spaces of chains take the form

$$C_i = Q_i^{\oplus N} = Q_i \otimes L. \quad (8)$$

Here, $L \equiv \mathbb{Z}_2^{\oplus N}$ is an N -dimensional vector space over \mathbb{Z}_2 . The N basis vectors of L represent N lattice points, such that linear combinations in L may be associated with the subset of lattice points that have non-zero coefficients. Intuitively, the embedding is realized by repeating the unit cell elements Q_i over each lattice point given by a displacement vector \mathbf{r} . We formalize this process in App. B.

B. Cell-vertex splitting

The splitting procedure of Nickerson and Bombín [6] may be phrased in terms of the unit cell complex introduced in Sec. III A. A visual example of a split in two dimensions can be seen in App. B, Fig. 14a and b; splitting each face of a square lattice diagonally results in the triangular lattice shown adjacently. Here, we review the general case of an n -split in three dimensions, with a simple split following from the case $n = 1$. Importantly, the splitting number n alone is not sufficient to uniquely characterize a split: one should also specify the new boundary relations between split vertices and edges. Denote the split vertex with \mathbf{v}_0 . The recipe for an n -split is then as follows:

1. Let $E' = \{(\mathbf{v}_j, \mathbf{v}_0) \mid \mathbf{v}_j \in N_0\}$ be the set of incident edges on \mathbf{v}_0 with neighborhood N_0 . Choose n disjoint subsets $E'_i \in E'$ ($i = 1 \dots n$) that will each connect to a new vertex.
2. Create n new vertices \mathbf{v}_i , and connect each \mathbf{v}_i to the incident edges E'_i . The \mathbf{v}_0 vertex connects to the remaining edges $E'_0 = E' \setminus \bigcup_i E'_i$, which might be the empty set.
3. Create n new edges $\mathbf{e}_i = (\mathbf{v}_i, \mathbf{v}_0)$. The corresponding boundary relations are encoded in $\partial_1^{[0]}$ of the unit cell complex with Miller index [0].
4. Fix the remaining boundary maps $\partial_2^{[r]}$. That is, for each \mathbf{v}_i and $\forall \mathbf{r}$, calculate the dual boundary $\mathbf{c}_2 \equiv \sum_{\mathbf{p}} \partial_2^{[\mathbf{p}]} \partial_3^{[\mathbf{r}-\mathbf{p}]} \mathbf{v}_i$. By the zero map conditions (Eq. (A2) in App. A), the right-hand side should be zero. That is, we connect faces $\mathbf{f}_j \in \mathbf{c}_2$ to the newly created edge \mathbf{e}_i with Miller index $\bar{\mathbf{r}}$.

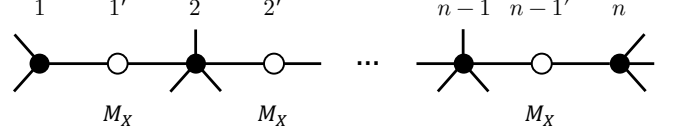


FIG. 3. A subgraph of $2n - 1$ qubits. Qubits marked primed indices are measured in the X basis. Unmeasured qubits are connected arbitrarily to the rest of the graph. If one assumes that every measurement outcome $m = 0$, the unmeasured qubits can be initialized in an n -partite Bell/GHZ state.

C. Face-edge splitting

The cell-vertex splitting procedure described before in Sec. III B changes both the number of syndromes and the connectivity between syndromes in the syndrome graph. We can define an additional splitting operation on faces (dual edges) of such a complex. We discuss this operation in this section.

Usually, a cluster state as described in Sec. II B is constructed with the aid of CZ gates on qubits initialized in the $|+\rangle$ state. Alternatively, one may replace CZ gates with other entangling operations that lead to the same stabilizer states.

Consider the subgraph of a graph state as in Fig. 3. Qubits at odd positions are marked with integers $i \in \{1, 2, \dots, n\}$ and their right neighbors at even positions with a primed index $i' \neq n'$. In this graph state, odd qubits have an arbitrary number of neighbors, whereas even qubits only neighbor the two odd qubits on either side. After measuring the even qubits i' in the Pauli- X basis, the post-measurement $m_{i'} = 0$ graph state stabilizers are

$$S_{\text{post}} = \langle X_{1'}, X_{2'}, \dots, X_{n-1'}, \prod_{i=1}^n (X_i \prod_{j \in N(i)} Z_j), Z_1 Z_2, Z_2 Z_3, \dots, Z_{n-1} Z_n \rangle. \quad (9)$$

Here, $N(i) = \{j \mid (i, j) \in E\}$ are the qubits connected to the odd qubit i *outside* the subgraph depicted in Fig. 3. The disentangled measured qubits play no further role in the graph state stabilizers. In practice, one may replace these “virtual” qubits with measurement outcomes $m = 0$ and initialize the unmeasured qubits in the state stabilized by Eq. (9)—the resulting state is the same. In constructing graph states, CZ gates transform an X -type stabilizer $\prod_{i=1}^n X_i \mapsto \prod_{i=1}^n (X_i \prod_{j \in N(i)} Z_j)$, whilst leaving the Z -type stabilizers untouched. This means that we can alternatively initialize the odd qubits i as an n -qubit $|\text{GHZ}_n\rangle$ state stabilized by $\langle \prod_{i=1}^n X_i, Z_1 Z_2, Z_2 Z_3, \dots, Z_{n-1} Z_n \rangle$, before applying the CZ gates to qubits outside the subgraph. If $n = 2$, one can initialize with the bipartite Bell state $\langle X_1 X_2, Z_1 Z_2 \rangle$.

The above procedure shows how the subgraph in Fig. 3

need not be initialized through CZ gates, so long as there is a protocol that can create $|\text{GHZ}_n\rangle$. Our new splitting procedure on faces produces subgraphs like Fig. 3 in a systematic way. Importantly, the closed-cell stabilizers of the cluster state stay intact on the split geometry.

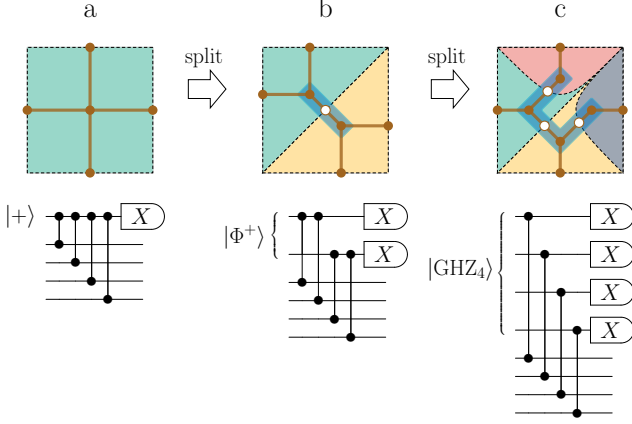


FIG. 4. Face splitting on a square. (a) Without a split, the square represents a monolithic cluster state without distributed entanglement. The corresponding circuit of $|+\rangle$ state initialization, CZ gates, and Pauli-X basis measurement is also drawn. (b) A hybrid approach. The distributed face qubit is split over two parties represented by two faces, forming a bipartite GHZ or Bell state. The parity check is performed through two bilocal CZ gates of each face to its neighboring qubits and two local measurements. (c) A fully distributed approach. A face is split into four parties that share a GHZ state. Each party performs a single CZ gate to its neighbor and performs a measurement.

Like a cell-vertex split, a face-edge split subdivides an existing face into two or more parts, adding new edges to separate the newly created faces. We give an example of this procedure on a square in Fig. 4. The newly created edges each support an additional qubit, always laying adjacent to two faces. Therefore, the subgraph supported by split edges and faces is a chain in the form of Fig. 3. We may replace the cluster state supported by the n connected faces with an n -partite GHZ state. Because the removed qubits correspond to “virtual” $m = 0$ measurement outcomes, their even parity plays no further role in the evaluation of error syndromes.

We may extend this procedure to both the primal and dual complex, producing GHZ states on both faces and edges. An example based on the cubic cluster state is given in Fig. 5. Starting from a monolithic architecture, a full 4-partite split of primal faces produces an architecture with nodes containing five qubits on a single edge and each of the adjacent split faces. Nodes are entangled with one another by GHZ states on every face. We may perform the same procedure for dual faces (primal edges), further reducing the number of qubits in each node to two.

We do not consider this in the rest of the paper but note that, instead of initializing the qubits introduced

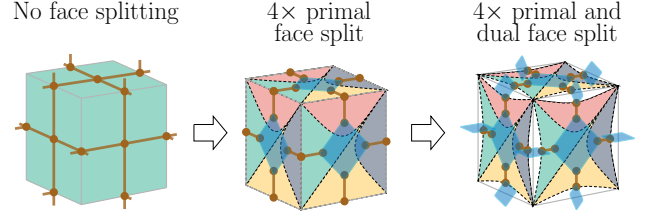


FIG. 5. Face-edge splitting on a cubic cell. Starting from the left, a monolithic cluster state is transformed into a distributed cluster state by a four-partite split for each of its faces (see also Fig. 4). Performing the same procedure for dual faces (primal edges) produces a cluster state that is fully distributed, where each network node contains two cluster state qubits.

by a face-edge split as an entangled state and measuring them out individually, one can alternatively initialize these qubits regularly in $|+\rangle$ and measure them out with a joint (*Type-II fusion measurement* [15–17]). This provides one with a method to transform (fault-tolerant) cluster states into so-called *fusion networks* that form the basis of *fusion-based quantum computing* [9].

D. Circuit-level and network noise

In this section, we describe the noise models used for monolithic and distributed threshold calculations. For the monolithic simulations, the entire cluster state is built from $|+\rangle$ state preparation, followed by CZ gates between every connected face-edge pair in the cluster state, concluded with a Pauli-X basis measurement of every qubit. In this model, we do not consider the effects of memory decoherence. For circuit-level noise, the following noise sources are included:

1. Noisy state preparation as a classical mixture $(1 - p_p)|+\rangle\langle+| + p_p|-\rangle\langle-|$. That is, with probability p_p , apply a phase-flip error on $|+\rangle$.
2. Two-qubit depolarizing noise following every CZ gate. That is, apply the channel $D(\rho) = (1 - p_g)\rho + p_g/15 \sum_{A,B} (A \otimes B)\rho(A \otimes B)^\dagger$, where $(A, B) \in \{\mathbb{I}, X, Y, Z\}^2 \setminus (\mathbb{I}, \mathbb{I})$ is a pair of Pauli matrices excluding identity.
3. Classical bit-flips following every Pauli-X basis measurement. That is, a measurement outcome m is flipped to $1 - m$ with probability p_m . The corresponding noise channel takes on the form $\tilde{P}^\pm(\rho) = (1 - p_m)P^\pm\rho P^\pm + p_m P^\mp\rho P^\mp$, where $P^\pm \equiv |\pm\rangle\langle\pm|$ are the desired projectors in the Pauli-X basis.

For the entangled states used in the distributed simulations, we assume that parties have shared access to

Bell states in the *Werner* or *isotropic* form

$$\rho_{p_n} = (1 - p_n) |\Phi^+\rangle\langle\Phi^+| + \frac{p_n}{3} |\Phi^-\rangle\langle\Phi^-| + \frac{p_n}{3} |\Psi^+\rangle\langle\Psi^+| + \frac{p_n}{3} |\Psi^-\rangle\langle\Psi^-|, \quad (10)$$

where $|\Phi^+\rangle = (|00\rangle + |11\rangle)/\sqrt{2}$, $|\Phi^-\rangle = Z|\Phi^+\rangle$, $|\Psi^+\rangle = X|\Phi^+\rangle$, and $|\Psi^-\rangle = X \otimes Z|\Phi^+\rangle$. We refer to p_n as the parameter that describes “network noise”. In this paper, we do not take into account specific physical systems. We justify using Werner states by noting that a depolarizing channel (*i.e.*, the most general noise channel) converts a perfect Bell state to a Werner state. Additionally, Werner states can be considered a general proxy for non-perfect Bell states, because every Bell state can be *twirled* [18] into a Werner state using local operations and classical communication.

To generate GHZ states from these Bell states, we use the very straightforward method based on local parity measurements [19]. Fundamentally, a GHZ state may be created from Bell pairs by a local projective measurement of the ZZ parity between two halves of two pairs shared by multiple parties. The circuits to create a 3-partite GHZ from 2 Bell states and a 4-partite GHZ state from 3 Bell states are drawn schematically in Fig. 6.

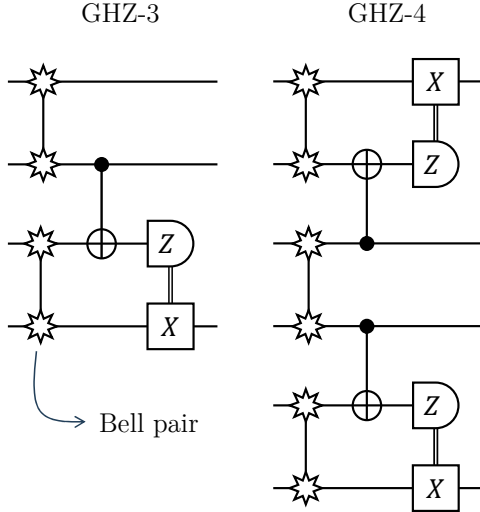


FIG. 6. Fusion circuits for 3- and 4-partite GHZ states. In each of the circuits, Bell states are fused through the application of local CX gates and a subsequent Z basis measurement of the target bits. The conditional bit-flips ensure that GHZ states have the desired form $|0\dots 0\rangle + |1\dots 1\rangle$.

We note that the circuits in Fig. 6 do not include distillation. Better quality GHZ states can be generated if the Bell states used to carry out the projective measurement are pre-distilled, or if GHZ states are distilled after creation. This, however, introduces a probabilistic factor to the GHZ creation protocols, which leads to higher numerical complexity in simulating the circuits.

E. Numerical tools and considerations

To decode error syndrome graphs, we have implemented a version of the *Union-Find* (UF) decoder [11]. This decoder is particularly attractive given its almost-linear time complexity and ease of implementation for both Pauli and erasure errors. Despite being a sub-optimal decoder, the phenomenological threshold for the cubic cluster state (2.6%) is very close to that of the *Minimum-Weight Perfect-Matching* (MWPM) decoder (2.9%) [6], which is in turn not far from the optimal threshold (3.3%) [20]. Furthermore, since the Union-Find decoder is maximum-likelihood over the erasure channel (due to its built-in *peeling* decoder [21]), we expect this performance gap to shrink further over noisy channels that are a combination of Pauli and erasure errors. We have specifically implemented the “weighted-growth” version of the Union-Find decoder, as introduced in the original Union-Find manuscript [11]. Our implementation can be found in the repository of Ref. [22].

Error models per unit cell are constructed with a circuit simulator. Unit cells are defined according to the description in Sec. III A, together with the splitting methods of Secs. III B and III C. The simulator is implemented as a classical efficient stabilizer simulator. Per operation, the simulator applies a full error channel as a series of Pauli operators, ending with a measurement in the n -qubit Pauli basis. The choice for Pauli noise is justified in these simulations because the input states are (convex combinations of) stabilizer states, the operation noise is typically Pauli noise and measurement noise is described by classical bit-flips. In App. C, we describe the details of constructing an error channel. Pauli twirling a state at the end of a series of non-Clifford operations is equivalent to Pauli twirling the individual operations before applying them. This allows us to also use the simulator in situations where one is interested in the Pauli-twirled version of the full error channel—in that case, it suffices to twirl the individual channel components before applying them.

The calculated error models are used to sample errors in Monte Carlo style on qubits of the full crystalline cluster state. These cluster states are constructed from the associated unit cell complex according to the crystal embedding procedure described in App. B. Per Monte Carlo sample, we decode the syndrome graph and assign a logical failure whenever there is a logical error for a single pair of logical X and Z operators of the channel—see Sec. II C for more details.

IV. RESULTS

The results are organized into three parts, where the noise models become increasingly complex. A summary of the most important thresholds found can be found in Fig. 7.

Secs. IV A and IV B provide thresholds for various ge-

ometries under a phenomenological noise model. In these sections, we reproduce earlier-known results and investigate the influence of a lattice boundary. For the phenomenological noise model, cluster states based on lattices that have a lower vertex degree are more resilient against errors. However, these cluster states typically require more two-qubit gates to construct. This aspect is not regarded by the phenomenological noise model.

To investigate the trade-off between noise resilience and noise introduced by constructing the cluster state, we investigate scenarios with circuit-level and network noise in Secs. IVC and IVD. Sec. IVC discusses numerical thresholds for monolithic (*i.e.*, non-distributed) architectures. These results are compared with the circuit-based error models for cluster states on a distributed network in Sec. IVD.

In the last part of this section, in Sec. IVE, we investigate how the GHZ success probability of distributed networks can be traded off against a higher erasure probability to achieve fault-tolerance against larger infidelity of the entangled states used.

A. Phenomenological thresholds

Before considering more realistic noise models, we first numerically evaluate erasure and phenomenological thresholds for several known lattices, which were constructed using cell-vertex splitting. These are the cubic, diamond, triamond, and so-called double-edge cubic cluster states. We go beyond the results in Ref. [6] by determining *fault-tolerant regions* against erasure and phenomenological errors. The results are depicted in Fig. 8. We consider a phenomenological error model. It corresponds to perfect state preparation of the cluster state followed by measurements that fail to report (*i.e.*, erase) an outcome with probability p_e or flip the outcome with probability p_m , both of which are i.i.d. In Fig. 8, we estimate a fault-tolerant region for the erasure probability and bit-flip probability. Data points of the fault-tolerant regions are calculated by sweeping over both error probabilities while keeping their ratio fixed. We use a different constant of proportionality at each data point on the fault-tolerant boundary.

The isolated thresholds found for both types of noise can be found in Fig. 7a. The thresholds reported are slightly higher than those reported in Ref. [6]. This can be attributed to a slightly different implementation of the Union-Find decoder—see Sec. IIIE for more details. Because phenomenological noise does not take into account higher error rates that arise with increasingly complex preparations of a cluster state, a fairer comparison is made by weighing each qubit error probability with the cluster state valency z , producing i.i.d. noise with probabilities zp_e and zp_g instead. Such a model is called *weighted phenomenological* error model in Ref. [6]. Because the cubic, diamond, triamond, and double-edge cubic lattices are regular with valency $z \in \{4, 6, 8, 10\}$ on

all qubits respectively, weighted thresholds may be calculated from unweighted thresholds by simply dividing by z .

B. Phenomenological thresholds with boundaries

The lattices considered above repeat periodically in all three spatial directions. It may be difficult to prepare such a cluster state if we take into account the connectivity of the qubits. We gauged how the performance of a cluster state is affected by the introduction of boundaries, under the same phenomenological noise model of both bit-flips with probability p_g and erasure errors with probability p_e . Each lattice is introduced to a smooth boundary along the $x = 0$ plane, and a rough boundary along the $y = 0$ plane. For the smooth boundary, we remove elements of the correlation surface defined by a *dual* logical membrane and its closure and introduce a boundary on the remaining dangling edges in the dual complex. The rough boundary is introduced in the same way, by swapping primal and dual notions.

In Figs. 7b and 8b-e, we show how the threshold values for the phenomenological and erasure noise models change with the introduction of the boundaries. The results show that differences in the phenomenological bit-flip threshold values are insignificant. For the erasure thresholds, only the cubic and double-edge cubic lattices are slightly affected. The results indicate that the introduction of boundaries affects the thresholds only minimally, at least for the lattices and noise models considered here. However, we found that boundaries *do* impose significantly weaker *sub-threshold scaling*. The sub-threshold scaling is the rate at which the logical error rate is suppressed below the threshold. Fig. 9 shows that boundaryless architectures have a favorable error probability suppression that is about 50 – 100% as effective as the same architecture with boundaries. These results are consistent with recent work, where it was also shown that the introduction of boundaries leaves the threshold nearly invariant, but negatively impacts error rate scaling by roughly the same factor [23].

C. Monolithic thresholds

To compare distributed thresholds against monolithic implementations of the same geometry, we first benchmark monolithic architectures with the circuit-level noise models from Sec. IIID. In the results that follow, we set all probabilities $p_p = p_g = p_m \equiv p_o$ equal (see Sec. IIID for the definition of the parameters), and sweep a threshold over the value of p_o . What remains is a specification of the ordering of CZ gates in the circuits, since pure CZ gates all commute with one another, whereas their noisy versions do not. Because no qubit can interact with two CZ gates simultaneously, a valid ordering may be extracted from an edge coloring of the correspond-

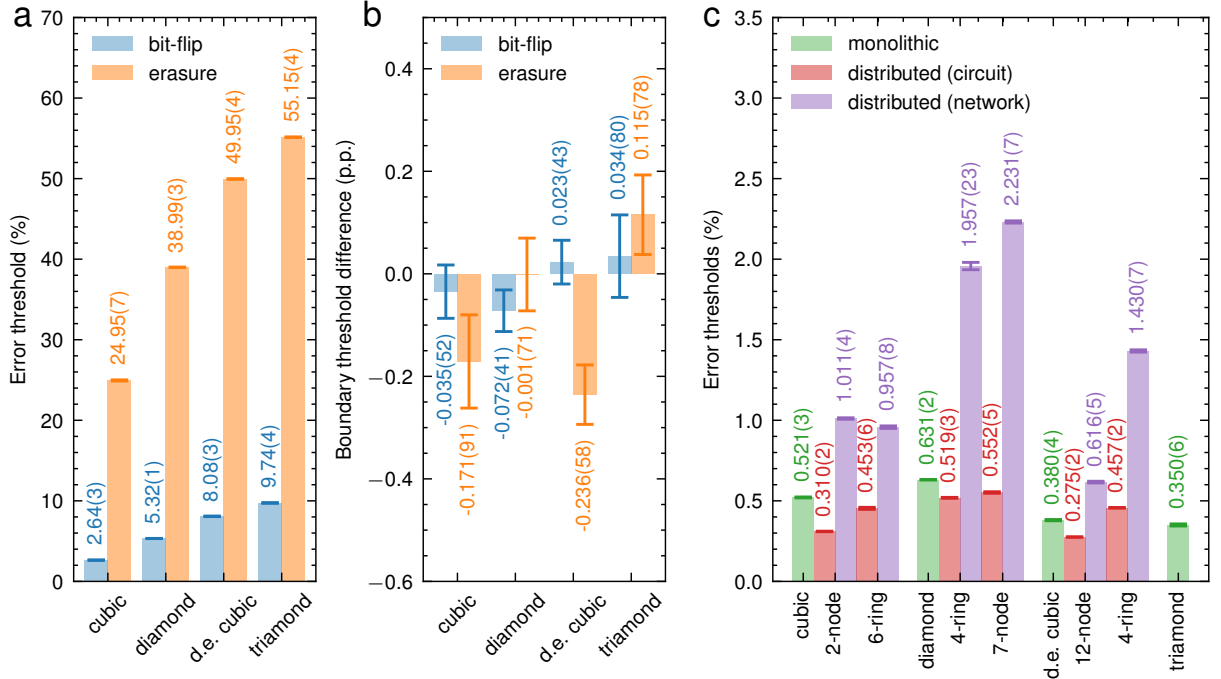


FIG. 7. Overview of the threshold values found in our study, for different lattices and noise models considered. **(a)** Summary of numerical thresholds for bit-flip and erasure-type phenomenological noise models of the four lattices described in this work. **(b)** Differences of estimated threshold values for the same lattices and error models as in (a) with boundaries, expressed in percentage points (p.p.). **(c)** Monolithic thresholds are gauged against circuit-level errors with a common error rate as defined in Secs. IIID and IV C, and similarly for distributed thresholds gauged under only circuit-level noise. Network thresholds are obtained from the model as defined in Sec. IIID. For the monolithic thresholds, we show only the highest threshold value found with alternative gate orderings (see Sec. IV C in the main text for details). Numbers above the bars indicate the threshold, with 95% confidence intervals around the value indicated as error bars and as decimals in parentheses.

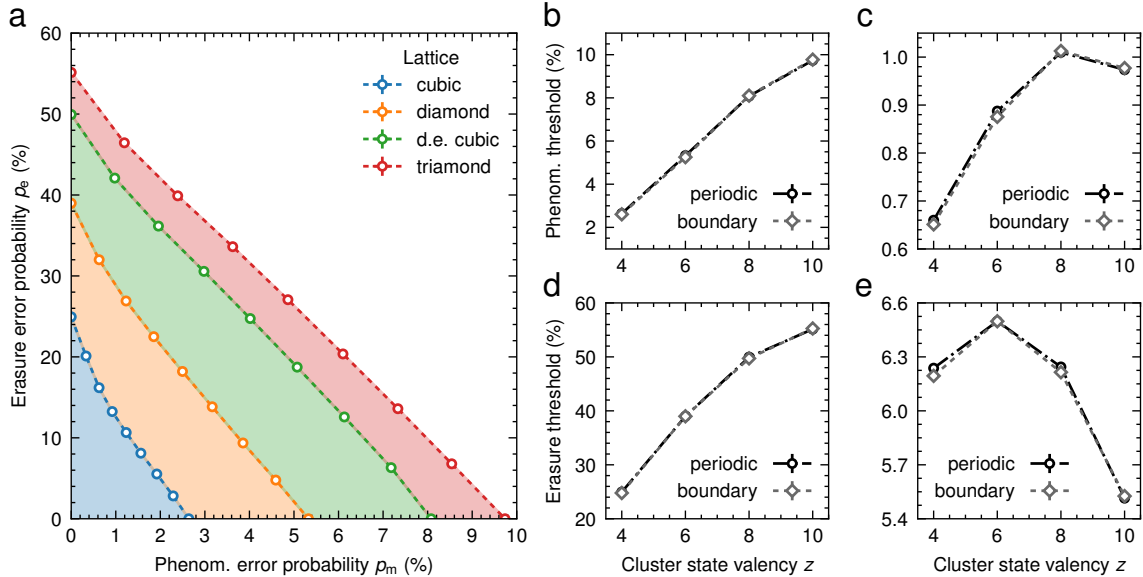


FIG. 8. Fault-tolerant regions of the cubic, diamond, double-edge (d.e.) cubic, and triangular lattices under phenomenological bit-flip and erasure noise. **(a)** Regions are estimated based on individual threshold values. **(b)** Thresholds $p_{m,th}$ under pure bit-flip noise as a function of cluster state valency. From left to right is the cubic, diamond, double-edge cubic, and triangular lattice. We provide thresholds with and without lattice boundaries—see Sec. IV B for more details. **(c)** Thresholds $p_{m,th}/z$ under pure weighted bit-flip noise, where z is the cluster state valency. **(d)** Thresholds $p_{e,th}$ under pure erasure noise. Results agree with known percolation thresholds for the given lattices. **(e)** Thresholds $p_{e,th}/z$ under pure weighted erasure noise.

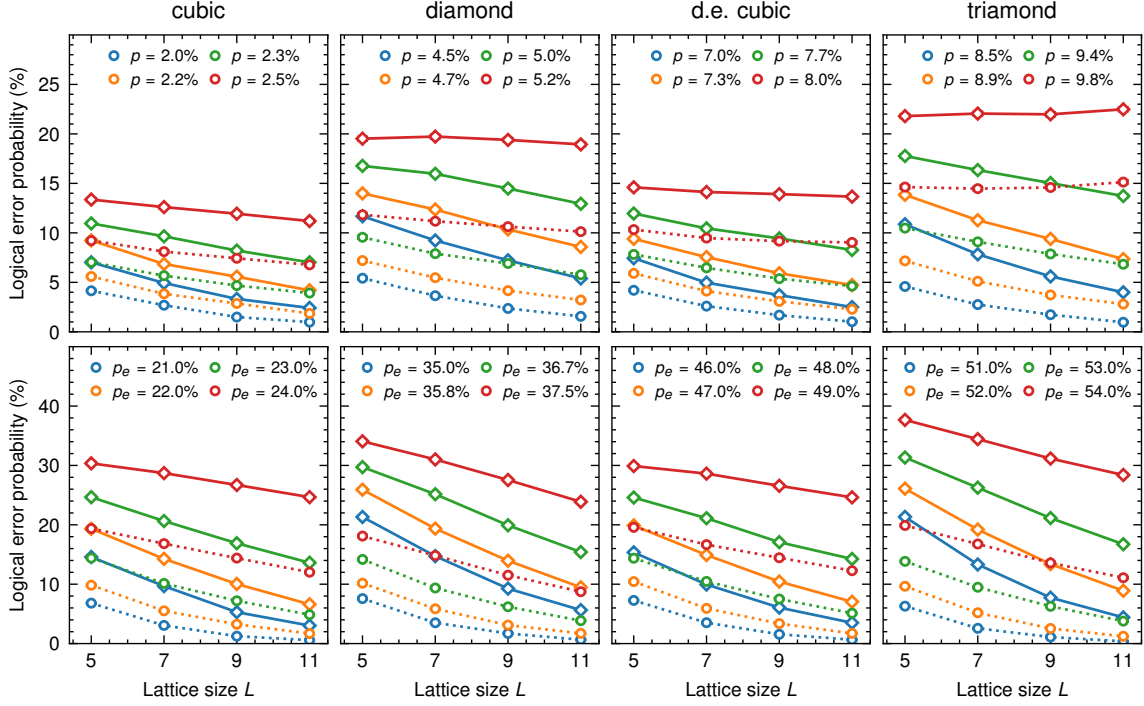


FIG. 9. Sub-threshold logical error rate scaling of cubic, diamond, double-edge (d.e.) cubic, and triamond lattices with and without boundaries for phenomenological bit-flip (top row) and erasure (bottom row) noise models. Data for periodic conditions has round markers with a dashed line. Although the thresholds are minimally affected, sub-threshold error probabilities of any lattice with a rough and smooth boundary are about 1.5 – 2 times the rate for the same lattice under periodic boundary conditions.

ing ∂_2 boundary map of the cluster state. This graph is bipartite by definition. Therefore, the chromatic index (*i.e.*, the minimum number of colors needed for an edge-coloring) equals the maximum degree of any vertex in the graph—*i.e.*, the maximum valency of the cluster state [24]. The cubic, diamond, double-edge cubic, and triamond lattices are all regular, and so their chromatic indices are 4, 6, 8, and 10, respectively.

For a chromatic index i and a given coloring, there are $i!$ different ways to order the edges and thus the CZ gates. Rotational and reflection symmetries of the cubic cluster state imply that there are only two colorings that correspond to a unique sequence of gates: a “(counter)clockwise” sequence going around a face, and a “zigzag” sequence jumping to opposite sides first. For diamond, double-edge-cubic and triamond lattices, the number of orderings quickly explode as $6! = 720$, $8! \approx 4 \times 10^4$ and $10! \approx 3.6 \times 10^6$ (not taking into account symmetries). We have not included an exhaustive search of all orderings and their corresponding thresholds up to symmetries, but only investigate a subset of orderings.

Monolithic thresholds with the (counter)clockwise orderings are shown in Fig. 10. An overview of all monolithic thresholds is included in Fig. 7c. For the cubic lattice, a zigzag ordering of CZ gates slightly outperforms the (counter)clockwise ordering. The diamond lattice outperforms a cubic cluster state, whereas the double-edge cubic lattice drops in performance for the orderings we considered. In all cases, (counter)clockwise orderings of the gates (at least in the primal complex) tend to produce lower thresholds than orderings that skip multiple edges at a time. We can intuitively understand this result by considering that a single Pauli- X error on a face qubit spreads through all subsequent CZ gates as correlated Pauli- Z errors to neighboring edge qubits. In a (counter)clockwise ordering, the Pauli- Z errors produce a single strand that wraps around the face. In a zigzag orientation, Z error strings can form disconnected chains, such that a single Pauli- X error produces multiple pairs of syndromes on different sides of the face. Initial numerical analysis shows that (counter)clockwise orderings indeed seem to result in, on average, longer error strings. The triamond lattice was gauged for a single ordering due to its complexity and performed poorly.

In the limiting case that all gate errors are Pauli- Z errors, the relative impact of gate order disappears, and thresholds coincide with the weighted phenomenological thresholds considered above. In Ref. [14], Newman *et al.* consider an error model that, after each CZ gate, applies an X -type error on a (primal) face qubit with probability p_X and a Z -type error on a (primal) edge qubit with probability p_Z . Their results show that the best-performing lattice in terms of threshold moves from higher to lower valency as Pauli- X errors start to dominate. Our results show a similar tendency under depolarizing gate noise for the higher valent double-edge cubic lattice and triamond lattices when compared to the lower valent cubic and diamond lattices. We can summarize

this finding by stating that as the cluster state valency increases, depolarizing noise incurs a larger cost on the threshold value. Under these noise models, there exists an optimal threshold resulting from a trade-off between the complexity of the geometry of the cluster state, and the structure of the noise created by the circuit. Noise bias is one example where this trade-off may be abused, by taking advantage of the architecture of the cluster state in either primal or dual lattice structures.

The monolithic cluster state thresholds do not compete with surface code monolithic thresholds, which are estimated at 0.90% and 0.95% under the same noise model [25]. It should be noted that the numbers for all non-cubic lattices provided here are likely sub-optimal, as we have only gauged the performance for a subset of CZ gate orderings. Nevertheless, estimates show that cluster states defined on the diamond lattice can outperform the cubic cluster state in the presence of circuit-level noise. For cluster states with even higher valencies, the cost of initialization negatively impacts the value of the threshold, consistent with the results under weighted phenomenological noise models. These results warrant further optimizations of the gate orderings and comparisons with other lattices.

D. Distributed thresholds

In this section, we investigate thresholds for distributed implementations of the cluster states. We use the face-edge splitting operation of Sec. III C to split faces of the cubic, diamond, and double-edge cubic lattices. The cubic lattice was gauged for two different splits: one along the diagonals, producing network nodes with six cluster qubits in a ring and entangled through Bell states, and one on the entire face, producing 2-qubit nodes that are entangled through 4-partite GHZ states. Both these architectures also appear in the context of fusion-based quantum computation in Bartolucci *et al.* [9]. The structure of the diamond lattice is more intricate, and we produce two different architectures that contain only 3-partite GHZ states (the 4-ring) and an architecture with a mixture of Bell states and 3-partite GHZ states. Architectures for the double-edge cubic lattice resemble the cubic lattice: the first contains only Bell states, whilst the other shares 4-partite GHZ states. All architectures are drawn schematically in Fig. 11. We present an investigation on the stabilizer fidelity of the distributed unit cell implementations of Fig. 11 in App. E.

In the distributed model, we set $p_p = p_g \equiv p_o$. On top of that, we use entangled states to connect the cluster states that are separated by the splits. For these states, we assume that nodes can prepare Werner states with fidelity $1 - p_n$ and that GHZ states are generated with the protocols of Fig. 6. In the absence of network links, this circuit-level model reduces to the model discussed in Sec. IV C.

Using this physical model, we numerically establish

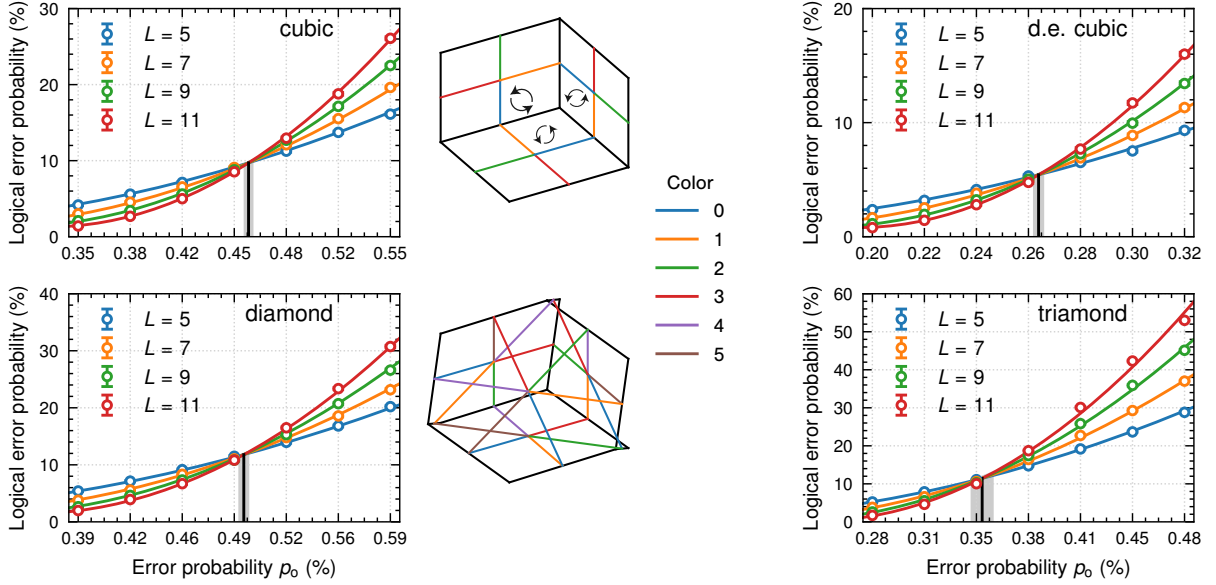


FIG. 10. Monolithic thresholds $p_{o,\text{th}}$ for specific gate orderings. The thresholds are calculated with, for the cubic and diamond unit cell, a straightforward order of CZ gates according to a “(counter)clockwise” coloring of the diagram on the right (see arrows in the diagram for the cubic lattice). We omit details on (counter)clockwise orderings for the double-edge cubic and triamond lattice. Gates of a single color are performed simultaneously for all unit cells in the lattice. Each data point constitutes 50000 samples. Error bars of the logical error rate are given as 95% confidence intervals but are too small to be discernible in most cases. The threshold value is highlighted with a 95% confidence interval based on its least-squares estimate of a second-order polynomial of the logical error rate around the threshold value—see App. D for details.

thresholds for the distributed lattices introduced above. In the same way as phenomenological bit-flip and erasure thresholds, we estimate fault-tolerant regions for error probabilities p_o and p_n . The results are shown in Fig. 12 and in the overview of Fig. 7c.

These results show that network error rate thresholds for both the 2-node and 6-ring cubic designs are similar. Even though higher-valent GHZ states tend to produce lower-quality stabilizers, we suspect that the superior performance of the 2-node design may be due to the size of its node. Because the 2-node architecture has only a single CZ gate per node, errors spread to only one adjacent qubit, as opposed to the 6-ring architecture that moves such an error to two adjacent qubits. Despite the more significant propagation of errors, the 6-ring architecture has a higher threshold against gate and measurement errors (the probability p_o) than the 2-node design. Most likely, this is because measurement errors have a higher influence in the 2-node architecture, which has more qubits. Both diamond architectures outperform the cubic lattice by a factor of roughly two. This result is promising, especially considering that the depolarizing thresholds also outperform cubic architectures, as was already the case for the monolithic thresholds discussed above.

For the double-edge cubic architectures, network error rate thresholds drop again. Nevertheless, the 4-ring design with GHZ states outperforms the bigger 12-node design, which we may explain in the same way as in the cubic case: in the 4-ring lattice, errors propagate to fewer

neighboring qubits compared to the 12-node architecture. An important difference with the cubic lattices is that the 4-ring design benefits enough from the lower error spreading to keep outperforming the 12-node design under pure gate and measurement noise, despite the additional qubit measurements in the 4-ring implementation.

Thresholds for the circuit-level noise rate $p_{o,\text{th}}$ of these architectures are not optimal, due to the way that the ordering of CZ gates affects the threshold. We find that, for all architectures considered, the values found for $p_{o,\text{th}}$ without network noise are similar to the monolithic thresholds. Similar to the earlier analysis of distributed designs, comparing distributed to monolithic architectures involves trading off less error propagation in the distributed architectures versus fewer measurement errors in the monolithic architecture.

E. Trade-off GHZ success rate and erasure probability

The results in the previous sections indicate that fault tolerance can be achieved with network error probabilities below $\sim 2\%$ —*i.e.*, with Bell state fidelities above $\sim 98\%$. This condition can be challenging from an experimental perspective. Fortunately, it is possible to boost the fidelity of entanglement before cluster state preparation through entanglement distillation. In this process, we can model distillation failures as erasures on the corresponding qubit(s), and use a suitable decoder, such as the

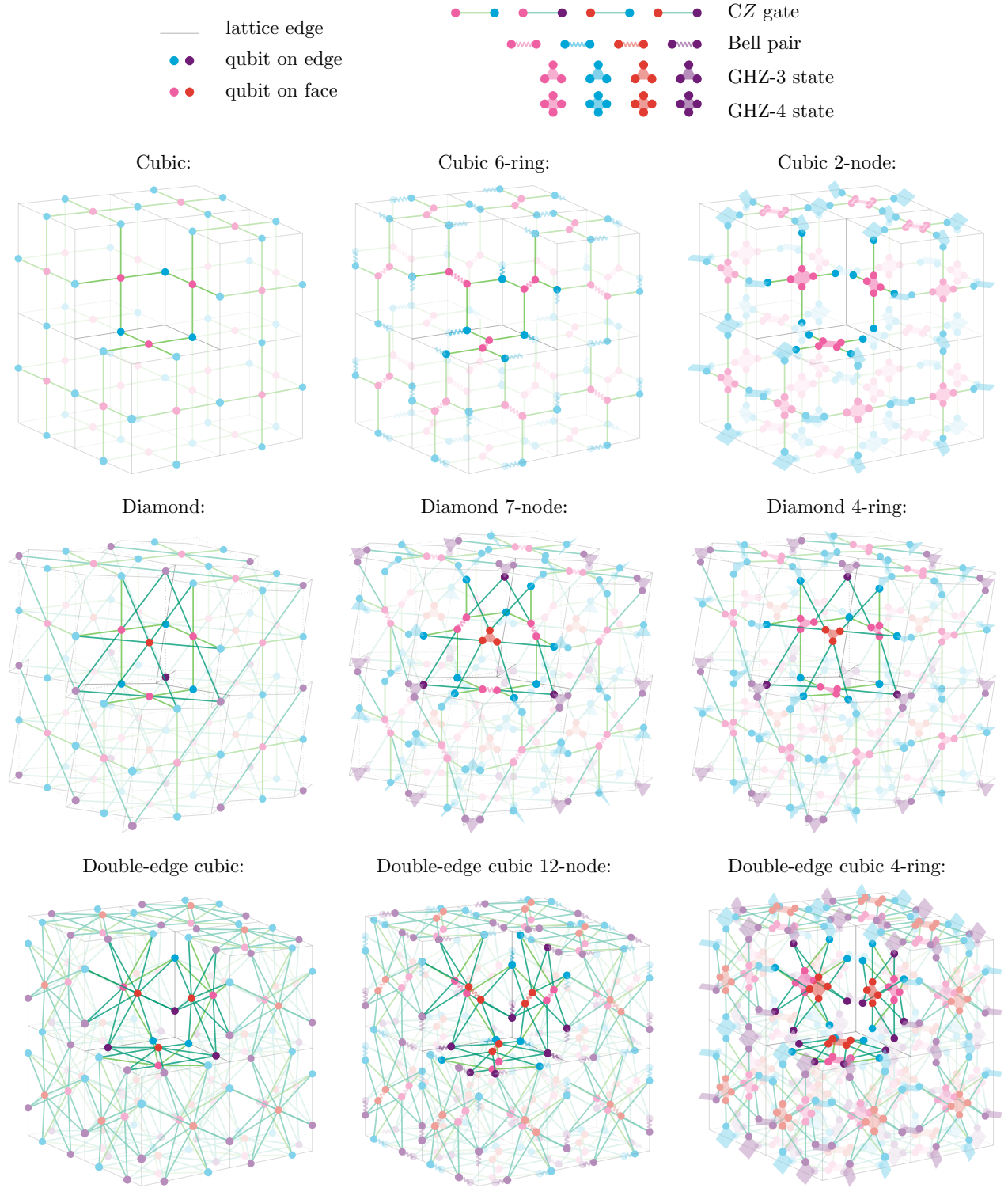


FIG. 11. Distributed architectures for various lattices, obtained through splitting the faces of the monolithic cluster state. A single-face split leads to a Bell state, whereas an n -split produces a GHZ state. They form connected components that are distinct nodes in a distributed network. We identify six different architectures: two each for the cubic, diamond, and double-edge cubic lattices. The left column shows purely monolithic cluster states. In the middle column, cluster states are initialized with Bell states on every face and edge, except for the diamond lattice that also contains weight-3 GHZ states. Architectures in the right column are initialized with 4-partite GHZ states for the cubic and double-edge cubic lattices, and 3-partite states for the diamond lattice. Architectures in the middle column tend to have larger nodes than those in the right column.

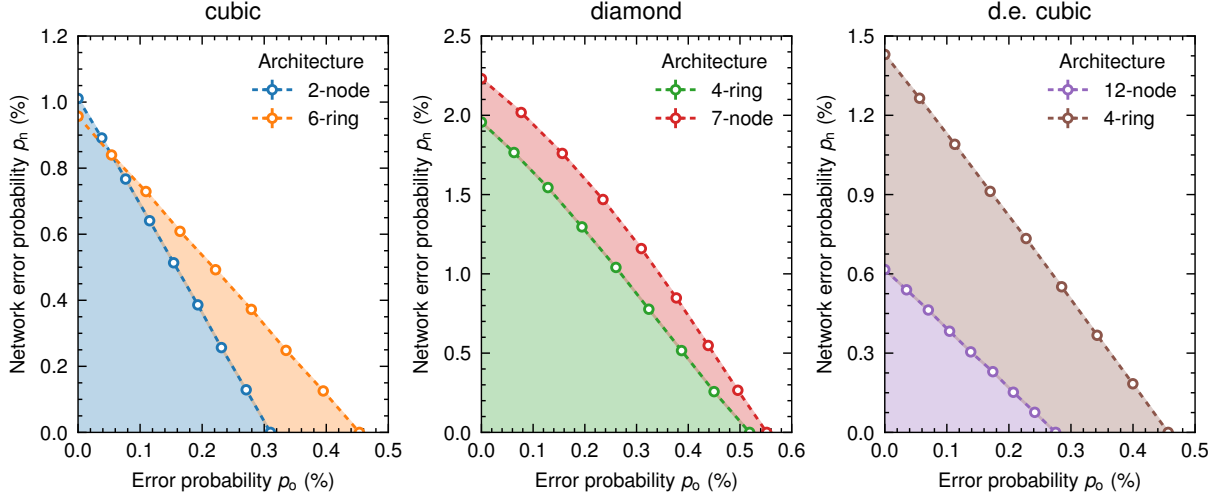


FIG. 12. Fusion-based thresholds for the six distributed architectures of the cubic, diamond, and double-edge cubic lattices. The circuit-level noise parameter p_o and network error rate p_n are both swept, and a fault-tolerant region is estimated over thresholds of both parameters. See main text for simulation details.

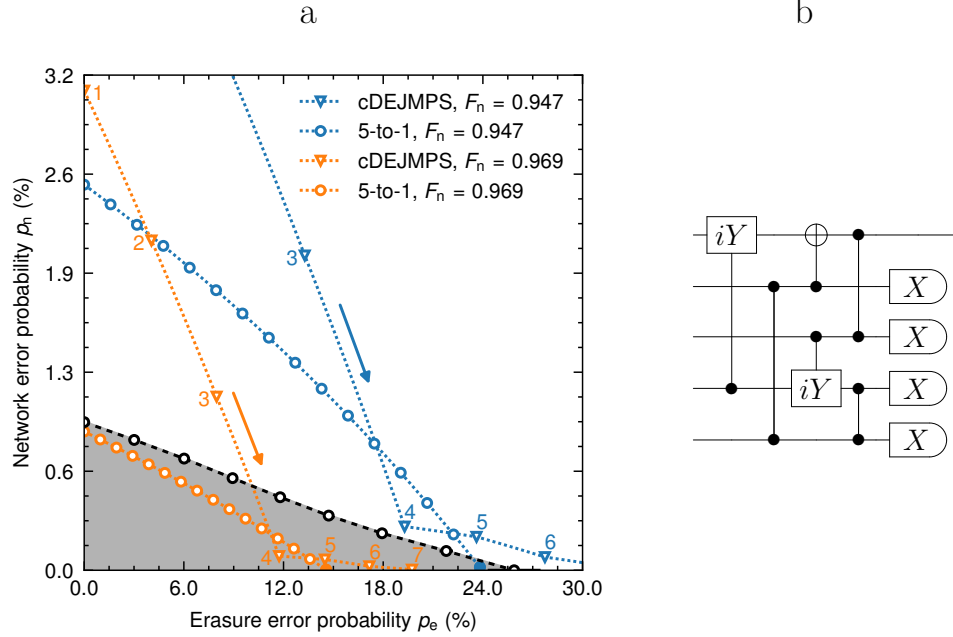


FIG. 13. **(a)** Trade-off between erasure and network error rate. The fault-tolerant region of the six-ring architecture is drawn (gray), where entangling links have a network error rate p_n and an independent probability p_e to fail. Gates and measurements are assumed to be noiseless. A failed link causes the edge to be erased—in the absence of network error, we obtain the phenomenological erasure threshold of the cubic lattice. The colored lines reflect infidelities and failure rates of entanglement distillation protocols that use identical copies of a Werner state with initial fidelity F_n to distill a single Bell state with higher fidelity—*i.e.*, with lower network error rate. Distillation may bring a non-fault-tolerant architecture into the fault-tolerant regime. Data marked with triangles correspond to concatenated DEJMPS (“cDEJMPS”) distillation protocols [26], where the numbers indicate how many Bell states are used to distill the final state. Data marked with colored circles is based on a distillation protocol (printed in panel b) that consumes five Bell states. For this protocol, the data point on the bottom right of each curve corresponds to a variant where, per Bell state, only coinciding measurement outcomes are accepted. The data point on the top left of each curve corresponds to a variant where one accepts all measurement outcomes. Intermediate protocols accept a subset of non-coinciding measurement results. **(b)** Possible circuit for the 5-to-1 distillation protocol. The circuit is based on the decoding circuit of the 5-qubit error-correction code—see Ref. [27] for more details. Each line corresponds to half of a Bell state—*i.e.*, the same circuit has to be applied to the qubits that form the other half of these Bell states.

Union-Find decoder, to deal with these qubit erasures. A suitable decoder can jointly correct errors and erasures. Phenomenological erasure thresholds can be roughly one order of magnitude higher than bit-flip error thresholds, so one may reasonably expect that trading in failed distillation attempts for higher-quality links will be worth the cost.

We make this argument more quantitative in Fig. 13, using the cubic 6-ring architecture as an example. In this architecture, every Bell pair is successfully heralded with probability $1 - p_e$ and network error rate p_n , and discarded with probability p_e . The fault-tolerant region is simulated in the same way as before, this time sweeping over the network and erasure error rates, where circuit-level noise is not taken into account—*i.e.*, p_o is set to $p_o = 0$. We can apply an entanglement distillation protocol to distill each Bell state used in the cluster state by consuming multiple Bell states with initial fidelity $F_n = 1 - p_n$. Distillation protocols that we consider are (concatenated versions of) the DEJMPS protocol [26] and the 5-to-1 protocol [27] of Fig. 13b. The output fidelities and failure rates of these distillation protocols are shown in the graph for two different values of the initial fidelity F_n . For these protocols, there is a trade-off between fidelity and failure rate. Importantly, we can directly link this trade-off to the trade-off between erasure and network error rate—this makes it possible to move the state inside the fault-tolerant region. For the distillation protocols considered, the first crossing with this region happens for an initial network error probability of $p_n \approx 5\%$, which is a five-fold increase of the approximate 6-ring threshold value of 1% without distillation. These results are particularly promising for the diamond architecture, where both the network error threshold of roughly 2% and the erasure threshold of 40% are higher than the cubic lattice.

V. CONCLUSION

In this paper, we have provided several tools and numerical analyses to explore fault-tolerant measurement-based quantum computing architectures built from smaller units, in the context of modular, distributed, or networked computing. This was achieved with a method that allows us to distribute a fault-tolerant cluster state over multiple parties. This method augments the splitting procedure of previous work by Nickerson and Bombin [6]. In the distributed context, the resulting states will lead to entanglement in the form of a Bell or GHZ state, such that existing methods for state generation and distillation may be used to design such a fault-tolerant architecture.

The performance of various three-dimensional cluster state architectures was studied through numerical evaluation of their fault-tolerant thresholds, which quantify the rate of specific sources of error below which fault-tolerant computation is possible. We find that the di-

amond lattice outperforms the traditional cubic cluster state for monolithic architectures suffering standard noise models—*i.e.*, the noise models described in Sec. III D. It should be mentioned that even better results may be achieved with different permutations of the entangling CZ gates during cluster state preparation. In the cases we have considered, (counter)clockwise orderings of CZ gates tend to produce lower thresholds.

Furthermore, we have gauged the performance of the same lattices in a distributed setting. For designs based on the cubic lattice, error thresholds of the network noise are around 1%. We consider two different designs of distributed cluster states defined on top of a diamond lattice, which outperform cubic thresholds roughly by a factor of two. Using a cubic architecture, we show how entanglement distillation may bring a non-fault-tolerant design into the fault-tolerant regime by trading in network noise for erasure errors in the cluster state. Combined with favorable erasure thresholds of the diamond lattice, these results indicate that distributed fault-tolerant cluster states may outperform topological error-correction codes, and warrant additional numerical simulations of these distributed networks in the presence of entanglement distillation.

There are several potential avenues for further investigation. On the one hand, our circuit-based qubit error models are limited to depolarizing and erasure-type noise, but one may estimate fault-tolerant thresholds for models that more accurately represent errors in present-day quantum hardware. Furthermore, it would be interesting to consider protocols that cannot be described as a simple sequence of instructions but contain dependencies and branches that split based on intermediate decisions—as, *e.g.*, protocols that contain entanglement distillation. Previous results in the field apply pre-calculated error models (*i.e.*, quantum channels) on a unit cell level, and randomly sample from the error model during Monte Carlo threshold calculations [25, 28]. This is possible because the fault-tolerant protocol consists of identical rounds that are applied over time, where each round is split up and grouped into multiple sub-rounds that act on disjoint subsets of qubits, so that the entire protocol may be simulated as distinct sections that are separated in both space and time. This is not necessarily the case for the three-dimensional cluster states that we consider here, in which case the entire cluster state should be simulated at a global level, *i.e.* without pre-calculating error models of individual sections [29].

As an alternative to measurement-based quantum computation, one may consider so-called fusion-based quantum computation [9]. It combines a low circuit depth with the topological features of cluster states. The fundamental operations in fusion-based quantum computation are resource-state generation and fusions. Fault-tolerant fusion-based architectures rely heavily on resilience against erasure, and the structures considered in this work are a natural candidate to consider in this alternative framework of computation.

ACKNOWLEDGMENTS

The authors would like to thank Michael Newman, Naomi Nickerson, Tim Taminiau, and Barbara Terhal for helpful feedback and/or discussions. We gratefully acknowledge support from the joint research program “Modular quantum computers” by Fujitsu Limited and Delft University of Technology, co-funded by the Netherlands Enterprise Agency under project number PPS2007. This work was supported by the Netherlands Organization for Scientific Research (NWO/OCW), as part of the Quantum Software Consortium Program under Project 024.003.037/3368. This work was partially supported by the JST Moonshot R&D program under Grant JP-MJMS226C. We thank SURF (www.surf.nl) for the support in using the National Supercomputer Snellius.

Appendix A: Quotient boundaries in unit cell complex

By the prescription of Sec. III A, the unit cell complex is described as a sequence of vector spaces

$$Q_3 \xrightarrow{\partial_3^{[r]}} Q_2 \xrightarrow{\partial_2^{[r]}} Q_1 \xrightarrow{\partial_1^{[r]}} Q_0,$$

with each Q_i over the field \mathbb{Z}_2 and with quotient boundaries $\partial_i^{[r]} : Q_i \mapsto Q_{i-1}$. Similar to the equivalence between \bar{C}_i and C_{D-i} , we define $\bar{Q}_i \cong Q_{D-i}$. One may verify that the dual quotient boundaries $\bar{\partial}_i^{[r]} : \bar{Q}_i \mapsto \bar{Q}_{i-1}$ are related to primal boundaries as

$$\bar{\partial}_i^{[r]} = \left(\partial_{D+1-i}^{[r]} \right)^T. \quad (\text{A1})$$

Importantly, the translation vector \mathbf{r} is also reversed to $\bar{\mathbf{r}}$. By dualizing boundary maps of unit cell complex directly, one may construct a representation of the dual unit cell without redefining it from the dual crystal. The zero map conditions $\partial_{i-1}\partial_i = 0$ take the form of

$$\sum_p \partial_{i-1}^{[p]} \partial_i^{[r-p]} = 0 \quad \forall \mathbf{r}. \quad (\text{A2})$$

The proof is given in App. B.

It is convenient to represent the underlying unit cell complex as a *labeled* graph, which is essentially a sparse representation of its boundaries as *arcs* and basis elements as *nodes*. (We use nomenclature *nodes* and *arcs* for such a graph, to make the distinction between vertices and edges of the chain complex.) Every basis element $(\mathbf{q}_i)_n \in Q_i$ is mapped to a node $q_{i,n}$, with two nodes $q_{i,n} \rightarrow_{[r]} q_{i-1,m}$ connected by an $[r]$ -labelled arc if the m th matrix element of the quotient boundary $\partial_i^{[r]}$ equals one. The maps $\partial_i^{[r]}$ thus form the *biadjacency* matrices between the nodes of Q_i and Q_{i-1} . We note that this description, including its labeling, resembles the *vector method* for describing three-periodic networks as a

quotient graph [30], except that the nodes of our quotient graph also represent higher-dimensional elements in a chain complex, such as edges, faces, and cells for a three-dimensional complex. Examples of the square, triangular, and cubic lattice are given in Fig. 14.

Appendix B: Crystal embeddings

In Fig. 15, we show an intuitive interpretation of how the vector spaces C_i of the full crystalline chain complex are constructed with the vector spaces Q_i of the unit cell complex of App. A. To construct the boundary maps ∂_i of the full crystal from the quotient boundary maps $\partial_i^{[r]}$, we let $\partial_i^{(\mathbf{n},\mathbf{m})} : C_i \mapsto C_{i-1}$ be the boundary map that applies $\partial_i^{[\mathbf{n}-\mathbf{m}]}$ from cell $Q_i^{(\mathbf{m})}$ to cell $Q_{i-1}^{(\mathbf{n})}$ and is zero everywhere else:

$$\partial_i^{(\mathbf{n},\mathbf{m})} = \partial_i^{[\mathbf{n}-\mathbf{m}]} \otimes e_{nm}. \quad (\text{B1})$$

Here, $e_{nm} : L \mapsto L$ is a *matrix unit*, i.e., an $N \times N$ matrix with a one at indices n, m and zero elsewhere. Then the boundary maps of the embedding are given as a sum

$$\partial_i = \sum_{m,n} \partial_i^{(\mathbf{n},\mathbf{m})} = \sum_{m,n} \left(\partial_i^{[\mathbf{n}-\mathbf{m}]} \otimes e_{nm} \right). \quad (\text{B2})$$

Because most $\partial_i^{[\mathbf{n}-\mathbf{m}]}$ are zero, we can substitute $\mathbf{r} = \mathbf{n} - \mathbf{m}$ and sum \mathbf{r} only over the non-zero maps $\partial_i^{[\mathbf{r}]}$, leading to

$$\partial_i = \sum_{m,r} \left(\partial_i^{[\mathbf{r}]} \otimes e_{m+\mathbf{r},m} \right) = \sum_r \partial_i^{[\mathbf{r}]} \otimes \left(\sum_m e_{m+\mathbf{r},m} \right). \quad (\text{B3})$$

In the second equality, the sum over \mathbf{m} may be carried over to the right by distributivity of the tensor product over addition. This last term is a permutation matrix with a single one in each row and column; it represents a *translation* of a lattice point \mathbf{m} to $\mathbf{m} + \mathbf{r}$. Denote this term as $T_{\mathbf{r}} \equiv \sum_m e_{m+\mathbf{r},m}$, such that the embedding is given as

$$\partial_i = \sum_{\mathbf{r}} \partial_i^{[\mathbf{r}]} \otimes T_{\mathbf{r}}. \quad (\text{B4})$$

Intuitively, the crystal boundary is formed by “gluing” the boundaries between the unit cells $Q_i^{(\mathbf{m})}$ and $Q_{i-1}^{(\mathbf{m}+\mathbf{r})}$ at lattice points \mathbf{m} and $\mathbf{m} + \mathbf{r}$ according to the map $\partial_i^{[\mathbf{r}]}$, and repeating this process for every non-trivial quotient boundary map. The zero map conditions for quotient boundaries (Eq. (A2)) follow trivially. Because matrix units multiply as $e_{ij}e_{kl} = \delta_{jk}e_{il}$, the multiplication of two permutation matrices

$$\begin{aligned} T_{\mathbf{p}}T_{\mathbf{q}} &= \sum_{m,n} e_{m+\mathbf{p},m}e_{n+\mathbf{q},n} = \sum_{m,n} \delta_{m,n+\mathbf{q}}e_{m+\mathbf{p},n} \\ &= \sum_n e_{n+\mathbf{p}+\mathbf{q},n} = T_{\mathbf{p}+\mathbf{q}} \end{aligned} \quad (\text{B5})$$

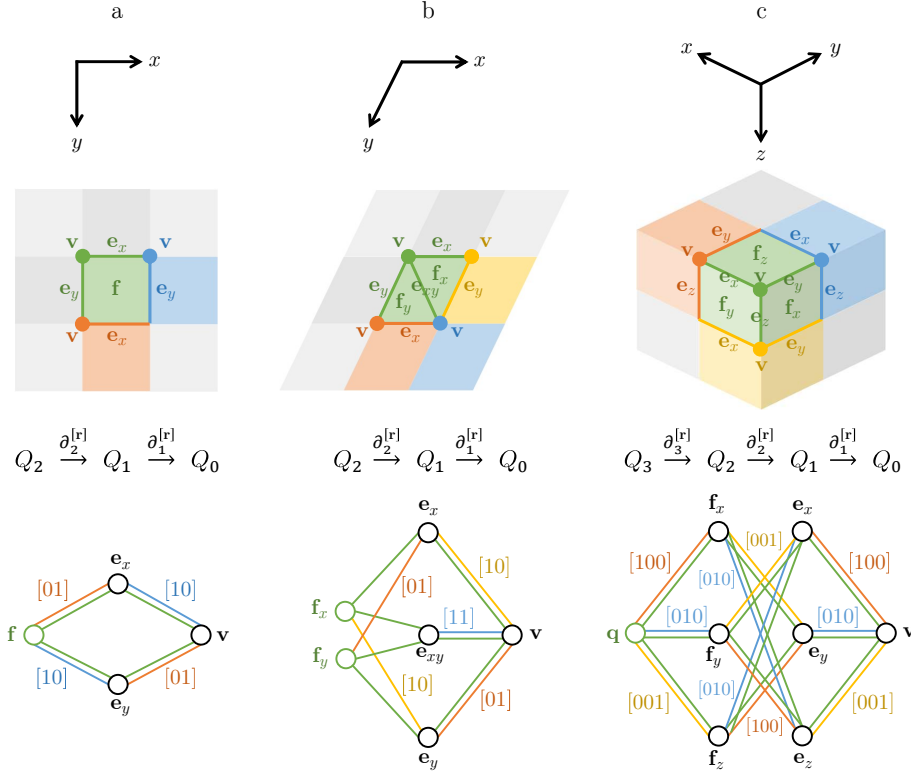


FIG. 14. Unit cell complexes as a labeled graph. Unlabelled edges correspond to a Miller index containing only zeros. **(a)** A square lattice. There is one face, two edges, and one vertex. The face \mathbf{f} is connected twice to both \mathbf{e}_x and \mathbf{e}_y within the unit cell ($[00]$) and outside of it ($[01]$ and $[10]$, respectively). Because the complex is self-dual, similar relationships hold for its vertex \mathbf{v} . **(b)** A triangular lattice. This lattice can be created by splitting the faces of the square lattice: per unit cell, the triangular lattice has one extra face and one extra edge compared to the square lattice. From the asymmetry in the quotient boundary maps it is clear that this lattice is not self-dual. **(c)** A cubic lattice. Per unit cell, there is one cell \mathbf{q} ; three faces \mathbf{f}_x , \mathbf{f}_y , and \mathbf{f}_z ; three edges \mathbf{e}_x , \mathbf{e}_y , and \mathbf{e}_z ; and one vertex \mathbf{v} .

represents the sum of their translations. Combining this result with the embedded boundaries (Eq. (B4)) directly, the composition of two maps equals

$$\partial_{i-1}\partial_i = \sum_{p,q} \partial_{i-1}^{[p]} \partial_i^{[q]} \otimes T_{\mathbf{p}+\mathbf{q}} = \sum_{\mathbf{r}} \left(\sum_p \partial_{i-1}^{[p]} \partial_i^{[\mathbf{r}-\mathbf{p}]} \right) \otimes T_{\mathbf{r}}, \quad (\text{B6})$$

where we have substituted $\mathbf{r} = \mathbf{p} + \mathbf{q}$ in the last equality. This map is the zero map if and only if the term in brackets is zero for all \mathbf{r} , which is exactly the result stated in Eq. (A2).

The above recipe for a crystal embedding may be expressed as a composition of *direct products* between two graphs, given the following correspondences:

1. The matrix $\partial_i^{[r]}$ is the biadjacency matrix of the subgraph $G[\partial_i^{[r]}]$ of the unit cell complex induced by edges with label $[r]$. This definition is consistent with the graph description given in Sec. A.
2. The matrix $T_{\mathbf{r}}$ is the biadjacency matrix of the subgraph of the lattice $H[T_{\mathbf{r}}]$ induced by edges that translate each lattice point by \mathbf{r} . That is, each lat-

tice point is represented by a node \mathbf{m} and connected by an arc to the translated node $\mathbf{m} + \mathbf{r}$.

The tensor products $\partial_i^{[r]} \otimes T_{\mathbf{r}}$ inside the embedding (Eq. (B4)) are direct products of the corresponding edge-induced subgraphs $G[\partial_i^{[r]}] \times H[T_{\mathbf{r}}]$. The sum over labels \mathbf{r} , which adds together adjacency matrices of the products modulo 2, composes the edge sets of the corresponding graphs as a disjunctive union. In this way, the entire crystal complex may be constructed directly as a graph from a given unit cell and a lattice of arbitrary size.

Appendix C: Characterization of noisy channels

The general process in our simulations can be described as an n -qubit quantum circuit C composed of Clifford operations, ending in a projective Pauli basis measurement $P^{\mathbf{m}}$ with outcomes $\mathbf{m} = \{m_1, m_2, \dots, m_l\}$ of (part) of the evolved state. For the sake of completeness, we also assume that the circuit operates on an ancillary input system A with a stabilizer state $|\psi_0\rangle$. Such a circuit might represent an entanglement distillation circuit, operating

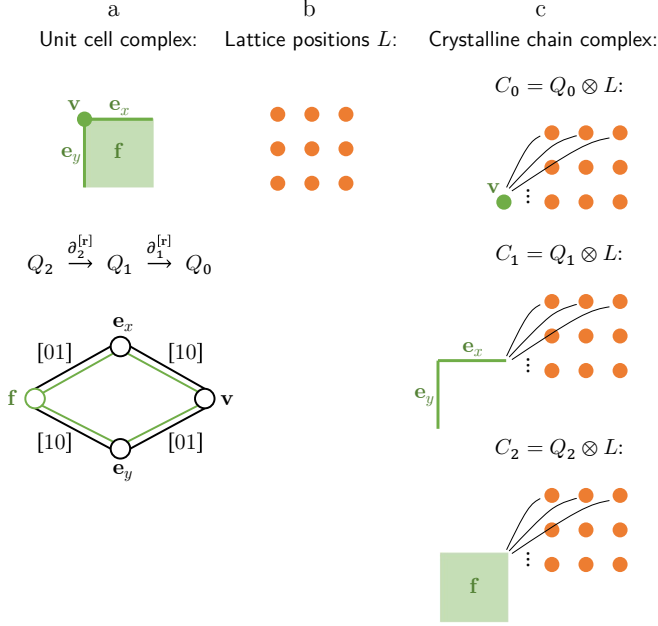


FIG. 15. Abstract interpretation of how the full crystalline chain complex is created out of the unit cell complex introduced in App. A. **(a)** We use the example of the unit cell of the square lattice—see Fig. 14 for more details. **(b)** As described in Sec. III A of the main text, we use an N -dimensional vector space $L \equiv \mathbb{Z}_2^{\oplus N}$ with as basis vectors the N lattice positions of the lattice. **(c)** The vector spaces C_i of the full crystalline chain complex are realized with the *graph product* between L and the vector spaces Q_i of the unit cell complex. The full boundary maps ∂_i are constructed from the quotient boundary maps $\partial_i^{[r]}$ according to Eq. (B3).

on a mixed Bell pair ρ and an ancillary Bell pair to distill it with. Alternatively, it may represent the action of a measurement-based fault-tolerant channel, where ρ is the input code space, $|\psi_0\rangle$ is the state of all ancillary qubits in the channel, C represents the CZ gates of the cluster state and $P^{\mathbf{m}}$ is the final X -basis measurement of every ancillary qubit. The output code space is then (up to normalization) given by $\mathcal{E}_{\mathbf{m}}(\rho)$.

We consider three sources of noise, depicted schematically in Fig. 16. First of all, noisy ancillary input may not be a pure stabilizer state $|\psi_0\rangle$, but a mixture ρ_A of possibly non-stabilizer states. Secondly, the circuit C consists of imperfect operations, which we assume as ideal operations followed by a mixture of Pauli gates. Lastly, the projectors $P^{\mathbf{m}}$ may produce a “wrong” outcome $\tilde{\mathbf{m}}$, which we model as a perfect operation $P^{\mathbf{m}}$ followed by classical bit-flips on \mathbf{m} .

Arbitrary noisy ancillary input states ρ_A that differ from the noiseless input $|\psi_0\rangle$ cannot be simulated efficiently. If, on the other hand, ρ_A may be approximated as a Pauli channel \mathcal{P} acting on $|\psi_0\rangle$, the Pauli operators may be pushed through the circuit in the same way as Pauli noise coming from imperfect gates. The key idea is to pre-process ρ_A by *twirling* [18] with the stabilizers

$s_k \in S_0$, where S_0 is the stabilizer group describing the state $|\psi_0\rangle$, i.e., to apply a trace-preserving channel \mathcal{T} as

$$\mathcal{T}(\rho_A) = \frac{1}{|S_0|} \sum_{s_k \in S_0} s_k \rho_A s_k. \quad (\text{C1})$$

In the context of Pauli twirling the input state, \mathcal{T} corresponds to a Pauli channel \mathcal{P} acting on the noiseless input state $|\psi_0\rangle$, with elements of a *destabilizer* group D_0 of S_0 acting as the Pauli operators of the channel. A destabilizer group D_0 associated with a stabilizer group S_0 is a subgroup of the full Pauli group. It has the same size as S_0 and is generated with a set of operators that each anti-commute with a different generator of S_0 and commute with all other generators of S_0 [31]. A destabilizer group D_0 of S_0 can be used to decompose ρ_A in a basis of states $\{d_k |\psi_0\rangle\}_{d_k \in D_0}$ and write

$$\rho_A = \sum_{d_m, d_p \in D_0} \lambda_{mp} d_m |\psi_0\rangle\langle\psi_0| d_p. \quad (\text{C2})$$

We can now use this to write $\mathcal{T}(\rho_A)$ as

$$\mathcal{T}(\rho_A) = \sum_{d_k \in D_0} p_k d_k |\psi_0\rangle\langle\psi_0| d_k = \mathcal{P}(|\psi_0\rangle\langle\psi_0|). \quad (\text{C3})$$

Here, the prefactors p_k are given by $p_k = \lambda_{kk} = \langle\psi_0| d_k \rho_A d_k |\psi_0\rangle$. The trace $\text{Tr}[\mathcal{T}(\rho_A)] = \sum_{d_k \in D_0} p_k = 1$ is preserved, such that prefactors p_k that sum to unity may be interpreted as the probability of applying some p_k in the Pauli channel \mathcal{P} . We see that twirling ρ_A over the group S_0 removes all off-diagonal elements of the state in the $\{d_k |\psi_0\rangle\}_{d_k \in D_0}$ basis.

This shows how we may approximate noisy ancillary input state ρ_A as a mixture of the ideal state $|\psi_0\rangle$ that is depolarized by a Pauli channel \mathcal{P} . In the same way, we may twirl non-Pauli noisy processes occurring during the application of the Clifford circuit C to a Pauli form. Operators P_i can now be propagated through the circuit C , forming another set of Pauli strings $P_j = C P_i C^\dagger$. Pauli operators P_j now appearing after C may each be split up into a string $P_j^{(B)} \equiv P'_j$ appearing on the ancillary B

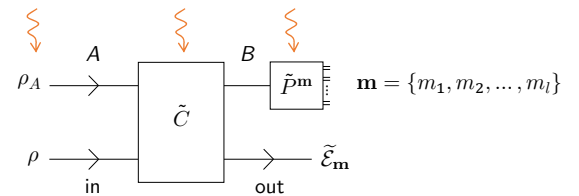


FIG. 16. A noisy channel that can be characterized efficiently, given that ρ_A is a convex combination of stabilizer states, \tilde{C} is the ideal circuit C with Pauli noise, and $\tilde{P}^{\mathbf{m}}$ are ideal projectors followed by classical bit-flips of the outcome \mathbf{m} . Under suitable assumptions of the noise models, the $\tilde{\mathcal{E}}_{\mathbf{m}}$ operation may be expressed as a mixture of ideal $\mathcal{E}_{\mathbf{m}}$ and Pauli operations.

system and a string $P_j^{(\text{out})} \equiv P_j''$ appearing on the output system as $P_j \equiv P_j' \otimes P_j''$. The string P_j' will commute with some projectors $P_j^{m_k} P_j' = P_j' P_j^{m_k}$ (where $m_k \in \mathbf{m}$), but anticommute with others as $P_j^{m_{k'}} P_j' = P_j' P_j^{(m_{k'}+1)}$. The addition of two classical bits $a + b$ here is understood modulo 2, *i.e.*, $a + 1$ represents a bit-flip of a . We summarize both cases as a commutation relation $P^{\mathbf{m}} P_j' = P_j' P^{\mathbf{m}+\mathbf{m}_j}$, where \mathbf{m}_j is a string of errors that represents the bit-flips due to the individual commutation relations above.

One can now derive an expression for the noisy operation $\tilde{\mathcal{E}}_{\mathbf{m}}$ as a mixture of ideal operations with the addition of probabilistic Pauli strings:

$$\tilde{\mathcal{E}}_{\mathbf{m}}(\rho) = \sum_j p_j P_j'' \mathcal{E}_{\mathbf{m}+\mathbf{m}_j}(\rho) P_j'. \quad (\text{C4})$$

The noisy channel $\tilde{\mathcal{E}}_{\mathbf{m}}$ is a mixture of ideal operations $\mathcal{E}_{\mathbf{m}+\mathbf{m}_j}$ and Pauli noise, with the mixture arising due to classical bit-flips \mathbf{m}_j that act on \mathbf{m} . Because we assumed each operation $\mathcal{E}_{\mathbf{m}}$ to be efficiently simulatable, the mixture may also be simulated efficiently.

On top of this, to account for faulty measurements, we assume that each projector P^{m_k} has a fixed probability p_m of reporting the wrong outcome $\tilde{m}_k \equiv m_k + 1$ (modulo 2), such that the noisy measurement channel \tilde{P}^{m_k} is given by a mixture

$$\tilde{P}^{m_k}(\rho) = (1 - p_m) P^{m_k} \rho P^{m_k} + p_m P^{m_k+1} \rho P^{m_k+1}. \quad (\text{C5})$$

For all measurement outcomes $\mathbf{m} = \{m_1, m_2, \dots, m_l\}$ this corresponds to the channel

$$\begin{aligned} \tilde{P}^{\mathbf{m}}(\rho) &= \sum_f p_f P^{\mathbf{m}+\mathbf{m}_f} \rho P^{\mathbf{m}+\mathbf{m}_f}, \\ p_f &\equiv (p_m)^{h(\mathbf{m}_f)} (1 - p_m)^{l-h(\mathbf{m}_f)}, \end{aligned} \quad (\text{C6})$$

where $h(\mathbf{m}_f)$ is the *Hamming weight* of the binary string \mathbf{m}_f with measurement errors. The additional mixing of measurement outcomes does not change the form of the noisy channel as in Eq. (C4) but introduces additional terms $\mathcal{E}_{\mathbf{m}+\mathbf{m}_j+\mathbf{m}_f}$ with prefactors that reflect the probability of applying a particular configuration of measurement errors:

$$\tilde{\mathcal{E}}_{\mathbf{m}}(\rho) = \sum_{j,f} p_j p_f P_j'' \mathcal{E}_{\mathbf{m}+\mathbf{m}_j+\mathbf{m}_f}(\rho) P_j'. \quad (\text{C7})$$

Appendix D: Statistical error in threshold values

To fit the thresholds, we assume a second-order polynomial model of the logical error probability p_L around

the threshold crossing p_{th} of the form [32]

$$p_L(p, L) = p_{L,\text{th}} + c_1 (p - p_{\text{th}}) L^{1/\nu} + c_2 (p - p_{\text{th}})^2 L^{2/\nu}, \quad (\text{D1})$$

where L corresponds to the lattice size used with the error probability p , and $p_{L,\text{th}}$, p_{th} , c_1 , c_2 , and ν are fitting parameters. The fit is obtained through non-linear least-squares minimization of the residuals $R_i = (p_L(p, L) - \hat{p}_i) / \hat{\sigma}_i$ for every data point i with the observed logical error rate \hat{p}_i and the associated standard deviation $\hat{\sigma}_i$. Confidence intervals for threshold crossings p_{th} are taken from the standard errors of the least-squares approximation.

In the results of Sec. IV, error bars of the logical error rate are given as 95% confidence intervals. Sometimes, they are too small to be discernible. The threshold value is highlighted with a 95% confidence interval based on its least-squares estimate of a second-order polynomial of the logical error rate around the threshold value.

Appendix E: Stabilizer fidelities for distributed setting

In this appendix, we gauge the fidelity of the stabilizer operator supported by each face, for several face splittings in the distributed setting. For these results, we model the entangled states as Bell and GHZ states that are depolarized to a diagonal form. For these states, the primary component $S_0 \equiv \langle X_0 \dots X_{n-1}, Z_0 Z_1, \dots, Z_0 Z_{n-1} \rangle$ has probability $1 - p'_n$ and all off-diagonal terms $\langle \pm X_0 \dots X_{n-1}, \pm Z_0 Z_1, \dots, \pm Z_0 Z_{n-1} \rangle$ with at least one negative sign are uniformly distributed with probabilities $p'_n / (2^n - 1)$. With $n = 2$, this state corresponds to the Werner state of Eq. (10). Furthermore, just as in Sec. IV D, we assume that state preparation and measurements invert the state with probability $p_m \equiv p_o$, and that CZ gates are followed by a depolarizing channel with probability $p_g \equiv p_o$.

For each of the geometries, the stabilizer fidelity of one of the faces is calculated under this circuit-based noise model. Fidelities are calculated as the overlap of the resulting mixed state with the ideal cluster state stabilizer. A strong simulation for each of the circuits provides an exact closed-form expression of the fidelity. We show these results for each of the lattices and the three different protocols (monolithic, Bell, and GHZ) corresponding to the three columns of Fig. 17. Based on these results, we see that an increase in the number of splittings produces worse stabilizer fidelities. These differences are more pronounced for the 4-partite GHZ states when compared to the 3-partite GHZ state in diamond. Distributed architectures are likely unable to compete with monolithic protocols, unless we combine entanglement generation protocols with better-quality GHZ states through entanglement distillation.

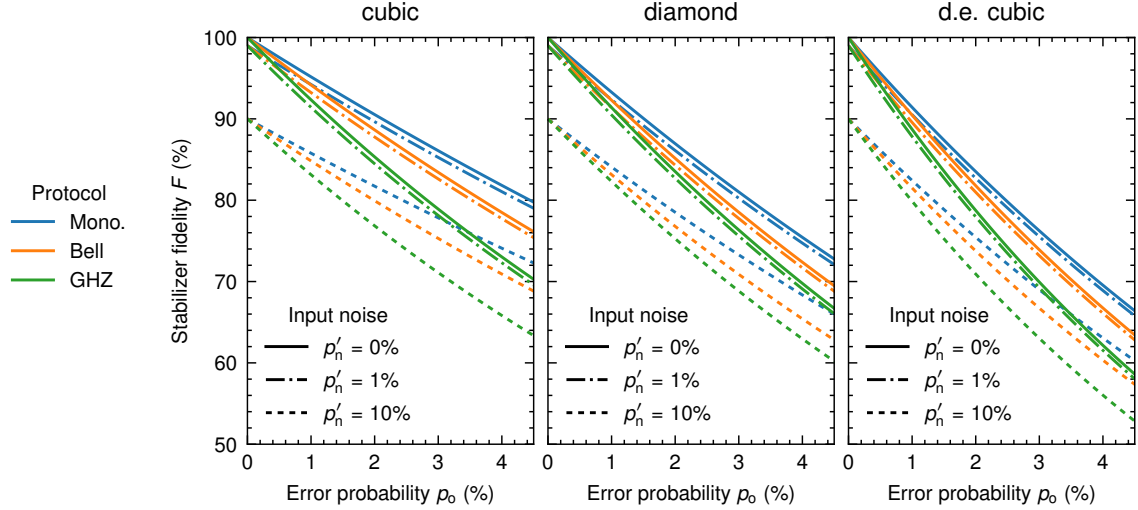


FIG. 17. Stabilizer fidelities of monolithic and distributed protocols for various architectures. The three protocols per architecture correspond to the three columns of Fig. 11, where the GHZ state is defined as weight-4 for cubic and double-edge cubic architectures and weight-3 for the diamond lattice, and a face with a Bell pair split is chosen for the diamond 7-node lattice. The noise probability p_o describes both depolarizing gate noise with probability p_g and faulty measurements with probability p_m . The quality of the network link is parameterized by p'_n (see main text for details). Since we are only looking at the errors arising on a single face, the ordering of CZ gates is not relevant for the stabilizer fidelity of that same face.

-
- [1] K. Fujii, *Quantum Computation with Topological Codes: From Qubit to Topological Fault-Tolerance*, Springer-Briefs in Mathematical Physics, Vol. 8 (Springer, Singapore, 2015).
- [2] A. Bolt, G. Duclos-Cianci, D. Poulin, and T. M. Stace, Foliated quantum error-correcting codes, *Physical Review Letters* **117**, 070501 (2016).
- [3] R. Raussendorf, J. Harrington, and K. Goyal, A fault-tolerant one-way quantum computer, *Annals of Physics* **321**, 2242 (2006).
- [4] S. D. Barrett and T. M. Stace, Fault tolerant quantum computation with very high threshold for loss errors, *Physical Review Letters* **105**, 200502 (2010).
- [5] H. Bombin, I. H. Kim, D. Litinski, N. Nickerson, M. Pant, F. Pastawski, S. Roberts, and T. Rudolph, *Interleaving: Modular Architectures for Fault-Tolerant Photonic Quantum Computing*, arXiv:2103.08612 [quant-ph].
- [6] N. Nickerson and H. Bombin, *Measurement Based Fault Tolerance beyond Foliation*, arXiv:1810.09621 [quant-ph].
- [7] J. M. Auger, H. Anwar, M. Gimeno-Segovia, T. M. Stace, and D. E. Browne, Fault-tolerant quantum computation with nondeterministic entangling gates, *Physical Review A* **97**, 030301 (2018).
- [8] D. Herr, A. Paler, S. J. Devitt, and F. Nori, A local and scalable lattice renormalization method for ballistic quantum computation, *npj Quantum Information* **4**, 1 (2018).
- [9] S. Bartolucci, P. Birchall, H. Bombin, H. Cable, C. Dawson, M. Gimeno-Segovia, E. Johnston, K. Kielsing, N. Nickerson, M. Pant, F. Pastawski, T. Rudolph, and C. Sparrow, Fusion-based quantum computation, *Nature Communications* **14**, 912 (2023).
- [10] D. D. Awschalom, R. Hanson, J. Wrachtrup, and B. B. Zhou, Quantum technologies with optically interfaced solid-state spins, *Nature Photonics* **12**, 516 (2018).
- [11] N. Delfosse and N. H. Nickerson, Almost-linear time decoding algorithm for topological codes, *Quantum* **5**, 595 (2021).
- [12] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2000).
- [13] H. J. Briegel and R. Raussendorf, Persistent entanglement in arrays of interacting particles, *Physical Review Letters* **86**, 910 (2001).
- [14] M. Newman, L. A. de Castro, and K. R. Brown, Generating fault-tolerant cluster states from crystal structures, *Quantum* **4**, 295 (2020).
- [15] D. E. Browne and T. Rudolph, Resource-efficient linear optical quantum computation, *Physical Review Letters* **95**, 010501 (2005).
- [16] W. P. Grice, Arbitrarily complete Bell-state measurement using only linear optical elements, *Physical Review A* **84**, 042331 (2011).
- [17] F. Ewert and P. van Loock, 3/4-efficient Bell measurement with passive linear optics and unentangled ancillae, *Physical Review Letters* **113**, 140403 (2014).
- [18] C. H. Bennett, D. P. DiVincenzo, J. A. Smolin, and W. K. Wootters, Mixed-state entanglement and quantum error correction, *Physical Review A - Atomic, Molecular, and Optical Physics* **54**, 3824 (1996).
- [19] N. Nickerson, *Practical Fault-Tolerant Quantum Computing*, Doctoral thesis, Imperial College London, London, United Kingdom (2015).
- [20] T. Ohno, G. Arakawa, I. Ichinose, and T. Matsui, Phase structure of the random-plaquette Z2 gauge model: Accuracy threshold for a toric quantum memory, *Nuclear*

- Physics B **697**, 462 (2004).
- [21] N. Delfosse and G. Zémor, Linear-time maximum likelihood decoding of surface codes over the quantum erasure channel, *Physical Review Research* **2**, 033042 (2020).
 - [22] Y. van Montfort, S. de Bone, and D. Elkouss, Data/software underlying the publication: Fault-tolerant structures for measurement-based quantum computation on a network, 4TU.ResearchData, <https://doi.org/10.4121/929e24f9-31fa-4816-99fa-3356e272df43> (2024).
 - [23] H. Bombín, C. Dawson, R. V. Mishmash, N. Nickerson, F. Pastawski, and S. Roberts, Logical blocks for fault-tolerant topological quantum computation, *PRX Quantum* **4**, 020303 (2023).
 - [24] D. König, Graphok es matrixok, *Matematikai és Fizikai Lapok* **38**, 116 (1931).
 - [25] N. H. Nickerson, Y. Li, and S. C. Benjamin, Topological quantum computing with a very noisy network and local error rates approaching one percent, *Nature Communications* **4**, 1756 (2013).
 - [26] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera, Quantum privacy amplification and the security of quantum cryptography over noisy channels, *Physical Review Letters* **77**, 2818 (1996).
 - [27] K. Goodenough, S. de Bone, V. L. Addala, S. Krastanov, S. Jansen, D. Gijswijt, and D. Elkouss, Near-Term n to k Distillation Protocols Using Graph Codes, *IEEE Journal on Selected Areas in Communications* **42**, 7, 1830 (2024).
 - [28] N. H. Nickerson, J. F. Fitzsimons, and S. C. Benjamin, Freely scalable quantum technologies using cells of 5-to-50 qubits with very lossy and noisy photonic links, *Physical Review X* **4**, 041041 (2014).
 - [29] C. Gidney, M. Newman, and M. McEwen, Benchmarking the planar honeycomb code, *Quantum* **6**, 813 (2022).
 - [30] S. J. Chung, T. Hahn, and W. E. Klee, Nomenclature and generation of three-periodic nets: The vector method, *Acta Crystallographica Section A: Foundations of Crystallography* **40**, 42 (1984).
 - [31] S. Aaronson and D. Gottesman, Improved simulation of stabilizer circuits, *Physical Review A* **70**, 052328 (2004).
 - [32] C. Wang, J. Harrington, and J. Preskill, Confinement-Higgs transition in a disordered gauge theory and the accuracy threshold for quantum memory, *Annals of Physics* **303**, 31 (2003).