

A graph-based formalism for surface codes and twists

Rahul Sarkar¹ and Theodore J. Yoder²

¹Institute for Computational and Mathematical Engineering, Stanford University, Stanford, CA 94305

²IBM T.J. Watson Research Center, Yorktown Heights, NY

Twist defects in surface codes can be used to encode more logical qubits, improve the code rate, and implement logical gates. In this work we provide a rigorous formalism for constructing surface codes with twists generalizing the well-defined homological formalism introduced by Kitaev for describing CSS surface codes. In particular, we associate a surface code to *any* graph G embedded on *any* 2D-manifold, in such a way that (1) qubits are associated to the vertices of the graph, (2) stabilizers are associated to faces, (3) twist defects are associated to odd-degree vertices. In this way, we are able to reproduce the variety of surface codes, with and without twists, in the literature and produce new examples. We also calculate and bound various code properties such as the rate and distance in terms of topological graph properties such as genus, systole, and face-width.

Contents

1	Introduction	3
2	Preliminaries	4
2.1	Graph embeddings	5
2.2	General rotation systems	5
2.3	Dual graph and checkerboardability	7
2.4	Homology of graph embeddings	9
2.5	Covering maps and contractible loops	10
3	Majorana and qubit surface codes from rotation systems	11
3.1	Majorana operators and codes	11
3.2	Majorana surface codes defined on embedded graphs	12
3.3	From Majorana surface codes to qubit surface codes	15
3.4	Relation to homological surface codes	20
4	Locating logical operators	21
4.1	Checkerboardability with defects	22
4.2	Characterizing logical operators as cycles in the decoding graph	24
4.3	Doubled graphs	27

Rahul Sarkar: rsarkar@stanford.edu,

Theodore J. Yoder: ted.yoder@ibm.com,

4.4	Face-width as a lower-bound on code distance	30
4.5	Logical operators from trails in the original graph	32
5	Code examples	34
5.1	Rotated toric codes	34
5.2	General rotated toric codes	36
5.3	New hyperbolic codes	40
5.4	Two more ways to generalize the triangle code	43
6	Open problems	44
	Appendices	49
A	Graph embeddings from oriented rotation systems	49
B	Chain complexes	51
C	Paulis with arbitrary commutation patterns and CALs	52
C.1	Basic definitions	52
C.2	Constructing a list of Paulis given desired commutation relations	54
C.3	Properties of cyclically anticommuting lists of Paulis	59
D	Majorana surface code lemmas and proofs	67
E	Equivalence of Majorana and qubit surface codes	69
F	A more general family of cyclic qubit stabilizer codes	74
F.1	A four parameter cyclic code family	74
F.2	A two parameter cyclic code family	81
G	Embedding of the medial graph	85
H	Using GAP to find normal subgroups	86

1 Introduction

Since their introduction [1], surface codes have been an essential tool in the quantum engineer's fight against noise, combining a high threshold with relatively simple 2-dimensional qubit connectivity. These codes also exemplify a surprising connection between quantum error-correction and topology. In effect, one associates a CSS quantum code to a graph embedded on a 2-dimensional manifold by assigning qubits to edges and stabilizers to vertices and faces of the graph. The homology of the surface guarantees that the stabilizers commute and that non-trivial logical operators correspond to homologically non-trivial cycles in the graph or its dual.

However, the simple 2-dimensional connectivity comes with a drawback. Like all 2-dimensional stabilizer codes, surface codes are limited by the Bravyi-Poulin-Terhal bound [2], which says that a code family that uses N qubits to encode K qubits with distance D must satisfy $cKD^2 \leq N$ for some constant c . For instance, the rotated surface code [3] achieves $c = 1$ in the plane, while the square-lattice toric code [1] achieves $c = 1$ on the torus.

Motivated in part by improving the constant c , twist defects have been introduced to surface codes [4, 5]. Fundamentally, twist defects are local regions where Pauli errors can create (or destroy) unpaired excitations of both types, X and Z , say, whereas elsewhere in the code the parities of these excitations are conserved. However, twist defects appear in several quantitatively different forms. For example, sometimes weight-five stabilizers are involved [4, 5], sometimes only weight-four as in the usual surface code [6, 7], and sometimes twists are embodied by a combination of weight-two and weight-six stabilizers [8]. Qualitatively, these defects behave the same, with M defects adding $(M - 2)/2$ logical qubits to the code and with non-trivial logical operators forming paths between pairs of defects.

Clever placement and implementation of twist defects can lead to savings in qubit count by producing codes with $N = cKD^2$ for relatively small constant c . For instance, in the plane, $c = 3/4$ [6] and $c = 1/2$ [9] have been achieved for surface codes, and $c = 3/8$ for color codes [9]. However, there is not a rigorous formalism for constructing surface codes with twists comparable to the well-defined homological formalism introduced by Kitaev for describing CSS surface codes [1]. Such a formalism would be helpful for looking for better surface codes with perhaps even smaller constants c .

Our main goal in this work is to introduce such a formalism, capable of describing surface codes with and without twist defects in a unified manner. Again we associate a code to an embedded graph, but find it most natural to do this in a different way than the homological formalism described above. Instead of placing qubits on edges, we place them on vertices. Only faces of the graph support stabilizers in our description, and odd-degree vertices act as the twist defects. Placing qubits on vertices is not a substantially new idea (for instance, appearing for the degree-4 case in [3]), but we go beyond prior works in applying the construction to arbitrary embedded graphs, yielding new codes with improved parameters. We highlight several code families we discuss in this work in Table 1.

Let us briefly summarize the main results of each section. Section 2 rigorously defines embedded graphs and related notation used throughout the paper. Our description of choice for an embedded graph is via a rotation system. Effectively, this is a combinatoric description of the embedded graph, which abstracts away unnecessary details about how exactly the graph is drawn on the surface. What remains is just the adjacency of the graph components (e.g. vertices, edges, and faces). The rotation system description is very similar to the formalisms used for hyperbolic codes in, for example, [10, 11] but slightly more general as it applies to non-regular graphs. Although the rotation system description may be overkill for individual codes, for which a picture may suffice, we believe that creating a more comprehensive theory of surface codes deserves such a precise framework. Having the rotation system description is also convenient for computer-assisted searches for codes, as we do for hyperbolic manifolds in Section 5.3. We identify a key property of embedded graphs called checkerboardability, which means one can two-color faces of the graph such that adjacent faces are differently colored.

In Section 3, we give our surface code construction. We actually provide two equivalent descrip-

tions, one using Majoranas to encode qubits and one using qubits to encode qubits. The Majorana description is similar to that used to describe surface codes in [12, 13, 14]. We place two Majoranas on each edge and a single Majorana at each odd-degree vertex. Stabilizers, products of Majoranas, are associated to faces and vertices of the graph. Identifying qubit subspaces at the vertices turns the Majorana code into a qubit surface code with qubits on the vertices, as we described above. A significant result in this section is the computation of the number of encoded qubits in Theorem 3.3. This depends on the genus of the surface, its orientability, the number of odd-degree vertices, and the checkerboardability of the surface. Finally, we note that, when the graph is checkerboardable, Kitaev’s homological construction, applied to a different but related graph, can be used to describe the same code.

Our focus in Section 4 is on finding logical operators and bounding the code distance. A key idea here is to derive a “decoding” graph from the original embedded graph. Vertices in the decoding graph represent stabilizers, edges represent Pauli errors, and cycles represent logical operators. In principle, one can perform perfect matching in the decoding graph to recover from errors, though for non-checkerboardable codes this may not succeed in achieving the code distance. Like Kitaev’s homological construction, we wish to associate homologically non-trivial cycles to non-trivial logical operators. This is easily done in the checkerboardable case of our construction. However, in the non-checkerboardable case, the decoding graph might not embed in the same manifold as the original graph. In this case, we instead find a way to embed the decoding graph in a manifold of potentially higher genus in such a way that this correspondence of homological and logical non-triviality is maintained. Bounds on the code distance follow in terms of lengths of cycles in the decoding graph, and we provide efficient algorithms to compute those bounds.

Finally, Section 5, we provide several code families that serve as examples of our construction. The first family, rotated square-lattice codes on the torus [12], serves as a familiar introduction. The second family introduces general rotations of the square-lattice on the torus to get codes like those in [15]. Included in this case is a family of cyclic codes that contains the famous 5-qubit quantum code as a member. The third family consists of hyperbolic codes defined using regular tilings of high genus surfaces. Because we are using our qubit-on-vertices definition of surface codes, this leads to new codes, different from those in, for instance [16, 17]. In these first examples, we take pains to calculate the code distances formally. A final section of examples consists of a planar code family generalizing the triangle code [6] and an embedding of stellated codes [9] on higher genus surfaces. In these cases, we do not formally prove the code distances but provide conjectures. We believe that stellated codes on higher genus surfaces provide the smallest known constant c , approaching $1/4$, for codes with stabilizers that are mostly weight four and vanishingly many stabilizers of weight five.

Code	Section	Notes	References
Majorana surface codes	Sec. 3, Fig. 2	Encodes qubits into Majoranas	[12, 13, 18]
Rotated toric codes	Sec. 5.1, Fig. 15	Encode 1 or 2 qubits on a torus	[12]
General rotated toric codes	Sec. 5.2, Fig. 6	Contains the $[[5, 1, 3]]$ code	[15, 19]
New hyperbolic codes	Sec. 5.3, Fig. 18, Tab. 3	Regular tilings, qubits on vertices	–
Circle-packing codes	Sec. 5.4, Fig. 19	Planar codes with $N \approx KD^2/2$	–
Stellated codes	Fig. 5	Planar codes with $N \approx KD^2/2$	[9]
High-genus stellated codes	Sec. 5.4, Fig. 20	$N \approx KD^2/4$ on 2D manifolds	–

Table 1: A summary of code families that we discuss in this work, some appearing for the first time here. Except for the Majorana codes, these are codes that use qubits to protect logical qubits.

2 Preliminaries

In this section, we first provide an introduction to the combinatorial description of graph embeddings (also called *maps*) on closed surfaces. We refer to this description as a *rotation system*, following terminology in [20, 21, 22]. We recall that a *closed surface* is a 2-dimensional, compact,

and connected topological manifold without boundary (see [23, 24] for an introduction to topological manifolds), and henceforth we simply say a *manifold* to mean a closed surface (unless specified otherwise). All manifolds admit a unique smooth structure up to diffeomorphisms [25], and by the *closed surface classification theorem* [26, 23, 27], each manifold is homeomorphic to a space of one of three types — (a) Type I: \mathbb{S}^2 , (b) Type II: a connected sum of one or more copies of \mathbb{T}^2 , (c) Type III: a connected sum of one or more copies of $\mathbb{R}\mathbb{P}^2$. This greatly simplifies the picture when we physically think of “drawing” a graph on a manifold, more formally captured using the concept of graph embeddings. A rotation system is simply a combinatorial description of any such graph embedding. To garner the most intuition, we believe that rotation systems are most clearly presented by first defining them for the general case of embeddings in orientable and non-orientable manifolds (Types I, II, III), and then discussing the case of embeddings in orientable manifolds (Types I, II) as a special case in Appendix A, which is a lot easier to visualize. To a large extent, the content of this section follows [20, 28, 29]. Graph theoretic terminology appearing in this section is borrowed from [30, 31].

This section is structured as follows. We begin by discussing graph embeddings in Section 2.1. General rotation systems are then introduced in Section 2.2. In the remainder of this section, we introduce a range of definitions and terminologies from topological graph theory, essential for the subsequent discussions in this paper. In Section 2.3, we introduce the concept of graph checkerboardability. Next in Section 2.4, we discuss the notion of \mathbb{F}_2 -homology of a graph embedding, and in particular introduce the notion of the homological systole. Finally, in Section 2.5 we discuss covering maps and contractible loops on a manifold.

2.1 Graph embeddings

Let us first cover graphs and their embeddings. A graph $G(V, E)$ is a collection of vertices V and edges E . We will mostly deal with finite graphs, i.e. $1 \leq |V| < \infty$, and $1 \leq |E| < \infty$, and explicitly state when this assumption is violated. An edge adjacent to a single vertex is a *loop*. If an edge is not a loop, then it is adjacent to exactly two vertices. The *degree* of a vertex v is denoted $\deg(v)$, and it is the number of edges adjacent to it, with loops counted with multiplicity two. We assume in this section that G is a connected graph.

A graph embedding requires that we actually draw the graph on a manifold \mathcal{M} . There may be multiple ways to do this even for a single graph and a fixed manifold. Generally, this drawing procedure is described by a *graph embedding map* $\Gamma : V \cup E \rightarrow \mathcal{M}$ that assigns unique points in \mathcal{M} to each $v \in V$, and *arcs* in \mathcal{M} to each $e \in E$. If $e \in E$ is not a loop adjacent to distinct vertices v_1 and v_2 , then the arc assigned to it must be the image of a homeomorphism $\gamma : [0, 1] \rightarrow \mathcal{M}$, with *endpoints* $\gamma(0) = \Gamma(v_1)$ and $\gamma(1) = \Gamma(v_2)$. If e is a loop adjacent to a vertex v , then the arc assigned to it is the image of a homeomorphism $\gamma : \mathbb{S}^1 \rightarrow \mathcal{M}$, with both endpoints given by $\gamma(0) = \Gamma(v)$ (here \mathbb{S}^1 is parameterized in polar coordinates). Moreover, two arcs may not intersect except possibly at their endpoints.

The set $\mathcal{F} = \mathcal{M} \setminus \Gamma(E)$ is a collection of regions called the *faces* of the embedding, and two points $p, q \in \mathcal{F}$ belong to the same face if and only if there exists a continuous map $\gamma' : [0, 1] \rightarrow \mathcal{F}$ such that $\gamma'(0) = p$, and $\gamma'(1) = q$. We say that an embedding of G is a *2-cell embedding* if and only if every face is homeomorphic to the open disc $\mathcal{B}(0, 1) = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 < 1\}$. If the closure of every face is homeomorphic to the closed disc $\mathcal{B}(0, 1)$, and the set difference of the closure of the face and the face itself is homeomorphic to the circle \mathbb{S}^1 , we say that it is a *closed 2-cell embedding* or a *strong embedding*. Every connected graph admits a 2-cell embedding in some orientable manifold (see Theorem A.16. in [22]), and also in some non-orientable manifold (see Theorem 3.4.3 in [31]). One can see a graph (a square lattice) embedded in the torus in Fig. 21 in Appendix A, while a graph embedded in the projective plane is shown in Fig. 1 below.

2.2 General rotation systems

The goal of this section is to establish an equivalence between graphs embedded in manifolds (both

orientable and non-orientable) and a combinatorial object called a general rotation system. The description of a graph embedding in the previous section requires a variety of homeomorphisms whose exact details, topologically speaking, do not matter. The rotation system description dispenses with those details by breaking the embedded graph down into a set of *flags* H , four for every edge, and describing vertices, edges, and faces of the graph embedding as sets of these flags. Each edge should be thought of as two *half-edges*¹, and of the four flags per edge, each half-edge corresponds to exactly two of the flags. The two flags per half-edge effectively introduce an orientation to the half-edge, with the flags corresponding to the two “sides” of the manifold, and it is this notion that allows us to handle both orientable and non-orientable manifolds simultaneously. Fig. 1 makes it clear how flags can be visualized in the embedding. The formal definition of a general rotation system is as follows.

Definition 2.1. A *general rotation system* is a quadruple $R = (H, \lambda, \rho, \tau)$, where H is a nonempty, finite set of flags, and λ, ρ, τ are permutations on H satisfying the properties

- (i) λ, ρ , and τ are fixed-point-free involutions, such that $h, \lambda h, \rho h$ and τh are distinct for every $h \in H$.
- (ii) $\lambda\tau = \tau\lambda$, or equivalently, $\lambda\tau$ is an involution.
- (iii) the *monodromy* group $M(R) = \langle \lambda, \rho, \tau \rangle$ acts transitively on H .

The involutions λ, ρ, τ permute flags as shown in Fig. 1. We define vertices, edges, and faces as orbits of $\langle \rho, \tau \rangle$, $\langle \lambda, \tau \rangle$, and $\langle \rho, \lambda \rangle$ respectively. From properties (i) and (ii) of Definition 2.1, orbits of $\langle \lambda, \tau \rangle$ have the form $\{h, \lambda h, \tau h, \lambda\tau h : h \in H\}$, i.e. are exactly of size four, and so $|H| \equiv 0 \pmod{4}$. Moreover, one deduces from property (i) that the orbits of $\langle \rho, \lambda \rangle$ and $\langle \rho, \tau \rangle$ have even sizes, with the canonical form of an orbit of size $2n + 4$ given by $\{h, \lambda h, \rho\lambda h, \lambda\rho\lambda h, \dots, (\lambda\rho)^{n+1}\lambda h : h \in H\}$, and $\{h, \tau h, \rho\tau h, \tau\rho\tau h, \dots, (\tau\rho)^{n+1}\tau h : h \in H\}$ respectively.

The sets of vertices, edges, and faces, which we denote V , E , and F respectively, define a graph embedding and satisfy Euler’s formula

$$\chi = |V| - |E| + |F|, \quad (1)$$

where χ is the Euler characteristic of the manifold, i.e. if g is the manifold’s genus (orientable or non-orientable), then $\chi = 2 - 2g$ if the manifold is orientable, and $\chi = 2 - g$ if it is non-orientable. Note that property (iii) ensures that G is a connected graph. We will write $G(V, E, F)$ to denote such an embedded graph, as compared to $G(V, E)$ which only denotes the graph without the embedding.

The orbits of the free group $\langle \tau \rangle$ are exactly the half-edges, and additionally we define *sectors* as the orbits of the free group $\langle \rho \rangle$. As τ and ρ are fixed-point-free involutions, these orbits are just two flags each. We denote a half-edge as $[h]_\tau = \{h, \tau h\}$, and a sector as $[h]_\rho = \{h, \rho h\}$.

Finally, we note that, in Definition 2.1, the requirement of distinctness in property (i) is unlike prior literature (see e.g. [28]), but conveniently rules out the possibilities of having degree-1 vertices (because $\rho h \neq \tau h$), unpaired half-edges ($\lambda h \neq \tau h$), and faces bounded by just one edge ($\rho h \neq \lambda h$), all of which are unnecessary and distracting for the later definitions of quantum codes.

While Definition 2.1 captures graph embedding on both orientable and non-orientable manifolds, the particular case of orientable manifolds is much simpler. This is carried out in Appendix A, and the relevant combinatorial object there is the oriented rotation system, defined in Definition A.1. However, as mentioned before, this is a special case of the general rotation system, and thus Definition 2.1 generalizes Definition A.1. Suppose we have an oriented rotation system $R_O = (H_O, \nu, \epsilon)$. There is a corresponding general rotation system $R = (H, \lambda, \rho, \tau)$ where $H = H_O \times \{1, -1\}$ and for $(h, i) \in H$,

$$\lambda(h, i) = (\epsilon h, -i), \quad \rho(h, i) = (\nu^i x, -i), \quad \tau(h, i) = (h, -i). \quad (2)$$

¹Half-edges are formally defined for the special case of oriented rotation systems in Appendix A, and Fig. 21 shows an example.

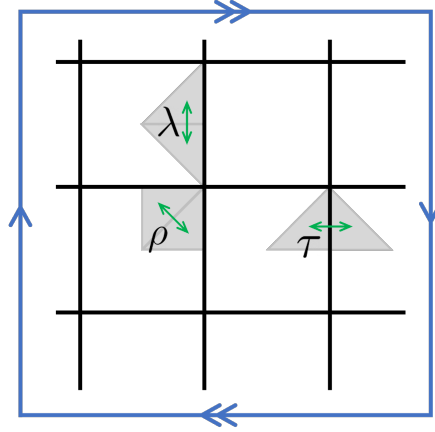


Figure 1: We show a graph embedded on the projective plane. A general rotation system is defined by a set of flags (gray triangles) and three permutations: λ swaps flags along the same side of an edge, ρ swaps them within a face adjacent to a vertex, and τ swaps them across an edge.

This definition ensures that the general rotation system represents the same embedded graph as the given oriented rotation system.

A general rotation system (H, λ, ρ, τ) describes a graph embedded in an orientable manifold if and only if H can be partitioned into two sets $H_{\pm 1}$ such that λ , ρ , and τ each map elements of either set to elements of the other set. Otherwise, the graph is embedded in a non-orientable manifold. If the manifold is orientable, there are at least two oriented rotation systems describing the same graph embedding, on the same set of half-edges. Take $H_O = H/\tau$ and let $[h]_\tau = \{h, \tau h\} \in H_O$ be a generic half-edge. Define $\nu[h]_\tau = [\rho\tau h]_\tau$, $\nu'[h]_\tau = [\tau\rho h]_\tau$, and $\epsilon[h]_\tau = [\lambda\tau h]_\tau$. Then, both (H_O, ν, ϵ) and (H_O, ν', ϵ) are oriented rotation systems for the same embedded graph.

Adjacency is an important concept for graphs and their components. Using the general rotation system description, flags are the primitives, and we say a flag is adjacent to a vertex, edge, or face $x \in V \cup E \cup F$, if it is contained in x . We define adjacency matrices in terms of flags. The vertex-flag adjacency matrix $A \in \mathbb{F}_2^{|V| \times |H|}$ has $A_{vh} = 1$ if and only if $h \in v$. Similarly define an edge-flag adjacency matrix $B \in \mathbb{F}_2^{|E| \times |H|}$, and a face-flag adjacency matrix $C \in \mathbb{F}_2^{|F| \times |H|}$. Although flags are primitives, we can also define adjacency of the more typical graph components. For example, we say that a vertex $v \in V$ and an edge $e \in E$ are adjacent if $v \cap e \neq \emptyset$. Likewise, adjacency of edges and faces, and of vertices and faces can be defined by non-trivial intersection. Note that each edge in an embedded graph is adjacent to at least one and at most two faces. The degree of a vertex v is half the number of flags it is adjacent to, $\deg(v) = |v|/2$, and this definition coincides with the one in Section 2.1. We also say that two vertices (resp. two faces) are adjacent, if there exists an edge adjacent to both vertices (resp. both faces).

2.3 Dual graph and checkerboardability

If an embedded graph $G(V, E, F)$ is described by the general rotation system (H, λ, ρ, τ) , then the *embedded dual graph* $\bar{G}(\bar{V}, \bar{E}, \bar{F})$ of G is defined by the general rotation system (H, τ, ρ, λ) , i.e. by simply exchanging the permutations λ and τ . This implies that \bar{G} has a vertex (resp. face) for every face (resp. vertex) of G , while $|E| = |\bar{E}|$, and we conclude that the Euler characteristic of G and \bar{G} are the same. Note also that the graph \bar{G} is connected (by properties of a general rotation system). Moreover \bar{G} defines an embedding in an orientable manifold if and only if G does too — thus the graphs G and \bar{G} are embedded in the same manifold. The edges of \bar{G} have a natural interpretation: for every edge $e \in E$ adjacent to faces $f, f' \in F$, not necessarily distinct, there is an edge $\bar{e} \in \bar{E}$ between the vertices of \bar{G} corresponding to f and f' (in particular if $f = f'$ then \bar{e} is a loop). The dual of the graph \bar{G} is isomorphic to the graph G . There are several more modifications of a graph G that we use in the paper. In Table 2, we list these and the sections where they are

Name	Original	Dual	Medial	Decoding	Doubled	Face-vertex
Notation	G	\overline{G}	\tilde{G}	G_{dec}	G^2	G_{fv}
Section	2.1	2.3	3.4	4.2	4.3	4.4

Table 2: Embedded graph G and the derived graphs used in this paper. Without loss of generality for constructing codes (see the end of Section 3.2), we assume G has no vertices of degree less than three.

defined.

Before proceeding any further, given a graph embedding $G(V, E, F)$, we define the face-edge adjacency matrix $\Phi = \frac{1}{2}CB^\top \in \mathbb{F}_2^{|F| \times |E|}$ (with this matrix multiplication followed by division performed over the integers, and then reduced modulo two), which captures the adjacency of the faces and edges of G . It is worth pointing out the structure of the matrix Φ . First note that each column of B^\top contains exactly four non-zeros, as each edge $e \in E$ is a set of four flags. Now there are two cases: (i) the two sides of e belong to distinct faces $f, f' \in F$, and (ii) e is surrounded by a single face on both sides, i.e. e is a subset of that face. In the former case, $\Phi_{fe} = \Phi_{f'e} = 1$ are the only non-zero entries of the column of Φ corresponding to e , while in the latter case $\Phi_{fe} = 0$ for all $f \in F$. The face-edge adjacency matrix Φ can also be interpreted in homological language directly, and this is mentioned in Appendix B.

We now discuss an important property of embedded graphs from the perspective of quantum codes, called checkerboardability.

Definition 2.2. A graph embedding G is *checkerboardable* if the dual graph \overline{G} is bipartite.

An equivalent, and a more intuitive way of stating this definition is the following lemma.

Lemma 2.1. *A graph embedding $G(V, E, F)$ described by a general rotation system (H, λ, ρ, τ) , is checkerboardable if and only if the faces of G can be two-colored, such that for every flag $h \in H$, h and τh are differently colored.*

Proof. One direction of this equivalence is clear. Suppose \overline{G} is bipartite, and we color the two disjoint vertex sets of \overline{G} , corresponding to the bipartition differently. Using the correspondence between the faces of G and vertices of \overline{G} , we get a two-coloring of F , and hence of the flags H . Now pick any flag $h \in H$, and let $h \in e \in E$. Then $h, \tau h$ must belong to distinct faces $f, f' \in F$ respectively, otherwise if e is adjacent to a single face then \overline{G} will contain a loop (as mentioned above), and hence cannot be bipartite. Since f and f' are both adjacent to e , they must be differently colored. For the converse, suppose we are given a two-coloring of F (which now induces a coloring of the vertices of \overline{G}) such that h , and τh are differently colored for all $h \in H$. Consider $\overline{G}(\overline{V}, \overline{E})$, and partition the vertices as $\overline{V} = \overline{V}_w \sqcup \overline{V}_b$, where each disjoint partition contains all vertices of \overline{G} of a single color. Assume that \overline{G} is not bipartite with this bipartition. If \overline{G} has a loop then there exists $e \in \overline{E}$ adjacent to a single face. If \overline{G} has no loop, then either \overline{V}_w or \overline{V}_b is not an independent set; so there exist distinct faces $f, f' \in F$ of the same color, adjacent to an edge $e \in E$. In both cases, all four flags of e have the same color which is a contradiction. Thus \overline{G} is bipartite. \square

There are numerous other equivalent statements of checkerboardability. We give two more below that we find useful. Note that the only permutation out of λ, ρ, τ that moves flags across edges of the graph is τ . Therefore using Lemma 2.1, we have the following:

Lemma 2.2. *A graph embedding $G(V, E, F)$ described by general rotation system $R = (H, \lambda, \rho, \tau)$, is checkerboardable if and only if we can partition H into two disjoint sets H_w and H_b such that (i) λ and ρ map both sets to themselves, and (ii) τ maps elements of either set to the other set.*

Proof. Suppose first that G is checkerboardable. Then there is a two-coloring of the faces of F such that for every $h \in H$, h and τh are differently colored. Defining H_w and H_b to be the sets of flags of each color then shows that these sets have the required properties (i) and (ii), proving the

only if direction. Now assume that H can be partitioned into disjoint sets H_w and H_b satisfying the two properties. If $f \in F$, by property (i) all flags of f must belong to either H_w or H_b . Let us color all flags of H_w and H_b black and white respectively, which induces a two-coloring of all the faces of G . Property (ii) now implies that for every $h \in H$, h and τh are differently colored, proving that G is checkerboardable. \square

In this paper we adopt the notation that $\vec{1}$ and $\vec{0}$ are the row vectors of all 1s and all 0s respectively. The arguments above show that in a checkerboard coloring, each edge is adjacent to exactly one face of each color (white and black, say). Thus, checkerboardability implies the existence of a vector $x \in \mathbb{F}_2^{|F|}$ (e.g. $x_f = 1$ if and only if face f is colored white) such that $x\Phi = \vec{1}$. In fact we also have that $(\vec{1} - x)\Phi = \vec{1}$, and these are the only two vectors with this property. This is summarized in the following lemma.

Lemma 2.3. *Let $G(V, E, F)$ be a graph embedding described by the general rotation system (H, λ, ρ, τ) , and let Φ be its face-edge adjacency matrix. Then*

- (a) *G is checkerboardable if there exists $x \in \mathbb{F}_2^{|F|}$ such that $x\Phi = \vec{1}$.*
- (b) *If G is checkerboardable then the set $\{x \in \mathbb{F}_2^{|F|} : x\Phi = \vec{1}\}$ has exactly two distinct elements x and x' (different from $\vec{0}$ and $\vec{1}$), which satisfy $x + x' = \vec{1}$. Thus the partitioning of the flags into the two disjoint sets as described in Lemma 2.2 is unique.*

Proof. (a) Suppose there exists x such that $x\Phi = \vec{1}$, and define $F_b = \{f \in F : x_f = 1\}$, and $F_w = F \setminus F_b$. Color the faces in F_b and F_w differently. Assume for contradiction that $h \in H$ is a flag such that h and τh have the same color, and let $h \in e \in E$. Since h and λh belong to the same face, this implies that all four flags of e have the same color. But this implies $\Phi_{fe} = 0$ for all $f \in F$ (by properties of Φ discussed above), and so $(x\Phi)_e = \vec{0}$, which is a contradiction.

(b) Assume G is checkerboardable, which also implies that every column of Φ has exactly two non-zeros (or equivalently each edge $e \in E$ is adjacent to two distinct faces), and so $\vec{1}\Phi = \vec{0}$. We already know from the discussion in the paragraph before this lemma that there exists $y \in \mathbb{F}_2^{|F|}$ such that $y\Phi = \vec{1}$. Thus we also conclude that $(\vec{1} - y)\Phi = \vec{1}$. Now suppose there is a vector $x \in \mathbb{F}_2^{|F|}$ such that $x\Phi = \vec{1}$, and define a disjoint partition $F = F_w \sqcup F_b$ of the faces of G , as in the proof of (a). This induces a disjoint vertex partition $\bar{V} = \bar{V}_w \sqcup \bar{V}_b$ of the dual graph $\bar{G}(\bar{V}, \bar{E})$. Then combining the arguments of the proof of (a), and the proof of the converse part of Lemma 2.1 shows that \bar{G} is bipartite. Since, \bar{G} is connected, the bipartition is unique, and so x is equal to either y or $\vec{1} - y$. This proves the first part, and the second part follows from it easily. \square

A consequence of Lemma 2.3(b) is that the checkerboard coloring of a checkerboardable graph is essentially unique up to a permutation of the colors. One easy way a graph can fail to be checkerboardable is if it contains an odd-degree vertex v . Then, $(\rho\tau)^{\deg(v)}$ would map $h \in v$ to itself, but also because τ is applied an odd number of times, it would map h from one set, say H_w , to the other H_b (where H_w and H_b are as in Lemma 2.2), a contradiction. If $g = 0$ (the graph is planar, i.e. embedded on the 2-sphere \mathbb{S}^2), we also have the well known converse: a planar graph embedding is checkerboardable if all vertices have even degree.

2.4 Homology of graph embeddings

One topological invariant that is important for this paper (and for other quantum codes such as homological codes) is that of \mathbb{F}_2 -homology. We introduce this concept briefly here, and refer the reader to [32] for a more comprehensive treatment of homology. To do this in the context of embedded graphs, we will first introduce some graph theoretic terminology. We prefer this approach of presenting homological concepts in graph theoretic language as it places a lighter burden on the reader than having to know algebraic topology; however, the corresponding connections with known objects in algebraic topology (specifically chain complexes) are described in Appendix B.

Recall first that a *trail* t in a graph is a sequence of vertices and edges,

$$t = (v_0, e_1, v_1, e_2, \dots, v_{\ell-1}, e_\ell, v_\ell), \quad (3)$$

where edge e_i is adjacent to vertices v_{i-1} and v_i , and each edge is distinct. Trails can be open (if $v_0 \neq v_\ell$) or closed (if $v_0 = v_\ell$). Open trails have endpoints v_0 and v_ℓ . Closed trails are also called cycles. Including the vertices in the specification of a trail is standard but superfluous — a trail t can be represented by a vector $t \in \mathbb{F}_2^{|E|}$ such that $t_i = 1$ if and only if the i^{th} edge of the graph is in the trail. A *path* is a trail without repeated vertices, except the first and last if it is a closed trail. The set of trails generates a group $\mathcal{T}(G)$ with the group operation the symmetric difference of the trails or, equivalently, the addition modulo two of the vectors $y \in \mathbb{F}_2^{|E|}$ representing the trails. Because each edge is a trail, $\mathcal{T}(G)$ is isomorphic to the vector space $\mathbb{F}_2^{|E|}$, and a generating set of $\mathcal{T}(G)$ (or equivalently a basis for $\mathbb{F}_2^{|E|}$) can be constructed to consist only of paths.

Now let $G(V, E, F)$ be an embedded graph on a manifold, either orientable or non-orientable. Two subgroups of $\mathcal{T}(G)$ are particularly important in the definition of \mathbb{F}_2 -homology: the subgroup $\mathcal{Z}(G)$ of all the cycles of G , and the subgroup $\mathcal{B}(G)$ generated by the cycles of G that are *boundaries* of faces of the embedded graph. Formally

$$\begin{aligned} \mathcal{Z}(G) &= \langle c : c \text{ is a cycle of } G \rangle, \\ \mathcal{B}(G) &= \langle c : c \text{ is a cycle of } G, \exists f \in F \text{ such that } \forall \text{ edges } e \in c, \Phi_{f_e} = 1 \rangle. \end{aligned} \quad (4)$$

We call cycles in $\mathcal{B}(G)$ homologically trivial and cycles in $\mathcal{Z}(G)$ but not in $\mathcal{B}(G)$ homologically non-trivial. The homological systole of the graph, $\text{hsys}(G)$, is the length of the shortest homologically non-trivial cycle. The first homology group over \mathbb{F}_2 is the quotient group (or equivalently as the quotient vector space)

$$H_1(G, \mathbb{F}_2) = \mathcal{Z}(G)/\mathcal{B}(G). \quad (5)$$

It turns out that for two different graphs G, G' embedded in the same manifold \mathcal{M} , $H_1(G, \mathbb{F}_2) \cong H_1(G', \mathbb{F}_2)$. Thus in the rest of the paper, we will simply write $H_1(\mathcal{M})$ for any given graph embedding $G(V, E, F)$ in \mathcal{M} .

2.5 Covering maps and contractible loops

A cycle drawn on a manifold \mathcal{M} is contractible if it can be continuously deformed to a point and non-contractible otherwise. Homological non-triviality implies non-contractibility [33]. However, the converse is not true – consider for instance a cycle winding twice around a torus, which is non-contractible but homologically trivial. Similar to the homological case, we let $\text{sys}(G)$ denote the length of the shortest non-contractible cycle in a graph G embedded on \mathcal{M} and note $\text{hsys}(G) \geq \text{sys}(G)$. A manifold is simply-connected if all cycles are contractible.

We say a connected manifold \mathcal{M}' is a cover of \mathcal{M} if there is a map $\pi : \mathcal{M}' \rightarrow \mathcal{M}$ such that each open disk $U \subseteq \mathcal{M}$ has a pre-image $\pi^{-1}(U)$ that is a union of disjoint open disks, each mapped homeomorphically onto U by π [34]. We say π is a covering map. An l -fold cover is one in which $\pi^{-1}(U)$ is a union of l open disks for all U . It is also acceptable to call a 2-fold cover a double cover, a 3-fold cover a triple cover, etc.

A universal cover \mathcal{U} of a manifold \mathcal{M} is a cover that covers all other covers of \mathcal{M} . Equivalently, the universal cover of \mathcal{M} is the unique simply-connected cover of \mathcal{M} (see for instance Chapter III.4 of [34]). Compact manifolds are universally covered by either the sphere (if \mathcal{M} is the sphere or projective plane) or the plane (otherwise).

A curve drawn in \mathcal{M}' can be projected down to \mathcal{M} by applying π . Conversely, a curve $\gamma : [0, 1] \rightarrow \mathcal{M}$ can be *lifted* to \mathcal{M}' . Roughly one constructs a lift by applying π^{-1} to γ . However, since π^{-1} is one-to-many, more precisely a lift of γ is defined to be any compact, continuous curve in \mathcal{M}' that maps to γ when applying π .

An interesting connection presents itself between contractibility and the universal cover. A cycle in \mathcal{M} is contractible if and only if its lift into the universal cover \mathcal{U} is also a cycle. Non-contractible curves, on the other hand, lift to curves with two distinct endpoints [33].

3 Majorana and qubit surface codes from rotation systems

In this section, we define our framework for constructing qubit surface codes from rotation systems. Majorana surface codes appear as a useful intermediary in the construction. So, building on the last section's introduction to rotation systems, the logical progression of this section is roughly

$$\begin{array}{ccccc} \text{Rotation system} \setminus \text{graph embedding} & \longrightarrow & \text{Majorana surface code} & \longrightarrow & \text{Qubit surface code} \\ \text{(Section 2)} & & \text{(Section 3.2)} & & \text{(Section 3.3)} \end{array} .$$

Before diving into surface codes, Section 3.1 gives a quick introduction to Majorana fermions and Majorana fermion codes in general. Afterward, in Section 3.4, we show how our framework for qubit surface codes generalizes the standard homological definition [1], in which qubits are placed on edges, while vertices and faces support stabilizers. We assume some familiarity of the reader with the Pauli group, and Appendix C contains a brief recap.

3.1 Majorana operators and codes

To introduce Majorana operators in this subsection, we largely follow and summarize [35]. For even integer m , the Majorana operators $\{\gamma_0, \gamma_1, \dots, \gamma_{m-1}\}$ are linear Hermitian operators acting on the fermionic Fock space $\mathcal{H}_{m/2} = \{|\vec{b}\rangle : \vec{b} \in \mathbb{F}_2^{m/2}\}$, or equivalently the $m/2$ -qubit complex Hilbert space, satisfying

$$\gamma_i^2 = I, \quad \gamma_i \gamma_j = -\gamma_j \gamma_i, \quad \forall 0 \leq i < j \leq m-1. \quad (6)$$

Eq. (6) ensures that each γ_i is distinct and different from I . The total number of Majorana operators is even because two Majoranas correspond to each fermion in the system. We define a group \mathcal{J}_m consisting of finite products of the Majorana operators γ_i , and phase factor $i = \sqrt{-1}$. By Eq. (6), this group is finite with size $|\mathcal{J}_m| = 2^{m+2}$. Elements of \mathcal{J}_m either commute or anticommute. We indicate an element of \mathcal{J}_m uniquely by $\eta\gamma_a$ where $\eta \in \{\pm 1, \pm i\}$, $a \in \mathbb{F}_2^m$, and $\gamma_a = \prod_{i:a_i=1} \gamma_i$ (this product is ordered so that the γ_i with smaller indices are on the left, and the empty product is defined to be equal to I), and the uniqueness of this representation follows from Eq. (6). We define the *support* of $\eta\gamma_a$ to be the set $\text{supp}(\eta\gamma_a) := \{i : a_i = 1\}$, and its *weight* $|\eta\gamma_a| := |\text{supp}(\eta\gamma_a)| = |a|$, where $|a|$ is the Hamming weight of a . The commutation of the elements of \mathcal{J}_m is now easily expressed as

$$\gamma_a \gamma_b = (-1)^{|a||b|+a \cdot b} \gamma_b \gamma_a = (-1)^{\xi(a,b)} \gamma_{a+b}, \quad \xi(a,b) = \sum_{i:b_i=1} |\{j > i : a_j = 1\}|, \quad (7)$$

where it is understood the operations “ \cdot ” and “ $+$ ” are performed over \mathbb{F}_2 (the quantity $|a||b|$ is computed over integers and reduced modulo 2). These vectors are assumed to have context-appropriate length.

Particular choices of Majorana operators can be made by associating them with appropriately chosen Pauli operators on $m/2$ qubits. This can be done in a variety of ways. A particularly famous example is the Jordan-Wigner transformation, which makes the association

$$\text{JW}(\gamma_{2k}) = X_k \prod_{i=0}^{k-1} Z_i, \quad \text{JW}(\gamma_{2k+1}) = Y_k \prod_{i=0}^{k-1} Z_i, \quad (8)$$

for all $0 \leq k \leq m/2 - 1$, where X_i, Y_i, Z_i are the Paulis acting on qubit i . One can check that the Paulis associated with the Majoranas obey Eq. (6).

Majorana fermion codes (see e.g. [35, 36]) are created by specifying a subgroup $\mathcal{S} \leq \mathcal{J}_m$ to be a stabilizer group. The corresponding codespace is defined as usual to be the $+1$ -eigenspace of all stabilizers, a subspace of $\mathcal{H}_{m/2}$. We require the stabilizer group be chosen so that

- (i) \mathcal{S} does not contain $-I$.

- (ii) Each $\gamma \in \mathcal{S}$ has even weight.

The first condition arises because we wish the codespace to be non-empty, and the second condition is imposed because we want the stabilizer operators to be physical, preserving fermion parity in the system. In fact, the first condition also implies that \mathcal{S} is Abelian and that all elements of \mathcal{S} are Hermitian. We note this fact and a few others in the following lemma, whose proof is in Appendix D.

Lemma 3.1. *Let \mathcal{S} be a subgroup of \mathcal{J}_m , and let $\mathcal{I} \subseteq \mathcal{J}_m$ be non-empty, such that elements of \mathcal{I} commute and are Hermitian. Then $\langle \mathcal{I} \rangle$ is Abelian and Hermitian, and moreover the following holds:*

- (a) *There exists a set \mathcal{I}' formed by multiplying each element of \mathcal{I} by either 1 or -1 , such that $-I \notin \langle \mathcal{I}' \rangle$.*
- (b) *If $-I \notin \mathcal{S}$, then \mathcal{S} is Abelian and Hermitian. Conversely, if \mathcal{S} is Abelian and Hermitian, then either $-I \notin \mathcal{S}$, or $\mathcal{S} = \langle \mathcal{S}', -I \rangle$ for some subgroup \mathcal{S}' of \mathcal{S} with $-I \notin \mathcal{S}'$.*

Because of Pauli representations of Majoranas like Eq. (8), it is clear that a Majorana fermion code on m Majoranas corresponds to a stabilizer code on $m/2$ qubits. Applying the Jordan-Wigner transformation, Eq. (8), to \mathcal{S} converts it to an Abelian group of Pauli operators on $m/2$ qubits, which has size upper bounded by $2^{m/2}$. Therefore, $|\mathcal{S}| \leq 2^{m/2}$. Thus, if \mathcal{S} is generated by $m/2 - k$ independent operators, then the code encodes k qubits (or $2k$ Majoranas). The centralizer $\mathcal{C}(\mathcal{S})$ of \mathcal{S} is the subset of \mathcal{J}_m that commutes with all elements of \mathcal{S} . The *distance* d of the Majorana code is the minimum weight of an element of $\mathcal{C}(\mathcal{S})$ that is not (up to factors of i) in \mathcal{S} , i.e. $d = \min\{|\gamma| : \gamma \in \mathcal{C}(\mathcal{S}) \setminus \langle \{iI\} \cup \mathcal{S} \rangle\}$.

Consider, for example, a simple Majorana code, generated by just one stabilizer:

$$\mathcal{S}_{\text{even}} = \langle i^{m/2} \gamma_0 \gamma_1 \dots \gamma_{m-1} \rangle. \quad (9)$$

The phase $i^{m/2}$ guarantees this stabilizer squares to I and not $-I$. This code encodes $k = (m/2 - 1)$ qubits into m Majoranas with distance 2, because the centralizer consists of all γ_a with even weight. The code $\mathcal{S}_{\text{even}}$ will play an important role in our converting Majorana codes defined on graphs to qubit stabilizer codes defined on graphs.

3.2 Majorana surface codes defined on embedded graphs

This subsection defines Majorana surface codes that generalize those in prior works [12, 13, 35, 18] by starting with an arbitrary embedded graph G given by a rotation system R . We define a Majorana stabilizer code $\mathcal{S}(R)$ (or equivalently $\mathcal{S}(G)$) by placing Majorana operators on each half-edge and each odd-degree vertex, and associating stabilizers to each vertex and face of the embedded graph. The formal definition is as follows.

Definition 3.1. Given a rotation system $R = (H, \lambda, \rho, \tau)$ and the associated embedded graph $G = (V, E, F)$ with M odd-degree vertices, associate a single Majorana $\gamma_{[h]_\tau}$ to each half-edge $[h]_\tau = \{h, \tau h\} \in H/\tau$ and a single Majorana $\bar{\gamma}_v$ to each odd-degree vertex in $v \in V$. Now to every vertex $v \in V$, associate a *vertex stabilizer* S_v by

$$S_v = \begin{cases} i^{\deg(v)/2} \left(\prod_{[h]_\tau \subseteq v} \gamma_{[h]_\tau} \right), & \deg(v) \text{ is even} \\ i^{(\deg(v)+1)/2} \left(\prod_{[h]_\tau \subseteq v} \gamma_{[h]_\tau} \right) \bar{\gamma}_v, & \deg(v) \text{ is odd} \end{cases} \quad (10)$$

and to every face $f \in F$, a *face stabilizer* S_f by

$$S_f = i^{|\{h \in f : \tau h \notin f\}|/2} \prod_{\{h \in f : \tau h \notin f\}} \gamma_{[h]_\tau}. \quad (11)$$

The order of Majoranas in these products is important. To clarify this, we give each Majorana a unique label from $\{0, 1, \dots, m-1\}$ satisfying certain rules, where $m = 2|E| + M$, and then

the products are sorted by ascending order in these labels. When G is not checkerboardable, the labeling can be arbitrary as long as it ensures that any two Majoranas on the same edge, e.g. $\gamma_{[h]_r}$ and $\gamma_{[\lambda h]_r}$, have successive labels. When G is checkerboardable, the labeling should additionally ensure that for all $0 \leq j \leq m/2 - 1$, the Majoranas corresponding to labels $2j + 1$ and $2j + 2$ (evaluating the labels modulo m) belong to half-edges which are subsets of the same vertex. We denote the group generated by all S_v and S_f as $\mathcal{S}(R)$ (or equivalently $\mathcal{S}(G)$). The Majorana code associated with R (or G) has the stabilizer group $\mathcal{S}(R)$.

We note a few easy consequences of this definition. First, notice that the total number of Majoranas is even since M is, which is true because $2|E| = \sum_{v \in V} \deg(v)$. Second, notice that the vertex and the face stabilizers have even weight, are Hermitian (since they square to I), and commute. The commutation follows from Eq. (7) since supports of S_v and $S_{v'}$ do not overlap for distinct $v, v' \in V$, supports of S_f and $S_{f'}$ for distinct $f, f' \in F$ overlap if and only if they share adjacent edges and therefore share an even number of Majoranas, and supports of S_v and S_f for $v \in V$ and $f \in F$ overlap only if v and f are both adjacent to one or more sectors in which case they share an even number of Majoranas (note that this holds even if v and f are both adjacent to any edge e such that e is not adjacent to any other face).

It turns out that the Majorana labeling scheme used in Definition 3.1 ensures that the stabilizer group $\mathcal{S}(R)$ has the property $-I \notin \mathcal{S}(R)$, which is formally proved below in Lemma 3.2(g), and since each stabilizer has even weight, so does every element of $\mathcal{S}(R)$; thus $\mathcal{S}(R)$ satisfies the two properties of a Majorana fermion code. If a different Majorana labeling was used, or if the products in Eqs. (10) and (11) were ordered differently then it is not necessarily true that the first property holds, but in that case by Lemma 3.1(a) one can multiply the stabilizers by either 1 or -1 and get a new set of stabilizers for which the property would still be true. By Lemma 3.1(b), $\mathcal{S}(R)$ is Abelian and Hermitian. Two examples of Majorana surface code stabilizers are shown in Fig. 2.

We note briefly that a labeling scheme satisfying the demands of Definition 3.1 exists in the checkerboardable case (the non-checkerboardable case is trivial). If G is checkerboardable it has no odd-degree vertices, so there exists an Euler cycle $(e_0, e_1, \dots, e_{|E|-1})$ that uses all the edges in E . Majoranas can be labeled in the order they are encountered by following this cycle.

The next lemma is useful to understand the dependence between the stabilizers of the Majorana surface code. Its proof is in Appendix D.

Lemma 3.2. *The stabilizers of the Majorana surface code for the embedded graph $G(V, E, F)$ satisfy*

- (a) *There is no non-empty subset $V' \subseteq V$ such that $\prod_{v \in V'} S_v = \pm I$.*
- (b) *There is no non-empty, proper subset $V' \subset V$ and no subset $F' \subseteq F$ such that $\prod_{v \in V'} S_v = \pm \prod_{f \in F'} S_f$. If there is any odd-degree vertex, then the statement holds for all non-empty subsets $V' \subseteq V$.*
- (c) *If $\prod_{v \in V} S_v = \prod_{f \in F'} S_f$ for some subset $F' \subseteq F$, then G is checkerboardable. If G is checkerboardable then $\prod_{v \in V} S_v = \prod_{f \in F'} S_f = \prod_{f \in F \setminus F'} S_f$ for a non-empty proper subset $F' \subset F$, which is determined uniquely up to taking complements.*
- (d) *There is no non-empty, proper subset $F' \subset F$ such that $\prod_{f \in F'} S_f = \pm I$.*
- (e) $\prod_{f \in F} S_f = I$.
- (f) $|F| = 1$ if and only if there is $f \in F$ such that $S_f = I$.
- (g) $-I \notin \mathcal{S}(G)$.

A consequence of Lemma 3.2(g) is that since $\mathcal{S}(R)$ is Abelian and Hermitian, it implies that a non-empty subset of the stabilizers is dependent if and only if the product of its elements is I . Let us determine the number of qubits encoded by this Majorana code. There are $|V| + |F|$ stabilizers as defined. However, not all these stabilizers are independent. If the total number of independent stabilizers is $|V| + |F| - \alpha$ for a non-negative integer α , then the number of encoded qubits is

$$K = (2|E| + M)/2 - (|V| + |F| - \alpha) = M/2 - \chi + \alpha, \quad (12)$$

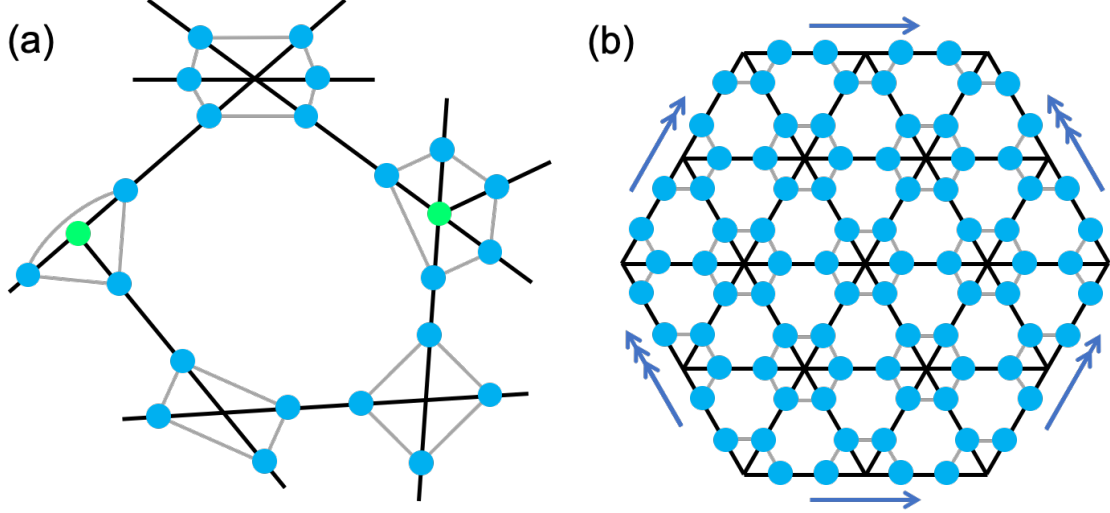


Figure 2: (a) An example section of a Majorana surface code. Majorana operators are placed on each half-edge (blue circles) and at each odd-degree vertex (green circles). The stabilizer associated to an even-degree vertex is the product of Majoranas around that vertex. The stabilizer associated to an odd-degree vertex is the product of Majoranas around that vertex and the Majorana located at that vertex. Gray lines cordon off the Majoranas involved in these stabilizers. The stabilizer associated to the pentagonal face is the product of the ten blue Majoranas on edges around that face. (b) A Majorana surface code of reference [18]. In our framework, it arises from a graph triangulating the torus. Here opposite sides of the hexagon are identified (directionally, as indicated by arrows) along with Majoranas on those edges. A total of 72 Majoranas survive this identification. All stabilizers associated to both vertices and faces are the product of six Majoranas.

using Eq. (1). The remaining work is in determining α by counting dependencies in the stabilizers, which can be done easily using Lemma 3.2.

Theorem 3.3. *A Majorana surface code encodes*

$$K = \left\{ \begin{array}{l} 2g \quad , \text{ orientable} \\ g \quad \quad , \text{ non-orientable} \end{array} \right\} + \left\{ \begin{array}{l} 0 \quad \quad , \text{ checkerboardable} \\ (M-2)/2 \quad , \text{ not checkerboardable} \end{array} \right\} \quad (13)$$

qubits, where conditions in brackets are properties of the rotation system (H, λ, ρ, τ) , or equivalently the graph embedding, defining the code.

Proof. We will argue that (i) $\alpha \in \{1, 2\}$, and (ii) $\alpha = 2$ if and only if the graph is checkerboardable. Once we have done so and because a checkerboardable graph does not contain any odd degree vertices, we can use Eq. (12) to complete the proof.

To prove (i), let $\mathcal{I} = \{S_v : v \in V\} \cup \{S_f : f \in F\}$ be the set of stabilizers, so $\langle \mathcal{I} \rangle = \mathcal{S}(R)$. By Lemma 3.2(e), one can always remove a single face stabilizer S_f so that $\langle \mathcal{I} \setminus S_f \rangle = \mathcal{S}(R)$, which implies $\alpha \geq 1$. If $\mathcal{I} \setminus S_f$ is not independent, by Lemma 3.2(a,b,d) it must be that $\prod_{v \in V} S_v = \prod_{f \in F'} S_f$ for some non-empty subset $F' \subseteq F \setminus f$. If this happens then we remove a vertex stabilizer S_v arbitrarily after which Lemma 3.2(b) guarantees that the set $\mathcal{I} \setminus \{S_f, S_v\}$ is independent. Thus $\alpha \leq 2$. To prove (ii), by the preceding arguments $\alpha = 2$ if and only if $\prod_{v \in V} S_v = \prod_{f \in F'} S_f$ for some non-empty proper subset $F' \subset F$, and the latter is true if and only if the graph is checkerboardable by Lemma 3.2(c). \square

We also point out that if the graph defining the Majorana surface code has vertices with degree one or degree two, then there are vertex stabilizers S_v that are the product of just two modes. These two modes and the stabilizer that is their product can be removed from the code without affecting the number of encoded qubits or code distance. What may be affected is the degree of protection arising from superselection rules [35], e.g. Kitaev's 1D chain [37] has many such weight-two stabilizers. Nevertheless, our main goal is to convert these Majorana surface codes to qubit codes where superselection is not a relevant protection. Thus, we assume from now on that our

graphs have no vertices of degree less than three.

3.3 From Majorana surface codes to qubit surface codes

In this subsection we replace Majorana operators with Paulis to make qubit surface codes on arbitrary graphs, thus slightly generalizing an idea from [12, 13, 14]. First, recognize that at each even (resp. odd) degree vertex v , because of the vertex stabilizer S_v , there is a copy of the code $\mathcal{S}_{\text{even}}$, Eq. (9), defined on $\deg(v)$ (resp. $\deg(v) + 1$) distinct Majorana modes and encoding $(\deg(v) - 2)/2$ (resp. $(\deg(v) - 1)/2$) qubits. Therefore, to define the qubit code given an embedded graph $G = (V, E, F)$, the first step is to place these many qubits at each vertex. We already assumed the graph has vertices of degree at least three, so there is at least one qubit at each vertex.

Now, consider a single vertex v . By definition, Paulis that act only on the qubits at v can be associated to logical operators of the code $\mathcal{S}_{\text{even}}$ located at v . We are most interested in the Paulis associated to the sector operators $q_{[h]_\rho} = i\gamma_{[h]_\tau}\gamma_{[\rho h]_\tau}$ for each flag $h \in v$. That is, $q_{[h]_\rho}$ is the product of the two Majoranas on half-edges adjacent to the sector. We are interested particularly in $q_{[h]_\rho}$ because the face stabilizers of the Majorana code can be alternatively written as the product of these sector operators,

$$S_f = \pm \prod_{[h]_\rho \subseteq f} q_{[h]_\rho}. \quad (14)$$

Therefore, choosing Paulis $q_{[h]_\rho}$ will also give S_f in terms of Paulis.

Let us list the sector operators $\{q_{[h]_\rho}, q_{[\tau \rho h]_\rho}, \dots, q_{[(\tau \rho)^{\deg(v)-1} h]_\rho}\}$ starting from some arbitrary $h \in v$. Notice that these operators have very specific commutation rules, with adjacent (and the first and last elements also considered adjacent) elements anticommute with one another, and all other pairs commuting. This leads us to study the following lists of Paulis.

Definition 3.2. A *cyclically anticommuting list* (CAL) acting on n qubits is a list of n -qubit Paulis $\{p_0, p_1, \dots, p_{\ell-1}\}$, in which for distinct i and j , p_i and p_j anticommute if and only if $i = j \pm 1 \pmod{\ell}$. A CAL of length $\ell \geq 1$ is called *extremal* if there does not exist a CAL of the same length acting on fewer qubits.

Clearly, the list of sector operators $\{q_{[h]_\rho}, q_{[\tau \rho h]_\rho}, \dots, q_{[(\tau \rho)^{\deg(v)-1} h]_\rho}\}$ is a CAL of length $\deg(v)$ acting on $(\deg(v) - 2)/2$ qubits if v has even degree or acting on $(\deg(v) - 1)/2$ qubits if v has odd degree. Let us establish existence of CALs with a construction.

Theorem 3.4. A CAL of length $\ell \geq 3$ acting on n qubits in which the Paulis have weight at most two exists if and only if

$$n \geq \begin{cases} (\ell - 2)/2, & \ell \text{ even,} \\ (\ell - 1)/2, & \ell \text{ odd.} \end{cases} \quad (15)$$

Proof. We construct extremal CALs with length ℓ acting on n qubits saturating the lower bound in Eq. (15). To get non-extremal CALs, more qubits can always be added by acting on them with identity, or just Pauli Z , for instance, so long as the commutation relations are unaffected. We note that extremal CALs of length three and four, acting on a single qubit, are $\{X, Y, Z\}$ and $\{X, Z, X, Z\}$. To construct extremal CALs of longer length, we define a composition operation \circ that takes a CAL of length ℓ (even) and a CAL of length ℓ' (either even or odd) to a CAL of length $\ell + \ell' - 2$:

$$\{p_0, \dots, p_{\ell-1}\} \circ \{p'_0, \dots, p'_{\ell'-1}\} := \{p_0, \dots, p_{\ell/2-2}, p_{\ell/2-1} p'_0, p'_1, p'_2, \dots, p'_{\ell'-2}, p'_{\ell'-1} p_{\ell/2}, p_{\ell/2+1} \dots p_{\ell-1}\}, \quad (16)$$

where the p_j s and p'_j s represent Paulis that act on n and n' qubits, respectively. Thus the CAL on the right acts on $n + n'$ qubits. Repeated application of this composition suffices to make CALs of any length out of the length three and four CALs given above. Composition preserves extremality — if n and n' are minimal qubit counts for CALs of lengths ℓ and ℓ' , then

$$n + n' = \begin{cases} (\ell + \ell' - 4)/2, & \ell' \text{ even,} \\ (\ell + \ell' - 3)/2, & \ell' \text{ odd} \end{cases} \quad (17)$$

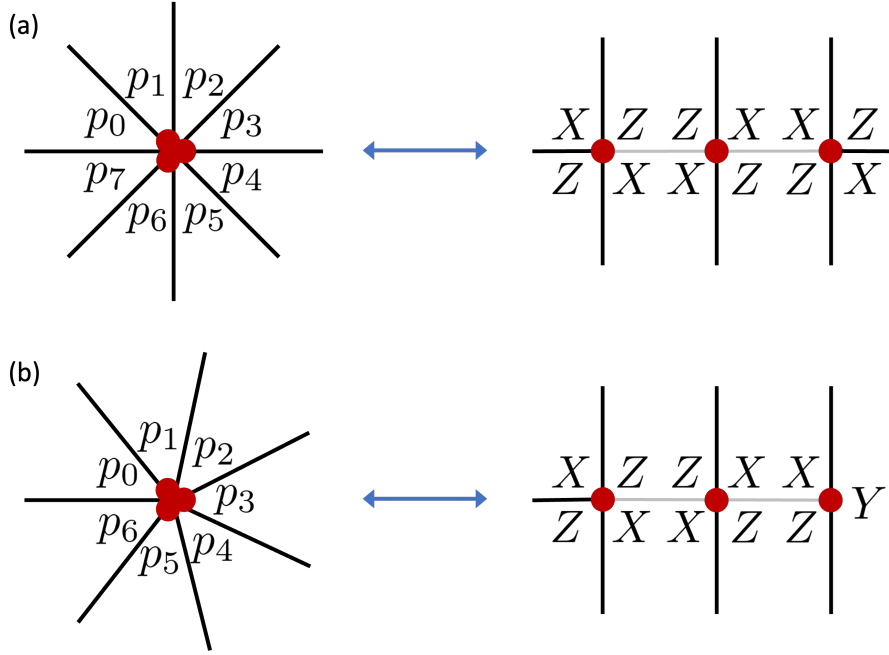


Figure 3: One way to create a CAL of length ℓ is to compose CALs of lengths three or four together. In (a), we compose three CALs with length four (right) to make a CAL with length eight (left) acting on three qubits (red circles). In (b), we use a similar composition of two length-four CALs and one length-three CAL to make a length-seven CAL. In both parts, to go from right to left, shrink the interior, light edges until all three qubits occupy the central vertex. Outgoing dark edges are meant to be identified in cyclical fashion between left and right, which in turn determines the identification of Paulis on the left and right. In terms of surface codes defined on graphs, this composition operation means that we can always decompose higher degree vertices into vertices with degrees just three or four. Still, we generally will include higher degree vertices in our drawings to highlight symmetries when possible.

is the minimum number of qubits needed for a length $\ell + \ell' - 2$ CAL. A graphical depiction of this composition operation is shown in Fig. 3. It should be clear that the construction results in CALs consisting of Paulis of weight just one or two.

There are no CALs of length ℓ acting on fewer qubits than in Eq. (15). Suppose, by way of contradiction, that a CAL $\{p_0, p_1, \dots, p_{\ell-1}\}$ acting on n qubits has length $\ell > 2n + 2$. Then, there is a set of Paulis

$$\left\{ a_j = \prod_{k=0}^j p_k : j \in \{0, 1, \dots, \ell - 2\} \right\} \quad (18)$$

with size $> 2n + 1$ in which all Paulis anticommute with one another. However, this contradicts known results on the maximum size of such anticommuting sets [38]. \square

CALs can be thoroughly studied in their own right. The interested reader can find much more discussion of CALs in Appendix C. For instance, we show some structural properties of extremal CALs, such as the following lemmas which we find useful (see Corollary C.10 and Corollary C.11).

Lemma 3.5. *If $\{p_0, p_1, \dots, p_{\ell-1}\}$ is an extremal CAL of length $\ell \geq 3$ acting on n qubits, then taking products of the p_i generates the entire n -qubit Pauli group (up to global phases).*

Lemma 3.6. *Suppose $\{p_0, p_1, \dots, p_{\ell-1}\}$ is an extremal CAL of length $\ell \geq 3$. Then $\prod_{j=0}^{\ell-1} p_j \propto I$ and, if ℓ is even, $\prod_{j=0}^{\ell/2-1} p_{2j} \propto I$ and $\prod_{j=0}^{\ell/2-1} p_{2j+1} \propto I$. Moreover, these are the only products that are proportional to identity.*

These lemmas lead to the following simple corollary.

Corollary 3.7. *Let $C = \{p_0, p_1, \dots, p_{\ell-1}\}$ be an extremal CAL of length $\ell \geq 3$ acting on n qubits. For any Pauli q acting on n qubits, define the (row) vector $C_q \in \mathbb{F}_2^\ell$ whose j^{th} element is 1 if and only if q and p_j anticommute. If ℓ is odd define $M^\top = (1 \ 1 \ \dots \ 1 \ 1) \in \mathbb{F}_2^{1 \times \ell}$ and if ℓ is even $M^\top = (\begin{smallmatrix} 1 & 1 & \dots & 1 & 1 \\ 1 & 0 & \dots & 1 & 0 \end{smallmatrix}) \in \mathbb{F}_2^{2 \times \ell}$. Then $C_q M = \vec{0}$. Moreover, if $x \in \mathbb{F}_2^\ell$ satisfies $xM = \vec{0}$, then there exists a (unique, up to phase) Pauli q such that $x = C_q$. Finally, $C_q = \vec{0}$ if and only if $q \propto I$.*

Proof. Lemma 3.6 implies that the last one (two) Paulis of a CAL are not independent from the others if ℓ is odd (even) and that these are the only dependencies. These dependencies result in $C_q M = 0$. Lemma 3.5 implies that the first $\ell - 1$ ($\ell - 2$) independent Paulis of the CAL generate the entire n -qubit Pauli group. Therefore, we are guaranteed the existence of a (unique, up to phase) Pauli q satisfying the commutation relations expressed by the first $\ell - 1$ ($\ell - 2$) bits of x and, since $xM = \vec{0}$, the final one (two) bits of x must be consistent with those commutations. Finally, $C_q = \vec{0}$ implies q commutes with the entire Pauli group, which is possible if and only if $q \propto I$. \square

With CALs forming the basis for the logical space of the Majorana code $\mathcal{S}_{\text{even}}$ we can complete our construction of qubit surface codes.

Definition 3.3. Suppose we have an embedded graph G described by some rotation system $R = (H, \lambda, \rho, \tau)$ and the vertices of G are at least degree three. At vertex v , place $N_v = \lceil (\deg(v) - 2)/2 \rceil$ qubits, and associate a Hermitian Pauli $q_{[h]_\rho}$ to each sector $[h]_\rho$ around that vertex. These Paulis $\{q_{[h]_\rho}, q_{[\tau\rho h]_\rho}, \dots, q_{[(\tau\rho)^{\deg(v)-1}h]_\rho}\}$ should form an extremal CAL of length $\deg(v)$ acting on the N_v qubits at v . To every face f of the graph we associate a stabilizer, the product of all Paulis associated to sectors $[h]_\rho \subseteq f$.

By the discussion starting this subsection, this definition ensures the codespaces of the Majorana code associated to G (Definition 3.1) and the qubit surface code associated to G (Definition 3.3) are the same, both encoding K qubits from Theorem 3.3, leading to the following corollary, which is formally proved in Appendix E.

Corollary 3.8. *A qubit surface code encodes*

$$K = \left\{ \begin{array}{ll} 2g & , \text{ orientable} \\ g & , \text{ non-orientable} \end{array} \right\} + \left\{ \begin{array}{ll} 0 & , \text{ checkerboardable} \\ (M - 2)/2 & , \text{ not checkerboardable} \end{array} \right\} \quad (19)$$

qubits, where conditions in brackets are properties of the rotation system (H, λ, ρ, τ) , or equivalently the graph embedding, defining the code.

Fig. 4, the qubit versions of the Majorana codes from Fig. 2, should help in understanding the correspondence of Majorana and qubit surface codes. Further examples of qubit surface codes fitting our definition and appearing in prior literature are shown in Fig. 5.

Notice that while a Majorana surface code has stabilizers associated to both vertices and faces, a qubit surface code only has stabilizers associated to faces. This is because we introduced just the right number of qubits at each vertex to automatically enforce the vertex stabilizers of the Majorana surface code. Alternatively, we introduced just enough physical qubits to span the codespaces of the Majorana codes $\mathcal{S}_{\text{even}}$ that exist at each vertex. This fact can also be illustrated with a simple qubit count – with one qubit for every two Majoranas (e.g. using the Jordan-Wigner transformation, Eq. (8)) we would have $(2|E| + M)/2 = |E| + M/2$ qubits (recall M is the number of odd-degree vertices), but instead we have used

$$N = \sum_{v \in V} N_v = M/2 + \sum_{v \in V} (\deg(v) - 2)/2 = |E| - |V| + M/2 \quad (20)$$

qubits. Each of the $|V|$ “missing” qubits is a degree of freedom eliminated by enforcing a stabilizer S_v .

For stabilizer code with stabilizer group \mathcal{S} , the *logical operators* are the elements of the centralizer $\mathcal{C}(\mathcal{S})$, i.e. the group of all Pauli operators that commute with all elements of \mathcal{S} . The centralizer includes the stabilizer group itself, and so we refer to elements of \mathcal{S} as trivial logical operators as

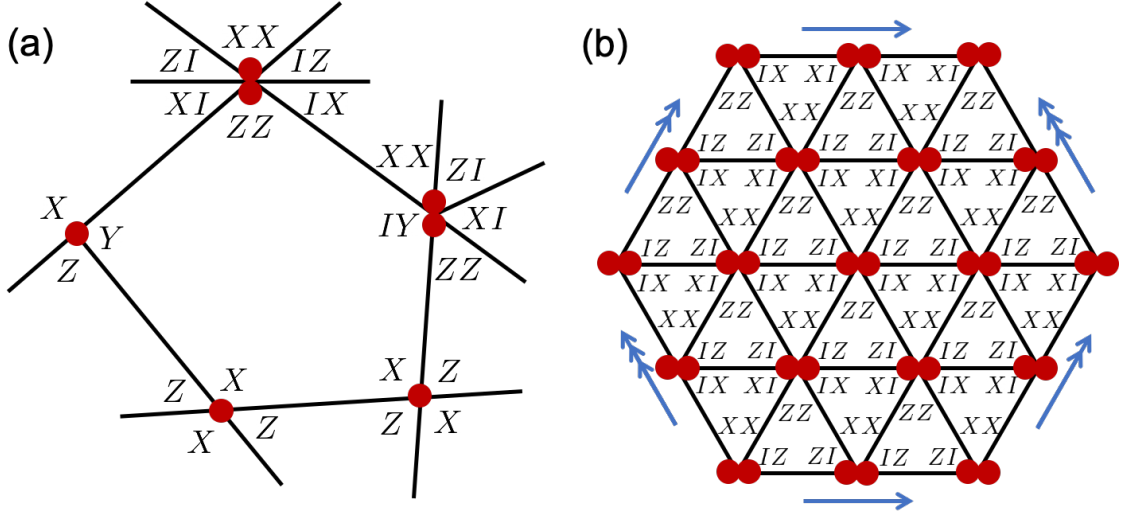


Figure 4: Qubit versions of the Majorana codes in Fig. 2. Place a single qubit on vertices with degrees three or four and a pair of qubits on vertices with degrees five or six. Around each vertex write a cyclically anticommuting list of Paulis (acting on qubits at that vertex). Each face represents a stabilizer defined as the product of all Paulis written within that face. Note that (b) depicts a toric code with $X^{\otimes 4}$ and $Z^{\otimes 4}$ stabilizers but on a lattice different from Kitaev's square lattice [1]. In this case, the code is $[[24, 2, 4]]$. The number of encoded qubits is clear by Corollary 3.8 as this graph embedding is checkerboardable.

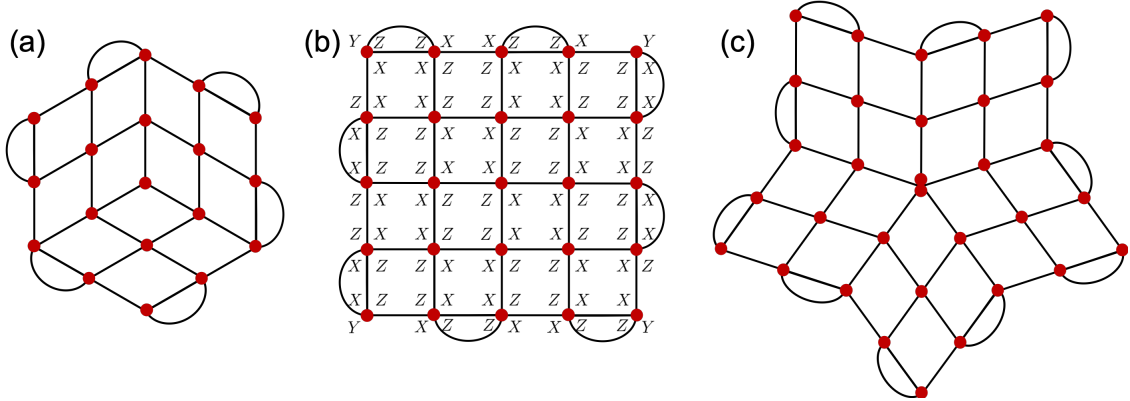


Figure 5: (a) The triangular surface code [6], (b) the rotated surface code [3] and (c) a stellated surface code [9] with even greater symmetry. Each code fits in our framework – in this case, by Definition 3.3 applied to these planar graphs. In (b), we show explicitly the assignment of CALs that gives the familiar rotated surface code. Notice that the outer face also gets assigned a stabilizer. Because of Lemma 3.2(e), the product of stabilizers on all non-outer faces is proportional to the stabilizer on the outer face.

they apply the logical identity to encoded qubits. The code distance D of a stabilizer code is the minimum weight of an element of $\mathcal{C}(\mathcal{S}) \setminus \{\{iI\} \cup \mathcal{S}\}$. We use the notation $[[N, K, D]]$ to concisely present the code parameters of a stabilizer code.

To further demonstrate Definition 3.3, in Fig. 6 we show a family of topological codes that includes the well-known $[[5, 1, 3]]$ code as its smallest member. Also included as a subset are the cyclic codes given in Example 11 and Figure 3 of [19]. The general construction gives cyclic codes defined on the torus. Consider the typical fundamental square for the torus, i.e. a unit square with opposite sides identified, and draw the lines $y = bx/a$ and $y = -ax/b$, where we assume $b > a \geq 1$ are integers and $\gcd(a, b) = 1$. These two lines intersect at $N = a^2 + b^2$ points within the fundamental square, including the single point at $(0, 0) = (0, 1) = (1, 0) = (1, 1)$. Interpret these intersections as vertices of a 4-regular graph with edges the line segments between them. With correct choice of CALs, so that each face stabilizer has two X s and two Z s (see Fig. 6 for the convention we use),

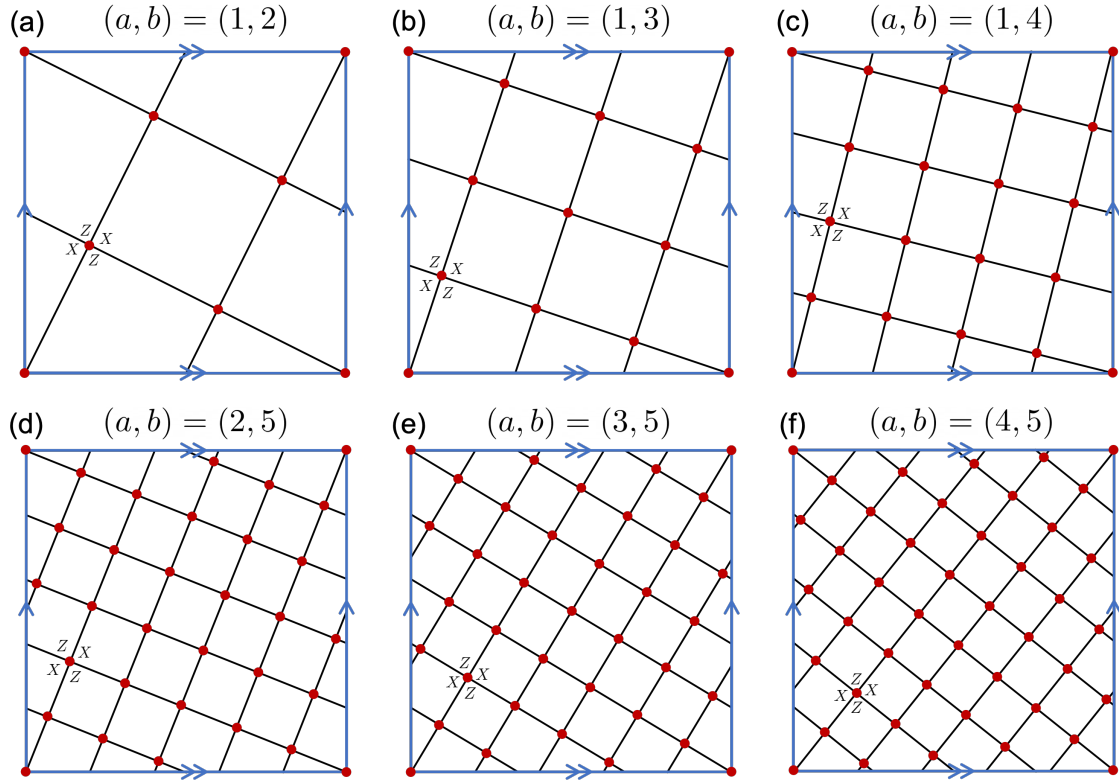


Figure 6: (a) The smallest error-correcting quantum code, the $[[5, 1, 3]]$ code, is a surface code defined by an embedding of the 5-vertex complete graph K_5 in the torus. There are different embeddings of K_5 in the torus, but no other gives a 5-qubit code with distance more than two. Parts (b-f) show members of the cyclic toric code family with more qubits. See Theorem 3.9 and the paragraph above it for the code definition and parameters. A demonstrative CAL is written around one vertex in each graph. The codes are cyclic when this same CAL is orientated similarly around each vertex.

and with qubits labeled in a sequence along the line $y = bx/a$ (moving outwards from the origin in the direction of increasing x), the surface code defined by this graph and Definition 3.3 is cyclic — an overcomplete generating set of the stabilizer group consists of $Z \otimes X \otimes I^{\otimes s} \otimes X \otimes Z \otimes I^{N-s-4}$ and its cyclic permutations, where $s = \lceil Nt/b \rceil - 2$ and t is the minimal positive integer such that $(ta + 1)/b$ is an integer.² We have the following parameters for these codes.

Theorem 3.9. *The cyclic toric code with integer parameters a and b , where without loss of generality $\gcd(a, b) = 1$ and $b > a \geq 1$, is a $[[N = a^2 + b^2, K, D]]$ code, with*

(a) *If N is odd, $K = 1$ and $D = a + b$,*

(b) *If N is even, $K = 2$ and $D = b$.*

The odd and even cases differ in the number of encoded qubits because of checkerboardability (the latter case being checkerboardable and the former not). Note that a consequence of the theorem is that the code distance D is always odd. The codes in Example 11 and Figure 3 of [19] are cyclic toric codes with $a = b - 1$.

We delay the proof of this theorem, in particular the code distances, until Section 5.2, after we have developed sufficient tools in Section 4. We point out that in two situations, the cyclic toric codes achieve $N = \frac{1}{2}KD^2 + O(D)$ — when $a = b - 1 = (D - 1)/2$ and when $b = D$ is odd with $a = 1$. Moreover, $1/2$ is the best constant achievable for the cyclic toric codes.

²The extended Euclidean algorithm on (a, b) can be used to determine t . A Bézout pair (x, y) of (a, b) is a pair of integers such that $xa + yb = \gcd(a, b)$. In particular, $(x, y) = (-t, (ta + 1)/b)$ is the unique Bézout pair of (a, b) in which $-b \leq x \leq 0$.

To conclude this section, we point out a corollary on the structure of qubit surface codes that follows from the proof of Theorem 3.4, and in particular the vertex decomposition demonstrated in Fig. 3. We use this corollary later to justify considering only graphs of low degree.

Corollary 3.10. *For any graph G , there is a choice of CALs such that the surface code associated with G is the same as the code associated with another graph G' that only has vertices with degrees three and four.*

3.4 Relation to homological surface codes

Finally, we want to point out the relation of Definition 3.3 with the well-known homological construction of surface codes [1, 39, 40]. In the homological construction, a single qubit is placed on each edge of a graph embedding $G = (V, E, F)$. Stabilizers correspond to each vertex $v \in V$ and face $f \in F$:

$$S'_v = \prod_{e \in v} X_e, \quad S'_f = \prod_{e \in f} Z_e, \quad (21)$$

where $e \in v$ and $e \in f$ are shorthand for saying edge e is adjacent to the vertex v or face f , and of course X_e, Z_e are Paulis on the qubit on that edge.

The *medial graph* \tilde{G} of an embedded graph $G = (V, E, F)$ has a vertex for each edge of G and connects them with edges such that vertices and faces of G form faces of \tilde{G} . See some examples in Fig. 7. It should be clear that the homological surface code defined on graph G is exactly the same code as our surface code, Definition 3.3, defined on the medial graph \tilde{G} . Thus, Definition 3.3 includes all homological codes.

We can formally define the medial graph using rotation systems.

Definition 3.4. The *medial rotation system* $\tilde{R} = (\tilde{H}, \tilde{\lambda}, \tilde{\rho}, \tilde{\tau})$ of a rotation system $R = (H, \lambda, \rho, \tau)$ is defined by setting $\tilde{H} = H \times \{1, -1\}$ and for $(h, j) \in \tilde{H}$ letting $\tilde{\tau}(h, j) = (h, -j)$, $\tilde{\lambda}(h, j) = (\rho(h), j)$, and

$$\tilde{\rho}(h, j) = \begin{cases} (\lambda(h), j), & \text{if } j = -1 \\ (\tau(h), j), & \text{if } j = 1 \end{cases}. \quad (22)$$

Some facts about medial graphs become apparent (we prove these in Appendix G). First, the medial graph embeds into the same manifold as the original graph. Second, medial graphs are also always 4-valent (every vertex is degree four). Third, medial graphs are always checkerboardable – e.g. a natural way to color faces in \tilde{G} is to color black faces that correspond to vertices of G and color white faces corresponding to faces of G . In fact, an embedded graph is the medial graph of another embedded graph if and only if it is 4-valent and checkerboardable. As a result, there are codes resulting from Definition 3.3 that cannot be described as homological codes.

The reader may also refer to [3] where checkerboardable, 4-valent graphs (i.e. medial graphs) are used to define homological surface codes directly and to [41] where this connection between homological codes and medial graphs is also pointed out.

Of course, forays have already been made beyond the strict homological surface code construction. We have already seen some examples in the forms of the triangular, rotated, and stellated codes in Fig. 5. Our goal however is a more rigorous framework that generalizes these specific instances.

One way in which the homological surface code construction can be rigorously generalized to produce more general planar codes is described by Freedman and Meyer [40]. Let us describe this construction using the language of medial graphs. Start with a homological surface code defined on an improper embedding of a graph in orientable surface of genus g . By an improper embedding, we mean one face is not an open disk but instead a g -punctured disk, e.g. if $g = 1$ the g -punctured face is a cylinder. Now, form the medial graph of this improperly embedded graph³. The medial graph will also be improperly embedded, but it is still 4-valent and checkerboardable. Because of the

³We defined the medial graph of an embedded graph. In this case, the graph is not properly embedded. When we construct the medial graph here, we do so by drawing proper, disk-like faces around each vertex.

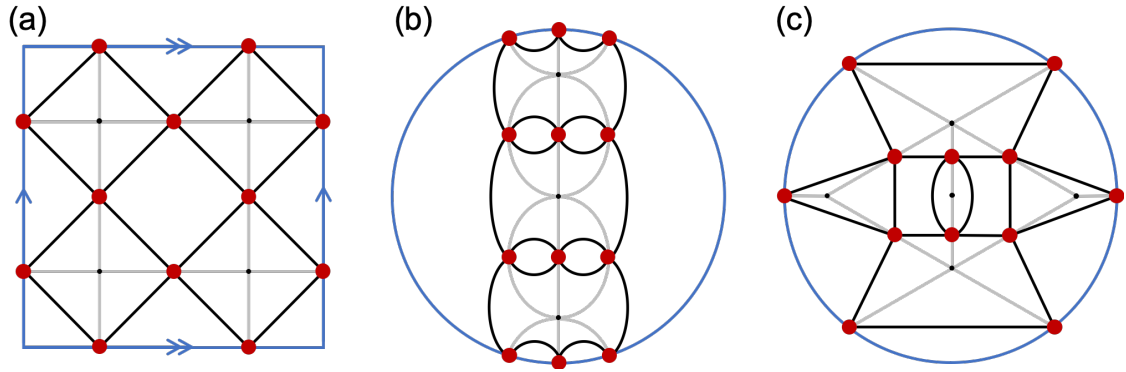


Figure 7: Graphs (gray edges with small, black vertices) on the torus (a) and projective plane, (b) and (c), are drawn with their medial graphs (black edges and large, red vertices). In (a), we get a $[[8, 2, 2]]$ toric code. In (b), we get Shor's $[[9, 1, 3]]$ code [42] with the gray graph coming from [40]. In (c), we show another $[[9, 1, 3]]$ code from [40].

redundancy in stabilizers (see proof of Theorem 3.3), we can delete, i.e. remove from the stabilizer group, any single black as well as any single white face without changing the code. Thus, delete the face that is g -punctured (say it is colored white), a black face sharing an adjacent edge, and any edges shared between them. The result is a planar graph with some degree-3 vertices (because of the deletion of edges). Examples producing the rotated surface code and the Bravyi-Kitaev planar code [39] are shown in Fig. 8.

Notably, this improper embedding procedure does not produce non-checkerboardable graph embeddings in surface of larger genus $g > 0$, so it does not produce the cyclic toric codes in Fig. 6(a,c,d,f) for instance. We have also not found a way to produce stellated codes with odd symmetry, e.g. parts (a) and (c) of Fig. 5, using the improper embedding technique, the apparent obstacle being the presence of an odd degree vertex in the center of the planar graph, i.e. not adjacent to the outer face. This appears an obstacle because the outer face and the odd degree vertices are effectively created by deleting faces and their shared adjacent edges from the improper embedding, with the result being that any odd degree vertices are adjacent to the outer face. Hence, we find this construction involving improper embeddings and face deletions to be more complicated than our construction while also producing fewer codes.

4 Locating logical operators

In this section, we discuss the structure of logical operators of the qubit surface codes of Definition 3.3. One of our main findings is that logical operators are exactly the homologically non-trivial cycles in a related graph called the decoding graph. This is probably not a huge surprise to someone familiar with homological codes, but there are some technical hurdles to overcome in proving it in our generalized setting. This includes the fact that the decoding graph for non-checkerboardable codes is not guaranteed to embed in the same manifold \mathcal{M} as the original graph. Without a graph embedding, there is no notion of homology, so finding an appropriate embedding of the decoding graph is essential to the proof.

In Section 4.1 we generalize checkerboardability to include *defects*, useful in the subsequent sections. In Section 4.2 we define the decoding graph, show that cycles in the decoding graph correspond to logical operators of the code (both trivial and non-trivial), and prove that it embeds in \mathcal{M} in the checkerboardable case. In Section 4.3, we define a manifold which is generally of higher genus than \mathcal{M} , in which we can embed the decoding graph, and prove that homological non-triviality in this new manifold is equivalent to logical non-triviality. Finally, in Section 4.5, we discuss how some logical operators, including all trivial ones, can be represented by paths in the original graph.

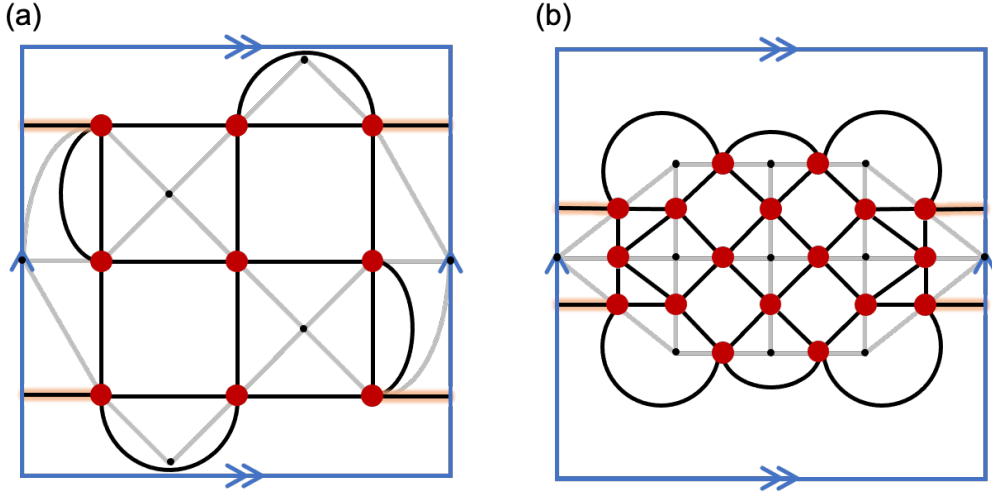


Figure 8: Improper embeddings of graphs (gray), and their medial graphs (black), on the torus. In both pictures, the face (of the medial graph embedding) intersecting the top and bottom frames is not an open disk but instead a cylinder. Deleting this face, the face intersecting the left and right frames, and their shared adjacent edges (highlighted) results in familiar planar codes. In (a) we find the $[[9, 1, 3]]$ rotated surface code and in (b) Bravyi and Kitaev's planar code [39], in this instance $[[18, 1, 3]]$.

4.1 Checkerboardability with defects

In Definition 2.2 we defined what it means for a graph embedding to be checkerboardable. In this section, we generalize this notion to include defects. Informally put, a defect is a set of edges that can be deleted from the graph embedding to make it checkerboardable. Defects have a few uses in finding or identifying logical operators in surface codes which are covered in the subsequent subsections.

To define checkerboardability with defects more rigorously, we use the matrix $\Phi \in \mathbb{F}_2^{|F| \times |E|}$ from Section 2.3, which encodes the edges belonging to each face of the embedding.

Definition 4.1. We say a graph embedding $G = (V, E, F)$ is checkerboardable with defect $\delta \in \mathbb{F}_2^{|E|}$ if there is a vector $x \in \mathbb{F}_2^{|F|}$ such that $x\Phi = \vec{1} + \delta$.

Clearly, being checkerboardable according to Definition 2.2 is equivalent to being checkerboardable with the trivial defect $\delta = \vec{0}$. The analogous lemma to Lemma 2.2 is as follows.

Lemma 4.1. G , described by rotation system $R = (H, \lambda, \rho, \tau)$, is checkerboardable with a defect if and only if we can partition H into two sets H_w and H_b such that λ and ρ map both sets to themselves, while τ maps elements of either set to the other set, except when those flags are part of an edge in the defect.

A given embedded graph does not have a unique defect. However, different choices of defect are related by the addition of rows of Φ . That is, by simple application of the definition, the following is true.

Lemma 4.2. Suppose a graph embedding $G = (V, E, F)$ is checkerboardable with defect δ_1 . Then, G is checkerboardable with defect δ_2 as well if and only if there exists x such that $x\Phi = \delta_1 + \delta_2$.

Moreover, it is an interesting fact that a defect cannot be just any subset of edges. In fact, a defect must be a sum of certain *trails* in the graph. Recall that $\mathcal{T}(G) \cong \mathbb{F}_2^{|E|}$ is the group of all trails in the graph. A subgroup of $\mathcal{T}(G)$ is generated by just the closed trails and trails whose endpoints are odd-degree vertices. We call this subgroup $\mathcal{T}_0(G)$.

Lemma 4.3. *Suppose the graph embedding $G = (V, E, F)$ is checkerboardable with defect $\delta \in \mathbb{F}_2^{|E|}$. Then, $\delta \in \mathcal{T}_0(G)$. Let $a_v = \sum_{e \ni v} \delta_e$ evaluated over \mathbb{F}_2 indicate the parity of the number of edges that are adjacent to $v \in V$ and in the defect. Then, $a_v = 1$ if and only if $\deg(v)$ is odd.*

Proof. The way that δ can fail to be in $\mathcal{T}_0(G)$ is if $a_v = 1$ for some even-degree vertex v . However, if this were the case, it would imply that G is not checkerboardable with defect δ because it fails to be so locally around vertex v . We would need to two-color faces around v such that faces on opposite sides of an edge $e \ni v$ are different colors if $\delta_e = 0$ and the same color if $\delta_e = 1$. This is impossible for an even-degree vertex with $a_v = 1$. This shows $\delta \in \mathcal{T}_0(G)$. The remaining claim, that $a_v = 1$ for odd-degree vertices v follows from a similar argument of colorability in the local neighborhood of v . \square

Finally, we point out that for any graph embedding G it is efficient (polynomial time in graph size) to perform two related tasks: (1) determine some $\delta \in \mathbb{F}_2^{|E|}$ such that G is checkerboardable with defect δ and (2) given a candidate $\delta \in \mathbb{F}_2^{|E|}$ determine whether G is checkerboardable with defect δ . We present an algorithm in Algorithm 1 that can perform both tasks via a greedy strategy. The inputs are, first, a representation of the faces in the graph embedding (to be concrete, we use the binary vectors $\phi_i \in \mathbb{F}_2^{|E|}$, $i = 1, 2, \dots, |F|$, that are the rows of Φ) and, second, a candidate defect δ . The outputs are a list of faces colored black, a list of those colored white, and a defect. The algorithm starts by arbitrarily coloring one face and continues by coloring faces that neighbor colored faces appropriately (the same color if they neighbor across a defect edge and differently otherwise). If ever a face cannot be colored either color without contradiction, edges are added or removed from the defect to make it work.

It is clear this algorithm solves task (1). To solve (2), we note the algorithm guarantees that the output defect γ is the same as the input δ if and only if δ is a defect. This solves (2). In the course of the algorithm, each edge is considered at most once in the while loop, the body of which takes $O(\max_i |\phi_i|)$ time using appropriate data structures,⁴ leading to a total time complexity of just $O(|E| \max_i |\phi_i|)$ for both tasks (1) and (2).

Algorithm 1 Returns checkerboard colors for each face and a list of edges in the defect

```

1: procedure CHECKERBOARD( $\{\phi_1, \phi_2, \dots, \phi_{|F|}\} \in \left(\mathbb{F}_2^{|E|}\right)^{|F|}$ ,  $\delta \in \mathbb{F}_2^{|E|}$ )
2:   Set  $B \leftarrow \{1\}$ ,  $W \leftarrow \emptyset$  # Sets of faces colored black and white
3:   Set  $\gamma \leftarrow \delta$  # This is a record of the defect edges (vector in  $\mathbb{F}_2^{|E|}$ )
4:   Set  $b \leftarrow \phi_1$  # This is a list of “boundary” edges of the faces already colored (vector in  $\mathbb{F}_2^{|E|}$ )
5:   For any edge  $e$  that is internal to a face (same face on both sides), set  $\gamma_e \leftarrow 1$ .
6:   While  $b$  is not  $\vec{0}$ :
7:     Let  $e$  be the first edge such that  $b_e = 1$ 
8:     If  $e$  is adjacent to a face  $f \notin B \cup W$  and a face  $g \in B \cup W$ :
9:       If  $g \in W$  and  $\gamma_e = 0$ , append  $f$  to  $B$ .
10:      If  $g \in B$  and  $\gamma_e = 1$ , append  $f$  to  $B$ .
11:      If  $f \notin B$ , append  $f$  to  $W$ .
12:      Set  $b_{e'} = 1$  for any  $e'$  that borders both  $f$  and a face  $f' \notin B \cup W$ .
13:     else:
14:       If the two faces adjacent to  $e$  are both in  $B$  or both in  $W$ , set  $\gamma_e \leftarrow 1$ . Else, set  $\gamma_e \leftarrow 0$ .
15:       Set  $b_e = 0$ 
16:   Return  $B, W, \gamma$ 

```

⁴In particular, one needs adjacency oracles that take a part of the graph embedding (e.g. a face) and return other adjacent parts (e.g. the adjacent edges). These oracles or similar are standard in the field of computational graph theory – for instance, the “doubly-connected edge list” data structure used in [43].

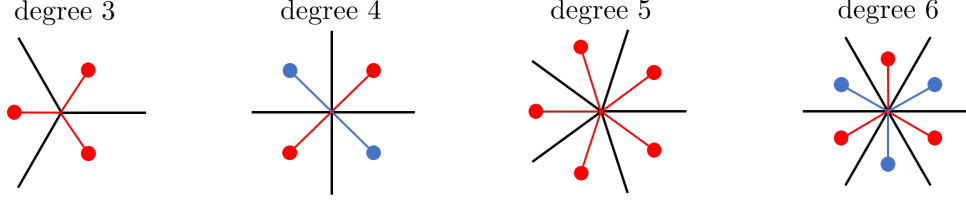


Figure 9: The decoding graph (red/blue) pictured locally around vertices of degrees 3 through 6. In odd-degree cases, all vertices in the surrounding faces connect to the central vertex. In the even-degree cases, there are two distinct central vertices, red and blue (not shown), and faces connect to these in alternating fashion.

4.2 Characterizing logical operators as cycles in the decoding graph

Recall that $\mathcal{C}(\mathcal{S})$ is the group of logical operators of the stabilizer code with stabilizer \mathcal{S} . We characterize $\mathcal{C}(\mathcal{S})$ for our surface codes as the cycles in a related graph, referred to here as the decoding graph.

Definition 4.2. Given an embedded graph $G = (V, E, F)$, an associated decoding graph G_{dec} is a (unembedded) graph with vertices V_{dec} of three types: (1) a vertex u_f associated to each face of G , (2) a vertex $w_v^{(0)}$ associated to each odd-degree vertex of G , and (3) two vertices $w_v^{(1)}$ and $w_v^{(-1)}$ associated to each even-degree vertex of G . For each sector $[h]_\rho$ of G , there is an edge $e_{[h]_\rho}$ in G_{dec} . Choose $v \in V$ and $f \in F$ such that $h \in v$ and $h \in f$. If v is odd-degree, $e_{[h]_\rho}$ connects vertices $w_v^{(0)}$ and u_f . If v is even-degree, $e_{[h]_\rho}$ connects vertices u_f to $w_v^{(j)}$ where j is such that the edges associated to the adjacent sectors, i.e. $e_{[\tau h]_\rho}$ and $e_{[\tau \rho h]_\rho}$, are incident to $w_v^{(-j)}$. All decoding graphs of G are equivalent up to relabeling vertices $w_v^{(1)}$ and $w_v^{(-1)}$, so we just speak of *the* decoding graph of G .

In Fig. 9, we draw the decoding graph locally around vertices of small degree. Our definition is slightly different from the decoding graphs commonly used to decode surface codes (e.g. [44]) in that we introduce vertices at the vertices of G as well as the faces. One reason we do this is so we can make precise statements about embedding the decoding graph or its connected components into various manifolds. A consequence of this decision however is that the decoding graph's systole is doubled compared to a definition not including vertices at the vertices of G . The reader knowledgeable in topological quantum error correction should not be concerned by the appearance of a factor of $1/2$ when we calculate the code distance in terms of this systole.

Consider now the problem of relating cycles in G_{dec} to elements of $\mathcal{C}(\mathcal{S})$. Notice that the decoding graph is bipartite, with vertices associated to faces of G (type 1 in the Definition) on one side and vertices associated to vertices of G (types 2 and 3) on the other. Therefore, any cycle in G_{dec} is even-length and can be broken into paths of consecutive edges $(e_{[h]_\rho}, e_{[k]_\rho})$, where h, k are flags in the same vertex v but not in the same sector. We let $\mathcal{T}_2(G_{\text{dec}}) \leq \mathcal{T}(G_{\text{dec}})$ denote the group generated by two-edge paths of G_{dec} .

Define a map $\sigma : \mathcal{T}_2(G_{\text{dec}}) \rightarrow \mathcal{C}(\mathcal{S})$ that translates each two-edge path $(e_{[h]_\rho}, e_{[k]_\rho})$ to a Pauli acting on qubits at the vertex v . This Pauli is chosen to anti-commute with only two elements, $q_{[h]_\rho}$ and $q_{[k]_\rho}$, of the CAL associated with v that defines the surface code on G (see Definition 3.3). This Pauli exists and is unique (up to phase) by Corollary 3.7.

Now we state some consequences of Corollary 3.7 for σ . First, if we ignore phases on the Paulis, σ is actually surjective – every Pauli up to phase is represented by some collection of two-edge paths in G_{dec} – and, second, σ is a homomorphism – for all $t_1, t_2 \in \mathcal{T}_2(G_{\text{dec}})$, we have $\sigma(t_1 + t_2) = \sigma(t_1)\sigma(t_2)$. Finally, $\sigma(t) \propto I$ if and only if t is the empty set.

By surjectivity of σ , if we have a logical operator $l \in \mathcal{C}(\mathcal{S})$, then it can be represented by a set of edges $E_l \subseteq E_{\text{dec}}$ in the decoding graph. Because the logical operator must commute with all stabilizers S_f , the number of edges from E_l incident to any vertex u_f of the decoding graph (i.e. a face in the original graph) must be even. Therefore, E_l is a sum of cycles in the decoding graph. The converse is also clearly true – any cycle c in the decoding graph maps to a logical operator

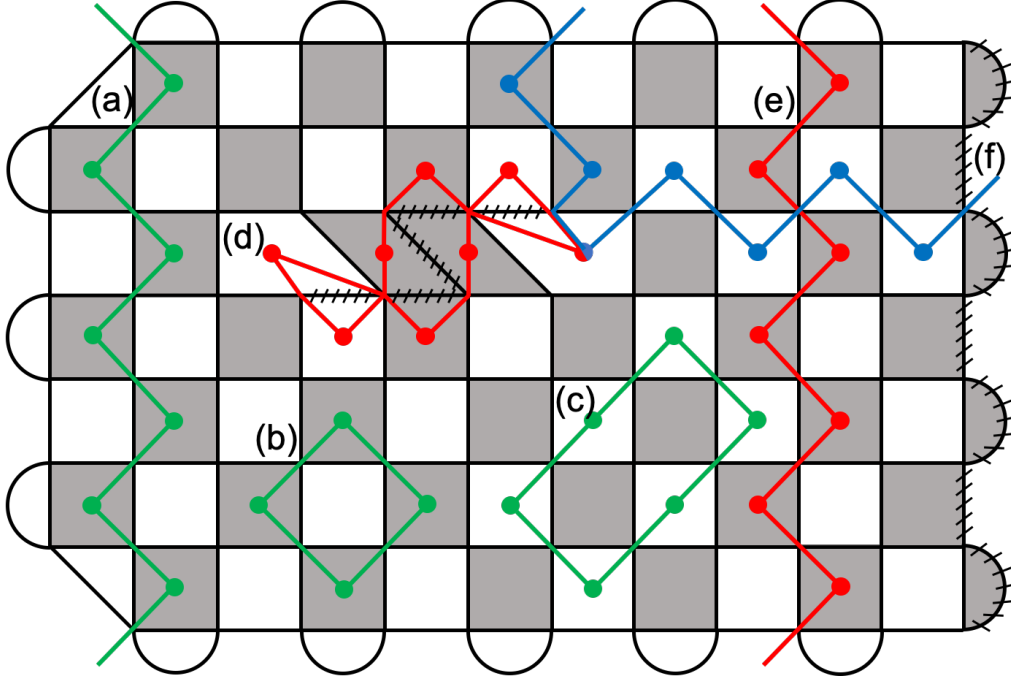


Figure 10: Example cycles in a decoding graph. The original graph has black edges, and stitched edges are part of a (particular choice of) defect. Faces are checkerboardable with this defect (the outer face should be colored black but is not shown). The entire decoding graph is not shown, but cycles labeled (a), (b), (c), (d), (e), and (f) illustrate some of it. Cycles (a), (b), (c) in green represent trivial logical operators (i.e. stabilizers), while cycles (d) and (e) in red and cycle (f) in blue represent non-trivial logical operators. We know they are non-trivial because (d) and (e) anticommute with (f). Cycles (a), (e), and (f) connect to the vertex in the outer face (not shown) via their dangling edges.

$\sigma(c) \in \mathcal{C}(S)$. Thus, we have a characterization of the logical operators of our codes.

Theorem 4.4. *Let $G = (V, E, F)$ be an embedded graph. Cycles in the decoding graph of G (Definition 4.2) represent all logical operators of the qubit surface code corresponding to G (Definition 3.3) via the map σ .*

See Fig. 10 for some examples of cycles in a decoding graph and their correspondence with logical operators.

In checkerboardable codes, we can say even more about locating nontrivial logical operators by relating them to the homology of the surface. For this to work, we must embed the decoding graph of the checkerboardable code into the same surface.

Lemma 4.5. *Suppose G is a checkerboardable graph embedded in a 2-manifold \mathcal{M} . Then the decoding graph G_{dec} consists of exactly two connected components, each of which can be embedded in the manifold \mathcal{M} .*

Proof. Suppose G is described by the rotation system (H, λ, ρ, τ) and, because it is checkerboardable, there is a partition $H = H_w \sqcup H_b$ as in Lemma 2.2. Of course, each vertex of G is necessarily even-degree. It should be clear from Fig. 9 that the decoding graph G_{dec} has two connected components, one corresponding to faces that are subsets of H_w and another corresponding to faces that are subsets of H_b .

Let $G_{\text{dec},w}$ and $G_{\text{dec},b}$ be the connected components of G_{dec} . These are described by rotation

systems $R_{\text{dec},w} = (H_w \times \{\pm 1\}, \lambda_{\text{dec}}, \rho_{\text{dec}}, \tau_{\text{dec}})$ and $R_{\text{dec},b} = (H_b \times \{\pm 1\}, \lambda_{\text{dec}}, \rho_{\text{dec}}, \tau_{\text{dec}})$, where

$$\lambda_{\text{dec}}(h, j) = (h, -j), \quad (23)$$

$$\rho_{\text{dec}}(h, j) = \begin{cases} (\tau\rho\tau(h), j), & j = -1 \\ (\lambda(h), j), & j = 1 \end{cases}, \quad (24)$$

$$\tau_{\text{dec}}(h, j) = (\rho(h), j). \quad (25)$$

This provides an explicit embedding of the components of G_{dec} .

We do still need to show that they are indeed embedded in the manifold \mathcal{M} and not some other one. Suppose G has sets of vertices, edges, faces $V, E, F = F_w \sqcup F_b$ (with this partition due to checkerboardability) and $G_{\text{dec},w}$ has sets $V_{\text{dec}}, E_{\text{dec}},$ and F_{dec} . One can check from the rotation system, Eqs. (23-25), that $|V_{\text{dec}}| = |F_w| + |V|$, $|E_{\text{dec}}| = \frac{1}{2} \sum_{v \in V} \deg(v) = |E|$, and $|F_{\text{dec}}| = |F_b|$. This implies the Euler characteristics of R and $R_{\text{dec},w}$ are the same.

We also need to show that $R_{\text{dec},w}$ is orientable if and only if R is. Note R being orientable means one can partition H into two sets $H_{\pm 1}$ such that λ, ρ, τ applied to an element of one set maps it to an element of the other set. Define $H'_{\pm 1} \subseteq H' := H_w \times \{\pm 1\}$ such that $(h, j) \in H'_{\text{dec},k}$ if and only if $h \in H_{jk}$. These sets clearly partition H' and $\lambda_{\text{dec}}, \rho_{\text{dec}},$ and τ_{dec} all map either set to the other. So R_{dec} is orientable. The other direction – if R_{dec} is orientable, then R is – follows a very similar argument that we leave for the reader. \square

Here we comment on the case where a checkerboardable graph G has only degree-4 vertices. In this case, the decoding graph has degree-2 vertices associated to the vertices of G . Suppose we remove these vertices, joining the edges incident to them into one edge. It is not hard to see that this vertex removal makes the connected components of G_{dec} into graphs Q and \bar{Q} (duals of one another) where the medial graph \tilde{Q} is equal to G . Therefore, in this checkerboardable and 4-valent case, the operation of building the decoding graph essentially inverts the operation of building the medial graph. Then Q is a graph that defines a homological code (see Section 3.4) equivalent to the code we have defined on G . For an illustration, refer back to Fig. 7: in each part of that figure, the gray edged graph is one connected component of the decoding graph of the black edged graph.

With the decoding graph embedded, we can relate homological and logical nontriviality in the checkerboardable case.

Theorem 4.6. *For checkerboardable graph G , a cycle c in G_{dec} is homologically nontrivial if and only if $\sigma(c)$ is a nontrivial logical operator of the surface code associated with G .*

Proof. The key fact to use is that a Pauli p is a stabilizer generator associated to a face in G if and only if $p \propto \sigma(c)$ for some facial cycle c in G_{dec} . In Fig. 11, we show this correspondence explicitly. We use this fact to complete the proof.

Let c be a homologically trivial cycle in G_{dec} . We need to show $\sigma(c)$ is a stabilizer. Since G_{dec} has two connected components, $c = c_w + c_b$ where c_w is a cycle in $G_{\text{dec},w}$ and c_b a cycle in $G_{\text{dec},b}$. Since $\sigma(c) = \sigma(c_w)\sigma(c_b)$, we just show $\sigma(c_w)$ is a stabilizer with an analogous argument for $\sigma(c_b)$. Since c_w is homologically trivial it is a sum of facial cycles $c_w = c_1 + c_2 + \dots + c_l$ and $\sigma(c_w) = \sigma(c_1)\sigma(c_2) \dots \sigma(c_l)$. By the fact in the previous paragraph, $\sigma(c)$ is indeed a stabilizer.

Likewise, if Pauli p is a stabilizer, then it is the product of stabilizer generators associated to faces of G , or $p = p_1 p_2 \dots p_l$. By fact (1), there are facial cycles c_1, c_2, \dots, c_l in G_{dec} such that $\sigma(c_j) = p_j$ and therefore a homologically trivial cycle $c = c_1 + c_2 + \dots + c_l$ so that $\sigma(c) = p$. \square

In a slight abuse of notation, we let $\text{hsys}(G_{\text{dec}})$ be the smaller of the homological systoles of the two connected components of G_{dec} embedded as in Lemma 4.5.

Corollary 4.7. *Suppose G is a checkerboardable, embedded graph and D is the code distance of the surface code defined by G . Then, $\frac{1}{2} \text{hsys}(G_{\text{dec}}) \leq D$ and if G is 4-valent, $\frac{1}{2} \text{hsys}(G_{\text{dec}}) = D$.*

Proof. For the proof of the bound, note that any non-trivial logical operator with weight D is represented by a homologically non-trivial cycle c confined to a single connected component of

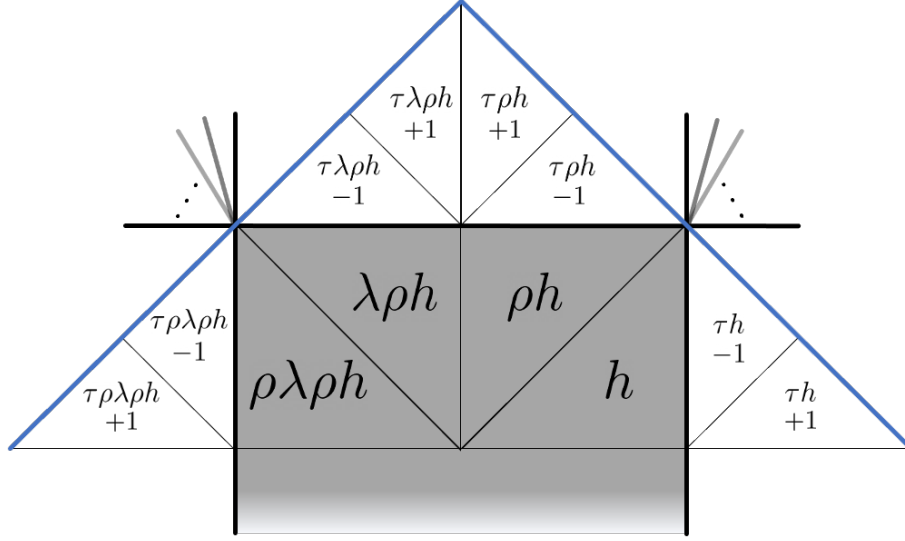


Figure 11: Picturing the correspondence between a face f of checkerboardable graph G (thick, black edges) and a face f' of a connected component of its decoding graph G_{dec} (blue edges). Notation comes from the proof of Lemma 4.5. Flags of G and G_{dec} are outlined by thin lines. Flags of G making up f include h , ρh , $\lambda\rho h$, and $\rho\lambda\rho h$. These each relate to two flags in the corresponding face f' . For instance, $h \in f$ becomes $(\tau h, \pm 1) \in f'$. If f is colored black in G , then f' is a face of $G_{\text{dec},w}$.

the decoding graph G_{dec} . Suppose c visits $n \leq D$ vertices of G_{dec} that are associated to vertices of G . There is a homologically non-trivial sub-cycle of c , call it c' , that visits each of the n vertices at most once and so has at most $2n$ edges. Since $\text{hsys}(G_{\text{dec}})$ lower bounds the length of c' , $\text{hsys}(G_{\text{dec}}) \leq 2n \leq 2D$.

Now suppose G is 4-valent. The shortest homologically non-trivial cycle c in G_{dec} is confined to just one connected component. By 4-valency, there is only one qubit at each vertex and each two-edge sub-path in c is mapped to single-qubit Pauli by σ . Since c has length $\text{hsys}(G_{\text{dec}})$, the weight of the non-trivial logical Pauli it represents is at most $\frac{1}{2}\text{hsys}(G_{\text{dec}})$, which therefore is an upper bound on D . \square

The equality statement in this corollary does not hold for graphs with higher degree vertices because, in that case, two-edge paths in the decoding graph may represent Paulis of weight greater than one. However, if one uses the CAL construction in Theorem 3.4 to assign Paulis to sectors, one can effectively decompose these higher-degree vertices into degree-4 vertices (see Corollary 3.10) and apply Corollary 4.7 to the resulting 4-valent graph.

4.3 Doubled graphs

Our goal in this section is to find some manifold to embed G_{dec} into when the original graph G is not checkerboardable. In general, this is a manifold with genus larger than the original surface. In particular, for any non-checkerboardable graph G embedded on a manifold \mathcal{M} , we show that there is a checkerboardable graph G^2 , referred to as a *doubled* graph, embedded on manifold \mathcal{M}^2 such that G_{dec} is isomorphic to any one of the two connected components of G_{dec}^2 . A similar doubled construction is given by Barkeshli and Freedman [45] in the context of topological phases.

Informally, the doubled manifold \mathcal{M}^2 is created by taking two copies of \mathcal{M} , cutting along the edges of a defect of the embedded graph G , and gluing the two copies together. This gluing is done in such a way that in crossing from one copy of \mathcal{M} to the other, a path ends up at the same place in the destination manifold as it would have in the original manifold if it were still intact. This means, for instance, that a defect trail between two odd degree vertices is cut open into a disk and glued with its corresponding disk on the other copy. In contrast, cycles in the defect that are

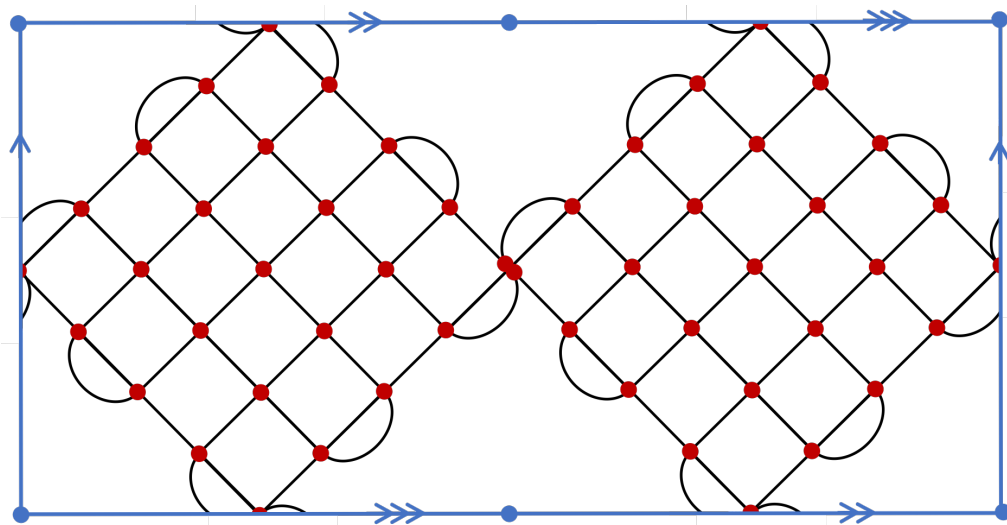


Figure 12: A doubled graph for the rotated surface code of Fig. 5(b) is embedded in the torus (six-sided blue border).

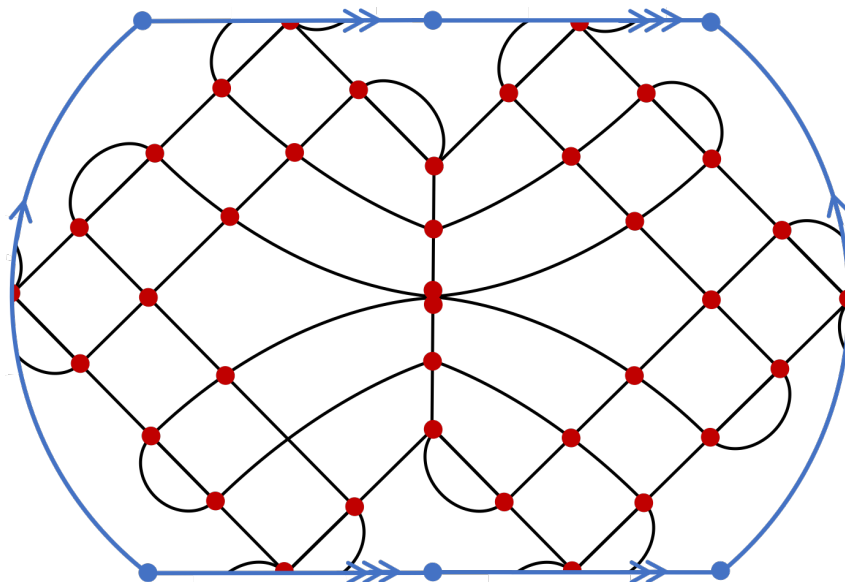


Figure 13: A doubled graph for the triangle code of Fig. 5(a) is embedded in the torus (six-sided blue border).

homologically non-trivial result in the handles or cross-caps of the surface being cut apart before they are glued with their copies. We provide a formal construction of the doubled graph in terms of rotation systems and a construction of the doubled manifold subsequently. We comment that the doubled graph G^2 depends on the choice of defect, but the topology of \mathcal{M}^2 does not.

Definition 4.3. The doubled rotation system $R^2 = (H', \lambda', \rho', \tau')$ of a rotation system $R = (H, \lambda, \rho, \tau)$ with defect δ is defined by setting $H' = H \times \{1, -1\}$ and for $(h, j) \in H'$ letting $\lambda'(h, j) = (\lambda(h), j)$, $\rho'(h, j) = (\rho(h), j)$, and

$$\tau'(h, j) = \begin{cases} (\tau(h), -j), & \exists e \in \delta \text{ s.t. } h \in e \\ (\tau(h), j), & \text{otherwise} \end{cases}. \quad (26)$$

One can study orbits of $\langle \lambda', \rho' \rangle$ to relate faces of G^2 with faces of G . Since neither λ' nor ρ' acting on $(h, j) \in H'$ flips the sign of j , there are two faces in G^2 for every face of G , one made up of flags with $j = 1$ and one with flags $j = -1$. It turns out we can checkerboard G^2 such that these two faces are oppositely colored.

Lemma 4.8. *For any embedded graph G , the doubled graph G^2 is checkerboardable.*

Proof. Let δ be the defect used to define G^2 from G , and let $R = (H, \lambda, \rho, \tau)$ be the rotation system of G . As G is checkerboardable with defect δ , there is a partition of $H = H_w \sqcup H_b$, where λ and ρ map either set to itself and τ maps an element h from either set to the other *except* when h is in an edge belonging to the defect (Lemma 4.1).

Now we can partition $H' = H \times \{1, -1\}$, the set of flags for the doubled rotation system $R^2 = (H', \lambda', \rho', \tau')$, into two sets

$$H'_w = \{(h, j) : h \in H_w, j = 1 \text{ or } h \in H_b, j = -1\}, \quad (27)$$

$$H'_b = \{(h, j) : h \in H_b, j = 1 \text{ or } h \in H_w, j = -1\}. \quad (28)$$

One can check that λ' and ρ' map either set to itself and τ' maps between the sets. Thus, G^2 is checkerboardable. \square

Just as there are two faces in G^2 for every face in G , it follows from Definition 4.3 that G^2 has two edges for every edge of G , two degree- $\deg(v)$ vertices for every even-degree vertex v of G , and one degree- $2\deg(v)$ vertex for every odd-degree vertex v of G . This counting allows us to establish the genus of G^2 .

Theorem 4.9. *Let G be a non-checkerboardable graph containing M odd-degree vertices.*

(a) *If G embeds into a genus g , orientable manifold, then G^2 embeds into a genus $2g + (M - 2)/2$, orientable manifold.*

(b) *Suppose G embeds into a genus g , non-orientable manifold.*

(i) *If G^2 embeds into an orientable manifold, its genus is $g + (M - 2)/2$.*

(ii) *If G^2 embeds into a non-orientable manifold, its genus is $2(g + (M - 2)/2)$.*

Proof. Part of this theorem that is implicit is that if G is orientable, then G^2 is orientable. Represent G and G^2 with rotation systems as in Definition 4.3. Then, G being orientable means there is a partition of $H = H_{+1} \sqcup H_{-1}$ such that λ , ρ , and τ all map elements of either set $H_{\pm 1}$ to elements of the other. Clearly, $H' = H'_{+1} \sqcup H'_{-1}$ can also be partitioned so that $(h, j) \in H'_k$ if and only if $h \in H_k$. As required for orientability, λ' , ρ' , and τ' all map elements of either set $H'_{\pm 1}$ to elements of the other.

The genus of G^2 can be obtained by counting. As discussed previously, the doubled graph has $|F'| = 2|F|$ faces, $|E'| = 2|E|$ edges, and $|V'| = 2|V| - M$ vertices where $|F|$, $|E|$, and $|V|$ are the face, edge, and vertex counts of G . The conclusions follow. \square

Because G^2 is checkerboardable, its decoding graph G^2_{dec} has two connected components, each embedding in the same manifold as G^2 .

Theorem 4.10. *Either connected component of G_{dec}^2 is isomorphic to G_{dec} . Moreover, homologically non-trivial cycles in G_{dec}^2 correspond to non-trivial logical operators of the surface code associated to G . If that code has distance D , then $\frac{1}{4}\text{hsys}(G_{\text{dec}}^2) \leq D \leq \frac{1}{2}\text{hsys}(G_{\text{dec}}^2)$.*

Proof. We describe the isomorphism by describing how vertices of G_{dec} and $G_{\text{dec},w}^2$ (one of the two connected components of G_{dec}^2) are mapped to each other. There are three maps that should be composed in the right way, the map from G to G_{dec} , the map from G to G^2 , and the map from G^2 to $G_{\text{dec},w}^2$.

First, note $G_{\text{dec},w}^2$ has a single vertex for each white face of G^2 (which is checkerboardable). Also, $G_{\text{dec},w}^2$ has a single vertex for each vertex of G^2 . Next, recall G_{dec} has a vertex for each face of G , two vertices for each even-degree vertex of G , and one vertex for each odd-degree vertex of G . Finally, G^2 has two (differently colored) faces for each face of G , two vertices for each even-degree vertex of G , and one vertex for each odd-degree vertex of G . With the vertices of G_{dec} and $G_{\text{dec},w}^2$ associated to one another by the implied map, vertex adjacency is also preserved, showing the isomorphism.

We argued in Lemma 4.8 that every face of G corresponds to two oppositely colored faces of G^2 . Since $G_{\text{dec},w}^2$ has faces corresponding to black faces of G^2 , these actually represent all the faces of G . Therefore, a cycle in $G_{\text{dec},w}^2$ is homologically non-trivial if and only if it is a non-trivial logical operator of the code. The upper bound on the code distance follows immediately.

The lower bound is a result of two vertices in G^2 representing each even-degree vertex of G . If a homologically non-trivial cycle in $G_{\text{dec},w}^2$ crosses both of these vertices, it may represent only one qubit of support in the corresponding logical operator. Thus, we have an additional factor of $1/2$ compared to the upper bound. \square

With the theory above, we have seen how the doubled manifold effectively describes the homology of nontrivial logical operators in a higher genus surface. One consequence for surface codes is that for every non-checkerboardable code defined by graph G with parameters $[[N, K, D]]$, there is a checkerboardable code defined by graph G^2 with parameters $[[2N, 2K, D']]$ where $D \leq D' \leq 2D$. The checkerboardable code has the same rate and at least as good a code distance, but with potentially higher weight stabilizers and larger genus. Thus, non-checkerboardable codes can offer improvements over checkerboardable codes given that stabilizer weight and genus are relevant parameters for practical implementations.

4.4 Face-width as a lower-bound on code distance

The *face-width* $\text{fw}(G)$ of an embedded graph G (with genus $g > 0$) is the minimum number of times any non-contractible cycle drawn on the manifold intersects the graph [33]. This can also be defined as the length of the shortest non-contractible cycle in a related graph, the face-vertex graph G_{fv} .

Definition 4.4. The face-vertex graph G_{fv} of an embedded graph G is an embedded graph possessing a vertex w_v for each vertex v of G and a vertex u_f for each face f of G . An edge is drawn between w_v and u_f if and only if v and f are adjacent in G . To specify the embedding, if G has rotation system (H, λ, ρ, τ) , the face-vertex graph has rotation system $(H_{\text{fv}}, \lambda_{\text{fv}}, \rho_{\text{fv}}, \tau_{\text{fv}})$, where $H_{\text{fv}} = H \times \{1, -1\}$ and

$$\lambda_{\text{fv}}(h, j) = (h, -j), \quad \rho_{\text{fv}}(h, j) = \begin{cases} (\tau(h), j), & j = -1 \\ (\lambda(h), j), & j = 1 \end{cases}, \quad \tau_{\text{fv}}(h, j) = (\rho(h), j). \quad (29)$$

We point out some easily-proved facts about the face-vertex graph. First, it is bipartite since vertices of type w_v are only ever connected to vertices of type u_f . Second, the face-vertex graph is the dual of the medial graph (Definition 3.4), $\overline{\overline{G}} = G_{\text{fv}}$. Third, the face-vertex graph is embedded in the same manifold as G . Fourth, if we recall that a graph G and its dual \overline{G} have isomorphic medial graphs, we also find that G and \overline{G} have isomorphic face-vertex graphs. Finally, the face-vertex

graph is the decoding graph (see Def. 4.2) with the vertices $w_v^{(\pm 1)}$ merged at each even-degree vertex v .

Because G_{fv} is bipartite, $\text{sys}(G_{\text{fv}})$ is even. Any cycle drawn on the manifold is homeomorphic to a cycle in the face-vertex graph. Hence, it is easy to see that $\text{fw}(G) = \frac{1}{2}\text{sys}(G_{\text{fv}})$. This is the most convenient definition of face-width for our purposes.

The rest of this subsection is devoted to proving the following.

Theorem 4.11. *Suppose G is an embedded graph with only even-degree vertices and genus $g > 0$. Let D be the code distance of the surface code associated to G . Then $\text{fw}(G) \leq D$.*

To prove this theorem, we need a simple lemma, which we prove separately.

Lemma 4.12. *Suppose G is a non-checkerboardable graph with only even-degree vertices embedded in manifold \mathcal{M} with genus $g > 0$. Then the doubled graph G^2 is embedded in a manifold \mathcal{M}^2 that is a double cover of \mathcal{M} .*

Proof. To create the manifold \mathcal{M}^2 from the rotation system description in Def. 4.3 (whose notation we import here), we glue the flags together according to λ', ρ', τ' . Suppose we map flags in $H' = H \times \{1, -1\}$ to flags in H , i.e. $\Pi(h, j) = h$ for $h \in H$ and $j = \pm 1$. This map extends to a map $\pi : \mathcal{M}^2 \rightarrow \mathcal{M}$ by simply mapping points in the flag (h, j) to points in flag h in homeomorphic fashion, i.e. if both (h, j) and h are viewed as identically-sized triangular patches of surface with boundaries associated to the λ, ρ , and τ involutions, then map the former to latter point-by-point. Note that $\Pi \circ \lambda' = \lambda \circ \Pi$, $\Pi \circ \rho' = \rho \circ \Pi$, and $\Pi \circ \tau' = \tau \circ \Pi$. This establishes that if two flags are glued together in \mathcal{M}^2 , their images are glued together in \mathcal{M} .

We should check that for all open disks U , $\pi^{-1}(U)$ is the union of two open disks, thus proving it is a double cover. This is clearly true for open disks contained entirely within a single flag h . We also look at open disks that straddle two flags (across a flag's edge) and open disks that straddle more than two flags (because they contain a flag's corner).

Suppose an open disk straddles two flags $h, \lambda h \in H$. Applying Π^{-1} to this pair of flags, we get either $(h, 1)$ and $\lambda'(h, 1) = (\lambda h, 1)$ or $(h, -1)$ and $\lambda'(h, -1) = (\lambda h, -1)$. Thus, π^{-1} gives two open disks, each straddling one of these pairs of flags in \mathcal{M}^2 .

The same argument holds for flags $h, \rho h$. Slightly more interesting is the case of flags $h, \tau h$. In this case, if the edge of the graph G containing h is in the defect, Π^{-1} gives either $(h, 1)$ and $(\tau h, -1)$ or $(h, -1)$ and $(\tau h, 1)$. If the edge is not in the defect, then the result is $(h, 1)$ and $(\tau h, 1)$ or $(h, -1)$ and $(\tau h, -1)$.

Finally, corners of the flags correspond to vertices, edges, and faces of the graphs. An open disk at the corner $\{h, \rho h, \lambda h, \rho \lambda h\}$, a face of the graph, is the most straightforward case since no defect is involved. Here Π^{-1} maps the face of G to two faces of G^2 , namely $\{(h, 1), (\rho h, 1), (\lambda h, 1), (\rho \lambda h, 1)\}$ and $\{(h, -1), (\rho h, -1), (\lambda h, -1), (\rho \lambda h, -1)\}$. Likewise, it is not hard to see that an edge (containing h) in G maps to two edges (the one containing $(h, 1)$ and the one containing $(h, -1)$) in G^2 , whether or not that edge is part of the defect. For the case of a vertex v in G , one must recall the fact that an even number of edges incident to v are part of the defect, Lemma 4.3. This is the only place where we are using that G has only even-degree vertices. Suppose $h \in v$. The vertices in G^2 that map to v under Π are the one containing $(h, 1)$ and the one containing $(h, -1)$, and these are distinct because of the aforementioned fact. \square

Proof of Theorem 4.11. If G is checkerboardable, then notice that $\text{fw}(G) = \frac{1}{2}\text{sys}(G_{\text{fv}}) \leq \frac{1}{2}\text{sys}(G_{\text{dec}}) \leq \frac{1}{2}\text{hsys}(G_{\text{dec}}) = D$, with the final equality following from Corollary 4.7.

Thus, we are left with the non-checkerboardable case. Lemma 4.12 says the doubled manifold \mathcal{M}^2 is a double cover of the manifold \mathcal{M} on which G is embedded. This means that the universal cover of \mathcal{M} (which is the plane $\mathcal{U} = \mathbb{R}^2$) is also the universal cover of \mathcal{M}^2 . This implies the existence of three covering maps, the projections $\pi_{21} : \mathcal{M}^2 \rightarrow \mathcal{M}$, $\pi_{U2} : \mathcal{U} \rightarrow \mathcal{M}^2$, and $\pi_{U1} : \mathcal{U} \rightarrow \mathcal{M}$ such that $\pi_{U1} = \pi_{21}\pi_{U2}$.

Now, choose a logical operator represented by some homologically non-trivial cycle c_2 (in the

decoding graph G_{dec}^2) drawn on \mathcal{M}^2 . Any lift $\pi_{U_2}^{-1}(c_2)$ is necessarily a non-closed curve in \mathcal{U} . We also see that $c_1 = \pi_{21}(c_2)$, which is easily seen to be homeomorphic to a cycle in G_{fv} , also lifts to $\pi_{U_2}^{-1}(c_2) = \pi_{U_1}^{-1}\pi_{21}(c_2)$, and therefore c_1 is non-contractible.

Finally, there must be some simple sub-cycle of c_1 , call it c'_1 , that is also non-contractible. It visits exactly $\frac{1}{2}\text{len}(c'_1)$ vertices and so this is at least the Pauli weight w of the logical operator it represents, $w \geq \frac{1}{2}\text{len}(c'_1) \geq \text{fw}(G)$. Since our arguments apply to all choices of c_2 (representing all non-trivial logical operators), the distance of the code is also lower bounded by $\text{fw}(G)$. \square

4.5 Logical operators from trails in the original graph

In this section, we provide another upper bound on the code distance of surface codes, complementing the bounds in Theorem 4.10. This bound comes from identifying logical operators with certain trails in the graph G that defines the surface code. Conveniently, we do not have to work with any derived graph, such as the decoding graph, to obtain these logical operators.

Recall that $\mathcal{T}_0(G)$ is the subgroup of $\mathcal{T}(G)$ that is generated by closed trails and open trails whose endpoints are odd-degree vertices. To each trail in $\mathcal{T}_0(G)$ we assign a logical operator, or equivalently an element of $\mathcal{C}(\mathcal{S})$. We define a homomorphism $\omega : \mathcal{T}_0(G) \rightarrow \mathcal{C}(\mathcal{S})$. It is easy to define ω using the Majorana code picture from Section 3.2. Let $t \in \mathcal{T}_0(G)$. If t is a closed trail, then we define $\omega(t)$ to be the product of all Majoranas on the edges in t . If t is an open trail, its endpoints are at odd degree vertices by definition, and we define $\omega(t)$ to be the product of all Majoranas on the edges in t and the two Majoranas at the odd-degree endpoints. By the arguments in Section 3.3, these products of Majoranas are equivalent to Paulis in the qubit surface code. We do not concern ourselves with the sign of the Pauli, and so the products can be taken in arbitrary order. Moreover, the Pauli $\omega(t)$ commutes with all stabilizers by construction and so is in $\mathcal{C}(\mathcal{S})$.

What is the weight of the Pauli $\omega(t)$? If t does not visit a vertex, clearly $\omega(t)$ is not supported on qubits at that vertex. Alternatively, if t traverses all edges adjacent to a vertex, then $\omega(t)$ is also not supported on qubits at that vertex. These are the only vertices lacking support – $\omega(t)$ is supported on qubits at all other vertices. If the underlying graph has only degree 3 or 4 vertices (higher degree vertices may be decomposed with Corollary 3.10), then the weight of $\omega(t)$ is equal to the number of vertices visited exactly once by t . Paths by definition visit vertices at most once, so that for a path $p \in \mathcal{T}_0(G)$, the weight of $\omega(p)$ is exactly the number of vertices visited by p . From this discussion, we can find the kernel of ω (for our usual assumption of a connected graph G).

$$\ker \omega = \{b \in \mathcal{T}_0(G) : \omega(b) = I\} = \{\vec{0}, \vec{1}\}. \quad (30)$$

This means the map ω is 2-to-1, ignoring phases on Paulis in $\mathcal{C}(\mathcal{S})$.

When is $\omega(t)$ a non-trivial logical operator? It turns out we can efficiently check this using the notion of checkerboardability and Algorithm 1. With some abuse of notation, we call t non-trivial if $\omega(t)$ is non-trivial.

Theorem 4.13. *Let $t \in \mathcal{T}_0(G)$. Then $\omega(t) \in \mathcal{C}(\mathcal{S})$ is a stabilizer if and only if G is checkerboardable with defect t or with defect $\vec{1} + t$.*

Proof. We start with the reverse direction. We are assured the existence of $x \in \mathbb{F}_2^{|F|}$ such that $x\Phi = \vec{1} + t$ or such that $x\Phi = t$. Let Φ_i denote the i^{th} row of Φ . Because ω is a homomorphism and $\omega(\vec{1}) = I$, we have

$$\omega(t) = \omega(x\Phi) = \omega\left(\sum_{i:x_i=1} \Phi_i\right) = \prod_{i:x_i=1} \omega(\Phi_i), \quad (31)$$

which is a stabilizer because each $\omega(\Phi_i)$ is the stabilizer S_i in Eq. (11) associated to the i^{th} face.

For the forward direction of the theorem, we start with $\omega(t) = \prod_{i:x_i=1} \omega(\Phi_i)$ for some $x \in \mathbb{F}_2^{|F|}$, since the faces generate the stabilizer group by definition. By Eq. (31) once again, we get $\omega(t) = \omega(x\Phi)$. This implies $t = x\Phi + b$ where $b \in \ker \omega$. By Eq. (30), we have $\vec{1} + t = x\Phi$ or $t = x\Phi$ or, equivalently, G is checkerboardable with defect t or with defect $\vec{1} + t$. \square

To summarize the above, for any graph G embedded in a 2-manifold defining surface code with stabilizer \mathcal{S} , we have shown

$$\mathcal{S} \leq \omega(\mathcal{T}_0(G)) \leq \mathcal{C}(\mathcal{S}). \quad (32)$$

We now point out a strengthening of this result when G is planar (i.e. $g = 0$).

Theorem 4.14. *Let G be a planar graph and \mathcal{S} be the stabilizer of the surface code defined on G according to Definition 3.3. Then $\mathcal{S} \leq \omega(\mathcal{T}_0(G)) = \mathcal{C}(\mathcal{S})$ (ignoring Pauli's phases).*

Proof. If G contains no odd-degree vertices, then $K = 0$ by Theorem 3.3 and so $\mathcal{S} = \mathcal{C}(\mathcal{S}) = \tau(\mathcal{T}_0(\mathcal{S}))$.

Otherwise, G contains an even number of odd-degree vertices $M \geq 2$. Construct a spanning tree of G rooted at one of these odd-degree vertices v . There is a path p_w from v to any other odd-degree vertex w contained within the spanning tree, and $\omega(p_w) \in \mathcal{C}(\mathcal{S})$. Notice that for $w \neq w'$, $\omega(p_w)$ anticommutes with $\omega(p_{w'})$, because, written as products of Majoranas, p_w and $p_{w'}$ have odd overlap – they share edges, two Majoranas each, as well as the Majorana located at odd-degree vertex v . Thus, we have a set of $M - 1$ logical Paulis $\mathcal{B} = \{\omega(p_w) : \text{odd-degree vertex } w \neq v\}$ that mutually anticommute. This implies, because the maximum size of an anticommuting set of K -qubit Paulis is $2K + 1$ [38], that $K \geq (M - 2)/2$. However, $K = (M - 2)/2$ is exactly the number of encoded qubits by Theorem 3.3, meaning \mathcal{B} is a basis of all logical operators. In turn, this implies $\mathcal{C}(\mathcal{S}) = \mathcal{S}\mathcal{B} \leq \omega(\mathcal{T}_0(G))$, which completes the proof. \square

The rest of this section concerns the actual calculation of an element of $\mathcal{T}_0(G)$ that visits the fewest number of vertices. First, note that there are bases of $\mathcal{T}_0(G)$ that consist only of trails that do not visit any vertex more than once, i.e. bases of paths. Let the length of a path be the number of vertices it visits. The total length of a path basis is the sum of lengths of all trails in the basis. This implies the existence of a path basis that minimizes the total length of the basis, called a minimum path basis. Such a minimum path basis must also contain the shortest non-trivial path (with non-triviality determined by Theorem 4.13). If it did not, we could replace some element of the basis with this shortest non-trivial path, obtaining another path basis with lower total weight.

This discussion of minimum path bases is mirrored for bases of $\mathcal{Z}(G)$, the cycle space of the graph. Likewise, there is a minimum simple cycle basis that consists of simple cycles (cycles without repeated vertices) and minimizes their combined length. Polynomial time algorithms for finding a minimum simple cycle basis have been known since Horton's algorithm [46], which has time complexity $O(|E|^3|V|)$ for a general graph $G = (V, E)$. Better algorithms now exist, including a $O(|E|^\Omega)$ time Monte-Carlo algorithm [47] (where Ω is the exponent of matrix multiplication) and $O(|E|^2|V|)$ [48] and $O(|E|^2|V|/\log|V| + |E||V|^2)$ [49] deterministic algorithms. We can use these algorithms to also find a minimum path basis and efficiently place an upper bound on the code distance.

Theorem 4.15. *For embedded graph G , let J be the number of vertices in a non-trivial path $p \in \mathcal{T}_0(G)$ visiting the fewest vertices. Then the surface code defined by graph G has distance $D \leq J$ and J can be calculated in polynomial time in the size of graph G .*

Proof. Since $\omega(p)$ is a non-trivial logical operator with weight J , clearly J upper bounds the code distance.

It remains to show J is efficient to calculate. Construct a graph G' from G by adding edges between all pairs of odd-degree vertices, a total of $M(M - 1)/2$ new edges if there are M odd-degree vertices. Call the set of new edges E_o . Find the minimum cycle basis B of G' using one of the efficient algorithms referenced above, e.g. [46]. Each cycle $b \in B$ that traverses an edge outside E_o corresponds to a path $q \in \mathcal{T}_0(G)$. In particular, q is formed from b by removing all edges from E_o . The length of b is the number of vertices visited by q . Thus, by replacing b with q we can turn the minimum simple cycle basis B into a minimum path basis B_0 . To calculate J , we just need to determine whether each path in B_0 is non-trivial using Theorem 4.13 and take the one visiting the fewest vertices. \square

Finally, we point out that the upper bound from Theorem 4.15 and the bounds from Theorem 4.10

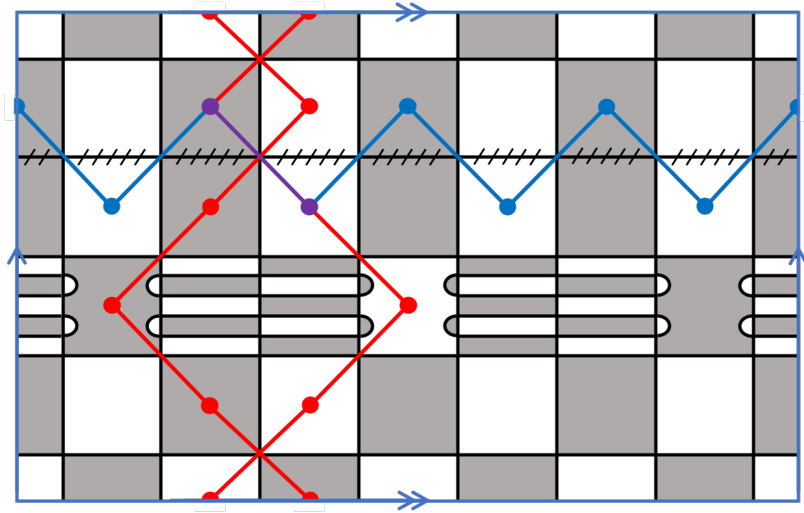


Figure 14: An example graph G in which the bounds in Theorems 4.10 and 4.15 are not saturated. The graph is largely a square lattice on the torus, but with obstructions along one row that force minimum weight logical operators to go around. We checkerboard the graph with stitched edges indicating the defect. The code encodes one logical qubit. The shortest non-trivial cycles in the decoding graph G_{dec} have length $L = 8$; an example is shown in blue (and purple where it overlaps the red cycle). The shortest non-trivial cycles in the original graph G also visit $J = 8$ vertices. Thus, Theorem 4.15 proves $4 \leq D \leq 8$. However, the distance of the code actually satisfies $D \leq 7$, as demonstrated by the red cycle in the decoding graph (which has length 10). Also, $D \geq 5$ follows from Lemma 5.1 and arguments similar to those in Section 5.1.

are not saturated in general. An example is shown in Fig. 14.

5 Code examples

In this section, we present five example code families. In the first three subsections we consider square lattice toric codes, rotated toric codes (with a subfamily of cyclic toric codes first mentioned in Section 3.3), and hyperbolic codes. For these codes we make some rigorous arguments about code parameters (especially the code distance) using the theorems from Section 4. In the final subsection, we present two more code families that generalize stellated codes [9].

5.1 Rotated toric codes

As the name suggests, this family of codes is created from simple square lattices on the torus. Up to local Cliffords, these are the codes defined by Wen [12]. Because of checkerboardability properties, the dimensions of the lattice are relevant to K as well as D . We let the lattice be $m \times n$, i.e. it takes m steps to traverse the torus in one direction and n to traverse the other direction. See Fig. 15, for two examples.

Note that a rotated toric code is only checkerboardable if both m and n are even. Therefore, by Corollary 3.8

$$K = \begin{cases} 2, & m, n \text{ are even,} \\ 1, & \text{otherwise.} \end{cases} \quad (33)$$

Since they are checkerboardable and four-valent, the $K = 2$ codes can also be described by the homological code model (see Section 3.4), while the $K = 1$ codes cannot.

What is the code distance of a $m \times n$ rotated toric code? It will surprise few to learn that it is

$$D = \min(m, n). \quad (34)$$

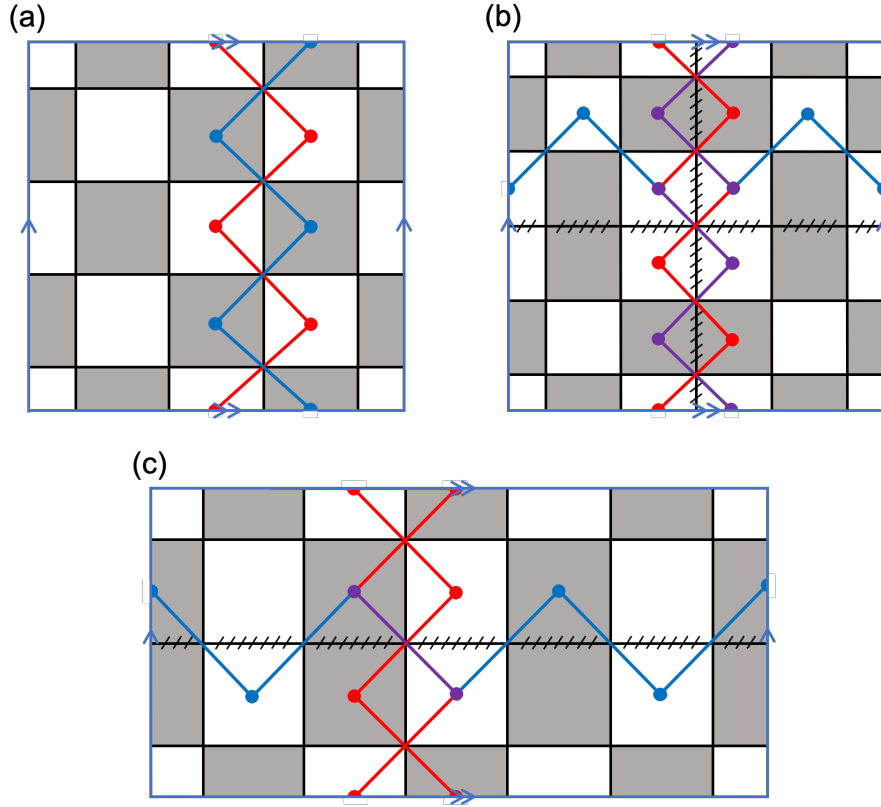


Figure 15: Three examples of rotated toric codes. (a) An even \times even code can be checkerboarded and encodes two logical qubits. We show two cycles, red and blue, in the decoding graph representing two commuting, non-trivial logical operators, e.g. \bar{X}_1 and \bar{Z}_2 . (b) An odd \times odd code can be checkerboarded with a defect (stitched edges) and encodes one logical qubit. We show two cycles, red and blue (with purple edges and vertices where they overlap), representing two anti-commuting logical operators, e.g. \bar{X} and \bar{Z} . (c) An odd \times even code is also checkerboardable with a defect and encodes one logical qubit.

However, we would like to prove this. We divide the proof into cases – even \times even, odd \times even, odd \times odd – though the proof is similar for each case. The general idea is to assume that there is a logical operator l that has weight less than $\min(m, n)$, then show that l commutes with all logical operators of the code, so it must be trivial (i.e. in the center of the group, i.e. a stabilizer). Thus, D is at least $\min(m, n)$. Separately, we can show there is some non-trivial logical operator with that weight.

The odd \times even case is slightly simpler so we start there. Say m (number of rows) is odd and n (number of columns) is even. There is one logical qubit. Any column c is a cycle, so $\omega(c)$ is a logical operator. Theorem 4.13 tells us $\omega(c)$ is a non-trivial logical operator, but we can also see this a different way. Let r be any row and r' be any cycle in the decoding graph such that the support of the logical operator $\sigma(r')$ is only within row r . Then $\sigma(r')$ and $\omega(c)$ anticommute and so are both non-trivial logical operators. Also, $\omega(c)$, $\sigma(r')$, and the stabilizers generate the entire group of logical operators, since there is only one encoded qubit. Fig. 15(c) shows an example of these two logical operators for $m = 3$, $n = 6$. Note, we can find such anticommuting pairs for any column c and row r . Therefore, if logical operator l has weight less than $\min(m, n)$, there is both a column c and a row r where it is not supported. Thus, it commutes with the entire logical group and must be trivial.

In the even \times even case, the only complication is that there are two logical qubits. Non-trivial logical operators can still be confined to a single column c and a single row r . There are two different cycles c' and c'' in the decoding graph such that support of $\sigma(c')$ and $\sigma(c'')$ lie within column c . Likewise, there are two different cycles r' and r'' . See Fig. 15(a). By their commutation relations,

we note that $\sigma(c'), \sigma(c''), \sigma(r'), \sigma(r'')$, and stabilizers generate the entire group of logical operators. Thus, as before, a logical operator l with weight less than $\min(m, n)$ will not have support in one column and row, thus commutes with all logical operators, and so must be trivial.

Finally, in the odd \times odd case, there is one logical qubit. Any column c_1 is a closed trail and $\omega(c_1)$ a non-trivial logical operator. However, the complementary logical operator is not confined to a single row. Instead, given any row r and column c_2 , there is a cycle z in the decoding graph such that $\sigma(z)$ has support only in r and c_2 . Moreover, $\omega(c_1)$ and $\sigma(z)$ anticommute no matter the row and columns chosen. Set $c_1 = c_2 = c$. See Fig. 15(b). Now the same argument from before can be applied: if logical operator l has weight less than $\min(m, n)$, then it is not supported in some column c and some row r , but then it commutes with all logical operators and is trivial.

The preceding proof technique can be summarized in a lemma that is potentially useful for lower bounding the code distance of any stabilizer code. If l is a logical Pauli operator of stabilizer code with stabilizer group \mathcal{S} , we let $D(l)$ be the minimum weight of an element of the coset $l\mathcal{S}$.

Lemma 5.1. *Let l be a logical Pauli operator for stabilizer code \mathcal{S} on N qubits and suppose there exists positive integer μ and collections $C_1, C_2, \dots, C_\mu \subseteq P(\{0, 1, \dots, N-1\})$ (where $P(S)$ denotes the power set of S) such that*

1. *For all C_i and all distinct $c, c' \in C_i$, $c \cap c' = \emptyset$,*
2. *For all choices of $c_i \in C_i$, there exists $l_0 \in l\mathcal{S}$, with $\text{supp}(l_0) \subseteq \bigcup_{i=1}^\mu c_i$.*

Then, for all logical Paulis l' that anticommute with l , $D(l') \geq \min_i |C_i|$.

Proof. Since logical operators l and l' anticommute, every element of $l\mathcal{S}$ anticommutes with every element of $l'\mathcal{S}$. Suppose, by way of contradiction, there is $l'_0 \in l'\mathcal{S}$ with weight $|l'_0| < \min_i |C_i|$. For all i , because the sets in each C_i are disjoint, there must be a set $c_i \in C_i$ such that $\text{supp}(l'_0) \cap c_i = \emptyset$. However, some $l_0 \in l\mathcal{S}$ exists such that $\text{supp}(l_0) \subseteq \bigcup_{i=1}^\mu c_i$. Therefore, $\text{supp}(l_0) \cap \text{supp}(l'_0) = \emptyset$. So l_0 and l'_0 do not anticommute, a contradiction. \square

The total distance of the code is $D = \min_{l \in \mathcal{C}(\mathcal{S}) \setminus \mathcal{S}} D(l)$. To lower bound the code distance it is sufficient to fix a basis of logical operators for a code, e.g. $\{\bar{X}_i, \bar{Z}_i\}$, and find appropriate collections satisfying the conditions of the lemma for each element of the basis.

For any rotated toric code, we need only two collections, C_1 the set of all columns and C_2 the set of all rows. With these collections, *every* non-trivial logical operator of the code satisfies the second condition of the lemma (see the constructions of these operators in Fig. 15), and we immediately get the lower bound $D \geq \min(m, n)$.

One thing to note when applying the lemma, is that because of the conditions on the collections C_i , N must be at least D^2/μ . Thus, the lemma is unlikely to be useful for codes with N scaling better than a constant times D^2 . Luckily, our codes must scale this way due to the Bravyi-Poulin-Terhal bound [2]. For instance, symmetric rotated toric codes with $m = n = D$ satisfy $N = KD^2$ if D is odd and $N = \frac{1}{2}KD^2$ if D is even. Because of the factor of $1/2$, the latter family has twice the code rate of the rotated surface code, Fig. 5(b).

5.2 General rotated toric codes

In this subsection we focus on a generalization of toric codes introduced by Kovalev and Pryadko [15] and prove Theorem 3.9. These codes are defined by two vectors, $L_1 = (a_1, b_1)$ and $L_2 = (a_2, b_2)$, where a_i, b_i are integers. Consider the infinite square lattice, and equate points $x, y \in \mathbb{R}^2$ if

$$x - y \in S(L_1, L_2) := \{m_1 L_1 + m_2 L_2 : m_1, m_2 \in \mathbb{Z}\}. \quad (35)$$

The vertices and edges of the infinite square lattice now map, using this equivalence, to vertices and edges of a finite graph $G(L_1, L_2)$ embedded on the torus. With single qubits at each vertex and stabilizers associated to faces, $G(L_1, L_2)$ defines a surface code via Def. 3.3. For instance, if

$b_1 = 0$ and $a_2 = 0$, these codes are just the square lattice toric codes discussed in the previous section. Referring back to Section 2.5, the infinite square lattice is acting as the universal cover of the graph $G(L_1, L_2)$ on the torus, with the covering map described by equating points as in Eq. (35).

Some of these rotated toric codes are equivalent despite being defined by different vectors L_1 and L_2 . The sets $S(L_1, L_2)$ and $S(L'_1, L'_2)$ are equal if and only if $L'_i = g_{i1}L_1 + g_{i2}L_2$ for some integer-valued matrix g with $\det(g) = \pm 1$ [15]. In these cases, the codes associated to graphs $G(L_1, L_2)$ and $G(L'_1, L'_2)$ are also equivalent.

Code parameters N and K are relatively obvious given our general framework. First, N can be calculated as the area of the parallelogram with sides L_1 and L_2 , or [15]

$$N = |L_1 \times L_2| = |a_1b_2 - b_1a_2|. \quad (36)$$

As also pointed out in [15], checkerboardability of the graph $G(L_1, L_2)$ depends on the parities of $\|L_1\|_1 = a_1 + b_1$ and $\|L_2\|_1 = a_2 + b_2$. A checkerboard coloring of the infinite square lattice can be mapped to a checkerboard coloring of the finite graph $G(L_1, L_2)$ if and only if, for all points $P \in \mathbb{R}^2$, the faces at points P , $P + L_1$, and $P + L_2$ are colored the same. Moreover, those points are colored the same if and only if $\|L_1\|_1$ and $\|L_2\|_1$ are both even integers. Thus, we have

$$K = \begin{cases} 1, & \|L_1\|_1 \text{ or } \|L_2\|_1 \text{ is odd,} \\ 2, & \|L_1\|_1 \text{ and } \|L_2\|_1 \text{ are even.} \end{cases} \quad (37)$$

With regards to distance D , we first relate the (non)-triviality of logical operators to the topological (non)-triviality of cycles in the decoding graph. To do this for non-checkerboardable codes, we need to use the doubled graph of Section 4.3. Even the doubled graph does not resolve the issue that cycle length in the non-checkerboardable code's decoding graph is not necessarily the Pauli weight (reflected in the gap in the upper and lower bounds of Theorem 4.10), and the next step is to resolve this.

Consider first the checkerboardable rotated toric codes as also considered in [15]. In this case, the faces of $G(L_1, L_2)$ are two-colorable, black and white. The decoding graph resolves into two connected components which are duals of one another: the black component has vertices associated to only black faces and the white component has vertices associated to only white faces. As both these connected components embed naturally onto the torus, with qubits associated to edges and stabilizers to vertices and faces, the homological description of toric codes applies [1]. We conclude that a cycle in the decoding graph is non-trivial if and only if it is homologically non-trivial, or equivalently, when mapped back to the infinite square lattice, it connects a point P with the point $P + m_1L_1 + m_2L_2$ where at least one of m_1 or m_2 is odd. Edges in the decoding graph follow the vectors $(1, 1)$, $(1, -1)$, $(-1, 1)$, or $(-1, -1)$. Therefore, to get from P to $P + m_1L_1 + m_2L_2$ takes $\|m_1L_1 + m_2L_2\|_\infty$ steps in the decoding graph. We conclude the code distance is [15]

$$D = \min_{\substack{m_1, m_2 \in \mathbb{Z} \\ (m_1, m_2) \neq (0, 0)}} \|m_1L_1 + m_2L_2\|_\infty. \quad (38)$$

We divide the non-checkerboardable case into two sub-cases. For now, assume $\|L_1\|_1$ is odd but $\|L_2\|_2$ is even. We claim that the decoding graph of $G(L_1, L_2)$, call it G_{dec} is isomorphic to either connected component of the decoding graph of the doubled graph $G(2L_1, L_2)$, a checkerboardable code. If one believes that $G(2L_1, L_2)$ is legitimately a doubled graph of $G(L_1, L_2)$, then this isomorphism follows from Theorem 4.10. However, we can be more explicit in the argument for this example.

Therefore, let us show that G_{dec} and the black component G'_{dec} of the decoding graph of $G(2L_1, L_2)$ are isomorphic. Showing it for the white component is analogous. To establish some coordinates, we imagine that graph $G(L_1, L_2)$ lies within the parallelogram P with corners $(0, 0)$, L_1 , $L_1 + L_2$, and L_2 . Similarly, the graph $G(2L_1, L_2)$ lies within the parallelogram P' with corners $(0, 0)$, $2L_1$, $2L_1 + L_2$, and L_2 . See Fig. 16(a).

Consider the graphs G_{dec} and G'_{dec} as a sets of points making up their edges and vertices. The idea is that any point p in G'_{dec} that is also in the parallelogram $P' \setminus P$ should be mapped to $p - L_1$.

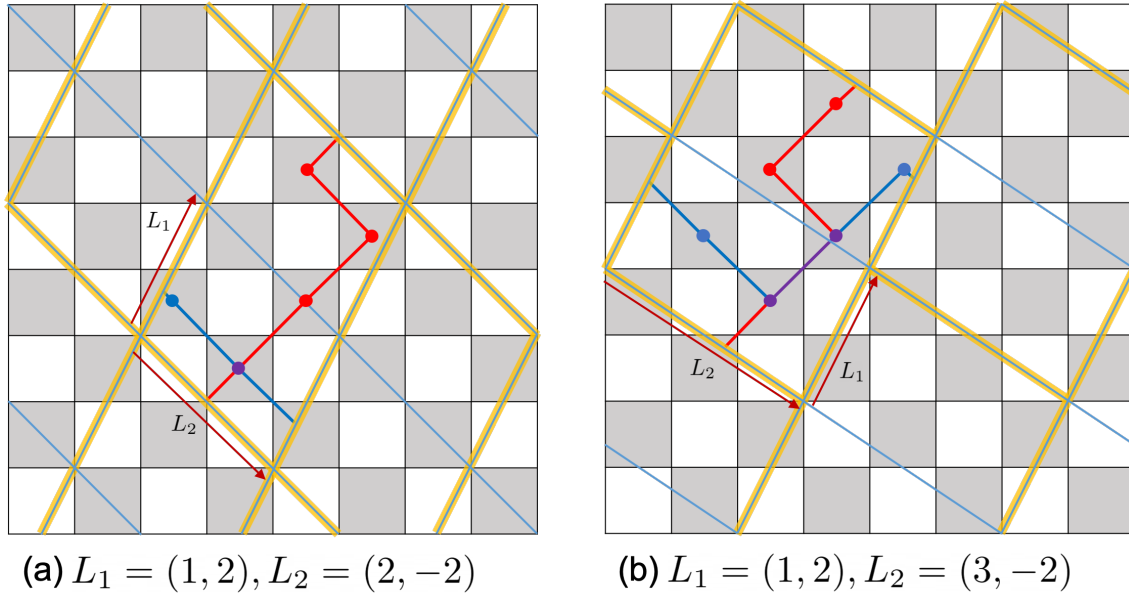


Figure 16: Regular tessellations of the infinite square lattice. Inside a single cell of a tessellation, the subgraph of the infinite square lattice defines a toric code. In fact, both images show two tessellations, one outlined in blue and one outlined in yellow. The blue tessellations have periodicity vectors L_1 and L_2 and define non-checkerboardable rotated toric codes. As argued in the text, the decoding graphs of these non-checkerboardable codes are isomorphic to a connected component of the decoding graphs of the corresponding checkerboardable codes defined by the yellow tessellations. Some paths (blue and red, purple where they overlap) in the yellow codes' decoding graphs are shown, which, when mapped to a single cell of the blue tessellation, are anticommuting logical operators.

This places it within P . It is easily seen that this map takes the points of G'_{dec} to the points of G_{dec} in one-to-one fashion and is a graph isomorphism.

The other crucial property of the map is that any cycle in G_{dec} that corresponds to single face stabilizer maps to a cycle in G'_{dec} corresponding to a single face stabilizer, and vice versa. In this way, we know that all trivial (both topologically and logically) cycles in G'_{dec} map to (logically) trivial cycles in G_{dec} . The cycle space of G'_{dec} is generated by these trivial cycles as well as two non-trivial cycles (because it is embedded on a torus). When mapped to G_{dec} , these two non-trivial cycles must be anticommuting logical operators of the code.

Now, we start the other sub-case of non-checkerboardable codes. Say both $\|L_1\|_1$ and $\|L_2\|_1$ are odd. To come up with a related checkerboardable code, we have to alter the rotated toric code construction slightly. Let $G_E(L_1, L_2)$ be the graph defined by equating points $x, y \in \mathbb{R}^2$ in the infinite square lattice such that

$$x - y \in S_E(L_1, L_2) := \{m_1 L_1 + m_2 L_2 : m_1, m_2 \in \mathbb{Z}, m_1 + m_2 \text{ is even}\}. \quad (39)$$

This corresponds to tiling the plane with offset parallelograms, or hexagons with opposite sides identified, see Fig. 16(b). Clearly, $G_E(L'_1, L'_2)$ is always a checkerboardable code for any L'_1 and L'_2 . We claim that either connected component of the decoding graph of $G_E(L_1, L_2)$ is isomorphic to the decoding graph of $G(L_1, L_2)$.

Again, let G_{dec} be the decoding graph of $G(L_1, L_2)$ and G'_{dec} be the black connected component of the decoding graph of $G_E(L_1, L_2)$. As before, we can imagine $G(L_1, L_2)$ lying within the parallelogram P and $G_E(L_1, L_2)$ lying within the parallelogram P' . The isomorphism is also the same map. Each point p in $P' \setminus P$ and in G'_{dec} gets mapped to a point $p - L_1$ in P . Once again this maps trivial cycles to trivial cycles. The two remaining generators of the cycle space of G'_{dec} , when mapped to G_{dec} , are the anticommuting logical operators of the non-checkerboardable code.

With this setup, we finish the proof of Theorem 3.9.

Proof of Theorem 3.9. The nontrivial part of Theorem 3.9 that remains is the proof of the distance. Here, in the special case of cyclic codes $L_1 = (a, b)$ and $L_2 = (-b, a)$ with $b > a > 0$ and $\gcd(a, b) = 1$. For checkerboardable codes, those in part (b) of the theorem, Eq. (38) applies and simplifies to $D = \max(a, b) = b$, as claimed.

The non-checkerboardable case, in which both $\|L_1\|_1$ and $\|L_2\|_1$ are odd, makes up the bulk of this proof. Above, we characterized the nontrivial logical operators as topologically nontrivial cycles in the decoding graph of a related checkerboardable code. We will take this a step further and change coordinates to align with the decoding graph itself. This amounts to rotating and re-scaling L_1 and L_2 .

$$L_1^{\text{dec}} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} L_1 = \left(\frac{1}{2}(b+a), \frac{1}{2}(b-a) \right), \quad (40)$$

$$L_2^{\text{dec}} = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix} L_2 = \left(-\frac{1}{2}(b-a), \frac{1}{2}(b+a) \right). \quad (41)$$

Again, points $x, y \in \mathbb{R}^2$ are equated if $x - y \in S_E(L_1^{\text{dec}}, L_2^{\text{dec}})$. It is now the decoding graph, rather than the doubled graph, that is represented by applying this equivalence to the square lattice.

We look at topologically nontrivial paths in the decoding graph. By symmetry, we just consider two cases: (1) paths from the origin $O = (0, 0)$ to $2L_1^{\text{dec}}$ and (2) paths from O to $L_1^{\text{dec}} + L_2^{\text{dec}}$. The shortest path from point $A \in \mathbb{R}^2$ to $B \in \mathbb{R}^2$ clearly has length equal to $\|B - A\|_1$. The complication is that, due to the graph isomorphism between non-checkerboardable and checkerboardable decoding graphs, edges located at points $e_1, e_2 \in \mathbb{R}^2$ act on the same qubit (one acts with Pauli X say, and the other with Pauli Z) if

$$e_2 - e_1 \in T(L_1^{\text{dec}}, L_2^{\text{dec}}) := \{m_1 L_1^{\text{dec}} + m_2 L_2^{\text{dec}} : m_1, m_2 \in \mathbb{Z}, m_1 + m_2 \text{ is odd}\}. \quad (42)$$

Thus, the Pauli weight of a path is at most $\|B - A\|_1$ but may be less. Eq. (42) implies that if e_1 is horizontal then e_2 is vertical and vice-versa. So if a path has, say, more horizontal edges than vertical edges, at most each vertical edge may be acting on the same qubit as a horizontal edge, and thus the Pauli weight is at least $\|B - A\|_\infty$. In summary, if we let $|P(A, B)|$ denote the Pauli weight of a minimum weight path from A to B , we have shown

$$\|B - A\|_\infty \leq |P(A, B)| \leq \|B - A\|_1. \quad (43)$$

Moreover, because adding an edge to a path never decreases the Pauli weight, there is a path with minimum Pauli weight that also has minimum length, namely, length $\|B - A\|_1$.

For case (1), we argue that the path with lowest Pauli weight has weight $|P(O, 2L_1^{\text{dec}})| = \|2L_1^{\text{dec}}\|_\infty = a + b$, matching the lower bound. We need only demonstrate an explicit path from O to $2L_1^{\text{dec}} = (a, b)$ with this weight. We can describe this with a string of symbols N and E representing moving north and east, respectively. So, for instance, $NENE = (NE)^2$ would mean moving north, then east, then north, then east. The path with minimum Pauli weight is then

$$(NE)^{\frac{1}{2}(b-a-1)} NE^a (EN)^{\frac{1}{2}(b-a-1)} EE^a. \quad (44)$$

Edges of opposite orientation act on the same qubit if they are separated in this sequence by $\frac{1}{2}(b+a-1)$ E edges and $\frac{1}{2}(b-a-1)$ N edges. For instance, the red edges in the sequence (44) act on the same $b-a$ qubits as the blue edges. Thus, while the sequence's length is $2b$, its Pauli weight is $2b - (b-a) = a+b$, as claimed. See Fig. 17 for an example.

For case (2), we argue that the path with lowest Pauli weight has weight $|P(O, L_1^{\text{dec}} + L_2^{\text{dec}})| = \|L_1^{\text{dec}} + L_2^{\text{dec}}\|_1 = a + b$, matching the upper bound on path weight. Therefore, we must reason that, in going from O to $L_1^{\text{dec}} + L_2^{\text{dec}} = (a, b)$ one can take no advantage of edges acting on the same qubit. According to the comment just below Eq. (43), we can restrict our investigation to minimum length paths. We argue that, for any $t \in T(L_1^{\text{dec}}, L_2^{\text{dec}})$ and any edge e , no minimum length path contains both e and $e + t$. If we show this claim for t , it implies the claim for $-t$. So assume $t = m_1 L_1^{\text{dec}} + m_2 L_2^{\text{dec}} = (t_x, t_y)$ and that, without loss of generality, $m_1 > m_2$. Note, because t associates edges of opposite orientation, both t_x and t_y are half-integers and so necessarily nonzero.

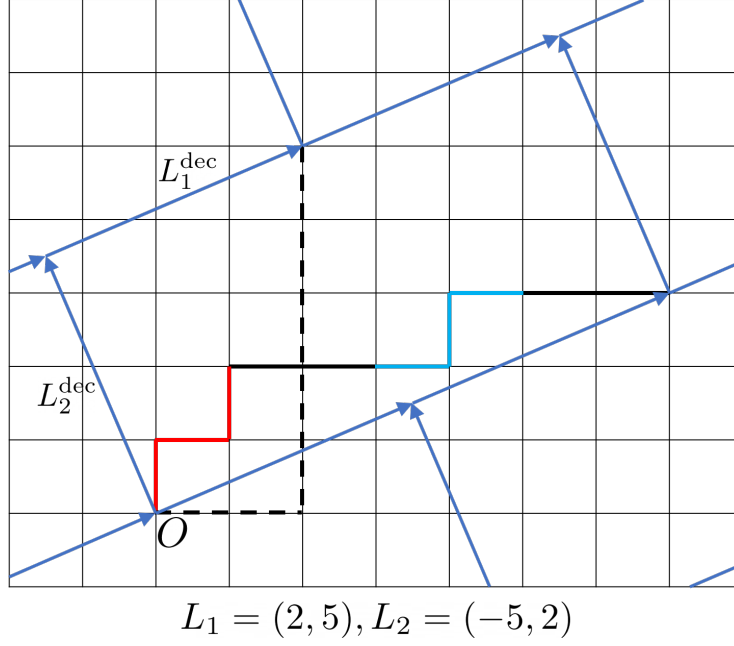


Figure 17: Illustrating the proof of Theorem 3.9 with a specific example. We show a path from O to $O + 2L_1^{\text{dec}}$ consisting of red, black, and blue edges. Each red edge can be paired with a blue edge that is located a vector L_1^{dec} away and acts on the same qubit. The dashed lines show a path from O to $O + L_1^{\text{dec}} + L_2^{\text{dec}}$ with minimal Pauli weight in which no two edges act on the same qubit. The logical operators represented by the solid and dashed paths anticommute.

We introduce three easily-verified conditions that preclude any shortest path from containing both edge e and $e+t$. If (A) $|t_x| > a$ or (B) $t_x t_y < 0$ or (C) $|t_y| > b$, then the shortest paths are restricted in this way and the shortest path's length equals its Pauli weight. We show all $t \in T(L_1^{\text{dec}}, L_2^{\text{dec}})$ fall into one of these two cases.

First, assume $m_1 > 0$. Then, because $m_1 > m_2$ and $b > a > 0$, $t_x > m_1 a \geq a$. This falls into case (A). Next, assume $m_2 < m_1 \leq 0$. We find $t_y \leq -a/2 < 0$. If $t_x > 0$, case (B) applies and we are done. So, suppose $t_x < 0$. Then $(m_1 + m_2) < -(m_1 - m_2)b/a$, which implies $|t_y|/b > (m_1 - m_2)(a^2 + b^2)/2ab > 1$, and we have case (C). \square

5.3 New hyperbolic codes

In this section, we discuss how one can construct hyperbolic quantum codes using Definition 3.3. We restrict to regular tilings of hyperbolic space, and as a result the set of codes we create here is incomparable with the set of regular homological codes created in, for instance, [11, 16, 17]. Codes in our set that are not in the regular homological set include non-checkerboardable codes, as well as some checkerboardable codes defined on graphs with even vertex degrees larger than four. Codes in the homological set not included in our set include those with face (e.g. Z -type) and vertex (e.g. X -type) stabilizers of different weight. In principle, using our code construction on irregular tilings would reproduce all homological hyperbolic codes and more, but irregular graphs are more difficult to study systematically than regular graphs, and so we do not consider them here.

The automorphism group $\text{Aut}(R)$ of a rotation system $R = (H, \lambda, \rho, \tau)$ is the group of all permutations of H that commute with λ , ρ , and τ [28]. Because of the transitivity of the monodromy group $M(R)$ on H , knowing where $\kappa \in \text{Aut}(R)$ sends a single flag $h \in H$ is enough to fix the action of κ on all of H . Therefore, the largest the automorphism group can be is $|H|$. If $|\text{Aut}(R)| = |H|$, then the rotation system is *regular*, and it represents a regular graph, i.e. all vertices have the same degree, say n , and all faces have the same number of adjacent edges, say m . These graphs are

Tiling	N	K	D	Generators	Orientable?
(5, 4)	20	5	4	$(\rho\tau\rho\lambda)^4, (\rho\lambda\tau)^5$	No
(4, 6)	6	3	2	$(\rho\lambda\tau)^3$	No
	15	6	2	$(\rho\tau\rho\lambda)^3, (\rho\lambda\tau)^5$	No
	24	9	3	$(\rho\lambda\tau\rho\lambda)^3, (\rho\tau\rho\lambda)^4, (\rho\lambda\tau)^6$	No
	30	11	3	$(\rho\tau\rho\lambda)^3$	Yes

Table 3: Some very small hyperbolic codes on non-checkerboardable graphs. The (5, 4) example in particular uses fewer qubits than any distance four hyperbolic code in [17].

called (m, n) -regular. A simple calculation and Eq. (1) shows

$$\frac{1}{2}\chi = \left(\frac{1}{m} + \frac{1}{n} - \frac{1}{2} \right) |E|. \quad (45)$$

All flags in regular rotation systems are equivalent up to automorphism. Therefore, it is sensible that there is a description without an explicit set of flags H . This description is realized by first specifying a free group along with identity relations on its generators:

$$\mathcal{F} = \langle \lambda, \rho, \tau | \lambda^2 = \rho^2 = \tau^2 = (\lambda\tau)^2 = (\lambda\rho)^m = (\rho\tau)^n = 1 \rangle. \quad (46)$$

However, only if $\chi > 0$, or equivalently, $1/m + 1/n > 1/2$, is the group \mathcal{F} finite. Additional identities are necessary to compactify infinite graphs. Compactified regular graphs are in one-to-one correspondence with normal subgroups of \mathcal{F} with finite index, and moreover these take the form of free groups with additional identities r_i imposed on the generators, e.g.

$$\mathcal{F}_c = \langle \lambda, \rho, \tau | \lambda^2 = \rho^2 = \tau^2 = (\lambda\tau)^2 = (\lambda\rho)^m = (\rho\tau)^n = 1 \text{ and } r_i(\lambda, \rho, \tau) = 1, \forall i \rangle. \quad (47)$$

To find normal subgroups of \mathcal{F}_c , one can use a computer algebra package such as GAP [50, 51]. Appendix H gives an example of using GAP for this purpose. We tabulate some small examples of non-checkerboardable hyperbolic codes in Table 3 and draw some explicit examples in Fig. 18.

Of course, non-checkerboardable codes like those in Table 3 and Fig. 18(a) are not equivalent to homological hyperbolic codes. What is less obvious is that even some checkerboardable codes, like the $[[32, 10, 4]]$ code shown in Fig. 18(b), are not equivalent to homological hyperbolic codes on *regular* tilings (though they are equivalent to a homological code on some irregular tiling). To see this, suppose we define a surface code in our framework on a (m, n) -regular tiling with $m > 4$ being even. Since $m > 4$ this is not a medial graph of any other embedded graph. However, we should also show that even after we decompose vertices into degree-four vertices using Fig. 3 the graph is still not a medial graph. Suppose it is, so that after the vertex decomposition we have a $(m', 4)$ -regular tiling. If before the decomposition we have V vertices and F faces, after the decomposition there are $V' = V + (n-4)V/2 = (n-2)V/2$ vertices and F faces. By regularity of the tilings, $F = nV/m = 4V'/m'$. This implies $m' = (2n-4)m/n$, which must be an integer. When $(2n-4)m/n$ is not an integer, it is not possible to end up with a medial graph after the vertex decomposition. An example is when $n = 4$ and $m = 6$, as for the code in Fig. 18(b).

Finally, we point out that the face-width lower bound on code distance, Theorem 4.11, is applicable to the hyperbolic codes in this section. Using it and explicitly finding logical operators with that weight, we can verify the distances of the non-checkerboardable codes in Table 3. The $[[32, 10, 4]]$ code in Fig. 18(b), being checkerboardable, has distance lower bounded by the shortest homologically non-trivial cycle in its decoding graph, Corollary 4.7. This gives $3 \leq D$, or, in other words, any non-trivial logical operator acts on qubits at three different vertices. However, we argue it must act on both qubits at at least one of these vertices, and thus be at least weight four. Focus on the X -type logical operators; the Z -type are analogous. Note each vertex is surrounded by three Z -type faces, IZ , ZI , and ZZ . Thus, the homologically non-trivial cycle with length three must visit a face of each type. At the vertex where it passes from the IZ to the ZI type face, it is supported on both qubits, namely as XX . Therefore, the minimum weight logical operator is at least weight four, and it is easy enough to find one with this weight.

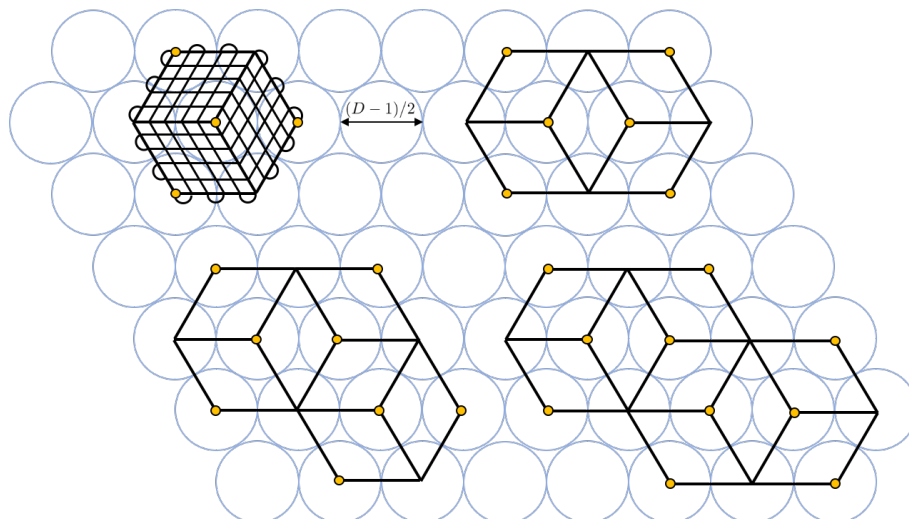


Figure 19: Circle-packing codes encoding $K = 1, 2, 3$, and 4 qubits in order of size. The packed circles in the background act as a coordinate system, and are not part of the graphs or codes. In the top left, we show the triangle code and the entire square lattices of qubits making it up. In other drawings, the lattice interiors are abstracted away. Odd-degree vertices, or twist defects, are shown as yellow circles. One can continue attaching square patches, following the pattern established in these first four examples, to grow the code down and to the right and obtain a family where $N = \frac{1}{4}(2K + 1)D^2$. Therefore, $c = (2K + 1)/4K$ approaches $1/2$ as K increases.

5.4 Two more ways to generalize the triangle code

In [6], the authors presented the family of triangle codes; the distance 5 version is pictured here in Fig. 5(a). This family has been generalized by Kesselring et al. to the stellated codes [9]. Stellated codes feature an improved constant in the relation $N = cKD^2$, namely $c = s/(2s - 2)$ for odd integers s specifying the degree of symmetry ($s = 3$ for the triangle code, $s = 5$ for the code in Fig. 5(c), etc.), and so c approaches $1/2$ as $s \rightarrow \infty$.

One curiosity in the stellated codes is that the qubit density in the center becomes large as s grows. One might wonder if there is a family of planar codes for which c approaches $1/2$ that does not have this singularity. There is indeed such a family of codes, provided one accepts a conjecture on its code distance.

We begin the construction of this family by imagining “gluing” square surface code patches, like that pictured in Fig. 5(b), together. For instance, one can glue three distance $D = 3$ square surface code patches together to obtain the $D = 5$ triangle code, Fig. 5(a), and five distance $D = 3$ patches to obtain the code in Fig. 5(c). To ensure that the qubit density is bounded by a constant throughout the resulting graph, we should not glue too many patches together around a single vertex. We glue at most four patches around a vertex, and the maximum vertex degree in the graph is six, leading to at most two qubits at a vertex.

Twist defects, the odd-degree vertices in the resulting graph, should be spaced apart by graph distance $D - 1$ to get a distance D code. Therefore, we imagine each square patch is $(D + 1)/2$ -by- $(D + 1)/2$ qubits in size, and that the only odd-degree vertices are located at two corners of the patch, diagonally opposite one another. If we glue the patches together accordingly, we get codes as in Fig. 19. We refer to this family as circle-packing codes, because of the way in which odd-degree vertices are closely packed. Because the addition of a square patch of roughly $D^2/4$ qubits adds one additional twist, and two additional twists are required to add a logical qubit, we have $N \rightarrow KD^2/2$ for these codes as K gets large.

A seemingly strange feature of the circle-packing codes is how they do not fill a 2-D area but are rather grown outward in one direction. However, one can easily see that filling a 2-D area with the same pattern of square patches results in a worse constant c . Suppose we have glued F

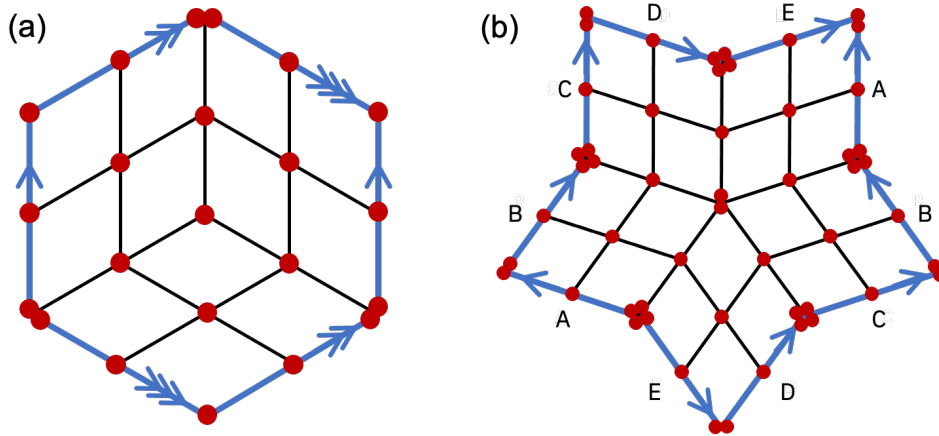


Figure 20: Two examples of stellated codes embedded in higher genus surfaces, (a) a $[[13, 2, 4]]$ code with $s = 3$ and $t = 2$ and (b) a $[[23, 4, 4]]$ code with $s = 5$ and $t = 2$. They are drawn within a $2s$ -gon with opposite sides identified. The $[[13, 2, 4]]$ code actually has optimal distance for a quantum code with $N = 13$ and $K = 2$ [52].

square patches together, filling a large 2-D area, so that the perimeter is much smaller than the volume. Then, because we can neglect the perimeter, there are $2F/3$ odd-degree vertices, implying $K \approx 2F/6$ and $N \approx F(D/2)^2 = 3KD^2/4$.

We note explicitly that we did not prove that the distance of the circle-packing codes is actually D , unlike the more exaggerated proofs in Secs. 5.1 and 5.2. We expect this is the code distance due to comparison with the similar codes in [6] and [9] and the confidence with which those authors state the code distances (though one might say they are also lacking rigorous proofs).

A second way we can generalize the stellated codes is by embedding them in higher genus ($g > 0$) surfaces. By Corollary 3.8, a larger genus can lead to more encoded qubits, assuming any reduction in the number of odd-degree vertices does not outweigh the effect. Our higher genus embeddings double the number of encoded qubits with small reductions in both N and D .

The idea of the higher genus embedding is to draw the stellated code with odd symmetry parameter s into a polygon with $2s$ sides and identify opposite sides of the polygon to make an orientable manifold. In fact, this manifold is a torus with $g = (s - 1)/2$ holes. This embedding results in two odd-degree vertices with degree s , a single vertex with degree $2s$, and all other vertices having degree four. Suppose the graph distance from the code's center to the boundary is t . Then, $N = st^2 + s - 2$, $K = s - 1$, and we conjecture $D = 2t$. This leads to $N = \frac{1}{4}(K + 1)D^2 + K - 1 = \frac{1}{4}KD^2 + O(D^2 + K)$, so that if both K and D are large, we have $c = 1/4$.

We believe the code distance is $2t$ because the code distance of a stellated code in the plane is $2t + 1$, the length of some logical operators that cross the code from one boundary to another. In the higher genus case, the qubits on the boundary are identified, reducing the weight of such operators by one. It is easy to see that $2t$ is indeed an upper bound on D by finding a logical operator (one crossing the code, as described) with that weight.

6 Open problems

In this work, we took a rigorous look at qubit surface codes defined from embedded graphs. We related them to Majorana surface codes, calculated the number of encoded qubits, and bounded the code distance in general with more specific cases worked out exactly.

There remain some interesting questions, however. First, it would be very nice to be able to calculate (efficiently) the code distance of an arbitrary surface code defined by Definition 3.3. We illustrated a prickly example in Fig. 14 and the codes in Section 5.4 lack rigorous distance proofs, for instance. If the distance cannot be efficiently calculated exactly, can a multiplicative approximation

be found better than the factor $1/2$ approximation of Theorem 4.10? Inapproximability (see e.g. [53]) in complexity theory might be applicable if indeed the exact distance calculation is NP-hard. We note that existing algorithms for exact distance calculation in the literature are limited to CSS surface codes [16] or require exponential time for some of the codes in our formalism [54].

A second interesting problem is to create a graph-based formalism for color code twists as described by Kesselring et al. [9]. By this we mean that one should be able to define twisted color codes on arbitrary embedded graphs, where the variety of twists correspond to (local) features of the graph, just as we found that surface code twists arise at odd-degree vertices. For instance, while untwisted color codes are defined from graphs with 3-colorable, even-degree faces, a twist may arise in a more general graph where a face has odd-degree.

A third direction would be to literally add a third direction: can a lattice-based formalism be used to describe the variety of surface codes and twist defects in 3-dimensions and beyond? Here things may in general get messy due to the lack of classification theorems like those for 1D and 2D codes [55, 56]. Still, we do not need to demand that the framework describe every 3D code by a 3D lattice, just that every 3D lattice gives rise to some 3D code. We probably would want some known family, like the toric code, to arise from the construction on the cubic lattice, and that something interesting (better code parameters, fractal logical operators, etc.) happen for other lattices.

Finally, probably the most practical open problem is to find lower bounds on the constant c in the relation $N = cKD^2$ [2] that holds for 2-dimensional codes. Note that $c(w, \mathcal{M})$ may depend on both the maximum stabilizer weight w and the manifold \mathcal{M} . Therefore, a concrete question is, for instance, limited to stabilizers of at most weight five in the plane \mathbb{R}^2 , what is the smallest possible value of $c(5, \mathbb{R}^2)$? We believe the best known is $c(5, \mathbb{R}^2)$ approaching $1/2$, see Figure 19. If one allows weight-six stabilizers, as in the color codes of [9], then $c(6, \mathbb{R}^2)$ can approach $1/4$. One can generally concatenate two 2-dimensional codes (e.g. surface codes) with the $[[4, 2, 2]]$ code to get a color code with $N' = 4N$, $K' = 2K$, $D' = 2D$, and $w' = 2w$ [57]. Therefore $c(2w, \mathcal{M}) \leq c(w, \mathcal{M})/2$ for any w and \mathcal{M} .

Acknowledgements

We are grateful to a number of people for helpful discussions, including Sergey Bravyi, Andrew Cross, Carrollan Helinski, Tomas Jochym-O'Connor, Alex Kubica, Andrew Landahl, and Guanyu Zhu. We would especially like to thank Leonid Pryadko and Alexey Kovalev for sharing ideas that led to the distance proof for cyclic toric codes. Rahul Sarkar was partially supported by the Schlumberger Innovation Fellowship. Ted Yoder was partially supported by the IBM Research Frontiers Institute.

References

- [1] Alexei Yu. Kitaev. “Fault-tolerant quantum computation by anyons”. *Annals of Physics* **303**, 2–30 (2003). doi: [10.1016/s0003-4916\(02\)00018-0](https://doi.org/10.1016/s0003-4916(02)00018-0).
- [2] Sergey Bravyi, David Poulin, and Barbara Terhal. “Tradeoffs for reliable quantum information storage in 2D systems”. *Physical Review Letters* **104**, 050503 (2010). doi: [10.1103/physrevlett.104.050503](https://doi.org/10.1103/physrevlett.104.050503).
- [3] Héctor Bombín and Miguel A. Martin-Delgado. “Optimal resources for topological two-dimensional stabilizer codes: Comparative study”. *Physical Review A* **76**, 012305 (2007). doi: [10.1103/physreva.76.012305](https://doi.org/10.1103/physreva.76.012305).
- [4] Héctor Bombín. “Topological order with a twist: Ising anyons from an abelian model”. *Physical Review Letters* **105**, 030403 (2010). doi: [10.1103/physrevlett.105.030403](https://doi.org/10.1103/physrevlett.105.030403).
- [5] Alexei Kitaev and Liang Kong. “Models for gapped boundaries and domain walls”. *Communications in Mathematical Physics* **313**, 351–373 (2012). doi: [10.1007/s00220-012-1500-5](https://doi.org/10.1007/s00220-012-1500-5).

- [6] Theodore J. Yoder and Isaac H. Kim. “The surface code with a twist”. *Quantum* **1**, 2 (2017). doi: [10.22331/q-2017-04-25-2](https://doi.org/10.22331/q-2017-04-25-2).
- [7] Andrew J. Landahl. “The surface code on the rhombic dodecahedron” (2020). doi: [10.48550/arXiv.2010.06628](https://doi.org/10.48550/arXiv.2010.06628).
- [8] Anirudh Krishna and David Poulin. “Topological wormholes: Nonlocal defects on the toric code”. *Physical Review Research* **2**, 023116 (2020). doi: [10.1103/physrevresearch.2.023116](https://doi.org/10.1103/physrevresearch.2.023116).
- [9] Markus S. Kesselring, Fernando Pastawski, Jens Eisert, and Benjamin J. Brown. “The boundaries and twist defects of the color code and their applications to topological quantum computation”. *Quantum* **2**, 101 (2018). doi: [10.22331/q-2018-10-19-101](https://doi.org/10.22331/q-2018-10-19-101).
- [10] Nicolas Delfosse. “Tradeoffs for reliable quantum information storage in surface codes and color codes”. In 2013 IEEE International Symposium on Information Theory. Pages 917–921. IEEE (2013). doi: [10.1109/isit.2013.6620360](https://doi.org/10.1109/isit.2013.6620360).
- [11] Nikolas P. Breuckmann and Barbara M. Terhal. “Constructions and noise threshold of hyperbolic surface codes”. *IEEE transactions on Information Theory* **62**, 3731–3744 (2016). doi: [10.1109/tit.2016.2555700](https://doi.org/10.1109/tit.2016.2555700).
- [12] Xiao-Gang Wen. “Quantum orders in an exact soluble model”. *Phys. Rev. Lett.* **90**, 016803 (2003). doi: [10.1103/PhysRevLett.90.016803](https://doi.org/10.1103/PhysRevLett.90.016803).
- [13] Alexei Kitaev. “Anyons in an exactly solved model and beyond”. *Annals of Physics* **321**, 2–111 (2006). doi: [10.1016/j.aop.2005.10.005](https://doi.org/10.1016/j.aop.2005.10.005).
- [14] Sergey Bravyi, Matthias Englbrecht, Robert König, and Nolan Peard. “Correcting coherent errors with surface codes”. *NPJ Quantum Information* **4**, 55 (2018). doi: [10.1038/s41534-018-0106-y](https://doi.org/10.1038/s41534-018-0106-y).
- [15] Alexey A. Kovalev and Leonid P. Pryadko. “Improved quantum hypergraph-product LDPC codes”. In 2012 IEEE International Symposium on Information Theory Proceedings. Pages 348–352. IEEE (2012). doi: [10.1109/isit.2012.6284206](https://doi.org/10.1109/isit.2012.6284206).
- [16] Nikolas P. Breuckmann, Christophe Vuillot, Earl Campbell, Anirudh Krishna, and Barbara M. Terhal. “Hyperbolic and semi-hyperbolic surface codes for quantum storage”. *Quantum Science and Technology* **2**, 035007 (2017). doi: [10.1088/2058-9565/aa7d3b](https://doi.org/10.1088/2058-9565/aa7d3b).
- [17] Nikolas P. Breuckmann. “Homological quantum codes beyond the toric code”. PhD thesis. RWTH Aachen University. (2017). doi: [10.18154/RWTH-2018-01100](https://doi.org/10.18154/RWTH-2018-01100).
- [18] Sagar Vijay, Timothy H. Hsieh, and Liang Fu. “Majorana fermion surface code for universal quantum computation”. *Physical Review X* **5**, 041038 (2015). doi: [10.1103/physrevx.5.041038](https://doi.org/10.1103/physrevx.5.041038).
- [19] Alexey A. Kovalev, Ilya Dumer, and Leonid P. Pryadko. “Design of additive quantum codes via the code-word-stabilized framework”. *Physical Review A* **84**, 062319 (2011). doi: [10.1103/PhysRevA.84.062319](https://doi.org/10.1103/PhysRevA.84.062319).
- [20] Saul Stahl. “The embeddings of a graph—a survey”. *Journal of Graph Theory* **2**, 275–298 (1978). doi: [10.1002/jgt.3190020402](https://doi.org/10.1002/jgt.3190020402).
- [21] Saul Stahl. “Generalized embedding schemes”. *Journal of Graph Theory* **2**, 41–52 (1978). doi: [10.1002/jgt.3190020106](https://doi.org/10.1002/jgt.3190020106).
- [22] Marcus Schaefer. “Crossing numbers of graphs”. CRC Press. (2018). First edition. doi: [10.1201/9781315152394](https://doi.org/10.1201/9781315152394).
- [23] John Lee. “Introduction to topological manifolds”. Volume 202 of Graduate Texts in Mathematics. Springer Science & Business Media. (2011). doi: [10.1007/978-1-4419-7940-7](https://doi.org/10.1007/978-1-4419-7940-7).
- [24] John Lee. “Introduction to smooth manifolds”. Volume 218 of Graduate Texts in Mathematics. Springer Science & Business Media. (2012). Second edition. doi: [10.1007/978-1-4419-9982-5](https://doi.org/10.1007/978-1-4419-9982-5).
- [25] Tibor Radó. “Über den Begriff der Riemannschen Fläche”. *Acta Litt. Sci. Szeged* **2**, 101–121 (1925). url: <https://zbmath.org/51.0273.01>.
- [26] Herbert Seifert and William Threlfall. “Seifert and Threlfall: A textbook of topology”. Volume 89 of Pure and Applied Mathematics. Elsevier. (1980). doi: [10.1016/s0079-8169\(08\)x6140-9](https://doi.org/10.1016/s0079-8169(08)x6140-9).

- [27] George K. Francis and Jeffrey R. Weeks. “Conway’s ZIP proof”. *The American Mathematical Monthly* **106**, 393–399 (1999). doi: [10.1080/00029890.1999.12005061](https://doi.org/10.1080/00029890.1999.12005061).
- [28] Roman Nedela and Martin Škoviera. “Regular maps on surfaces with large planar width”. *European Journal of Combinatorics* **22**, 243–262 (2001). doi: [10.1006/eujc.2000.0441](https://doi.org/10.1006/eujc.2000.0441).
- [29] Jozef Širáň. “Triangle group representations and their applications to graphs and maps”. *Discrete Mathematics* **229**, 341–358 (2001). doi: [10.1016/s0012-365x\(00\)00215-6](https://doi.org/10.1016/s0012-365x(00)00215-6).
- [30] Reinhard Diestel. “Graph theory”. Volume 173 of Graduate Texts in Mathematics. Springer Berlin, Heidelberg. (2000). Fifth edition. doi: [10.1007/978-3-662-53622-3](https://doi.org/10.1007/978-3-662-53622-3).
- [31] Jonathan L Gross and Thomas W Tucker. “Topological graph theory”. Dover books on mathematics. Dover Publications. (2001).
- [32] Allen Hatcher. “Algebraic topology”. Cambridge University Press. (2002).
- [33] Sergio Cabello and Bojan Mohar. “Finding shortest non-separating and non-contractible cycles for topologically embedded graphs”. *Discrete & Computational Geometry* **37**, 213–235 (2007). doi: [10.1007/s00454-006-1292-5](https://doi.org/10.1007/s00454-006-1292-5).
- [34] Glen E. Bredon. “Topology and geometry”. Volume 139 of Graduate Texts in Mathematics. Springer Science & Business Media. (2013). First edition. doi: [10.1007/978-1-4757-6848-0](https://doi.org/10.1007/978-1-4757-6848-0).
- [35] Sergey Bravyi, Barbara M. Terhal, and Bernhard Leemhuis. “Majorana fermion codes”. *New Journal of Physics* **12**, 083039 (2010). doi: [10.1088/1367-2630/12/8/083039](https://doi.org/10.1088/1367-2630/12/8/083039).
- [36] Sagar Vijay and Liang Fu. “Quantum error correction for complex and Majorana fermion qubits” (2017). doi: [10.48550/arXiv.1703.00459](https://doi.org/10.48550/arXiv.1703.00459).
- [37] Alexei Yu. Kitaev. “Unpaired Majorana fermions in quantum wires”. *Physics-Uspekhi* **44**, 131 (2001). doi: [10.1070/1063-7869/44/10S/S29](https://doi.org/10.1070/1063-7869/44/10S/S29).
- [38] Rahul Sarkar and Ewout van den Berg. “On sets of maximally commuting and anticommuting Pauli operators”. *Research in the Mathematical Sciences* **8**, 14 (2021). doi: [10.1007/s40687-020-00244-1](https://doi.org/10.1007/s40687-020-00244-1).
- [39] Sergey B. Bravyi and Alexei Yu. Kitaev. “Quantum codes on a lattice with boundary” (1998). doi: [10.48550/arXiv.quant-ph/9811052](https://doi.org/10.48550/arXiv.quant-ph/9811052).
- [40] Michael H. Freedman and David A. Meyer. “Projective plane and planar quantum codes”. *Foundations of Computational Mathematics* **1**, 325–332 (2001). doi: [10.1007/s102080010013](https://doi.org/10.1007/s102080010013).
- [41] Jonas T. Anderson. “Homological stabilizer codes”. *Annals of Physics* **330**, 1–22 (2013). doi: [10.1016/j.aop.2012.11.007](https://doi.org/10.1016/j.aop.2012.11.007).
- [42] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. *Physical Review A* **52**, R2493–R2496 (1995). doi: [10.1103/PhysRevA.52.R2493](https://doi.org/10.1103/PhysRevA.52.R2493).
- [43] Mark De Berg, Marc Van Kreveld, Mark Overmars, and Otfried Schwarzkopf. “Computational geometry”. Springer. (1997). First edition. doi: [10.1007/978-3-662-03427-9](https://doi.org/10.1007/978-3-662-03427-9).
- [44] Eric Dennis, Alexei Kitaev, Andrew Landahl, and John Preskill. “Topological quantum memory”. *Journal of Mathematical Physics* **43**, 4452–4505 (2002). doi: [10.1063/1.1499754](https://doi.org/10.1063/1.1499754).
- [45] Maïssam Barkeshli and Michael Freedman. “Modular transformations through sequences of topological charge projections”. *Physical Review B* **94**, 165108 (2016). doi: [10.1103/PhysRevB.94.165108](https://doi.org/10.1103/PhysRevB.94.165108).
- [46] Joseph Douglas Horton. “A polynomial-time algorithm to find the shortest cycle basis of a graph”. *SIAM Journal on Computing* **16**, 358–366 (1987). doi: [10.1137/0216026](https://doi.org/10.1137/0216026).
- [47] Edoardo Amaldi, Claudio Iuliano, Tomasz Jurkiewicz, Kurt Mehlhorn, and Romeo Rizzi. “Breaking the $O(m^2n)$ barrier for minimum cycle bases”. In *Algorithms - ESA 2009*. Pages 301–312. Springer Berlin Heidelberg (2009). doi: [10.1007/978-3-642-04128-0_28](https://doi.org/10.1007/978-3-642-04128-0_28).
- [48] Telikepalli Kavitha, Kurt Mehlhorn, Dimitrios Michail, and Katarzyna Paluch. “A faster algorithm for minimum cycle basis of graphs”. In *International Colloquium on Automata, Languages, and Programming*. Pages 846–857. Springer Berlin Heidelberg (2004). doi: [10.1007/978-3-540-27836-8_71](https://doi.org/10.1007/978-3-540-27836-8_71).

- [49] Kurt Mehlhorn and Dimitrios Michail. “Minimum cycle bases: Faster and simpler”. *ACM Transactions on Algorithms (TALG)* **6**, 8 (2010). doi: [10.1145/1644015.1644023](https://doi.org/10.1145/1644015.1644023).
- [50] The GAP Group. “GAP – Groups, Algorithms, and Programming, Version 4.11.0”. (2020). url: <https://www.gap-system.org>.
- [51] Friedrich Rober. “LINS – a gap package”. (2020). url: <https://github.com/FriedrichRober/LINS>.
- [52] Markus Grassl. “Bounds on the minimum distance of linear codes and quantum codes”. Online available at <http://www.codetables.de> (2007). Accessed on 2020-11-03.
- [53] Luca Trevisan. “Inapproximability of combinatorial optimization problems”. In *Paradigms of Combinatorial Optimization*. Chapter 13, pages 381–434. John Wiley & Sons, Ltd (2014). doi: [10.1002/9781119005353.ch13](https://doi.org/10.1002/9781119005353.ch13).
- [54] ChunJun Cao, Michael J Gullans, Brad Lackey, and Zitao Wang. “Quantum Lego expansion pack: Enumerators from tensor networks” (2023). doi: [10.48550/arXiv.2308.05152](https://doi.org/10.48550/arXiv.2308.05152).
- [55] Héctor Bombín. “Structure of 2D topological stabilizer codes”. *Communications in Mathematical Physics* **327**, 387–432 (2014). doi: [10.1007/s00220-014-1893-4](https://doi.org/10.1007/s00220-014-1893-4).
- [56] Jeongwan Haah. “Algebraic methods for quantum codes on lattices”. *Revista Colombiana de Matemáticas* **50**, 299–349 (2016). doi: [10.15446/revcolma.v50n2.62214](https://doi.org/10.15446/revcolma.v50n2.62214).
- [57] Ben Criger and Barbara Terhal. “Noise thresholds for the $[[4, 2, 2]]$ -concatenated toric code”. *Quantum Information & Computation* **16**, 1261–1281 (2016). doi: [10.26421/qic16.15-16-1](https://doi.org/10.26421/qic16.15-16-1).
- [58] Daniel Gottesman. “Stabilizer codes and quantum error correction”. PhD thesis. California Institute of Technology. (1997). doi: [10.7907/rzr7-dt72](https://doi.org/10.7907/rzr7-dt72).
- [59] Pascual Jordan and Eugene Paul Wigner. “Über das Paulische Äquivalenzverbot”. *Zeitschrift für Physik* **47**, 631–651 (1928). doi: [10.1007/BF01331938](https://doi.org/10.1007/BF01331938).
- [60] Pavel Hrubeš. “On families of anticommuting matrices”. *Linear Algebra and its Applications* **493**, 494–507 (2016). doi: [10.1016/j.laa.2015.12.015](https://doi.org/10.1016/j.laa.2015.12.015).
- [61] Henry S Warren. “Hacker’s delight”. Addison-Wesley Professional. (2012). 2nd edition.

Appendices

A Graph embeddings from oriented rotation systems

The goal of this appendix is to establish an equivalence between graphs embedded in orientable manifolds and a combinatorial object called an oriented rotation system. It is easier to see this equivalence in one direction — start with an embedded graph in an orientable manifold and develop the oriented rotation system description. We do this first below, and sketch a more formal argument for the converse direction after that.

As was already mentioned before in Section 2.2, the oriented rotation system is a special case of the general rotation system. Since the manifold is orientable, one can unambiguously define a “side” of the manifold, and thus one does not need to maintain a flag for each half-edge (defined below in Definition A.1), one for each side of the manifold. Thus one can simply work with the half-edges, two for every edge of the graph. As in the case of general rotations systems, the vertices, edges, and faces of the graph embedding will be sets of these half-edges.

Definition A.1. An *oriented rotation system* is a triple $R_O = (H_O, \nu, \epsilon)$, where H_O is a finite set of *half-edges* (sometimes called *darts*), and two permutations $\nu : H_O \rightarrow H_O$ and $\epsilon : H_O \rightarrow H_O$ satisfying the additional properties

- (i) ϵ is a fixed-point-free involution, meaning $\epsilon^2 = \text{Id}$, and $\epsilon h \neq h$ for all $h \in H_O$.
- (ii) The free group $\langle \nu, \epsilon \rangle$ generated by ν and ϵ , acts transitively on H_O . This means that for all $h, h' \in H_O$, one can find $\kappa \in \langle \nu, \epsilon \rangle$ such that $\kappa h = h'$.

We now define the three sets

$$\begin{aligned} V &= \{v \subseteq H_O : v \text{ is an orbit of the free group } \langle \nu \rangle\}, \\ E &= \{e \subseteq H_O : e \text{ is an orbit of the free group } \langle \epsilon \rangle\}, \\ F &= \{f \subseteq H_O : f \text{ is an orbit of the free group } \langle \nu, \epsilon \rangle\}. \end{aligned} \tag{48}$$

We call elements of these sets the *vertices*, *edges*, and *faces* of the oriented rotation system, respectively. So, condition (i) ensures edges are sets of exactly two half-edges, and thus $|H_O|$ is even, while condition (ii) ensures the graph is connected.

Thus given an embedded graph $G(V, E)$ in an oriented manifold \mathcal{M} , to get the oriented rotation system (H_O, ν, ϵ) corresponding to it, we first create a set of half-edges H_O with each edge in E contributing two half-edges. Let $e \in E$ with corresponding half-edges $h_e, h'_e \in H_O$. The permutation ϵ is defined so that $\epsilon h_e = h'_e$, and $\epsilon h'_e = h_e$ for every $e \in E$, and note that property (i) of Definition A.1 is satisfied. If e is adjacent to vertices $v, v' \in V$ ($v = v'$ if e is a loop), we assign h_e and h'_e to v and v' respectively. Thus each vertex v gets assigned exactly $\deg(v)$ half-edges, and conversely each half-edge in H_O is assigned to exactly one vertex in V . This partitions H_O into disjoint subsets indexed by the vertices. Now choose a continuously varying outward normal on \mathcal{M} , which is possible since it is orientable, fix a vertex $v \in V$, and let $(H_O)_v$ be the half-edges assigned to it. We can enumerate the edges adjacent to v in counterclockwise order with respect to the outward normal, where an edge appears twice successively in this enumeration if it is a loop. Then replacing each edge in this ordering by the corresponding half-edge that it contributes to v , gives a counterclockwise ordering of $(H_O)_v$. The permutation ν for $(H_O)_v$ is defined to be the cyclic permutation given by this counterclockwise ordering, and repeating this at every vertex completely specifies ν . It follows easily that property (ii) of Definition A.1 is satisfied if G is connected. This completes the construction of the oriented rotation system (H_O, ν, ϵ) . See Fig. 21 for an example. However, the description we just provided fails to capture graph embeddings in non-orientable manifolds, because in that case we cannot choose a continuously varying outward normal.

Next, we show how to get a graph embedding on an orientable manifold starting from an oriented rotation system $R_O = (H_O, \nu, \epsilon)$. Recall that the rotation system gives rise to a set of vertices V ,

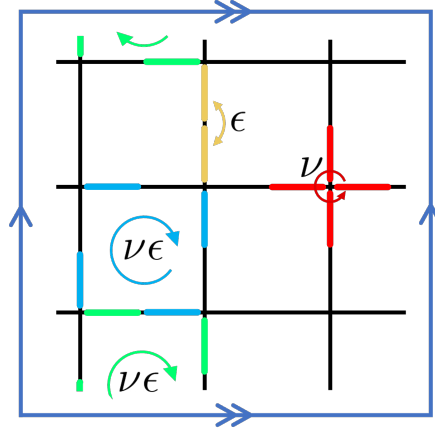


Figure 21: A graph embedded onto the torus and its oriented rotation system. An orbit of $\langle \nu \rangle$, an orbit of $\langle \epsilon \rangle$, and two orbits of $\langle \nu \epsilon \rangle$ are shown. These correspond to an edge, a vertex, and two faces of the graph embedding.

edges E , and faces F , which are orbits of $\langle \nu \rangle$, $\langle \epsilon \rangle$ and $\langle \nu \epsilon \rangle$ respectively. The graph specified by R_O , consists of vertices and edges labeled by the elements of V and E respectively (in a bijective fashion), so we can use the notation $G(V, E)$ to denote the graph without any ambiguity. A vertex $v \in V$ and edge $e \in E$ are adjacent if $v \cap e \neq \emptyset$. One notes that property (ii) of Definition A.1 guarantees that G is connected.

In addition to specifying the graph connectivity, R_O also specifies an embedding of G , where the faces of the graph embedding are in bijection with the elements of F — this justifies the use of the notation $G(V, E, F)$ to denote the graph embedding. We will now briefly sketch how to obtain this embedding starting from R_O . For this we establish some additional notation: for every half-edge $h \in H_O$, let $[h]_\nu \in V$ be the unique vertex that contains h and, likewise, let $[h]_\epsilon \in E$ be the unique edge that contains h . Adjacency of vertices, edges and faces for oriented rotation systems are defined by non-trivial intersection (just as for general rotation systems), so for e.g. a vertex v and a face f are adjacent if and only if $v \cap f \neq \emptyset$.

The construction starts by assigning an open disc $\mathcal{B}_f(0, 1) \subseteq \mathbb{R}^2$ to each face $f \in F$ of R_O . For a face f , let us order the half-edges in it as $\{h_0, \dots, h_{|f|-1}\}$, where $h_i = (\nu \epsilon)^i h$ for any $h \in f$ chosen arbitrarily. Now for the closed disc $\overline{\mathcal{B}_f(0, 1)}$, first let S_f^1 denote the boundary circle parameterized in polar coordinates, and place $|f|$ distinct points $0 = \theta_0 < \dots < \theta_{|f|-1} < 2\pi$ on S_f^1 . Let $\omega_0, \dots, \omega_{|f|-1}$ denote the open segments that result defined as $\omega_i = (\theta_i, \theta_{i+1})$ for $0 \leq i \leq |f| - 2$, and $\omega_{|f|-1} = (\theta_{|f|-1}, 2\pi)$, and then define a marking function $M_f : \bigcup_{i=0}^{|f|-1} \{\theta_i, \omega_i\} \rightarrow V \cup H_O$ as

$$M_f(x) = \begin{cases} [h_i]_\nu & \text{if } x = \theta_i \\ h_i & \text{if } x = \omega_i. \end{cases} \quad (49)$$

The combined effect of the marking functions for all the faces is that each half-edge h gets associated with exactly one open segment of a boundary circle S_f^1 for some face f , while each vertex v gets associated with exactly $|v|$ points in different boundary circles. Finally, we take the disjoint union space $\bigsqcup_{f \in F} \overline{\mathcal{B}_f(0, 1)}$ and define an identification rule \sim for points on the boundary circles as follows:

- (i) For every half-edge $h \in H_O$, if (θ_a, θ_b) and (θ'_a, θ'_b) are the open segments associated with h and ϵh respectively, identify these segments using the map $(\theta_a, \theta_b) \ni x \mapsto x(\theta'_b - \theta'_a)/(\theta_a - \theta_b) + (\theta_a \theta'_a - \theta_b \theta'_b)/(\theta_a - \theta_b) \in (\theta'_a, \theta'_b)$.
- (ii) If points θ, θ' are associated with vertices v, v' respectively, then they are identified if and only if $v = v'$.

The quotient space $\mathcal{M} = \bigsqcup_{f \in F} \overline{\mathcal{B}_f(0, 1)} / \sim$ that results under this identification is an orientable manifold (we omit the topological details), and $G(V, E)$ is 2-cell embedded in it. The graph embedding map Γ is obtained as follows: (a) if a vertex $v \in V$ is associated with points $\{\theta_1, \dots, \theta_{|v|}\}$,

then $\Gamma(v)$ equals the equivalence class of these points under \sim , (b) if a half-edge h is associated with an open segment (θ_a, θ_b) then for the corresponding edge $e = \{h, \epsilon h\}$, $\Gamma(e)$ equals the closure in \mathcal{M} of the equivalence class of (θ_a, θ_b) under \sim .

B Chain complexes

In this appendix, we translate the discussion of Section 2.4 into the language of chain complexes. Any graph embedding $G(V, E, F)$ can be associated with a *chain complex*, consisting of 0-chains, 1-chains, and 2-chains. A 2-chain is a formal linear combination of the faces of the graph embedding with coefficients in \mathbb{F}_2 , that is an expression of the form

$$\sum_{f \in F} c_f f, \quad c_f \in \mathbb{F}_2, \quad (50)$$

and the set of all 2-chains form an \mathbb{F}_2 -vector space of dimension $|F|$, which we will denote as $C_2(G)$. Similarly, 1-chains and 0-chains are defined to be formal linear combinations of edges and vertices of the graph respectively, with coefficients in \mathbb{F}_2 . The set of all 1-chains, which we will denote $C_1(G)$, and the set of all 0-chains, which we will denote $C_0(G)$, form \mathbb{F}_2 -vector spaces of dimensions $|E|$ and $|V|$ respectively. It is clear that the set of faces, edges, and vertices form a basis for $C_2(G)$, $C_1(G)$, and $C_0(G)$ respectively.

Associated to these vector spaces are two linear maps, called *boundary maps* ∂_2 and ∂_1 . To define these maps, we need the face-edge adjacency matrix Φ defined in Section 2.3, and also need the edge-vertex adjacency matrix $\tilde{\Phi} \in \mathbb{F}_2^{|E| \times |V|}$, which we define to be $\tilde{\Phi} = \frac{1}{2}BA^\top$ (recall that A, B are defined in Section 2.2), with the result reduced modulo 2. Note that $|e \cap v| = 0, 2$, or 4 , with the case $|e \cap v| = 4$ corresponding to the case when both the endpoints of e are incident on v . Thus given an edge e and a vertex v , we have that $\tilde{\Phi}_{ev} = 0$ if $|e \cap v| = 0$ or 4 , and $\tilde{\Phi}_{ev} = 1$ if $|e \cap v| = 2$. The maps ∂_1 and ∂_2 are now defined by their action on the basis elements of their domains as follows, with the convention that an empty sum is the zero vector:

$$\begin{aligned} \partial_2 : C_2(G) &\rightarrow C_1(G), \quad \partial_2(f) := \sum_{\{e: \Phi_{fe}=1\}} e, \quad \forall f \in F, \\ \partial_1 : C_1(G) &\rightarrow C_0(G), \quad \partial_1(e) := \sum_{\{v: \tilde{\Phi}_{ev}=1\}} v, \quad \forall e \in E, \end{aligned} \quad (51)$$

and these maps are extended uniquely by linearity to the whole domain. This also shows that the matrices $\tilde{\Phi}^\top$ and Φ^\top are the matrix representations of the linear maps ∂_1 and ∂_2 respectively. With these boundary maps defined, we have the chain complex given by the diagram

$$C_2(G) \xrightarrow{\partial_2} C_1(G) \xrightarrow{\partial_1} C_0(G). \quad (52)$$

The first homology group over \mathbb{F}_2 can now be equivalently defined to be the quotient vector space

$$H_1(G, \mathbb{F}_2) = \frac{\text{kernel}(\partial_1)}{\text{range}(\partial_2)}. \quad (53)$$

This definition makes sense because $\partial_1 \circ \partial_2 = 0$, which can be verified directly by showing $\sum_e \Phi_{fe} \tilde{\Phi}_{ev} = 0$ modulo two for all $f \in F$, $v \in V$. Calculate

$$\sum_{e \in E} \Phi_{fe} \tilde{\Phi}_{ev} = \sum_{e: \Phi_{fe}=1} \tilde{\Phi}_{ev} = \sum_{e: \Phi_{fe}=1} \frac{1}{2} \sum_{h \in H} B_{eh} A_{vh} \quad \text{mod } 2 \quad (54)$$

$$= \sum_{e: \Phi_{fe}=1} |\{[h]_\tau \in H/\tau : [h]_\tau \subseteq e, v\}| \quad \text{mod } 2 \quad (55)$$

$$= |\{[h]_\tau \in H/\tau : [h]_\tau \subseteq v, |[h]_\tau \cap f| = 1\}| \quad \text{mod } 2 \quad (56)$$

$$= |\{h \in H : h \in v, f\}| \quad \text{mod } 2 = 0. \quad (57)$$

Going from Eq. (54) to Eq. (55) involves noticing that if $B_{eh}A_{vh} = 1$ then $B_{e,\tau h}A_{v,\tau h} = 1$, $B_{eh} = 1$, and $A_{vh} = 1$. If $A_{vh} = 1$, then $[h]_\tau = \{h, \tau h\} \subseteq v$ by definition of vertex-flag adjacency matrix A . Likewise, if $B_{eh} = 1$, then $[h]_\tau \subseteq e$ by definition of edge-flag adjacency matrix B . To show Eq. (56) use the definition of Φ to conclude that any edge $e = \{h, \tau h, \lambda h, \tau \lambda h\}$ for which $\Phi_{fe} = 1$ shares exactly one of h and τh with f . Because we evaluate the set size modulo 2, we can count flags shared between v and f directly as in Eq. (57). We conclude the number of shared flags is even by noting that for every flag $h \in v, f$, there is another flag $\rho h \in v, f$.

One should also note the following. Let us decompose the set of edges E into disjoint sets E' and E'' , where $E' = \{e \in E : \Phi_{fe} = 0 \ \forall f \in F\}$, and $E'' = E \setminus E'$. Then defining the subspaces $C'_1(G)$ (resp. $C''_1(G)$) to be formal linear combinations of edges in E' (resp. E'') with coefficients in \mathbb{F}_2 , we see that $C_1(G)$ can be written as a direct sum $C_1(G) = C'_1(G) \oplus C''_1(G)$. From the definition of ∂_2 in Eq. 51, it is then clear that $\text{range}(\partial_2) \subseteq C''_1(G)$.

Since trails are elements of $\mathbb{F}_2^{|E|}$, we now see that they are 1-chains. However, not every 1-chain is a trail. On the other hand, every edge is a trail and an 1-chain, and thus the group $\mathcal{T}(G)$ can be canonically identified with the \mathbb{F}_2 -vector space $C_1(G)$, such that trails are identified with their corresponding 1-chains, and the identification is a group isomorphism. Once this identification is made, we can trace through the definitions of $\mathcal{Z}(G)$ and $\mathcal{B}(G)$ in Eq. (4), and that of ∂_1 and ∂_2 in Eq. (51), to conclude that $\mathcal{Z}(G)$ is identified with $\text{kernel}(\partial_1)$, and $\mathcal{B}(G)$ is identified with $\text{range}(\partial_2)$. Thus cycles of G are elements of $\text{kernel}(\partial_1)$ and boundaries of faces of G are elements of $\text{range}(\partial_2)$. This proves the equivalence of the two definitions of $H_1(G, \mathbb{F}_2)$ in Eq. (5) and Eq. (53).

C Paulis with arbitrary commutation patterns and CALs

In this appendix we first do a very brief review of the Pauli group and recall some of its basic properties. We then analyze the problem of finding a list of Paulis given a desired commutation pattern. As a special case, we study the case of cyclically anticommuting list of Paulis in some detail, which is a key tool in the construction of the qubit surface codes of Section 3.3.

C.1 Basic definitions

A *multiset* M is a set with repeated elements. The *multiplicity* $\nu(m)$ of an element $m \in M$ is defined as the number of occurrences of m in M . For a set A , if $m \in M$ implies $m \in A$, then we say $M \subseteq A$. If M' is another multiset, then we say $M' \subseteq M$ if and only if the following condition holds: for each $m \in M'$ with multiplicity $\nu'(m)$, it must hold that $m \in M$ and $\nu'(m) \leq \nu(m)$. If $M' \subseteq M$ and $M' \neq M$ then we write $M' \subset M$. An ordered multiset is a *list*.

The Pauli group \mathcal{P}_1 on one qubit is the 16 element group consisting of the 2×2 identity matrix I_2 , and the Pauli matrices

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \text{and} \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad (58)$$

together with all possible phase factors of ± 1 and $\pm i$. The n -Pauli group \mathcal{P}_n on n qubits ($n \geq 1$), with qubits numbered $0, \dots, n-1$, is now defined to be the group

$$\mathcal{P}_n = \{\eta (\sigma_0 \otimes \dots \otimes \sigma_{n-1}) : \eta \in \{\pm 1, \pm i\}, \sigma_j \in \{I_2, X, Y, Z\} \ \forall j\}, \quad (59)$$

where \otimes denotes the Kronecker product of matrices. The *support* of a Pauli $p = \eta (\sigma_0 \otimes \dots \otimes \sigma_{n-1}) \in \mathcal{P}_n$, which we define as $\text{supp}(p) := \{0 \leq j \leq n-1 : \sigma_j \neq I_2\}$, indicates all those qubits where it acts by either X , Y or Z . For brevity we will often simply list the Pauli matrices acting on the support of a Pauli, with a subscript per matrix indicating the qubit where it acts. For example, $p = iX_0Y_2Z_3$ indicates that p acts on qubits 0, 2 and 3 with X , Y and Z respectively, and with I_2 on all remaining qubits. Another equivalent notation for representing p (frequently used in Appendix F) is to drop the \otimes symbol, which gives a string of length n with a phase factor in front. This is called the string representation of Paulis. In a string representation, since there is

never any chance for confusion, I_2 is replaced by I to simplify the notation. For example if $n = 5$, the Pauli $iX_0Y_2Z_3$ has the string representation $iX I Y Z I$. Moreover one can optionally combine alphabets appearing consecutively in a string representation, for which we give an example: for $n = 5$, if $XX I I Z$ is the string representation of a Pauli, then it is also equivalently written in compressed form as $X^{\otimes 2} I^{\otimes 2} Z$. We define the abelian group $\hat{\mathcal{P}}_n$ to be the quotient group $\mathcal{P}_n / \langle iI \rangle$, i.e. if $\hat{p} \in \hat{\mathcal{P}}_n$, then it is an equivalence class of the form $\hat{p} = \{p, -p, ip, -ip\} = \langle iI, p \rangle$ for some $p \in \mathcal{P}_n$, where I is the identity element of \mathcal{P}_n , the $2^n \times 2^n$ identity matrix. In this appendix and also in the next, if $p \in \mathcal{P}_n$, then $[p]$ will refer to the equivalence class of p in $\hat{\mathcal{P}}_n$, and the support of $[p]$ is defined to be the support of p . The *weight* of any element of \mathcal{P}_n or $\hat{\mathcal{P}}_n$ is defined to be the number of qubits in its support. For any $[p] \in \hat{\mathcal{P}}_n$, where $p = \eta (\sigma_0 \otimes \cdots \otimes \sigma_{n-1}) \in \mathcal{P}_n$, we define the *restriction* of p (resp. $[p]$) to qubit j for any $0 \leq j \leq n-1$, to be $\sigma_j \in \mathcal{P}_1$ (resp. $[\sigma_j] \in \hat{\mathcal{P}}_1$). An element $p \in \mathcal{P}_n$ (resp. $\hat{p} \in \hat{\mathcal{P}}_n$) is called X -type if and only if the restriction of p (resp. \hat{p}) to qubit j is either I_2 or X (resp. $[I_2]$ or $[X]$), for all $0 \leq j \leq n-1$. We similarly define Y -type and Z -type elements of \mathcal{P}_n and $\hat{\mathcal{P}}_n$.

\mathcal{P}_n is a non-Abelian group of size 4^{n+1} , and each element of \mathcal{P}_n is traceless, while we have $|\hat{\mathcal{P}}_n| = 4^n$. Any two elements $p, q \in \mathcal{P}_n$ either commute or anticommute, and we will use the notations $[p, q] = 0$ and $\{p, q\} = 0$, to denote these two cases respectively. Note that $\{p, q\}$ can also refer to the set containing the elements p and q , but this difference will either be clear from context, or we will explain the usage whenever there is a chance for confusion. For convenience, we define a function $\text{Com} : \mathcal{P}_n \times \mathcal{P}_n \rightarrow \mathbb{F}_2$, that will be useful in some places, as

$$\text{Com}(p, q) = \frac{1}{2} (1 - \text{Tr}(p q p^\dagger q^\dagger) / 2^n), \quad (60)$$

which satisfies $\text{Com}(p, q) = 0$ if and only if $[p, q] = 0$. Thus Com is a symmetric function of its arguments. Two elements $\hat{p}, \hat{q} \in \hat{\mathcal{P}}_n$ will be said to commute (resp. anticommute)⁵ if and only if for any chosen representatives $p \in \hat{p}$ and $q \in \hat{q}$, p and q commute (resp. anticommute), and we will use the same notation as above to express commutativity. For any non-empty, ordered multiset $\mathcal{H} = \{\hat{p}_1, \dots, \hat{p}_\ell\} \subseteq \hat{\mathcal{P}}_n$ such that $|\mathcal{H}| = \ell$, we define following [38], the *commutativity map* with respect to \mathcal{H} , to be the function $\text{Com}_{\mathcal{H}} : \hat{\mathcal{P}}_n \rightarrow \mathbb{F}_2^\ell$,

$$(\text{Com}_{\mathcal{H}}(\hat{q}))_i = \text{Com}(q, p_i), \text{ where } \hat{q} = [q], \hat{p}_i = [p_i], \quad \forall 1 \leq i \leq \ell, \quad (61)$$

and we say that $\hat{q} \in \hat{\mathcal{P}}_n$ generates the *commutativity pattern* $\text{Com}_{\mathcal{H}}(\hat{q})$ with respect to \mathcal{H} .

Commutativity of Paulis can also be represented in yet another equivalent form. Since the phase factors of two Paulis $p, q \in \mathcal{P}_n$ do not influence their commutativity, it is often convenient to represent a Pauli $p = \eta (\sigma_0 \otimes \cdots \otimes \sigma_{n-1})$ in symplectic notation [58] ignoring the phase factor η , that is as a binary (row) vector $v_p \in \mathbb{F}_2^{2n}$ such that for $0 \leq i \leq n-1$, $(v_p)_i = 1$ if and only if $\sigma_i \in \{X, Y\}$, and for $n \leq i \leq 2n-1$, $(v_p)_i = 1$ if and only if $\sigma_{i-n} \in \{Y, Z\}$. As binary vectors, Paulis p and q commute if and only if $v_p \Lambda v_q^\top = 0$ where $\Lambda = \begin{pmatrix} 0 & I_n \\ I_n & 0 \end{pmatrix} \in \mathbb{F}_2^{2n \times 2n}$ written using $n \times n$ blocks [58], I_n is the $n \times n$ identity matrix, and all operations are performed over \mathbb{F}_2 .

A list $\mathcal{H} = \{p_0, \dots, p_{\ell-1}\} \subseteq \mathcal{P}_n$ is called *anticommuting* (resp. *commuting*) if and only if $\{p_i, p_j\} = 0$, for all $i \neq j$ (resp. $[p_i, p_j] = 0$ for all i, j). Recall from Definition 3.2 that a CAL on n qubits is a list $\mathcal{C} = \{p_0, \dots, p_{\ell-1}\} \subseteq \mathcal{P}_n$, with the property that for distinct i and j , $\{p_i, p_j\} = 0$ if and only if $j = (i \pm 1) \bmod \ell$. Singleton lists and the empty list are considered to be anticommuting lists and CALs by convention. Similarly we also say that a list $\mathcal{H} \subseteq \hat{\mathcal{P}}_n$ is anticommuting (resp. commuting) if the resulting list of \mathcal{P}_n obtained by replacing each element in \mathcal{H} by exactly one of its representatives (which can be chosen arbitrarily) is anticommuting (resp. commuting). We define a CAL of $\hat{\mathcal{P}}_n$ analogously to be a list $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$, such that the list obtained by replacing each element of \mathcal{C} by a representative is a CAL in \mathcal{P}_n , and it is called extremal if it has length $\ell \geq 1$, and there is no CAL of same length on fewer qubits.

We will define the product of all elements in a list $\mathcal{H} \subseteq \mathcal{P}_n$ to be $\prod \mathcal{H} := \prod_{p \in \mathcal{H}} p$, (this product is ordered). Product of multisets of $\hat{\mathcal{P}}_n$ is defined in a similar fashion, except the ordering does

⁵This notion of commutativity of elements of $\hat{\mathcal{P}}_n$ is different from that due to the group operation on $\hat{\mathcal{P}}_n$. Elements of $\hat{\mathcal{P}}_n$ always commute under the group operation on $\hat{\mathcal{P}}_n$.

not matter as all elements of $\hat{\mathcal{P}}_n$ commute under the group operation. By convention the product of the empty set is defined to be the identity element $I \in \mathcal{P}_n$ (resp. $[I] \in \hat{\mathcal{P}}_n$). Given a multiset $\mathcal{H} = \{p_0, \dots, p_{\ell-1}\} \subseteq \mathcal{P}_n$, the symplectic representation of Paulis allows us to define its *dimension*. Writing each Pauli in the symplectic notation gives a matrix $H \in \mathbb{F}_2^{\ell \times 2n}$ whose each row is a Pauli in \mathcal{H} , and we define $\dim(\mathcal{H}) := \text{rank}(H)$. In words, $\dim(\mathcal{H})$ is the size of the smallest subset of \mathcal{H} that generates all its elements up to phase factors — thus it satisfies $\dim(\mathcal{H}) = \min\{|\mathcal{H}'| : \mathcal{H}' \subseteq \mathcal{H}, \langle \mathcal{H}', iI \rangle = \langle \mathcal{H}, iI \rangle\}$. We say \mathcal{H} is *independent* if and only if $\dim(\mathcal{H}) = |\mathcal{H}|$. Dimension and independence of multisets of $\hat{\mathcal{P}}_n$ are now analogously defined in terms of the dimension and independence of the multiset of \mathcal{P}_n , obtained by choosing representatives. Finally, if $\hat{p} \in \hat{\mathcal{P}}_n$, and $\hat{q} \in \hat{\mathcal{P}}_m$, the Kronecker product of \hat{p} and \hat{q} , is defined as $\hat{p} \otimes \hat{q} = [p \otimes q] \in \hat{\mathcal{P}}_{n+m}$, where $p \in \hat{p}$, and $q \in \hat{q}$.

C.2 Constructing a list of Paulis given desired commutation relations

Suppose $C \in \mathbb{F}_2^{\ell \times \ell}$ is a matrix specifying the commutation relations for an unknown list of Paulis $\{p_0, \dots, p_{\ell-1}\} \subseteq \mathcal{P}_n$ (not necessarily all distinct), that is $C_{ij} = \text{Com}(p_i, p_j)$. Hence, C must be symmetric and must have zero diagonal (i.e. $C_{ii} = 0$ for all i). In this subsection, we answer the following questions:

- (i) From C alone, can we find Paulis p_i that obey the specified commutation relations?
- (ii) What is the minimum number of qubits n required to achieve C ?

The key idea to answering these questions is to simultaneously row and column reduce the matrix C . We start by analyzing a symmetric rank-revealing factorization algorithm, which is important in the sequel. We make use of the elementary row operations $E^{(h,k)} \in \mathbb{F}_2^{\ell \times \ell}$, defined by $E_{ij}^{(h,k)} = 1$ if and only if $i = j$, or $(i, j) = (h, k)$. If $h \neq k$, when a matrix is multiplied by $E^{(h,k)}$ on the left, its k^{th} row is added to its h^{th} row, while when multiplied by $(E^{(h,k)})^\top$ on the right, its k^{th} column is added to its h^{th} column. We will refer to the latter as elementary column operations. Note also that $E^{(h,k)}$ is self-inverse. If $h = k$, $E^{(h,h)}$ equals the identity matrix. Now consider Algorithm 2, where all operations occur over the field \mathbb{F}_2 (note that Algorithm 2 may be of independent interest elsewhere).

Algorithm 2

- 1: **procedure** LBLDECOMPOSE(C) $\triangleright C \in \mathbb{F}_2^{\ell \times \ell}$ is symmetric with zero diagonal
 - 2: Set $B \leftarrow C$ and $L \leftarrow I$
 - 3: For $i = 1$ to $\ell - 1$:
 - 4: Find the smallest index j_i such that $B_{ij_i} = 1$. If none exists, set $j_i \leftarrow i$.
 - 5: While there exists $k > i$ such that $B_{kj_i} = 1$:
 - 6: $B \leftarrow E^{(k,i)} B (E^{(k,i)})^\top$ \triangleright Set $B_{kj_i} = B_{j_i k} = 0$
 - 7: $L \leftarrow L E^{(k,i)}$
 - 8: Return B and L
-

Lemma C.1. *Let $C \in \mathbb{F}_2^{\ell \times \ell}$ with $\ell \geq 1$, such that $C = C^\top$ and $C_{ii} = 0$ for all i , and consider Algorithm 2. Let $B^{(0)} = C$, and for $i \geq 1$, let $B^{(i)}$ denote the matrix B after the end of iteration i of the for-loop. For $i \geq 0$, let $B^{(i,k)}$ denote the upper-left $k \times k$ block of $B^{(i)}$, i.e. all rows and columns numbered 1 to k . Then for all $1 \leq i \leq \ell - 1$*

- (a) $B^{(i)}$ is symmetric, has zero diagonal, and $B^{(i,i+1)} = B^{(r,i+1)}$ for all $r \geq i$. Moreover the upper-left $i \times i$ block of $B^{(i-1)}$ is unchanged during for-loop iteration i .
- (b) If $j_i \neq i$, then $B_{is}^{(r)} = B_{si}^{(r)} = 0$ and $B_{ij_i}^{(r)} = B_{j_i i}^{(r)} = 1$ for all $r \geq i - 1$ and $s < j_i$. If $j_i = i$, then $B_{is}^{(r)} = B_{si}^{(r)} = 0$ for all $r \geq i - 1$ and $1 \leq s \leq \ell$.

- (c) If $j_i \neq i$, then $B_{sj_i}^{(r)} = B_{j_i s}^{(r)} = 0$, for all $r \geq i$ and $s > i$.
- (d) If $B_{rs}^{(i,i)} = 1$, then it is the only non-zero element of the r^{th} and s^{th} rows and columns of $B^{(i)}$.
- (e) Each row and column of $B^{(i,i+1)}$ contains at most a single 1.
- (f) $j_{j_i} = i$.

The output of the algorithm $L, B \in \mathbb{F}_2^{\ell \times \ell}$ satisfy $C = LBL^\top$, where L is an invertible lower-triangular matrix, B is symmetric and has zero diagonal, and each row and column of B contains at most a single 1. The algorithm runs in $O(\ell^3)$ time.

Proof. Note that by construction, throughout the algorithm we have that L is lower triangular and invertible (since the $E^{(k,i)}$ matrix inside the while loop is lower triangular and invertible), and also that $C = LBL^\top$. The claims about the output B of the algorithm follow as $B = B^{(\ell-1)}$ and from (a) and (e). Also it is clear that the worst case run time of the algorithm is $O(\ell^3)$, where we must use the sparsity of $E^{(k,i)}$ while performing the matrix multiplication inside the while loop (otherwise we will get a $O(\ell^5)$ algorithm). Note also that the symmetry of $B^{(i-1)}$ dictates that in the case $j_i = i$, the while loop will exit immediately.

We now prove (a)-(f).

- (a) Let \tilde{L} denote the matrix L after the end of for-loop iteration i of Algorithm 2. Then $C = \tilde{L}B^{(i)}\tilde{L}^\top$ and so $B^{(i)}$ is symmetric, for all $1 \leq i \leq \ell - 1$. $B^{(i)}$ has zero diagonal because for all $1 \leq r \leq \ell$

$$B_{rr}^{(i)} = \sum_{xy} \tilde{L}_{rx}^{-1} C_{xy} \tilde{L}_{ry}^{-1} = \sum_x \tilde{L}_{rx}^{-1} C_{xx} + \sum_{x < y} \tilde{L}_{rx}^{-1} \tilde{L}_{ry}^{-1} (C_{xy} + C_{yx}) = 0. \quad (62)$$

Now note that during any for-loop iteration i , in each iteration of the while loop, multiplying $E^{(k,i)}$ on the left does not change any row of B with index 1 to i , while multiplying by $(E^{(k,i)})^\top$ on the right does not change any column of $E^{(k,i)}B$ with index 1 to i . It immediately follows that the upper-left $i \times i$ block of $B^{(i-1)}$ is unchanged during for-loop iteration i . We also easily deduce from this that $B^{(i,i+1)} = B^{(r,i+1)}$ for all $r \geq i$.

- (b) First notice that from the beginning of for-loop iteration i , all rows numbered 1 to i are unaffected by elementary row operations alone. If $j_i = i$, it means that at the start of for-loop iteration i , the i^{th} row of $B^{(i-1)}$ is zero. Any elementary column operations will not change this row subsequently. If instead $j_i \neq i$, then at the start of for-loop iteration i we have $B_{i_s}^{(i-1)} = 0$ for all $s < j_i$, and $B_{ij_i}^{(i-1)} = 1$. In this case, any subsequent elementary column operations will not change columns 1 to j_i of the i^{th} row. The conclusion now follows by symmetry using (a).

For parts (c)-(f), we will first prove a claim. We claim that for all $1 \leq i \leq \ell - 1$, Algorithm 2 maintains the following invariants after for-loop iteration i :

- (i) For all $1 \leq k \leq i$, $B_{sj_k}^{(i)} = B_{j_k s}^{(i)} = 0$, for all $s > k$.
- (ii) For every $1 \leq r, s \leq i$ such that $B_{rs}^{(i)} = 1$, the only non-zero element of the r^{th} and s^{th} rows and columns of $B^{(i)}$ are $B_{rs}^{(i)} = B_{sr}^{(i)} = 1$. If in addition $r = i$, then the i^{th} row and column of $B^{(i-1)}$ do not change during the iteration, while if $1 \leq r, s < i$, then the r^{th} row and the s^{th} column of $B^{(i-1)}$ do not change during iteration i .
- (iii) If $j_i \leq i$, then $j_{j_i} = i$.

Proof of claim. It is helpful to make an observation about what happens during for-loop iteration i , assuming $j_i \neq i$. One then easily checks that in an iteration of the while loop, the effect of the update $B \leftarrow E^{(k,i)}B(E^{(k,i)})^\top$ on the j_i^{th} column of B is to set the (k, j_i) entry to zero without affecting any other entries, for some $k > i$. By symmetry the only effect on the j_i^{th} row of B is to

set the (j_i, k) entry to zero. Thus at the end of for-loop iteration i , the j_i^{th} column of $B^{(i)}$ satisfies $B_{sj_i}^{(i)} = 0$ for all $s > i$, and $B_{ij_i}^{(i)} = 1$.

Returning to the proof of the claim, notice that for $i = 1$ (i)-(iii) are clearly true: if $j_1 \neq 1$ then (i) holds by the observation in the previous paragraph, and if $j_1 = 1$ then it is true by (b), (ii) is vacuous as $B_{11}^{(1,1)} = 0$ by (a), and for (iii) note that $j_1 \leq 1$ implies $j_1 = 1$, thus $j_{j_1} = j_1 = 1$. Now for the inductive hypothesis, assume that the invariants are true after for-loop iteration $i - 1$, for some $1 < i \leq \ell - 1$, and we want to show that they are also true after iteration i . We do this separately in three steps.

Step 1: Let us first show that (iii) holds. If $j_i = i$ then it clearly holds, so assume $j_i \leq i - 1$. Now at the start of for-loop iteration i we have by symmetry $B_{j_i i}^{(i-1)} = 1$. Assume for contradiction that $j_{j_i} \neq i$, and consider the situation during for-loop iteration j_i . If $j_{j_i} > i$, then by (b) we would get $B_{j_i j_i}^{(i-1)} = 0$, so it must be the case that $j_{j_i} < i$. But then we get $B_{j_i j_i}^{(i-1)} = 1$, again by (b). Thus the j_i^{th} row of $B^{(i-1)}$ contains two non-zero elements, which contradicts (ii) of the inductive hypothesis.

Step 2: We now show that (ii) holds after for-loop iteration i , first in the limited setting $1 \leq r, s < i$. By symmetry, it suffices to prove that whenever $B_{rs}^{(i)} = 1$, it is the only non-zero element of the r^{th} row of $B^{(i-1)}$ which does not change during the iteration, since one can repeat the same argument for the element $B_{sr}^{(i)} = 1$ in the s^{th} row. Assuming such an element $B_{rs}^{(i)} = 1$ exists, we have from (a) that $B_{rs}^{(i-1)} = 1$, and by (ii) of the inductive hypothesis it is the only non-zero element of the r^{th} row of $B^{(i-1)}$. Now for every while loop iteration within for-loop iteration i , the r^{th} row of $B^{(i-1)}$ is not modified by elementary row operations alone, by the same argument as in the proof of (b), or by elementary column operations as $B_{ri}^{(i-1)} = 0$ (this continues to be true by (a) during the iteration). Thus the r^{th} row of $B^{(i-1)}$ is unchanged proving this case.

It remains to prove that (ii) also holds for any non-zero element in the i^{th} row or column of $B^{(i,i)}$. For this consider the i^{th} row of $B^{(i)}$. If $j_i \geq i$, then we are again done using (b) as the i^{th} row and column of $B^{(i,i)}$ is zero, so suppose $j_i < i$. We will show that in this case the only non-zero element of the i^{th} row and j_i^{th} column of $B^{(i)}$ is $B_{ij_i}^{(i)} = 1$, and the i^{th} row of $B^{(i-1)}$ is unchanged during the for-loop, and then we are done by symmetry. We now have a series of consequences. By invariant (iii), which we have already proved above to hold after for-loop iteration i , we have $j_{j_i} = i$. Applying (i) of the inductive hypothesis then gives $B_{si}^{(i-1)} = 0$ for all $s > j_i$, and using symmetry of $B^{(i-1)}$ we conclude that the only non-zero element of the i^{th} row of $B^{(i-1)}$ is $B_{ij_i}^{(i-1)} = 1$. Next using exactly the same argument as in the last paragraph (for the case $1 \leq r, s < i$), we conclude that the i^{th} row of $B^{(i-1)}$ is unchanged during for-loop iteration i . Now consider the j_i^{th} column of $B^{(i-1)}$. By (ii) of the inductive hypothesis and (a) we first have $B_{sj_i}^{(i-1)} = B_{sj_i}^{(i)} = 0$ for all $s < i$, and by the observation stated above we also have $B_{sj_i}^{(i)} = 0$ for all $s > i$. This proves that the only non-zero element in the j_i^{th} column of $B^{(i)}$ is $B_{ij_i}^{(i)} = 1$.

Step 3: Finally we show that (i) holds. If $k = i$, then the case $j_i \neq i$ is already true by the observation made above, while the case $j_i = i$ is true by (b). So fix some $1 \leq k < i$. If $j_k \leq i$ then by (b) we have $B_{k,j_k}^{(i)} = 1$, so we are done because invariant (ii) holds after iteration i . If $j_i = i$, then $B^{(i-1)} = B^{(i)}$ and we are done using (i) of the inductive hypothesis. So the interesting case is when $j_i \neq i$ and $j_k > i$. Now there are two subcases $j_i > k$ and $j_i < k$ which need to be handled separately (note that the case $j_i = k$ is ruled out by (b)). First suppose $j_i > k$, and consider the j_k^{th} column of $B^{(i-1)}$. Since by (i) of the inductive hypothesis we have $B_{ij_k}^{(i-1)} = B_{j_i j_k}^{(i-1)} = 0$, elementary row operations in the while loops during for-loop iteration i do not change this column (this uses $B_{ij_k}^{(i-1)} = 0$), while elementary column operations never act on the column as $B_{j_k j_i}^{(i-1)} = 0$ (this uses symmetry and $B_{j_i j_k}^{(i-1)} = 0$). Thus the j_k^{th} columns are the same for both $B^{(i-1)}$ and $B^{(i)}$, and this case is proved. Finally consider the case $j_i < k$. Notice that since $j_i < k < i$ and we have proved that (iii) holds after for-loop iteration i , we have $j_{j_i} = i$, and then by (i) of the induction hypothesis we get $B_{si}^{(i-1)} = 0$ for all $s > j_i$. Moreover since we also proved that (ii) holds, we know

that the i^{th} column of $B^{(i-1)}$ does not change during the for-loop. Now again consider the j_k^{th} column of $B^{(i-1)}$, focusing only on the rows numbered k to ℓ . As before, this part of the column is unchanged by elementary row operations during for-loop iteration i , so we only need to argue that elementary column operations also do not affect it. But this immediately follows from the pattern of zeros in the i^{th} column of $B^{(i-1)}$, which finishes the proof of the claim.

We now finish up the proof of the lemma.

- (c) This follows directly from (i) of the above claim.
- (d) This is a sub-part of invariant (ii) of the claim.
- (e) Suppose this is not true, and let $k = i + 1$. Then by (ii) of the claim and symmetry of $B^{(i)}$, we must have at least two non-zero elements in the k^{th} row of $B^{(i,k)}$, one of which must be $B_{kj_k}^{(i,k)}$, and let the other non-zero element be $B_{kh}^{(i,k)}$, and note that $j_k < h < k$ (by definition of j_k). But by invariant (iii) applied to the k^{th} for-loop, we also have $j_{j_k} = k$. Since $j_k \leq i$, we can now apply (i) of the claim, but this gives a contradiction as $B_{hk}^{(i,k)} = 1$ by symmetry.
- (f) If $j_i \leq i$, then this is just the statement of (iii) of the claim. So let $j_i > i$, and suppose $j_{j_i} \neq i$. Then after for-loop iteration j_i , we have $B_{j_i j_i}^{(j_i)} = 1 = B_{i j_i}^{(j_i)}$, by (b) and (i) of the claim. But this contradicts invariant (ii). □

From the properties of B in the above lemma, we deduce the following additional simple lemma.

Lemma C.2. *Let $B \in \mathbb{F}_2^{\ell \times \ell}$ be a symmetric matrix with zero diagonal and at most a single 1 in each row and column. Let $|B| = \sum_{ij} B_{ij}$ be the total number of ones. Then $|B|$ is even and $\text{rank}(B) = |B|$. In particular $\text{rank}(C) = |B|$ is even.*

Proof. It is clear that $|B|$ is even due to symmetry of B and it having zero diagonal. Since each row has at most a single 1, B has exactly $|B|$ non-zero rows, all of which are independent since each column of B has at most a single 1. This shows $\text{rank}(B) = |B|$. Finally, since $C = LBL^\top$ and L is invertible, $\text{rank}(C) = \text{rank}(B)$. □

Let us apply this linear algebra to our study of Paulis. Let $P \in \mathbb{F}_2^{\ell \times 2n}$ be the matrix representing the list of Paulis $\{p_0, p_1, \dots, p_{\ell-1}\}$ (which we are looking for) in symplectic notation. Therefore, $\{p_0, p_1, \dots, p_{\ell-1}\}$ satisfies the commutation relations specified by C if and only if $P\Lambda P^\top = C$. If $C = 0$, it suffices to just use 1 qubit and we can take $p_0 = \dots = p_{\ell-1} = I_2$. So let us assume that $C \neq 0$. Let us suppose that we decompose $C = LBL^\top$ as in Lemma C.1. As we now explain, we can easily create a list of Paulis $\{q_0, q_1, \dots, q_{\ell-1}\}$ with commutation relations given by B , or equivalently a matrix $Q \in \mathbb{F}_2^{\ell \times 2n}$ such that $Q\Lambda Q^\top = B$. For each pair (i, j) such that $i < j$ and $B_{ij} = 1$, we introduce a qubit (call it qubit (i, j)), and let $q_i = X_{(i,j)}$ and $q_j = Z_{(i,j)}$ act on that qubit. For all zero rows i of B , we set $q_i = I_2$. We have thus used $n = \text{rank}(B)/2 = \text{rank}(C)/2$ qubits to make the Paulis $\{q_0, q_1, \dots, q_{\ell-1}\}$, and one easily verifies that this list of Paulis satisfy the commutation relations specified by B . Now $Q\Lambda Q^\top = B$ implies $(LQ)\Lambda(LQ)^\top = LBL^\top = C$, so $P = LQ$ represents a list of Paulis satisfying the commutation relations specified by C . It is also easy to check that $\text{rank}(P) = \text{rank}(LQ) = \text{rank}(Q) = 2n = \text{rank}(C)$. This discussion leads to the following theorem.

Theorem C.3. *Let $C \in \mathbb{F}_2^{\ell \times \ell}$ be symmetric with zero diagonal. A list of n -qubit Paulis $\mathcal{C} = \{p_0, p_1, \dots, p_{\ell-1}\} \subseteq \mathcal{P}_n$ such that $C_{ij} = \text{Com}(p_i, p_j)$ for all $0 \leq i, j \leq \ell - 1$, exists if and only if $n \geq \max(1, \text{rank}(C)/2)$. Moreover, when \mathcal{C} exists, $\dim(\mathcal{C}) \geq \text{rank}(C)$, with equality if $n = \text{rank}(C)/2$.*

Proof. If $C = 0$, then as argued above $n = 1$ is the minimum number of needed qubits to achieve the commutation relations specified by C ; thus the first statement is true. So assume $C \neq 0$, which implies $\text{rank}(C) \geq 2$ by Lemma C.2. The discussion in the previous paragraph shows the construction of \mathcal{C} when $n = \text{rank}(C)/2$, and this clearly suffices as if $n > \text{rank}(C)/2$, then one can simply act by I_2 on the extra qubits. We argue the converse, i.e. for the minimality of n ,

by contradiction. Let $P \in \mathbb{F}_2^{\ell \times 2n}$ be the matrix such that $PAP^\top = C$ with $n < \text{rank}(C)/2$. Let $C = LBL^\top$ from Lemma C.1. Then $L^{-1}P$ represents a list of Paulis with commutation relations given by B . But by the structure of B , this implies $\text{rank}(B)/2 = \text{rank}(C)/2$ anticommuting pairs of Paulis (i.e. the Paulis can be grouped in pairs that anticommute with each other but commute with all other Paulis in the list), which is known to be impossible on fewer than $\text{rank}(C)/2$ qubits.⁶

Now suppose $\mathcal{C} \subseteq \mathcal{P}_n$ exists satisfying the commutation relations specified by C , where we already know $n \geq \max(1, \text{rank}(C)/2)$. Representing the Paulis in \mathcal{C} by a matrix $P \in \mathbb{F}_2^{\ell \times 2n}$ again gives $PAP^\top = C$. Thus $\text{rank}(C) \leq \text{rank}(P) = \dim(\mathcal{C})$. If in addition $2n = \text{rank}(C)$, then $\text{rank}(P) \leq \min(\ell, \text{rank}(C)) \leq \text{rank}(C)$, which implies $\dim \mathcal{C} = \text{rank}(C)$. \square

We demonstrate the use of Theorem C.3 with a couple of examples, the first of which is well known, and the second one concerns CALs, which will be discussed much more thoroughly in the next subsection.

Example 1. *Suppose*

$$C = \begin{pmatrix} 0 & 1 & 1 & & 1 \\ 1 & 0 & 1 & \dots & 1 \\ 1 & 1 & 0 & & 1 \\ & \vdots & & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 0 \end{pmatrix} \in \mathbb{F}_2^{\ell \times \ell}, \quad \ell \geq 2, \quad (63)$$

so that we are looking for a list of anticommuting Paulis. It is well known that

$$\text{rank}(C)/2 = \begin{cases} (\ell - 1)/2, & \ell \text{ odd} \\ \ell/2, & \ell \text{ even} \end{cases}. \quad (64)$$

Then, the construction from Theorem C.3 with $n = \text{rank}(C)/2$, returns the Jordan-Wigner [59] list — i.e. for $h \in \{0, 1, \dots, 2\lfloor \ell/2 \rfloor - 1\}$,

$$p_h = \begin{cases} X_{h/2} \prod_{i=0}^{h/2-1} Y_i, & h \text{ even} \\ Z_{(h-1)/2} \prod_{i=0}^{(h-3)/2} Y_i, & h \text{ odd} \end{cases}, \quad (65)$$

and, if ℓ is odd, the final Pauli $p_{\ell-1} = \prod_{i=0}^{n-1} Y_i$.

Example 2. *Suppose*

$$C = \begin{pmatrix} 0 & 1 & 0 & & 1 \\ 1 & 0 & 1 & \dots & 0 \\ 0 & 1 & 0 & & 0 \\ & \vdots & & \ddots & \vdots \\ 1 & 0 & 0 & \dots & 0 \end{pmatrix} \in \mathbb{F}_2^{\ell \times \ell}, \quad \ell \geq 3, \quad (66)$$

so that we are looking for a cyclically anticommuting list of Paulis. We note that the rank of C satisfies

$$\text{rank}(C)/2 = \begin{cases} (\ell - 1)/2, & \ell \text{ odd} \\ (\ell - 2)/2, & \ell \text{ even} \end{cases}. \quad (67)$$

The construction from Theorem C.3, again with $n = \text{rank}(C)/2$ qubits, now gives the list

$$\begin{cases} \left\{ X_0, Z_0, X_0 X_1, Z_1, \dots, X_{n-2} X_{n-1}, Z_{n-1}, Y_{n-1} \prod_{i=0}^{n-2} Z_i \right\}, & \ell \text{ odd}, \\ \left\{ X_0, Z_0, X_0 X_1, Z_1, \dots, X_{n-2} X_{n-1}, Z_{n-1}, X_{n-1}, \prod_{i=0}^{n-1} Z_i \right\}, & \ell \text{ even}. \end{cases} \quad (68)$$

By Theorem C.3, $\dim(\mathcal{C}) = 2n$ in the preceding examples, so it ensures that $\langle \mathcal{C}, iI \rangle = \mathcal{P}_n$. However, we can even generalize the construction to include $\dim(\mathcal{C})$ as an input parameter.

⁶Notice that, ignoring phases, the size of the group generated by n anticommuting pairs of Paulis is 4^n , since every paired Pauli is necessarily independent. Therefore, the Paulis must act on at least n qubits.

Theorem C.4. *Suppose $C \in \mathbb{F}_2^{\ell \times \ell}$ is symmetric and has zero diagonal, and let $k \in \mathbb{Z}$ such that $\text{rank}(C) \leq k \leq \ell$. A list of Paulis $\mathcal{C} = \{p_0, p_1, \dots, p_{\ell-1}\} \subseteq \mathcal{P}_n$ satisfying (a) $C_{ij} = \text{Com}(p_i, p_j)$ for all $0 \leq i, j \leq \ell - 1$, and (b) $\dim(\mathcal{C}) = k$ exists if and only if $n \geq \max(1, k - \text{rank}(C)/2)$.*

Proof. First consider the case $C = 0$, then as argued in Theorem C.3 we need $n \geq 1$. But it is well known that on n qubits, the maximum size of an independent commuting list of Paulis is n , so $k \leq n$, and this also proves the converse for this case. For the rest of the proof assume $C \neq 0$, which implies $\text{rank}(C) \geq 2$ by Lemma C.2. Let $[\ell] = \{0, 1, \dots, \ell - 1\}$. We begin the same way as Theorem C.3, decomposing $C = LBL^\top$ and creating a set of Paulis $q_i = X_{(i,j)}$, $q_j = Z_{(i,j)}$ for all pairs (i, j) such that $i < j$ and $B_{ij} = 1$. Let $\mathcal{I} \subseteq [\ell]$ be the indices of the rows in B that are all zero, and partition \mathcal{I} into two sets \mathcal{I}_1 and \mathcal{I}_2 such that $|\mathcal{I}_1| = k - \text{rank}(C)$. For each element $i \in \mathcal{I}_1$ we introduce a qubit (labeled by i) and set $q_i = Z_i$. We now have $n = k - \text{rank}(C)/2$ qubits. We form $Q \in \mathbb{F}_2^{\ell \times 2n}$ representing the Pauli list $\{q_0, q_1, \dots, q_{\ell-1}\}$ as symplectic vectors and calculate $P = LQ$ representing the desired list $\mathcal{C} = \{p_0, p_1, \dots, p_{\ell-1}\}$. Note $Q\Lambda Q^\top = B$ by construction and so $P\Lambda P^\top = LBL^\top = C$. Also note that since $\dim(\{q_0, q_1, \dots, q_{\ell-1}\}) = \text{rank}(Q) = k$ by construction and L is invertible, $\dim(\mathcal{C}) = \text{rank}(P) = k$. This shows the existence of \mathcal{C} when $n = k - \text{rank}(C)/2$, and also covers the case $n \geq k - \text{rank}(C)/2$, since one can act on the extra qubits by identity.

It remains to prove the converse. This argument by contradiction is analogous to that in Theorem C.3. If $P \in \mathbb{F}_2^{\ell \times 2n'}$ represents the list of Paulis on $n' < n$ qubits satisfying $P\Lambda P^\top = C$ and $\text{rank}(P) = k \geq \text{rank}(C)$, then $Q = L^{-1}P$ represents the list of n' -qubit Paulis such that $Q\Lambda Q^\top = B$ and $\text{rank}(Q) = k$. However, the structure of B implies that there is an independent and fully commuting subset of $\{q_0, q_1, \dots, q_{\ell-1}\}$ with size $k - \text{rank}(C)/2$ which is again known to be impossible on $n' < k - \text{rank}(C)/2$ qubits [38]. \square

C.3 Properties of cyclically anticommuting lists of Paulis

We will work exclusively in this subsection with the group $\hat{\mathcal{P}}_n$, to not have to keep track of the phase factors of the Paulis. CALs of \mathcal{P}_n were introduced in Definition 3.2, and CALs of $\hat{\mathcal{P}}_n$ were defined in Section C.1. Here we present some more properties of CALs of $\hat{\mathcal{P}}_n$, that we know of currently. Translating these results to CALs of \mathcal{P}_n is obvious. We recall the commutation relations that a CAL satisfies: if $\mathcal{C} = \{\hat{p}_0, \hat{p}_1, \dots, \hat{p}_{k-1}\} \subseteq \hat{\mathcal{P}}_n$ is a CAL, then for distinct i and j

$$\begin{cases} \{\hat{p}_i, \hat{p}_j\} = 0, & \text{if } i = j \pm 1 \pmod k \\ [\hat{p}_i, \hat{p}_j] = 0, & \text{otherwise.} \end{cases} \quad (69)$$

This subsection is organized as follows: Lemma C.5 - Lemma C.6 points out the relationship between CALs and anticommuting lists, and shows that all elements of a CAL must be unique if the CAL has length greater than four, Theorem C.8 - Corollary C.11 addresses the question of dimension and independence of a CAL, and finally Lemma C.13 - Theorem C.15 looks at whether it is possible to create a CAL, where the representative of each element of the CAL is a Kronecker product of only X and Z Pauli matrices, and possibly the 2×2 identity matrix I_2 .

Recall from the proof of Theorem 3.4, that on $n = 1$ qubits, $\mathcal{C} = \{[X], [Z], [X], [Z]\}$ is a CAL. Clearly this has repeating elements. A natural question is whether it is possible to have CALs of longer length with repeating elements. The next lemma completely addresses this question.

Lemma C.5. *Let $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$ be a CAL, for $n \geq 1$. If \mathcal{C} has repeating elements, then \mathcal{C} has one of the following forms*

$$\mathcal{C} = \begin{cases} \{\hat{p}, \hat{q}, \hat{p}, \hat{q}\} & \text{with } \{\hat{p}, \hat{q}\} = 0, \\ \{\hat{p}, \hat{q}, \hat{p}, \hat{s}\} & \text{with } \{\hat{p}, \hat{q}\} = \{\hat{p}, \hat{s}\} = [\hat{q}, \hat{s}] = 0, \\ \{\hat{p}, \hat{q}, \hat{s}, \hat{q}\} & \text{with } \{\hat{p}, \hat{q}\} = \{\hat{q}, \hat{s}\} = [\hat{p}, \hat{s}] = 0, \end{cases} \quad (70)$$

where $\hat{p}, \hat{q}, \hat{s} \in \hat{\mathcal{P}}_n$, and distinct. The last two cases can only happen if $n > 1$.

Proof. First notice that if \mathcal{C} has any of the forms in Eq. (70), then the commutation relations imply that \mathcal{C} is indeed a CAL. Now we claim that if \mathcal{C} has at least one repeating element, then $|\mathcal{C}| = 4$. We first prove this claim. If $|\mathcal{C}| \leq 1$, then there can obviously be no repeating element. If $2 \leq |\mathcal{C}| \leq 3$, then \mathcal{C} is an anticommuting list, and so again there can be no repeating element. Now suppose that $|\mathcal{C}| \geq 5$, and $\mathcal{C} = \{\hat{p}_0, \dots, \hat{p}_{k-1}\}$ with repeating elements $\hat{p}_\ell = \hat{p}_{\ell'}$, for $0 \leq \ell < \ell' \leq k$. Notice first that $\ell' \notin \{\ell + 1, (\ell - 1) \bmod k\}$, because that would imply that \hat{p}_ℓ anticommutes with itself. Then there must exist an element $\hat{p}_m \in \mathcal{C}$, $m \in \{\ell + 1, (\ell - 1) \bmod k\}$, such that $\{\hat{p}_\ell, \hat{p}_m\} = 0$, and $[\hat{p}_{\ell'}, \hat{p}_m] = 0$. The existence of \hat{p}_m follows from the fact that \mathcal{C} is a CAL, and $|\mathcal{C}| \geq 5$. But this is a contradiction because it implies that \hat{p}_m commutes and anticommutes with the same element, and therefore $|\mathcal{C}| < 5$. This proves the claim.

Using the claim, we can suppose that $\mathcal{C} = \{\hat{p}, \hat{q}, \hat{r}, \hat{s}\}$, for $\hat{p}, \hat{q}, \hat{r}, \hat{s} \in \hat{\mathcal{P}}_n$, where all of them may not be distinct, and satisfies the CAL commutation relations in Eq. (69). We already know that $\hat{p} \neq \hat{q}$, $\hat{q} \neq \hat{r}$, $\hat{r} \neq \hat{s}$, and $\hat{s} \neq \hat{p}$, because otherwise Eq. (69) is violated. Now there are two cases: (i) $\hat{p} = \hat{r}$, and (ii) $\hat{p} \neq \hat{r}$. Assuming the first case $\hat{p} = \hat{r}$, we have two subcases: either $\hat{q} = \hat{s}$ which gives that $\mathcal{C} = \{\hat{p}, \hat{q}, \hat{p}, \hat{q}\}$, or $\hat{q} \neq \hat{s}$ which gives $\mathcal{C} = \{\hat{p}, \hat{q}, \hat{p}, \hat{s}\}$, and both are thus of the forms in Eq. (70). Next assume the second case $\hat{p} \neq \hat{r}$, and thus all \hat{p}, \hat{q} , and \hat{r} are distinct. Since there is at least one repeating element in \mathcal{C} , this means that \hat{s} equals either \hat{p}, \hat{q} , or \hat{r} . But we already argued above that $\hat{p} \neq \hat{s}$ and $\hat{r} \neq \hat{s}$, and so in fact $\hat{q} = \hat{s}$. This gives that $\mathcal{C} = \{\hat{p}, \hat{q}, \hat{r}, \hat{q}\}$, which is the third form in Eq. (70). The proof is finished by noting that when $n = 1$, only the first case in Eq. (70) can happen. \square

The next lemma concerns the maximum possible length of a CAL on n qubits. The arguments in the proof of parts (b) and (e) of the lemma already appear in the proof of Theorem 3.4, when the length of the CAL is at least three. We record the lemma here for completeness, as it covers the case when the length is less than three. Also part (c) shows how to get a CAL from an anticommuting list of $\hat{\mathcal{P}}_n$.

Lemma C.6. *Let $\mathcal{C} = \{\hat{p}_0, \dots, \hat{p}_{\ell-1}\} \subseteq \hat{\mathcal{P}}_n$ be a CAL, and $|\mathcal{C}| \geq 1$.*

- (a) *The list $\mathcal{C}' = \{\hat{p}'_0, \dots, \hat{p}'_{\ell-1}\}$ defined as $\hat{p}'_i = \hat{p}_j$ with $j = (i + r) \bmod \ell$, is also a CAL for all translations $r \in \mathbb{Z}$.*
- (b) *If $|\mathcal{C}| = 1$, let $\mathcal{A} = \emptyset$, otherwise let $\mathcal{A} = \{\hat{q}_0, \dots, \hat{q}_{\ell-2}\}$ defined as $\hat{q}_i = \prod_{j=0}^i \hat{p}_j$, for all $0 \leq i \leq \ell - 2$. Then \mathcal{A} is an anticommuting list. Thus an anticommuting list exists of size $|\mathcal{C}| - 1$.*
- (c) *Let $\mathcal{A} = \{\hat{q}_0, \dots, \hat{q}_{k-1}\} \subseteq \hat{\mathcal{P}}_n$ be an anticommuting list, and $|\mathcal{A}| \geq 2$. If we define $\mathcal{D} = \{\hat{d}_0, \dots, \hat{d}_k\}$ by $\hat{d}_0 = \hat{q}_0$, $\hat{d}_k = \hat{q}_{k-1}$, and $\hat{d}_{i+1} = \hat{q}_i \hat{q}_{i+1}$ for all $0 \leq i \leq k - 2$, then \mathcal{D} is a CAL.*
- (d) *$|\mathcal{C}| \leq 2n + 2$. Thus there exists a CAL on n qubits of every length $0 \leq \ell' \leq 2n + 2$.*
- (e) *The minimum number n of qubits needed to create a CAL of length ℓ' is*

$$n = \begin{cases} 1 & \text{if } \ell' \leq 2, \\ (\ell' - 1)/2 & \text{if } \ell' > 2 \text{ and odd,} \\ (\ell' - 2)/2 & \text{if } \ell' > 2 \text{ and even.} \end{cases} \quad (71)$$

- (f) *$\mathcal{C} = \{[I]\}$ if and only if $[I] \in \mathcal{C}$.*

Proof. (a) This is easily checked and follows from the commutation relations of the elements of \mathcal{C} .

- (b) If $|\mathcal{C}| = 1$, $\mathcal{A} = \emptyset$ is anticommuting by convention. Otherwise take any two distinct elements $\hat{q}_i, \hat{q}_j \in \mathcal{A}$, and without loss of generality assume that $i < j$, and thus $\hat{q}_j = \hat{q}_i \prod_{k=i+1}^j \hat{p}_k$. From the commutation relations of CAL, we have that $\{\hat{p}_i, \hat{p}_{i+1}\} = 0$, $[\hat{p}_i, \hat{p}_{k'}] = 0$ for all $i + 1 < k' \leq j$, and $[\hat{p}_k, \hat{p}_{k'}] = 0$ for all $0 \leq k < i$, and $i + 1 \leq k' \leq j$. Then $\hat{q}_i \hat{q}_j = \hat{q}_i (\hat{q}_i \prod_{k=i+1}^j \hat{p}_k) = -(\hat{q}_i \prod_{k=i+1}^j \hat{p}_k) \hat{q}_i = -\hat{q}_j \hat{q}_i$. Thus \mathcal{A} is an anticommuting list of size $|\mathcal{C}| - 1$.

- (c) As in (b), one easily checks that the elements of \mathcal{D} satisfy the CAL commutation relations of Eq. 69.
- (d) For the sake of contradiction, assume that $|\mathcal{C}| \geq 2n + 3$. Then by (b) there exists an anticommuting list \mathcal{A} of size at least $2n + 2$. But it is a well-known fact that the largest size of an anticommuting list of $\hat{\mathcal{P}}_n$ is $2n + 1$ (see for example [38] and Proposition 9 in [60]), which is a contradiction. For the last part, if $\ell' \leq 2$, then choose any anticommuting list of size ℓ' . If $\ell' > 2$, choose an anticommuting list of size $\ell' - 1$, and then apply (c).
- (e) The case $\ell' \geq 3$ follows from (d) (and has been proved previously). For the case $\ell' \leq 2$, note that the set $\{[X]\}$ is a CAL of length 1, and the set $\{[X], [Y]\}$ is a CAL of length 2.
- (f) Suppose $[I] \in \mathcal{C}$ and $\mathcal{C} \neq \{[I]\}$. Then \mathcal{C} must contain an element \hat{p} that anticommutes with $[I]$ which is impossible. □

It is clear from the above lemma that extremal CALs of length $\ell \geq 3$ correspond precisely to the last two cases of Eq. (71), while extremal CALs of length $1 \leq \ell \leq 2$ correspond to the first case. We now turn to the question of the dimension of a CAL. For the next few proofs, it will be useful to have the following result, that is proved in Section 4 of [38]. It uses the notion of commutativity maps in parts (e)-(g), that we defined in Section C.1.

Lemma C.7. *Let $\mathcal{A} = \{\hat{p}_0, \dots, \hat{p}_{\ell-1}\} \subseteq \hat{\mathcal{P}}_n$ be an anticommuting list, and let $Com_{\mathcal{A}}$ denote the commutativity map with respect to \mathcal{A} . Then*

- (a) $\prod \mathcal{A} \neq [I]$ implies \mathcal{A} is an independent generator of the subgroup $\langle \mathcal{A} \rangle$ of order 2^ℓ .
- (b) For any non-empty list $\mathcal{H} \subset \mathcal{A}$, we have $\prod \mathcal{H} \neq [I]$, so \mathcal{H} is independent.
- (c) $\prod \mathcal{A} = [I]$ implies ℓ is odd.
- (d) Suppose \mathcal{A} is non-empty, independent and ℓ is even. Then for every $v \in \mathbb{F}_2^\ell$, there exists a unique element $\hat{p} \in \langle \mathcal{A} \rangle$ such that $Com_{\mathcal{A}}(\hat{p}) = v$.
- (e) Suppose \mathcal{A} is independent, $\mathcal{A} \neq \{[I]\}$, and ℓ is odd. Then for every $v \in \mathbb{F}_2^\ell$ containing an odd number of zeros, there exists exactly two elements $\hat{p}, \hat{q} \in \langle \mathcal{A} \rangle$ such that $Com_{\mathcal{A}}(\hat{p}) = Com_{\mathcal{A}}(\hat{q}) = v$.
- (f) Suppose \mathcal{A} is independent, $\mathcal{A} \neq \{[I]\}$, and ℓ is odd. Then each coset \mathcal{H} of $\langle \mathcal{A} \rangle$ in $\hat{\mathcal{P}}_n$ has the property that either for all $\hat{q} \in \mathcal{H}$, $Com_{\mathcal{A}}(\hat{q})$ has an odd number of zeros, or for all $\hat{q} \in \mathcal{H}$, $Com_{\mathcal{A}}(\hat{q})$ has an even number of zeros. Moreover the number of cosets of each type are equal.

We now give the answer to the question of the dimension of a CAL. The next two theorems cover the cases when the CAL has lengths odd and even respectively, and the corollary after that specializes these results to the case of an extremal CAL. It is important to remember for the next proofs that by Lemma C.6(f), whenever $[I] \notin \mathcal{C} \subseteq \hat{\mathcal{P}}_n$, the dimension of a CAL \mathcal{C} is the minimum number of elements of \mathcal{C} that generates $\langle \mathcal{C} \rangle$.

Theorem C.8. *Let $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$ be a CAL, and suppose $|\mathcal{C}|$ is odd.*

- (a) *The dimension of \mathcal{C} satisfies*

$$\dim(\mathcal{C}) = \begin{cases} |\mathcal{C}| & \text{if } \prod \mathcal{C} \neq [I], \\ |\mathcal{C}| - 1 & \text{if } \prod \mathcal{C} = [I]. \end{cases} \quad (72)$$

- (b) *If $|\mathcal{C}| \geq 3$, it holds that $\mathcal{C} \setminus \{\hat{p}\}$ is independent for every $\hat{p} \in \mathcal{C}$.*
- (c) *For all non-empty multisets $\mathcal{H} \subset \mathcal{C}$, we have $\prod \mathcal{H} \neq [I]$.*

Proof. In this proof let $\mathcal{C} = \{\hat{p}_0, \dots, \hat{p}_{2k}\}$ where $k \geq 0$, so $|\mathcal{C}| = 2k + 1$. The case $\mathcal{C} = \{[I]\}$ is true for part (a) as $\dim(\{[I]\}) = 0$, and also obvious for parts (b)-(c). So for the rest of the proof assume that $\mathcal{C} \neq \{[I]\}$.

(a) Since $\dim(\mathcal{C}) \leq |\mathcal{C}|$, it is easily seen that proving Eq. (72), is equivalent to proving the following two conditions:

- (i) $\dim(\mathcal{C}) \geq |\mathcal{C}| - 1$,
- (ii) $\dim(\mathcal{C}) = |\mathcal{C}| - 1$ if and only if $\prod \mathcal{C} = [I]$.

If $|\mathcal{C}| = 1$, then $\dim(\mathcal{C}) = 1$, and $\prod \mathcal{C} \neq [I]$, and so (i) and (ii) are satisfied. Now assume $k \geq 1$. Construct the anticommuting list $\mathcal{A} = \{\hat{q}_0, \dots, \hat{q}_{2k-1}\}$ similarly as in Lemma C.6(b). Notice that because $|\mathcal{A}|$ is even, it follows from Lemma C.7(c) that $\prod \mathcal{A} \neq [I]$, and so \mathcal{A} is an independent generator of $\langle \mathcal{A} \rangle$ by Lemma C.7(a). We also have $\langle \mathcal{C} \setminus \{\hat{p}_{2k}\} \rangle = \langle \mathcal{A} \rangle$, which then implies that $\mathcal{C} \setminus \{\hat{p}_{2k}\}$ is independent, because $|\mathcal{C} \setminus \{\hat{p}_{2k}\}| = 2k$ and $|\langle \mathcal{C} \setminus \{\hat{p}_{2k}\} \rangle| = |\langle \mathcal{A} \rangle| = 2^{2k}$. It follows that $\dim(\mathcal{C}) \geq |\mathcal{C}| - 1$, finishing the proof of (i).

For the proof of (ii), first assume that $\prod \mathcal{C} = [I]$. But this implies that $\dim(\mathcal{C}) < |\mathcal{C}|$, and hence combining with (i) this gives us that $\dim(\mathcal{C}) = |\mathcal{C}| - 1$. To prove the converse, for contradiction assume that $\prod \mathcal{C} \neq [I]$, which is equivalent to the assumption that $\hat{p}_{2k} \neq \prod(\mathcal{C} \setminus \{\hat{p}_{2k}\}) = \hat{q}_{2k-1}$. We have already proved in the previous paragraph that \mathcal{A} is an independent list, and it can also be checked that $\{\hat{p}_{2k}, \hat{q}_i\} = 0$, for all $0 \leq i \leq 2k - 2$, and $[\hat{p}_{2k}, \hat{q}_{2k-1}] = 0$. Since $|\mathcal{A}|$ is even, it follows from Lemma C.7(d) that \hat{q}_{2k-1} is the unique element in $\langle \mathcal{A} \rangle$ generating the commutativity pattern $\text{Com}_{\mathcal{A}}(\hat{p}_{2k})$ with respect to \mathcal{A} . But $\hat{p}_{2k} \neq \hat{q}_{2k-1}$ by assumption, implying that $\hat{p}_{2k} \notin \langle \mathcal{A} \rangle = \langle \mathcal{C} \setminus \{\hat{p}_{2k}\} \rangle$. So \mathcal{C} is an independent list, because we also proved in the previous paragraph that $\mathcal{C} \setminus \{\hat{p}_{2k}\}$ is independent, and it follows that $\dim(\mathcal{C}) = |\mathcal{C}|$, which is a contradiction.

- (b) We have already proved in (a) that $\mathcal{C} \setminus \{\hat{p}_{2k}\}$ is independent. The fact that $\mathcal{C} \setminus \{\hat{p}_i\}$ is also independent for all $\hat{p}_i \in \mathcal{C}$ then follows by translating the elements of \mathcal{C} to get a relabeling of the CAL $\mathcal{C} = \{\hat{p}'_0, \dots, \hat{p}'_{2k}\}$ by Lemma C.6(a), where $\hat{p}'_j = \hat{p}_{i+j-2k}$ and indices are evaluated modulo $2k + 1$.
- (c) If $|\mathcal{C}| = 1$, there is nothing to prove. So let $\mathcal{H} \subset \mathcal{C}$ be a non-empty multiset, and $|\mathcal{C}| \geq 3$. Then there exists $\mathcal{C} \ni \hat{p} \notin \mathcal{H}$. So by (b), $\mathcal{C} \setminus \{\hat{p}\}$ is an independent multiset. Since $\mathcal{H} \subseteq \mathcal{C} \setminus \{\hat{p}\}$, $\prod \mathcal{H} \neq [I]$.

□

Theorem C.9. Let $\mathcal{C} = \{\hat{p}_\ell : 0 \leq \ell \leq 2k - 1\} \subseteq \hat{\mathcal{P}}_n$ be a non-empty CAL, with $|\mathcal{C}|$ even.

(a) Define the lists $\mathcal{C}_{\text{odd}} = \{\hat{p}_{2\ell+1} : 0 \leq \ell \leq k - 1\}$, and $\mathcal{C}_{\text{even}} = \{\hat{p}_{2\ell} : 0 \leq \ell \leq k - 1\}$. The dimension of \mathcal{C} satisfies

$$\dim(\mathcal{C}) = \begin{cases} |\mathcal{C}| & \text{if } \prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}, \text{ and } \prod \mathcal{C}_{\text{odd}} \neq \prod \mathcal{C}_{\text{even}}, \\ |\mathcal{C}| - 1 & \text{if } \prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}, \text{ and } \prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}}, \\ |\mathcal{C}| - 1 & \text{if either } \prod \mathcal{C}_{\text{odd}} = [I] \neq \prod \mathcal{C}_{\text{even}}, \text{ or } \prod \mathcal{C}_{\text{odd}} \neq [I] = \prod \mathcal{C}_{\text{even}}, \\ |\mathcal{C}| - 2 & \text{if } \prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}} = [I]. \end{cases} \quad (73)$$

In particular, if $|\mathcal{C}| = 2$, then $\dim(\mathcal{C}) = 2$, while if $|\mathcal{C}| = 2n \geq 4$, then $\dim(\mathcal{C}) < 2n$.

- (b) For all non-empty multisets $\mathcal{H} \subset \mathcal{C}$ such that $\mathcal{H} \neq \mathcal{C}_{\text{odd}}$ and $\mathcal{H} \neq \mathcal{C}_{\text{even}}$, we have $\prod \mathcal{H} \neq [I]$.
- (c) If $|\mathcal{C}| \geq 4$, it holds that $\mathcal{C} \setminus \{\hat{p}, \hat{q}\}$ is an independent multiset for all $\hat{p} \in \mathcal{C}_{\text{odd}}$, and $\hat{q} \in \mathcal{C}_{\text{even}}$.

Proof. (a) If $|\mathcal{C}| = 2$, then \mathcal{C} is an anticommuting list of two elements, and hence by Lemma C.7(a) and (d), we have that \mathcal{C} is independent. Thus $\dim(\mathcal{C}) = 2$, $\prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}$, and also $\prod \mathcal{C}_{\text{odd}} \neq \prod \mathcal{C}_{\text{even}}$, which proves the statement for $|\mathcal{C}| = 2$. For the rest of the proof assume $|\mathcal{C}| \geq 4$. Construct the anticommuting list $\mathcal{A} = \{\hat{q}_0, \dots, \hat{q}_{2k-2}\}$ similarly as in Lemma C.6(b). Then $\mathcal{A} \setminus \{\hat{q}_{2k-2}\}$ is independent by Lemma C.7(c), and since it is easily checked that $\langle \mathcal{A} \setminus \{\hat{q}_{2k-2}\} \rangle = \langle \mathcal{C} \setminus \{\hat{p}_{2k-2}, \hat{p}_{2k-1}\} \rangle$, we have that $\dim(\mathcal{C}) \geq |\mathcal{C}| - 2$.

First suppose that $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}} = [I]$. This implies that $\dim(\mathcal{C}_{\text{odd}}) \leq |\mathcal{C}_{\text{odd}}| - 1$, and $\dim(\mathcal{C}_{\text{even}}) \leq |\mathcal{C}_{\text{even}}| - 1$, and so $\dim(\mathcal{C}) \leq |\mathcal{C}| - 2$. This proves that $\dim(\mathcal{C}) = |\mathcal{C}| - 2$ in this case.

Next suppose that either $\prod \mathcal{C}_{\text{odd}} = [I] \neq \prod \mathcal{C}_{\text{even}}$, or $\prod \mathcal{C}_{\text{odd}} \neq [I] = \prod \mathcal{C}_{\text{even}}$. In the former case we have that $\dim(\mathcal{C}_{\text{odd}}) \leq |\mathcal{C}_{\text{odd}}| - 1$, while in the latter case we have that $\dim(\mathcal{C}_{\text{even}}) \leq |\mathcal{C}_{\text{even}}| - 1$, and so in both cases $\dim(\mathcal{C}) \leq |\mathcal{C}| - 1$. Let us now assume without loss of generality that $\prod \mathcal{C}_{\text{odd}} = [I] \neq \prod \mathcal{C}_{\text{even}}$, because the other case reduces to this one by translating \mathcal{C} by one element, and relabeling the elements as in the proof of Theorem C.8(b). Then a simple computation shows that $\prod \mathcal{A} = \prod \mathcal{C}_{\text{even}}$, and so $\prod \mathcal{A} \neq [I]$ by assumption, which implies that \mathcal{A} is an independent list by Lemma C.7(a). But as $\langle \mathcal{A} \rangle = \langle \mathcal{C} \setminus \{\hat{p}_{2k-1}\} \rangle$, this implies that $\dim(\mathcal{C}) \geq |\mathcal{C}| - 1$, and so in fact $\dim(\mathcal{C}) = |\mathcal{C}| - 1$.

Now assume that $\prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}$, and $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}}$. Because $\prod \mathcal{C}_{\text{even}} \neq [I]$, the reasoning in the previous paragraph gives us that $\dim(\mathcal{C}) \geq |\mathcal{C}| - 1$ because \mathcal{A} is independent, while we also have that $\prod \mathcal{C} = (\prod \mathcal{C}_{\text{odd}})(\prod \mathcal{C}_{\text{even}}) = [I]$ and thus $\dim(\mathcal{C}) \leq |\mathcal{C}| - 1$, which means that $\dim(\mathcal{C}) = |\mathcal{C}| - 1$.

Finally assume that $\prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}$, and $\prod \mathcal{C}_{\text{odd}} \neq \prod \mathcal{C}_{\text{even}}$. As in the last paragraph, we still have that \mathcal{A} is an independent anticommuting list. One can now verify that \hat{p}_{2k-1} , \hat{q}_{2k-2} and $\prod(\mathcal{A} \setminus \hat{q}_{2k-2})$ generate the same commutativity pattern with respect to \mathcal{A} , i.e. if \hat{p} is either \hat{p}_{2k-1} , \hat{q}_{2k-2} , or $\prod(\mathcal{A} \setminus \hat{q}_{2k-2})$, then $[\hat{p}, \hat{q}_{2k-2}] = 0$, and $\{\hat{p}, \hat{q}_i\} = 0$, for all $0 \leq i \leq 2k - 3$. Now notice that $\hat{p}_{2k-1}\hat{q}_{2k-2} = \prod \mathcal{C} \neq [I]$ as $\prod \mathcal{C}_{\text{odd}} \neq \prod \mathcal{C}_{\text{even}}$, and also $\hat{p}_{2k-1} \prod(\mathcal{A} \setminus \hat{q}_{2k-2}) = \prod \mathcal{C}_{\text{odd}} \neq [I]$ by assumption, and so an application of Lemma C.7(e) implies that $\hat{p}_{2k-1} \notin \langle \mathcal{A} \rangle$. This implies that in this case $\dim(\mathcal{C}) = |\mathcal{C}|$.

It remains to show that if $|\mathcal{C}| = 2n \geq 4$, then $\dim(\mathcal{C}) < 2n$. From what has already been proved above, it suffices to show that the conditions $\prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}$, and $\prod \mathcal{C}_{\text{odd}} \neq \prod \mathcal{C}_{\text{even}}$ cannot both be true. However, for contradiction assume that they in fact both hold. Then the discussion in the last paragraph applies, and so $\hat{p}_{2n-1} \notin \langle \mathcal{A} \rangle$, and $\dim(\mathcal{A}) = 2n - 1$. This also implies that there are exactly two cosets of $\langle \mathcal{A} \rangle$ in $\hat{\mathcal{P}}_n$, and so \hat{p}_{2n-1} must be in the coset different from $\langle \mathcal{A} \rangle$. But this is a contradiction by an application of Lemma C.7(f), which rules out any element in the other coset, that generate the same commutativity pattern as \hat{p}_{2k-1} , with respect to the elements of \mathcal{A} .

- (b) The statement is vacuous for $|\mathcal{C}| = 2$ as there are no non-empty proper multisets satisfying the conditions; so assume $|\mathcal{C}| \geq 4$. We will prove this using contradiction. So suppose $\mathcal{H} \subset \mathcal{C}$ is a non-empty multiset, $\mathcal{C}_{\text{even}} \neq \mathcal{H} \neq \mathcal{C}_{\text{odd}}$, and $\prod \mathcal{H} = [I]$. This means $\prod \mathcal{H}$ commutes with all elements of \mathcal{C} . Now there are two cases: (i) $\hat{p}_0 \in \mathcal{H}$, and (ii) $\hat{p}_0 \notin \mathcal{H}$. Since $\prod \mathcal{H}$ commutes with \hat{p}_1 , it must be that in the first case $\hat{p}_2 \in \mathcal{H}$, and in the second case $\hat{p}_2 \notin \mathcal{H}$. Continuing this argument iteratively for $\hat{p}_3, \dots, \hat{p}_{2k-3}$ then shows that (i) and (ii) are equivalent to the conditions $\mathcal{C}_{\text{even}} \subseteq \mathcal{H}$ and $\mathcal{C}_{\text{even}} \cap \mathcal{H} = \emptyset$ respectively. Similarly, since $\prod \mathcal{H}$ commutes with all elements of $\mathcal{C}_{\text{even}}$, one again concludes that exactly one of the two cases is true: either $\mathcal{C}_{\text{odd}} \subseteq \mathcal{H}$ or $\mathcal{C}_{\text{odd}} \cap \mathcal{H} = \emptyset$. Combining we have four cases: (i) $\mathcal{C}_{\text{odd}} \cup \mathcal{C}_{\text{even}} \subseteq \mathcal{H}$, (ii) $(\mathcal{C}_{\text{odd}} \cup \mathcal{C}_{\text{even}}) \cap \mathcal{H} = \emptyset$, (iii) $\mathcal{H} = \mathcal{C}_{\text{odd}}$, and (iv) $\mathcal{H} = \mathcal{C}_{\text{even}}$, all of which are impossible by assumption.

- (c) For contradiction assume that $\mathcal{C} \setminus \{\hat{p}, \hat{q}\}$ is not independent. Then there exists a multiset $\mathcal{H} \subseteq \mathcal{C} \setminus \{\hat{p}, \hat{q}\}$ such that $\prod \mathcal{H} = [I]$. But this is a contradiction by (b). \square

Corollary C.10. *Let $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$ be an extremal CAL of length $|\mathcal{C}| \geq 3$. Then the dimension of \mathcal{C} satisfies*

$$\dim(\mathcal{C}) = \begin{cases} |\mathcal{C}| - 1 & \text{if } |\mathcal{C}| \text{ odd,} \\ |\mathcal{C}| - 2 & \text{if } |\mathcal{C}| \text{ even.} \end{cases} \quad (74)$$

Thus \mathcal{C} generates the group $\hat{\mathcal{P}}_n$.

Proof. First assume that $|\mathcal{C}|$ is odd. Then from Theorem C.8 we have $\dim(\mathcal{C}) \geq |\mathcal{C}| - 1$. But because \mathcal{C} is an extremal CAL we also have that $\dim(\mathcal{C}) \leq |\mathcal{C}| - 1$, because the number of qubits used is $(|\mathcal{C}| - 1)/2$. Now assume that $|\mathcal{C}|$ is even. In this case Theorem C.9 gives that $\dim(\mathcal{C}) \geq |\mathcal{C}| - 2$, while again as \mathcal{C} is an extremal CAL, it must also be true that $\dim(\mathcal{C}) \leq |\mathcal{C}| - 2$. In both cases $\dim(\mathcal{C}) = n$, and so $\langle \mathcal{C} \rangle = \hat{\mathcal{P}}_n$. This proves the lemma. \square

As a consequence of the previous lemmas, we get the next corollary that states which multisets of an extremal CAL of length at least three, can possibly multiply to the identity.

Corollary C.11. *Let $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$ be an extremal CAL of length $|\mathcal{C}| \geq 3$. Then the following statements are true.*

- (a) *Let $|\mathcal{C}|$ be odd. If $\mathcal{H} \subseteq \mathcal{C}$ is a non-empty multiset, then $\prod \mathcal{H} = [I]$ if and only if $\mathcal{H} = \mathcal{C}$.*
- (b) *Let $|\mathcal{C}|$ be even. If $\mathcal{H} \subseteq \mathcal{C}$ is a non-empty multiset, then $\prod \mathcal{H} = [I]$ if and only if $\mathcal{H} = \mathcal{C}_{\text{odd}}$ or $\mathcal{H} = \mathcal{C}_{\text{even}}$, where \mathcal{C}_{odd} and $\mathcal{C}_{\text{even}}$ are defined as in Theorem C.9.*

Proof. (a) As \mathcal{C} is an extremal CAL, it is not independent by Corollary C.10. The result then follows directly from Theorem C.8(a),(c).

(b) As \mathcal{C} is an extremal CAL, $\dim(\mathcal{C}) = |\mathcal{C}| - 2$ by Corollary C.10. The result follows by Theorem C.9(a),(b). □

In the final part of this subsection, we turn to the question of whether it is possible to create CALs of $\hat{\mathcal{P}}_n$, such that the representatives of each element in the CAL are Kronecker products of I_2 and the Pauli matrices X and Z only, and not of Y . To motivate this section, note that on $n = 1$ qubit, one has the CALs of lengths $\ell = 1, 2$, and 4 given by $\{[X]\}$, $\{[X], [Z]\}$, and $\{[X], [Z], [X], [Z]\}$ respectively, satisfying this property. But there is no CAL of length $\ell = 3$ with this property. Also notice that if the CAL has length $\ell \geq 2$, then it is not possible that all its element representatives are Kronecker products of I_2 and just one other Pauli matrix, for example X . This is because all elements of the CAL constructed this way will commute with one another. So we definitely need to use at least two of the Pauli matrices, for example X and Z .

For what follows, we need a couple of simple lemmas. The next result is well-known (see for instance Chapter 5 in [61]).

Lemma C.12. *For any $v \in \mathbb{F}_2^p$, let $\text{Ham}(v) = \sum_j v_j$ denote its Hamming weight. Then if $u, v \in \mathbb{F}_2^p$ have even Hamming weight, so does $w = u + v$ (where the addition is over \mathbb{F}_2). Thus any binary sum of even Hamming weight vectors also has even Hamming weight.*

Proof. Let $J_u \subseteq \{0, \dots, p-1\}$ be the set of indices such that $u_j = 1$ if and only if $j \in J_u$, and similarly let $J_v \subseteq \{0, \dots, p-1\}$ be the set of indices such that $v_j = 1$ if and only if $j \in J_v$. Since addition is over \mathbb{F}_2 , this implies that $w_j = 1$ if and only if $j \in J_w = (J_u \cup J_v) \setminus (J_u \cap J_v)$. But $|J_w| = |J_u| + |J_v| - 2|J_u \cap J_v|$, and we also have that $|J_u| = \text{Ham}(u)$, and $|J_v| = \text{Ham}(v)$, which are even by assumption. So $|J_w|$ is even, and since $|J_w| = \text{Ham}(w)$, this proves the first part of the lemma. The last part now follows by induction. □

We also give a construction that allows us to take a CAL and create a CAL of smaller length on the same number of qubits, or a longer CAL on more qubits, as given in the lemma below. The first part of this construction will be used later.

Lemma C.13. *Let $\mathcal{C} = \{\hat{p}_0, \dots, \hat{p}_{k-1}\} \subseteq \hat{\mathcal{P}}_n$ be a CAL of length $k \geq 2$, and define the lists $\mathcal{C}_{\text{odd}} = \{\hat{p}_{2\ell+1} : 0 \leq \ell \leq \lfloor k/2 \rfloor - 1\}$ and $\mathcal{C}_{\text{even}} = \{\hat{p}_{2\ell} : 0 \leq \ell \leq \lfloor k/2 \rfloor - 1\}$. Consider the following constructions:*

(i) *CAL expansion* — *If k is odd, let $\hat{r} = [Z]$, $\hat{s} = [X]$, and if k is even, let $\hat{r} = [X]$, $\hat{s} = [Z]$. Define the list $\mathcal{C}' = \{\hat{p}'_0, \dots, \hat{p}'_{k+1}\} \subseteq \hat{\mathcal{P}}_{n+1}$ by $\hat{p}'_\ell = \hat{p}_\ell \otimes [I_2]$ for all $0 \leq \ell \leq k-3$, $\hat{p}'_{k-2} = \hat{p}_{k-2} \otimes \hat{r}$, $\hat{p}'_{k-1} = [I] \otimes \hat{s}$, $\hat{p}'_k = [I] \otimes \hat{r}$, and $\hat{p}'_{k+1} = \hat{p}_{k-1} \otimes \hat{s}$, where $[I] \in \hat{\mathcal{P}}_n$ is the identity element of $\hat{\mathcal{P}}_n$. Also define the lists $\mathcal{C}'_{\text{odd}} = \{\hat{p}'_{2\ell+1} : 0 \leq \ell \leq \lfloor k/2 \rfloor\}$ and $\mathcal{C}'_{\text{even}} = \{\hat{p}'_{2\ell} : 0 \leq \ell \leq \lfloor k/2 \rfloor\}$.*

(ii) *CAL contraction* — *Let $\alpha \geq 0$ be any integer such that $\beta := k - 2\alpha \geq 2$. Define the list $\tilde{\mathcal{C}} = \{\hat{q}'_0, \dots, \hat{q}'_{\beta-1}\} \subseteq \hat{\mathcal{P}}_n$ by $\hat{q}'_\ell = \hat{p}_\ell$ for all $0 \leq \ell \leq \beta-3$, $\hat{q}'_{\beta-2} = \prod_{\ell=0}^{\alpha} \hat{p}_{\beta-2+2\ell}$, and $\hat{q}'_{\beta-1} = \prod_{\ell=0}^{\alpha} \hat{p}_{\beta-1+2\ell}$, and also define the lists $\tilde{\mathcal{C}}_{\text{odd}} = \{\hat{q}'_{2\ell+1} : 0 \leq \ell \leq \lfloor \beta/2 \rfloor - 1\}$ and $\tilde{\mathcal{C}}_{\text{even}} = \{\hat{q}'_{2\ell} : 0 \leq \ell \leq \lfloor \beta/2 \rfloor - 1\}$.*

Then we have the following:

- (a) $\prod \mathcal{C}'_{\text{odd}} = \prod \mathcal{C}_{\text{odd}} \otimes [I_2]$, $\prod \mathcal{C}'_{\text{even}} = \prod \mathcal{C}_{\text{even}} \otimes [I_2]$, and $\prod \mathcal{C}' = \prod \mathcal{C} \otimes [I_2]$. If all elements of \mathcal{C}_{odd} (resp. $\mathcal{C}_{\text{even}}$) are of Z -type (resp. X -type), then so are all elements of $\mathcal{C}'_{\text{odd}}$ (resp. $\mathcal{C}'_{\text{even}}$).
- (b) If $k \geq 3$, then \mathcal{C}' is a CAL of length $k + 2$, and $\dim(\mathcal{C}') = \dim(\mathcal{C})$.
- (c) $\prod \tilde{\mathcal{C}}_{\text{odd}} = \prod \mathcal{C}_{\text{odd}}$, $\prod \tilde{\mathcal{C}}_{\text{even}} = \prod \mathcal{C}_{\text{even}}$, and $\prod \tilde{\mathcal{C}} = \prod \mathcal{C}$. If all elements of \mathcal{C}_{odd} (resp. $\mathcal{C}_{\text{even}}$) are of Z -type (resp. X -type), then so are all elements of $\tilde{\mathcal{C}}_{\text{odd}}$ (resp. $\tilde{\mathcal{C}}_{\text{even}}$).
- (d) If $\beta \geq 3$, then $\tilde{\mathcal{C}}$ is a CAL of length β , and $\dim(\tilde{\mathcal{C}}) = \dim(\mathcal{C})$.

Proof. (a) It is easy to verify that by construction, if all elements of \mathcal{C}_{odd} are Z -type, then the same is true for $\mathcal{C}'_{\text{odd}}$, and if all elements of $\mathcal{C}_{\text{even}}$ are X -type, then the same holds for all elements of $\mathcal{C}'_{\text{even}}$. Now $\prod \mathcal{C}'_{\text{odd}} = \prod_{\ell=0}^{\lfloor k/2 \rfloor} \hat{p}'_{2\ell+1} = \left(\prod_{\ell=0}^{\lfloor k/2 \rfloor - 2} \hat{p}_{2\ell+1} \otimes [I_2] \right) (\hat{p}_{2\lfloor k/2 \rfloor - 3} \otimes [Z]) ([I] \otimes [Z]) = \left(\prod_{\ell=0}^{\lfloor k/2 \rfloor - 1} \hat{p}_{2\ell+1} \right) \otimes [I_2]$, which shows $\prod \mathcal{C}'_{\text{odd}} = \prod \mathcal{C}_{\text{odd}} \otimes [I_2]$. A similar calculation also shows that $\prod \mathcal{C}'_{\text{even}} = \prod \mathcal{C}_{\text{even}} \otimes [I_2]$, and from both these we also conclude that $\prod \mathcal{C}' = \prod \mathcal{C} \otimes [I_2]$.

(b) It is straightforward to verify that when $k \geq 3$, and since \mathcal{C} is a CAL, the elements of \mathcal{C}' satisfy the CAL commutation relations (Eq. (69)). Now by (a), we have that $\prod \mathcal{C}' = [I] \in \hat{\mathcal{P}}_{n+1}$ if and only if $\prod \mathcal{C} = [I] \in \hat{\mathcal{P}}_n$. So when k is odd, we have from Theorem C.8(a) that $\dim(\mathcal{C}') = \dim(\mathcal{C})$. When k is even, it follows similarly using (a) the following: (i) $\prod \mathcal{C}_{\text{odd}} = [I]$ if and only if $\prod \mathcal{C}'_{\text{odd}} = [I]$, (ii) $\prod \mathcal{C}_{\text{even}} = [I]$ if and only if $\prod \mathcal{C}'_{\text{even}} = [I]$, and (iii) $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}}$ if and only if $\prod \mathcal{C}'_{\text{odd}} = \prod \mathcal{C}'_{\text{even}}$. Theorem C.9 now implies that $\dim(\mathcal{C}') = \dim(\mathcal{C})$.

(c) This also follows by construction: we have $\prod \tilde{\mathcal{C}}_{\text{odd}} = \left(\prod_{\ell=0}^{\lfloor \beta/2 \rfloor - 2} \hat{p}_{2\ell+1} \right) (\prod_{\ell=0}^{\alpha} \hat{p}_{2\lfloor \beta/2 \rfloor - 1 + 2\ell}) = \prod \mathcal{C}_{\text{odd}}$, and $\prod \tilde{\mathcal{C}}_{\text{even}} = \left(\prod_{\ell=0}^{\lceil \beta/2 \rceil - 2} \hat{p}_{2\ell} \right) (\prod_{\ell=0}^{\alpha} \hat{p}_{2\lceil \beta/2 \rceil - 2 + 2\ell}) = \prod \mathcal{C}_{\text{even}}$, and multiplying them together gives $\prod \tilde{\mathcal{C}} = \prod \mathcal{C}$. The last part is a simple observation.

(d) If $\beta \geq 3$, then one easily verifies that $\tilde{\mathcal{C}}$ is a CAL. The proof of the last part is similar to (b), after noting that (c) implies (i) $\prod \mathcal{C}_{\text{odd}} = \prod \tilde{\mathcal{C}}_{\text{odd}}$, and (ii) $\prod \mathcal{C}_{\text{even}} = \prod \tilde{\mathcal{C}}_{\text{even}}$. \square

We are now in a position to discuss the case of odd-length CALs, which is more interesting than the even-length case. It turns out that one must always use all the three Pauli matrices to construct an odd-length CAL of length at least three, if the CAL is not an independent set; otherwise it is possible to not use the Pauli matrix Y . This is proved in the next theorem.

Theorem C.14. *For all $n \geq 1$, and $1 \leq k \leq 2n + 1$ such that k is odd, we have the following:*

- (a) If $3 \leq k \leq 2n + 1$, then it is not possible to construct a CAL $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$, with $|\mathcal{C}| = k$, and $\prod \mathcal{C} = [I]$, such that the representatives of each element of \mathcal{C} are Kronecker products not involving the Pauli matrix Y . If $k = 1$, then $\mathcal{C} = \{[I]\}$ is a CAL whose element representatives are Kronecker products not involving any Pauli matrix.
- (b) If $1 \leq k \leq 2n - 1$, then it is always possible to construct a CAL $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$, with $|\mathcal{C}| = k$, and $\prod \mathcal{C} \neq [I]$, such that the representatives of each element of \mathcal{C} are Kronecker products not involving the Pauli matrix Y .

Proof. (a) The case $k = 1$ is obvious, so fix $n \geq 1$, $3 \leq k \leq 2n + 1$, such that k is odd. The proof is by contradiction. Suppose there exists a CAL $\mathcal{C} = \{\hat{p}_0, \dots, \hat{p}_{k-1}\} \subseteq \hat{\mathcal{P}}_n$ satisfying $|\mathcal{C}| = k$, and $\prod \mathcal{C} = [I]$, such that the representatives of each element of \mathcal{C} are Kronecker products not involving Y . Let $\hat{p}_\ell^{(j)}$ denote the restriction of \hat{p}_ℓ to qubit j , and let us also define the lists $\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(n)} \subseteq \hat{\mathcal{P}}_1$ by $\mathcal{C}^{(j)} = \{\hat{p}_\ell^{(j)} : 0 \leq \ell \leq k - 1\}$, for all $0 \leq j \leq n - 1$. We observe that $\prod \mathcal{C} = [I]$ implies $\prod \mathcal{C}^{(j)} = [I]$ for all j (here the latter $[I]$ denotes the identity element of $\hat{\mathcal{P}}_1$). Now by assumption we have that $[Y] \notin \mathcal{C}^{(j)}$, and so $[X]$ and $[Z]$ must occur an even number of times in each $\mathcal{C}^{(j)}$. Define the matrices $M, M^{(0)}, \dots, M^{(n-1)} \in \mathbb{F}_2^{k \times k}$ as

$$\begin{aligned}
M(r, s) &= \begin{cases} 0 & \text{if } [\hat{p}_r, \hat{p}_s] = 0 \\ 1 & \text{if } \{\hat{p}_r, \hat{p}_s\} = 0 \end{cases} \\
M^{(j)}(r, s) &= \begin{cases} 0 & \text{if } [\hat{p}_r^{(j)}, \hat{p}_s^{(j)}] = 0 \\ 1 & \text{if } \{\hat{p}_r^{(j)}, \hat{p}_s^{(j)}\} = 0 \end{cases}
\end{aligned} \tag{75}$$

for all $0 \leq r, s \leq k-1$, and $0 \leq j \leq n-1$. These matrices have the following properties:

- (i) $M, M^{(0)}, \dots, M^{(n-1)}$ are symmetric, and have zeros on the diagonal.
- (ii) The matrices satisfy the identity $M = \sum_{j=0}^{n-1} M^{(j)}$, where addition is performed over \mathbb{F}_2 .
This is because \hat{p}_r and \hat{p}_s commute if and only if their restrictions $\hat{p}_r^{(j)}$ and $\hat{p}_s^{(j)}$ for qubits $0 \leq j \leq n-1$, anticommute an even number of times.
- (iii) The upper triangular part of M contains exactly k non-zeros. This is simply a consequence of the commutation relations Eq. (69).
- (iv) For all $0 \leq j \leq n-1$, since the multiset $\mathcal{C}^{(j)}$ contains an even number of $[X]$ and $[Z]$, it follows from Eq. (75) that the number of non-zeros in $M^{(j)}$ is a multiple of 8. As $M^{(j)}$ is symmetric with zeros on the diagonal by property (i), we have that the number of non-zeros in the upper triangular part of $M^{(j)}$ is a multiple of 4.

By an application of Lemma C.12, property (iv) above implies that the number of non-zeros in the upper triangular part of $\sum_{j=0}^{n-1} M^{(j)}$ is even, but this is a contradiction to property (iii) which states that the number of non-zeros in the upper triangular part of M is k , which is odd by assumption. This gives the desired contradiction.

- (b) First notice that if there exists a CAL $\mathcal{C} = \{\hat{p}_\ell : 0 \leq \ell \leq k-1\} \subseteq \hat{\mathcal{P}}_n$ of odd length k and $\prod \mathcal{C} \neq I$, then it must necessarily satisfy $1 \leq k \leq 2n-1$, by Lemma C.6(d) and Corollary C.11(a), and this also implies the existence of a CAL of length k with the required properties on $n' > n$ qubits, because one can simply form the new CAL $\mathcal{C}' = \{\hat{p}_\ell \otimes [I] : 0 \leq \ell \leq k-1\}$, where $[I] \in \hat{\mathcal{P}}_{n'-n}$ is the identity element on $n' - n$ qubits. Thus for each odd $k \geq 1$, it suffices to find a CAL \mathcal{C}_k of length k , with the required properties on $(k+1)/2$ qubits. If $k = 1$, then $\mathcal{C}_1 = \{[X]\}$ is such a CAL. If $k = 3$, then $\mathcal{C}_3 = \{[X] \otimes [I_2], [Z] \otimes [Z], [Z] \otimes [X]\}$ is a CAL with the desired properties. One can now iteratively use the CAL expansion construction in Lemma C.13, to obtain \mathcal{C}_{k+2} from \mathcal{C}_k for all odd $k \geq 3$. The construction ensures that if the restriction of each element of \mathcal{C}_k to qubit j is not $[Y]$, for all $0 \leq j \leq (k-1)/2$, then the same is true for each element of \mathcal{C}_{k+2} for all qubit indices $0 \leq j \leq (k+1)/2$. \square

The final theorem of this section shows that for CALs of even-length, it always suffices to not use the Pauli matrix Y .

Theorem C.15. *For all $n \geq 1$, and for all integers k satisfying $2 \leq 2k \leq 2n+2$, there exists a CAL $\mathcal{C} \subseteq \hat{\mathcal{P}}_n$, with $|\mathcal{C}| = 2k$, such that the representatives of each element of \mathcal{C} are Kronecker products not involving the Pauli matrix Y . Moreover defining \mathcal{C}_{odd} and $\mathcal{C}_{\text{even}}$ according to Theorem C.9, one can choose \mathcal{C} to satisfy any of the cases*

- (i) $\prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}$, and $\prod \mathcal{C}_{\text{odd}} \neq \prod \mathcal{C}_{\text{even}}$
- (ii) $\prod \mathcal{C}_{\text{odd}} \neq [I] \neq \prod \mathcal{C}_{\text{even}}$ and $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}}$
- (iii) $\prod \mathcal{C}_{\text{odd}} = [I] \neq \prod \mathcal{C}_{\text{even}}$
- (iv) $\prod \mathcal{C}_{\text{odd}} \neq [I] = \prod \mathcal{C}_{\text{even}}$
- (v) $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}} = [I]$

if the case is allowed by Theorem C.9(a). Except for case (ii), one can also choose all elements of \mathcal{C}_{odd} to be of Z -type, and all elements of $\mathcal{C}_{\text{even}}$ to be of X -type.

Proof. As in the proof of Theorem C.14(b), notice that if we prove this theorem for any fixed values of k and n , then we have also proved the theorem for the same fixed k and for all $n' > n$. First consider CALs of length $2k = 2$. Then by Theorem C.9(a) and its proof, we know that only case (i) is possible. In this case the CAL $\mathcal{C} = \{[X], [Z]\}$ has the required properties for $n = 1$, so this proves the case $2k = 2$. For the rest of the proof, assume $2k \geq 4$. Also notice that the last statement is true because if it were possible to choose all elements of \mathcal{C}_{odd} and $\mathcal{C}_{\text{even}}$ to be of Z -type and X -type respectively, then the only way to have $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}}$ is to have $\prod \mathcal{C}_{\text{odd}} = \prod \mathcal{C}_{\text{even}} = [I]$, which is a contradiction.

We now adopt the notation $\mathcal{C}_{2k,n} \subseteq \hat{\mathcal{P}}_n$ to denote a CAL of length $2k$. Note that the minimum value of n for which $\mathcal{C}_{2k,n}$ can exist for cases (i)-(v) satisfying the properties of the theorem is given by $n \geq n_{2k}$, where $n_{2k} = k + 1$ for case (i), $n_{2k} = k$ for cases (ii)-(iv), and $n_{2k} = k - 1$ for case (v), all following from Lemma C.6(d) and Theorem C.9(a). We first exhibit examples of CALs \mathcal{C}_{4,n_2} existing for each case — these are (i) $\mathcal{C}_{4,3} = \{[X_1X_2], [Z_2], [X_2X_3], [Z_1Z_3]\}$, (ii) $\mathcal{C}_{4,2} = \{[X_1], [Z_1], [X_1X_2], [Z_1X_2]\}$, (iii) $\mathcal{C}_{4,2} = \{[X_1], [Z_1], [X_1X_2], [Z_1]\}$, (iv) $\mathcal{C}_{4,2} = \{[X_1], [Z_1], [X_1], [Z_1Z_2]\}$, and (v) $\mathcal{C}_{4,1} = \{[X], [Z], [X], [Z]\}$, for the cases (i)-(v) respectively. Now for each case we use the CAL expansion construction of Lemma C.13(i) to iteratively construct the CALs \mathcal{C}_{2k,n_k} for all $k \geq 2$, starting from \mathcal{C}_{4,n_2} . These CALs have the required properties in each case by Lemma C.13(a), since \mathcal{C}_{4,n_2} has them as well. The theorem is then proved by the first sentence of the previous paragraph. \square

D Majorana surface code lemmas and proofs

This section contains proofs of Lemmas 3.1 and 3.2, which establish properties of Majorana subgroups and Majorana surface codes. We also restate the Lemmas here.

Lemma 3.1. Let \mathcal{S} be a subgroup of \mathcal{J}_m , and let $\mathcal{I} \subseteq \mathcal{J}_m$ be non-empty, such that elements of \mathcal{I} commute and are Hermitian. Then $\langle \mathcal{I} \rangle$ is Abelian and Hermitian, and moreover the following holds:

- (a) There exists a set \mathcal{I}' formed by multiplying each element of \mathcal{I} by either 1 or -1 , such that $-I \notin \langle \mathcal{I}' \rangle$.
- (b) If $-I \notin \mathcal{S}$, then \mathcal{S} is Abelian and Hermitian. Conversely, if \mathcal{S} is Abelian and Hermitian, then either $-I \notin \mathcal{S}$, or $\mathcal{S} = \langle \mathcal{S}', -I \rangle$ for some subgroup \mathcal{S}' of \mathcal{S} with $-I \notin \mathcal{S}'$.

Proof. In this proof $\eta, \eta', \eta'' \in \{\pm 1, \pm i\}$, and $a, a', b \in \mathbb{F}_2^m$. We also note some useful facts for this proof which follow easily from Eqs. (6,7): if $x \in \mathcal{J}_m$ then (i) $x^{-1} = x^\dagger$, (ii) x is either Hermitian or skew-Hermitian, and (iii) $x^2 \in \{I, -I\}$, with $x^2 = I$ if and only if $x = x^\dagger$. Also note that for any non-empty $\mathcal{J} \subseteq \mathcal{J}_m$ whose elements commute and are Hermitian, if $x \in \mathcal{J}$, then by (i) and (iii) $x = x^\dagger = x^{-1}$, so all elements in $\langle \mathcal{J} \rangle$ are products of elements of \mathcal{J} . That $\langle \mathcal{I} \rangle$ is Abelian is clear since any element of $\langle \mathcal{I} \rangle$ is a product of the elements in \mathcal{I} . To see that $\langle \mathcal{I} \rangle$ is Hermitian, note that if $\langle \mathcal{I} \rangle \ni x = \prod_{y \in \mathcal{I}'} y$ for some $\mathcal{I}' \subseteq \mathcal{I}$, then $x^2 = \prod_{y \in \mathcal{I}'} y^2 = I$, and we conclude using (iii).

- (a) Let $|\mathcal{I}| = k \geq 1$. We prove the result by induction on $|\mathcal{I}|$. If $\mathcal{I} = \{x\}$ we choose $\mathcal{I}' = \{x\}$ if $x \neq -I$, else $\mathcal{I}' = \{I\}$, proving the base case. Now suppose the result is true for all \mathcal{I} such that $1 \leq |\mathcal{I}| \leq r < k$. If $|\mathcal{I}| = r + 1$, pick any $x \in \mathcal{I}$ and let $\mathcal{I}_1 = \mathcal{I} \setminus \{x\}$. By the inductive hypothesis, there exists \mathcal{I}'_1 formed by multiplying every element of \mathcal{I}_1 by either 1 or -1 , such that $-I \notin \langle \mathcal{I}'_1 \rangle$. If $x \in \langle \mathcal{I}'_1 \rangle$ (resp. $-x \in \langle \mathcal{I}'_1 \rangle$), choose $\mathcal{I}' = \mathcal{I}'_1 \cup \{x\}$ (resp. $\mathcal{I}' = \mathcal{I}'_1 \cup \{-x\}$) and we are done. If $x, -x \notin \langle \mathcal{I}'_1 \rangle$, choose $\mathcal{I}' = \mathcal{I}'_1 \cup \{x\}$. Now for contradiction, assume $-I \in \langle \mathcal{I}' \rangle$, so we have $-I = zx$ for some $z \in \langle \mathcal{I}'_1 \rangle$, using properties of \mathcal{I} (commuting and Hermitian) and (iii). Letting $z = \eta\gamma_a$, and $x = \eta'\gamma_{a'}$ then gives $-I = \eta\eta'(-1)^{\xi(a,a')}\gamma_{a+a'}$ using Eq. (7), and so $a = a'$. Combining with $z^2 = x^2 = I$ now gives $\eta = -\eta'$, or equivalently $z = -x$ which is a contradiction.
- (b) First suppose $-I \notin \mathcal{S}$. Note that if $x \in \mathcal{S}$ is not Hermitian, then $\mathcal{S} \ni x^2 = -I$ by (iii). Now suppose \mathcal{S} is non-Abelian. Then there exists $\eta\gamma_a, \eta'\gamma_{a'} \in \mathcal{S}$, such that $\gamma_a\gamma_{a'} = -\gamma_{a'}\gamma_a$. But

this implies that $\pm\eta''\gamma_b \in \mathcal{S}$, where $\eta''\gamma_b = \eta\eta'\gamma_a\gamma_{a'}$, and so both $\pm(\eta'')^2\gamma_b^2 \in \mathcal{S}$. But one of these must be $-I$, as $\gamma_b^2 = \pm I$, giving a contradiction. For the converse assume that $-I \in \mathcal{S}$ for Hermitian and Abelian \mathcal{S} . Consider the cosets of $\langle -I \rangle$ in \mathcal{S} , and let \mathcal{I} be the set formed by picking exactly one element from each coset. By (a), then there exists \mathcal{I}' such that $-I \notin \langle \mathcal{I}' \rangle$, and $\mathcal{S} = \langle \langle \mathcal{I}' \rangle, -I \rangle$.

□

For the next Lemma, refer to the definition of Majorana surface codes, Definition 3.1 for the notation.

Lemma 3.2. The stabilizers of the Majorana surface code for the embedded graph $G(V, E, F)$ satisfy

- (a) There is no non-empty subset $V' \subseteq V$ such that $\prod_{v \in V'} S_v = \pm I$.
- (b) There is no non-empty, proper subset $V' \subset V$ and no subset $F' \subseteq F$ such that $\prod_{v \in V'} S_v = \pm \prod_{f \in F'} S_f$. If there is any odd-degree vertex, then the statement holds for all non-empty subsets $V' \subseteq V$.
- (c) If $\prod_{v \in V} S_v = \prod_{f \in F'} S_f$ for some subset $F' \subseteq F$, then G is checkerboardable. If G is checkerboardable then $\prod_{v \in V} S_v = \prod_{f \in F'} S_f = \prod_{f \in F \setminus F'} S_f$ for a non-empty proper subset $F' \subset F$, which is determined uniquely up to taking complements.
- (d) There is no non-empty, proper subset $F' \subset F$ such that $\prod_{f \in F'} S_f = \pm I$.
- (e) $\prod_{f \in F} S_f = I$.
- (f) $|F| = 1$ if and only if there is $f \in F$ such that $S_f = I$.
- (g) $-I \notin \mathcal{S}(G)$.

Proof. We start with an observation that is needed for proofs of (c) and (g). For any subset $F' \subseteq F$ and any Majorana associated to a half-edge $\gamma_{[h]_\tau}$, the product $\prod_{f \in F'} S_f$ either contains both $\gamma_{[h]_\tau}$ and $\gamma_{[\lambda h]_\tau}$, or neither of them. Thus the Majoranas appearing in the product $\prod_{f \in F'} S_f$ can be grouped in pairs labeled by edges, which allows us to identify the product with a vector $y \in \mathbb{F}_2^{|E|}$, with $y_e = 1$ for $e = \{h, \lambda h, \tau h, \lambda \tau h\} \in E$, if and only if both $\gamma_{[h]_\tau}$ and $\gamma_{[\lambda h]_\tau}$ appear in the product. It follows that $y = x\Phi$, where we recall that $\Phi \in \mathbb{F}_2^{|F| \times |E|}$ is the face-edge adjacency matrix of the graph embedding, and $x \in \mathbb{F}_2^{|F|}$ satisfying $x_f = 1$ if and only if $f \in F'$. We now return to the proof.

- (a) This follows because all the vertex stabilizers are independent from each other, as they have disjoint supports.
- (b) First assume that $V' \subset V$ is a non-empty proper subset. We find an edge $e \in E$ with one endpoint in V' and one in $V \setminus V'$ (we use connectivity of G here), and notice that since face stabilizers either include or exclude both Majoranas on an edge together, there is no way to take a product of face stabilizers to include only one Majorana out of the two on edge e . This proves the first part. Now assume that there is at least one odd-degree vertex v , and suppose $\prod_{v \in V'} S_v = \prod_{f \in F'} S_f$ for some non-empty subset $V' \subseteq V$ and some subset $F' \subseteq F$. Then by the first part, $V' = V$, so the Majorana $\bar{\gamma}_v$ is present in the product $\prod_{v \in V'} S_v$, which is a contradiction since no face stabilizer contains $\bar{\gamma}_v$.
- (c) Suppose $\prod_{v \in V} S_v = \prod_{f \in F'} S_f$ for some subset $F' \subseteq F$. This cannot happen if there is any odd-degree vertex, as the product $\prod_{f \in F'} S_f$ cannot contain any Majorana associated to an odd-degree vertex. On the other hand, $\prod_{v \in V} S_v$ contains every Majorana associated to a half-edge, and so we can identify $\prod_{f \in F'} S_f$ with $\vec{1} \in \mathbb{F}_2^{|E|}$. Thus we have $x\Phi = \vec{1}$ where $x \in \mathbb{F}_2^{|F|}$ satisfying $x_f = 1$ if and only if $f \in F'$, so by Lemma 2.3(a) we conclude that G is checkerboardable. Now suppose that G is checkerboardable. So G has no odd-degree vertices, and by Lemma 2.3(b) there exist exactly two distinct vectors $x, x' \in \mathbb{F}_2^{|F|}$ (different from $\vec{0}$ and

$\vec{1}$) such that $x\Phi = x'\Phi = \vec{1}$, and $x + x' = \vec{1}$. Taking $F' = \{f \in F : x_f = 1\}$ we obtain a non-empty proper subset for which these properties imply $\prod_{v \in V} S_v = \eta \prod_{f \in F'} S_f = \eta' \prod_{f \in F \setminus F'} S_f$ where $\eta, \eta' \in \{\pm 1, \pm i\}$. Now $\eta, \eta' \neq \pm i$ as $\mathcal{S}(G)$ is Hermitian by Lemma 3.1, so it remains to show that $\eta, \eta' \neq -1$. We defer this remaining part to the proof of (g).

- (d) This is similar to (b), but for the dual graph embedding \bar{G} . Find a face $f' \in F'$ that shares an adjacent edge $e \in E$ with a face $f \notin F'$ (this uses connectivity of the dual graph), and notice that the Majoranas on edge e appear just once in the product, which can therefore not be identity.
- (e) Note that each Majorana $\gamma_{[h]\tau}$ is included in either two face stabilizers (if h and τh are in different faces) or none (if h and τh are in the same face). Moreover, Majoranas associated to odd-degree vertices are not included in any face stabilizer. Also note that for any flag $h \in H$, $i\gamma_{[h]\tau}\gamma_{[\lambda h]\tau}$ commutes with all Majoranas different from $\gamma_{[h]\tau}$ and $\gamma_{[\lambda h]\tau}$, and squares to I . Using these facts, a straightforward calculation shows that $\prod_{f \in F} S_f = I$.
- (f) This follows directly from (d) and (e).
- (g) Suppose this is false. Then $-I = \prod_{v \in V} S_v \prod_{f \in F'} S_f$ for some non-empty proper subset $F' \subset F$, as all other cases are ruled out by (a), (b), (d), (e), or equivalently $\prod_{v \in V} S_v = -\prod_{f \in F'} S_f$. By the same identification argument used in (c), we conclude that G is checkerboardable. We will now show that in fact $\prod_{v \in V} S_v = \prod_{f \in F'} S_f$, which will complete the proof. Note that this also finishes the remaining part of the proof of (c), because if either $\eta = -1$ or $\eta' = -1$, then $-I \in \mathcal{S}(G)$.

Since G is checkerboardable, it has no odd-degree vertices, and each edge in E is adjacent to exactly one face in F' . Thus let us identify the Majoranas by their labels: so define $\gamma_j := \gamma_{[h]\tau}$ if $\gamma_{[h]\tau}$ has label j . Let $\tilde{v} \in V$ be the vertex that contains the Majoranas γ_0 and $\gamma_{2|E|-1}$ in its vertex stabilizer. Then for each $v \in V \setminus \tilde{v}$, one has $S_v = \prod_{j \in I_v} (i\gamma_{2j+1}\gamma_{2j+2})$, and $S_{\tilde{v}} = (i\gamma_0) \left(\prod_{j \in I_{\tilde{v}}} i\gamma_{2j+1}\gamma_{2j+2} \right) \gamma_{2|E|-1}$, where for every $v \in V$, $I_v \subseteq \{0, \dots, |E| - 2\}$ contains those integers j such that γ_{2j+1} is associated to a half-edge that is a subset of v . Noticing that $(i\gamma_{2j+1}\gamma_{2j+2})$ commutes with all Majoranas different from γ_{2j+1} and γ_{2j+2} , we immediately get $\prod_{v \in V} S_v = i^{|E|} \gamma_0 \gamma_1 \dots \gamma_{2|E|-1}$. Similarly, each face stabilizer can be written as $S_f = \prod_{j \in I_f} i\gamma_{2j}\gamma_{2j+1}$ where $I_f \subseteq \{0, \dots, |E| - 1\}$ contains those integers j for which γ_{2j} belongs to an edge adjacent to f . It follows that $\prod_{f \in F'} S_f = i^{|E|} \gamma_0 \gamma_1 \dots \gamma_{2|E|-1}$, completing the proof. □

E Equivalence of Majorana and qubit surface codes

The goal of this section is to prove the equivalence between the Majorana surface code, defined on $m = M + 2|E|$ Majoranas (where m is even), and the qubit surface code, defined on $N = M/2 + |E| - |V|$ qubits, as explained in Section 3.3 (the quantities $M, |E|, |V|$ are defined in Section 3.2). The proof relies on some general facts that we also present here. We will use some of the notations and definitions introduced in Section C.1. Recall from Section 3.1 that the Jordan-Wigner transformation identifies the Pauli group $\mathcal{P}_{m/2}$ with the Majorana group \mathcal{J}_m , in a way that preserves commutation relations. To remain consistent with the notation used for Paulis, define $\text{Com} : \mathcal{J}_m \times \mathcal{J}_m \rightarrow \mathbb{F}_2$ as $\text{Com}(\gamma, \gamma') = \text{Com}(\text{JW}(\gamma), \text{JW}(\gamma'))$, where $\gamma, \gamma' \in \mathcal{J}_m$, and $\text{JW}(\cdot)$ is the Jordan-Wigner map (the phase factors of the elements of \mathcal{J}_m have been absorbed into γ, γ' and we continue to do so for the rest of this section⁷). Due to the Jordan-Wigner identification, Lemma 3.1 is also valid for the Pauli group. To simplify notation, let us define $r := m/2$.

Recall that a stabilizer group \mathcal{S} on r -qubits is a subgroup of \mathcal{P}_r satisfying $-I \notin \mathcal{S}$, which implies that $p^2 = I$ for all $p \in \mathcal{S}$, and \mathcal{S} is a commuting subgroup. Let $\mathcal{C}(\mathcal{S})$ be the centralizer of \mathcal{S} . If

⁷This is in contrast to the notation in Section 3.1 where the phase factors were made explicit.

$\dim(\mathcal{S}) = k$, for some $0 \leq k \leq r$, then $\dim(\mathcal{C}(\mathcal{S})) = 2r - k$. Let \mathcal{S}' be another stabilizer group of \mathcal{P}_r such that $\mathcal{S} \subseteq \mathcal{S}'$. Then by the definition of the centralizer we have $\mathcal{S}' \subseteq \mathcal{C}(\mathcal{S})$. Now suppose that $\kappa : \mathcal{C}(\mathcal{S}) \rightarrow \mathcal{P}_N$ is a homomorphism. Notice that $\pm iI \notin \kappa(\mathcal{S}')$, because if otherwise $\kappa(p) = \pm iI$ for some $p \in \mathcal{S}'$, then it implies $\kappa(I) = \kappa(p^2) = (\kappa(p))^2 = -I$. We say that κ is *commutation preserving* if $\text{Com}(p, q) = \text{Com}(\kappa(p), \kappa(q))$ for all $p, q \in \mathcal{C}(\mathcal{S})$. Let $\pi : \mathcal{P}_N \rightarrow \hat{\mathcal{P}}_N$ be the projection map taking each element of \mathcal{P}_N to its equivalence class in $\hat{\mathcal{P}}_N$. It is easily checked that π is a surjective homomorphism, so κ descends to a homomorphism $\hat{\kappa} := \pi \circ \kappa : \mathcal{C}(\mathcal{S}) \rightarrow \hat{\mathcal{P}}_N$. The homomorphism π is commutation preserving in the sense that for all $p, q \in \mathcal{P}_N$, $\text{Com}(p, q) = \text{Com}([p], [q])$. Thus if κ is commutation preserving also, then so is $\hat{\kappa}$, as for all $p, q \in \mathcal{C}(\mathcal{S})$, $\text{Com}(p, q) = \text{Com}(\hat{\kappa}(p), \hat{\kappa}(q))$. Moreover, if κ is surjective, it follows by surjectivity of π that $\hat{\kappa}$ is surjective. Below in Lemma E.1, we prove a result addressing the converse direction.

Lemma E.1. *Let \mathcal{S} be a stabilizer subgroup of \mathcal{P}_r , and let $\mu : \mathcal{C}(\mathcal{S}) \rightarrow \hat{\mathcal{P}}_N$ be a surjective, commutation preserving homomorphism. Then there exists a surjective homomorphism $\kappa : \mathcal{C}(\mathcal{S}) \rightarrow \mathcal{P}_N$ such that $\hat{\kappa} = \mu$, and $\kappa(\mathcal{S})$ is a stabilizer subgroup of \mathcal{P}_N . Moreover, for any such κ , if \mathcal{S}' is a stabilizer subgroup of \mathcal{P}_r satisfying $\mathcal{S} \subseteq \mathcal{S}'$, the following holds:*

- (a) κ is commutation preserving.
- (b) $\kappa(p) \notin \langle iI \rangle$ for all $p \in \mathcal{C}(\mathcal{S}) \setminus \langle iI, \mathcal{S} \rangle$.
- (c) The kernel of κ is \mathcal{S} , and $\kappa(iI) = \pm iI$.
- (d) $\kappa(\mathcal{S}')$ is a stabilizer subgroup of \mathcal{P}_N .
- (e) $\kappa(\mathcal{C}(\mathcal{S}')) = \mathcal{C}(\kappa(\mathcal{S}'))$.
- (f) $\mathcal{T}/\mathcal{S} \cong \kappa(\mathcal{T})$, and $\dim(\kappa(\mathcal{T})) = \dim(\mathcal{T}) - \dim(\mathcal{S})$, where \mathcal{T} is either \mathcal{S}' or $\mathcal{C}(\mathcal{S}')$. The stabilizer codes given by the stabilizer groups \mathcal{S}' in \mathcal{P}_r , and $\kappa(\mathcal{S}')$ in \mathcal{P}_N encode the same number of logical qubits.

Proof. For the proof, we first note some useful facts. Since μ is surjective and commutation preserving, we have $\mu(iI) = [I]$. This follows because if $\mu(iI) \neq [I]$, then by surjectivity there exists $p \in \mathcal{C}(\mathcal{S})$ such that $\text{Com}(\mu(p), \mu(iI)) = 1$, while $\text{Com}(p, iI) = 0$ giving a contradiction. Also recall that for any $p \in \mathcal{C}(\mathcal{S}) \setminus \langle iI, \mathcal{S} \rangle$, there exists $q \in \mathcal{C}(\mathcal{S}) \setminus \langle iI, \mathcal{S} \rangle$ such that $\text{Com}(p, q) = 1$. Now let $\dim(\mathcal{S}) = k \geq 0$. Then there exists an independent list $\mathcal{G} = \{p_1, \dots, p_k, p'_1, \dots, p'_{2r-2k}\}$ of commuting, Hermitian elements of \mathcal{P}_r , such that $\mathcal{S} = \langle p_1, \dots, p_k \rangle$, and $\mathcal{C}(\mathcal{S}) = \langle iI, \mathcal{G} \rangle$. Thus every element $p \in \mathcal{C}(\mathcal{S})$ can be uniquely represented in the form $p = (iI)^{\alpha_p} \prod_{y \in \mathcal{H}_p} y$, where $\alpha_p \in \{0, 1, 2, 3\}$, and $\mathcal{H}_p \subseteq \mathcal{G}$ are uniquely determined by p . The elements in this product representation appear in the same order as in the list \mathcal{G} , with the convention that the product of the empty subset is I .

We first show the existence of κ . Let us define a map $\psi : \mathcal{G} \rightarrow \mathcal{P}_N$ by letting $\psi(x)$ be any arbitrarily chosen Hermitian element in the equivalence class $\mu(x)$ for each $x \in \mathcal{G}$, and consider the list $\mathcal{T} = \{\psi(p_1), \dots, \psi(p_k)\}$. Since \mathcal{S} is a commuting subgroup and μ is a homomorphism, it follows that $\langle \mathcal{T} \rangle$ is a commuting subgroup. If $k \geq 1$, then \mathcal{T} is non-empty, and so by Lemma 3.1 we conclude $\langle \mathcal{T} \rangle$ is Hermitian, and the elements of \mathcal{T} can be multiplied by ± 1 to obtain a new list $\tilde{\mathcal{T}} = \{\tilde{\psi}(p_1), \dots, \tilde{\psi}(p_k)\}$ such that $-I \notin \langle \tilde{\mathcal{T}} \rangle$. Next define the map $\kappa : \mathcal{G} \cup \{iI\} \rightarrow \hat{\mathcal{P}}_N$ as follows

$$\kappa(x) = \begin{cases} \tilde{\psi}(x) & \text{if } x = p_1, \dots, p_k \\ \psi(x) & \text{if } x = p'_1, \dots, p'_{2r-2k} \\ iI & \text{if } x = iI, \end{cases} \quad (76)$$

and we claim that κ extends uniquely to a homomorphism $\kappa : \mathcal{C}(\mathcal{S}) \rightarrow \mathcal{P}_N$. For any such extension, if $x = (iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} y \in \mathcal{C}(\mathcal{S})$, then $\kappa(x) = (iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} \kappa(y)$, showing uniqueness. To prove existence, for any $x = (iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} y \in \mathcal{C}(\mathcal{S})$, define $\kappa(x) := (\kappa(iI))^{\alpha_x} \prod_{y \in \mathcal{H}_x} \kappa(y)$. This definition satisfies Eq. (76), so we need to verify that κ is a homomorphism. Let $x, x' \in \mathcal{C}(\mathcal{S})$ be given by $x = (iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} y$, and $x' = (iI)^{\alpha_{x'}} \prod_{y \in \mathcal{H}_{x'}} y$. Then $x'' := xx' = \left((iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} y \right) \left((iI)^{\alpha_{x'}} \prod_{y \in \mathcal{H}_{x'}} y \right) = (iI)^{\alpha_{x''}} \prod_{y \in \mathcal{H}_{x''}} y$, where $\mathcal{H}_{x''} = (\mathcal{H}_x \cup \mathcal{H}_{x'}) \setminus (\mathcal{H}_x \cap \mathcal{H}_{x'})$, and $\alpha_{x''} = (\alpha_x + \alpha_{x'} + 2t) \pmod{4}$,

with t determined by the commutativity of the elements of \mathcal{H}_x and $\mathcal{H}_{x'}$. Now from the definition of κ , we get $\kappa(xx') = (iI)^{\alpha_{x'}} \prod_{y \in \mathcal{H}_{x'}} \kappa(y)$. We also observe that since μ is commutation preserving, Eq. (76) implies $\text{Com}(x_1, x_2) = \text{Com}(\kappa(x_1), \kappa(x_2))$, for all $x_1, x_2 \in \mathcal{G} \cup \{iI\}$. This gives $\kappa(x)\kappa(x') = \left((iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} \kappa(y) \right) \left((iI)^{\alpha_{x'}} \prod_{y \in \mathcal{H}_{x'}} \kappa(y) \right) = (iI)^{\alpha_{x'}} \prod_{y \in \mathcal{H}_{x'}} \kappa(y)$, where in the last equality we have used that all elements of $\kappa(\mathcal{G})$ square to I (as they are Hermitian by choice). This shows $\kappa(xx') = \kappa(x)\kappa(x')$ proving the claim. Next observe that by construction $\kappa(\mathcal{S}) = \langle \tilde{\mathcal{T}} \rangle$, and so $\kappa(\mathcal{S})$ is a stabilizer subgroup of \mathcal{P}_N . We moreover have $\hat{\kappa} = \mu$, because for any $x = (iI)^{\alpha_x} \prod_{y \in \mathcal{H}_x} y \in \mathcal{C}(\mathcal{S})$, we have $\hat{\kappa}(x) = \prod_{y \in \mathcal{H}_x} \hat{\kappa}(y) = \prod_{y \in \mathcal{H}_x} \mu(y) = \mu(\prod_{y \in \mathcal{H}_x} y) = \mu(x)$, where the last equality uses $\mu(iI) = [I]$. Finally the condition $\kappa(iI) = iI$ in Eq. (76), together with the fact that μ is surjective, and $\hat{\kappa} = \mu$, imply that κ is surjective.

We now prove parts (a)-(f) of the lemma, assuming the existence of κ .

- (a) For any two elements $p, q \in \mathcal{C}(\mathcal{S})$, since $\hat{\kappa} = \mu$ and both μ and π are commutation preserving homomorphisms, we have $\text{Com}(p, q) = \text{Com}(\hat{\kappa}(p), \hat{\kappa}(q)) = \text{Com}(\kappa(p), \kappa(q))$.
- (b) If $p \in \mathcal{C}(\mathcal{S}) \setminus \langle iI, \mathcal{S} \rangle$, then there exists $q \in \mathcal{C}(\mathcal{S}) \setminus \langle iI, \mathcal{S} \rangle$ such that $\text{Com}(p, q) = 1$. As κ is commutation preserving by (a), we also have $\text{Com}(\kappa(p), \kappa(q)) = 1$, which is impossible if $\kappa(p) \in \langle iI \rangle$.
- (c) The center of $\mathcal{C}(\mathcal{S})$ is $\langle iI, \mathcal{S} \rangle$, and the center of \mathcal{P}_N is $\langle iI \rangle$. Since κ is a surjective homomorphism, this implies $\kappa(\langle iI, \mathcal{S} \rangle) \subseteq \langle iI \rangle$. Now fix any $p \in \mathcal{S}$. We have already argued in the paragraph preceding the lemma that $\kappa(p) \neq \pm iI$. Since $\kappa(\mathcal{S})$ is a stabilizer group, we also have $\kappa(p) \neq -I$. This shows $\kappa(\mathcal{S}) = \langle I \rangle$ (equivalently \mathcal{S} is in the kernel of κ). Now by the surjectivity of κ and (b), it follows that $\kappa(\langle iI, \mathcal{S} \rangle) = \langle iI \rangle$, and since $\mu(iI) = [I]$, we also have $\kappa(iI) \in \langle iI \rangle$. If $\kappa(iI) = \pm iI$, then $\kappa(\langle iI, \mathcal{S} \rangle) = \langle -I \rangle$, which is a contradiction. This proves $\kappa(iI) = \pm iI$. In each of these two possible cases, since $\kappa(\mathcal{S}) = \{I\}$, we have for any $p \in \langle iI, \mathcal{S} \rangle \setminus \mathcal{S}$, $\kappa(p) \neq I$, and this proves that the kernel of κ is exactly \mathcal{S} .
- (d) It suffices to show that $-I \notin \kappa(\mathcal{S}')$, as then Lemma 3.1(b) implies $\kappa(\mathcal{S}')$ is also commuting and Hermitian. If $\mathcal{S}' = \mathcal{S}$ then $\kappa(\mathcal{S}') = I$ by (c). Now assume $\mathcal{S}' \neq \mathcal{S}$, and for contradiction, assume there exists $p \in \mathcal{S}'$ such that $\kappa(p) = -I$. Then by (c), $p \in \mathcal{S}' \setminus \mathcal{S}$, and since \mathcal{S}' is a stabilizer group it must also hold that $p \notin \langle iI, \mathcal{S} \rangle$. But this contradicts (b).
- (e) As κ is a homomorphism, we have $\kappa(\mathcal{C}(\mathcal{S}')) \subseteq \mathcal{C}(\kappa(\mathcal{S}'))$. We want to show $\mathcal{C}(\kappa(\mathcal{S}')) \subseteq \kappa(\mathcal{C}(\mathcal{S}'))$. Pick any $p \in \mathcal{C}(\kappa(\mathcal{S}'))$. By surjectivity of κ , there exists $q \in \mathcal{C}(\mathcal{S})$ such that $\kappa(q) = p$. Then for all $x \in \mathcal{S}'$, since κ is commutation preserving, we have $\text{Com}(q, x) = \text{Com}(p, \kappa(x)) = 0$, which implies $q \in \mathcal{C}(\mathcal{S}')$.
- (f) Since $\mathcal{S} \subseteq \mathcal{S}'$, we have $\mathcal{S} \subseteq \mathcal{S}' \subseteq \mathcal{C}(\mathcal{S}') \subseteq \mathcal{C}(\mathcal{S})$. Let \mathcal{T} be either \mathcal{S}' or $\mathcal{C}(\mathcal{S}')$, and let $\kappa_{\mathcal{T}} : \mathcal{T} \rightarrow \kappa(\mathcal{T})$ be the restriction of κ to \mathcal{T} . Then $\kappa_{\mathcal{T}}$ is surjective, and the kernel of $\kappa_{\mathcal{T}}$ is \mathcal{S} . Thus we have the isomorphism $\mathcal{T}/\mathcal{S} \cong \kappa(\mathcal{T})$, which implies $\dim(\kappa(\mathcal{T})) = \dim(\mathcal{T}/\mathcal{S}) = \dim(\mathcal{T}) - \dim(\mathcal{S})$. Using this, we easily obtain the equality $\dim(\kappa(\mathcal{C}(\mathcal{S}'))) - \dim(\kappa(\mathcal{S}')) = \dim(\mathcal{C}(\mathcal{S}')) - \dim(\mathcal{S}')$, where the left (resp. right) hand side of the equation denotes twice the number of encoded qubits by the stabilizer code specified by $\kappa(\mathcal{S}')$ (resp. \mathcal{S}').

□

Similarly, a homomorphism $\mu : \mathcal{G} \rightarrow \hat{\mathcal{P}}_N$, where \mathcal{G} is a subgroup of \mathcal{J}_m , is called commutation preserving if for all $\gamma, \gamma' \in \mathcal{G}$, $\text{Com}(\mu(\gamma), \mu(\gamma')) = \text{Com}(\gamma, \gamma')$. A simple fact is the following: if $\mathcal{T} \subseteq \mathcal{G}$ such that $\langle \mathcal{T} \rangle = \mathcal{G}$, and $\mu : \mathcal{G} \rightarrow \hat{\mathcal{P}}_N$ is a homomorphism such that $\text{Com}(\gamma, \gamma') = \text{Com}(\mu(\gamma), \mu(\gamma'))$ for all $\gamma, \gamma' \in \mathcal{T}$, then μ is commutation preserving on \mathcal{G} . To see this, if $\gamma, \gamma' \in \mathcal{G}$, there exists $\gamma_1, \dots, \gamma_{k_\gamma}, \gamma'_1, \dots, \gamma'_{k_{\gamma'}} \in \mathcal{T}$ such that $\gamma = \prod_{s=1}^{k_\gamma} \gamma_s$, and $\gamma' = \prod_{t=1}^{k_{\gamma'}} \gamma'_t$. Then $\text{Com}(\mu(\gamma), \mu(\gamma')) = \text{Com}(\prod_{s=1}^{k_\gamma} \mu(\gamma_s), \prod_{t=1}^{k_{\gamma'}} \mu(\gamma'_t)) = \sum_{s=1}^{k_\gamma} \sum_{t=1}^{k_{\gamma'}} \text{Com}(\mu(\gamma_s), \mu(\gamma'_t)) = \sum_{s=1}^{k_\gamma} \sum_{t=1}^{k_{\gamma'}} \text{Com}(\gamma_s, \gamma'_t) = \text{Com}(\gamma, \gamma')$, where the first equality is true since μ is a homomorphism, and the sums are evaluated over \mathbb{F}_2 .

Let us discuss a consequence of Majorana and qubit partitioning on \mathcal{J}_m and $\hat{\mathcal{P}}_N$, respectively. For $m, N \geq \ell \geq 1$, let J_1, \dots, J_ℓ (resp. J'_1, \dots, J'_ℓ) be a partition of m Majoranas (resp. N qubits), and denote by \mathcal{J}_{m, J_k} (resp. $\hat{\mathcal{P}}_{N, J'_k}$) the subgroup of \mathcal{J}_m (resp. $\hat{\mathcal{P}}_N$) formed by those elements of \mathcal{J}_m

(resp. $\hat{\mathcal{P}}_N$) whose support is contained in J_k (resp. J'_k). We then have $\mathcal{J}_m = \langle \mathcal{J}_{m,J_1}, \dots, \mathcal{J}_{m,J_\ell} \rangle$ and the direct sum $\hat{\mathcal{P}}_N = \bigoplus_{k=1}^\ell \hat{\mathcal{P}}_{N,J'_k}$. Recall that a stabilizer group \mathcal{S} of \mathcal{J}_m satisfies $-I \notin \mathcal{S}$ such that all elements of \mathcal{S} have even weight. Now suppose we are given stabilizer groups $\mathcal{S}_1, \dots, \mathcal{S}_\ell$ of \mathcal{J}_m , such that for each k , $\mathcal{S}_k \subseteq \mathcal{J}_{m,J_k}$, and moreover $\mathcal{S} = \langle \mathcal{S}_1, \dots, \mathcal{S}_\ell \rangle$ is a stabilizer group. Then the centralizer of \mathcal{S} satisfies $\mathcal{C}(\mathcal{S}) = \langle \mathcal{C}(\mathcal{S}_1) \cap \mathcal{J}_{m,J_1}, \dots, \mathcal{C}(\mathcal{S}_\ell) \cap \mathcal{J}_{m,J_\ell} \rangle$. To prove this, note that for any fixed k , if $\gamma \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$, then from the support of γ we deduce using Eq. (7) that $\gamma \in \mathcal{C}(\mathcal{S})$, showing $\langle \mathcal{C}(\mathcal{S}_1) \cap \mathcal{J}_{m,J_1}, \dots, \mathcal{C}(\mathcal{S}_\ell) \cap \mathcal{J}_{m,J_\ell} \rangle \subseteq \mathcal{C}(\mathcal{S})$. Next let $\gamma \in \mathcal{C}(\mathcal{S})$. As γ is an element of \mathcal{J}_m , we have a decomposition $\gamma = \prod_{k=1}^\ell \gamma_k$, where for each k we have $\gamma_k \in \mathcal{J}_{m,J_k}$. Fixing a k , we note that for all $\gamma' \in \mathcal{S}_k$, $\text{Com}(\gamma, \gamma') = 0$ implies $\text{Com}(\gamma_k, \gamma') = 0$ (this uses $\text{Com}(\gamma_{k'}, \gamma') = 0$ for all $k' \neq k$ from support conditions), and this shows $\mathcal{C}(\mathcal{S}) \subseteq \langle \mathcal{C}(\mathcal{S}_1) \cap \mathcal{J}_{m,J_1}, \dots, \mathcal{C}(\mathcal{S}_\ell) \cap \mathcal{J}_{m,J_\ell} \rangle$. This discussion is useful in proving the following lemma:

Lemma E.2. *Suppose we are given*

- (i) $m, N \geq \ell \geq 1$, J_1, \dots, J_ℓ is a partition of m Majoranas, and J'_1, \dots, J'_ℓ is a partition of N qubits.
- (ii) $\mathcal{S}_1, \dots, \mathcal{S}_\ell$ are stabilizer groups of \mathcal{J}_m , satisfying $\mathcal{S}_k \subseteq \mathcal{J}_{m,J_k}$ for all k , and $\mathcal{S} = \langle \mathcal{S}_1, \dots, \mathcal{S}_\ell \rangle$ is a stabilizer group.
- (iii) For all k , and for all $\gamma \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$, γ has even weight.
- (iv) $\mu : \mathcal{C}(\mathcal{S}) \rightarrow \hat{\mathcal{P}}_N$ is a map with the properties: (a) $\mu(\mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}) \subseteq \hat{\mathcal{P}}_{N,J'_k}$ for each k , and (b) if $\gamma = \prod_{k=1}^\ell \gamma_k$ with each $\gamma_k \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$, then $\mu(\gamma) = \prod_{k=1}^\ell \mu(\gamma_k)$.

Denote by μ_k the restriction of μ to $\mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$. Then the following hold:

- (a) Two distinct μ cannot give rise to the same family $\{\mu_k\}_{k=1}^\ell$.
- (b) If $\ell > 1$, then for each k , we have $\mu_k(iI) = [I]$, and $\mu_k(\gamma) = \mu_k(\gamma')$ whenever $\gamma, \gamma' \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$, and γ, γ' are equivalent up to phase factors. Moreover, if we are given a family of maps $\mu_k : \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k} \rightarrow \hat{\mathcal{P}}_{N,J'_k}$ for $1 \leq k \leq \ell$, satisfying these two conditions, then there exists a corresponding μ satisfying (iv), and it is unique.
- (c) μ is a surjective, commutation preserving homomorphism if and only if $\mu_k : \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k} \rightarrow \hat{\mathcal{P}}_{N,J'_k}$ is a surjective, commutation preserving homomorphism, for each k .

Proof. (a) Since $\mathcal{C}(\mathcal{S}) = \langle \mathcal{C}(\mathcal{S}_1) \cap \mathcal{J}_{m,J_1}, \dots, \mathcal{C}(\mathcal{S}_\ell) \cap \mathcal{J}_{m,J_\ell} \rangle$, it implies that if $\gamma \in \mathcal{C}(\mathcal{S})$, then $\gamma = \prod_{k=1}^\ell \gamma_k$ with each $\gamma_k \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$. Then by property (b) of (iv), we get $\mu(\gamma) = \prod_{k=1}^\ell \mu_k(\gamma_k)$ showing that distinct μ cannot give rise to the same family $\{\mu_k\}_{k=1}^\ell$.

(b) If there exists k such that $\mu_k(iI) \neq [I]$, then property (a) of (iv) does not hold. Now suppose there exists k and $\gamma_k, \gamma'_k \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$, such that γ_k, γ'_k are distinct and equivalent up to phase, and $\mu_k(\gamma_k) \neq \mu_k(\gamma'_k)$. Consider any $\gamma \in \mathcal{C}(\mathcal{S})$ given by $\gamma = \prod_{s=1}^\ell \gamma_s$, where for all $s \neq k$, $\gamma_s \in \mathcal{C}(\mathcal{S}_s) \cap \mathcal{J}_{m,J_s}$ are chosen arbitrarily. By adjusting the phases, one also has $\gamma = \prod_{s=1}^\ell \gamma'_s$ with γ_s, γ'_s equivalent up to phase for all s . Then by property (b) of (iv) we get $\prod_{s=1}^\ell \mu_s(\gamma_s) = \prod_{s=1}^\ell \mu_s(\gamma'_s)$, and the support conditions imply $\mu_k(\gamma_k) = \mu_k(\gamma'_k)$ giving a contradiction. To prove the second part, assume we are given a family $\{\mu_k\}_{k=1}^\ell$ satisfying the two conditions of the first part. Take any $\gamma \in \mathcal{C}(\mathcal{S})$, and suppose $\gamma = \prod_{k=1}^\ell \gamma_k$ and $\gamma = \prod_{k=1}^\ell \gamma'_k$ are two different representations of γ , where $\gamma_k, \gamma'_k \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$ for all k . From support conditions, we conclude that γ_k, γ'_k are equivalent up to phase for each k . Thus $\prod_{k=1}^\ell \mu_k(\gamma_k) = \prod_{k=1}^\ell \mu_k(\gamma'_k)$, and so we can define $\mu(\gamma) := \prod_{k=1}^\ell \mu_k(\gamma_k)$ unambiguously. This definition of μ satisfies (iv), and it is unique by (a).

(c) First assume that μ is a surjective, commutation preserving homomorphism, and fix any k . Then if $\gamma, \gamma' \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m,J_k}$, we have $\text{Com}(\mu_k(\gamma), \mu_k(\gamma')) = \text{Com}(\mu(\gamma), \mu(\gamma')) = \text{Com}(\gamma, \gamma')$, and also $\mu_k(\gamma\gamma') = \mu(\gamma\gamma') = \mu(\gamma)\mu(\gamma') = \mu_k(\gamma)\mu_k(\gamma')$, showing that μ_k is a commutation preserving homomorphism. Next let $[q] \in \hat{\mathcal{P}}_{N,J'_k}$. By surjectivity of μ , we have $\gamma \in \mathcal{C}(\mathcal{S})$ such

that $\mu(\gamma) = [q]$. Since we can write $\gamma = \prod_{s=1}^{\ell} \gamma_s$ with $\gamma_s \in \mathcal{C}(\mathcal{S}_s) \cap \mathcal{J}_{m, J_s}$ for each s , it implies $\mu(\gamma) = \prod_{s=1}^{\ell} \mu(\gamma_s) = [q]$. But from property (a) of (iv), it must be that $\mu(\gamma_s) = [I]$ for all $s \neq k$, and so $\mu(\gamma_k) = [q]$ showing that μ_k is surjective. For the converse direction, suppose $\mu_k : \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m, J_k} \rightarrow \hat{\mathcal{P}}_{N, J'_k}$ is a surjective, commutation preserving homomorphism for all k . Take $\gamma, \gamma' \in \mathcal{C}(\mathcal{S})$, and we can write $\gamma = \prod_{k=1}^{\ell} \gamma_k$, and $\gamma' = \prod_{k=1}^{\ell} \gamma'_k$, with $\gamma_k, \gamma'_k \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m, J_k}$ for each k , and thus $\gamma\gamma' = \pm \prod_{k=1}^{\ell} \gamma_k \gamma'_k$. It follows that $\mu(\gamma\gamma') = \mu(\pm \gamma_1 \gamma'_1) \prod_{k=2}^{\ell} \mu(\gamma_k \gamma'_k) = \prod_{k=1}^{\ell} \mu_k(\gamma_k \gamma'_k) = \prod_{k=1}^{\ell} \mu_k(\gamma_k) \mu_k(\gamma'_k)$, where the first equality is by property (b) of (iv), and the second equality uses that $\mu_1(iI) = [I]$ as μ_1 is a surjective, commutation preserving homomorphism (see the first paragraph of the proof of Lemma E.1 for a similar reasoning). Similarly, $\mu(\gamma) = \prod_{k=1}^{\ell} \mu_k(\gamma_k)$ and $\mu(\gamma') = \prod_{k=1}^{\ell} \mu_k(\gamma'_k)$, from which we get $\mu(\gamma)\mu(\gamma') = \mu(\gamma\gamma')$. We also have $\text{Com}(\mu(\gamma), \mu(\gamma')) = \sum_{k=1}^{\ell} \text{Com}(\mu_k(\gamma_k), \mu_k(\gamma'_k)) = \sum_{k=1}^{\ell} \text{Com}(\gamma_k, \gamma'_k) = \text{Com}(\gamma, \gamma')$, where the last equality is by (iii). This shows that μ is a commutation preserving homomorphism. To prove μ is surjective, let $[q] \in \hat{\mathcal{P}}_N$. Then by the direct sum representation of $\hat{\mathcal{P}}_N$, one can write $[q] = \prod_{k=1}^{\ell} [q_k]$ with $[q_k] \in \hat{\mathcal{P}}_{N, J'_k}$ for all k . By surjectivity of μ_k , we have that for all k , there exists $\gamma_k \in \mathcal{C}(\mathcal{S}_k) \cap \mathcal{J}_{m, J_k}$ satisfying $\mu_k(\gamma_k) = [q_k]$. Defining $\gamma = \prod_{k=1}^{\ell} \gamma_k$, gives $\mu(\gamma) = \prod_{k=1}^{\ell} \mu_k(\gamma_k) = \prod_{k=1}^{\ell} [q_k] = [q]$. \square

Using the lemmas and facts presented above in this section, it is now easy to show that the Majorana and the qubit surface codes of Section 3.2 and Section 3.3 respectively, encode the same number of qubits. Recall that the Majorana code is defined by the stabilizer group $\mathcal{S}(G) \subseteq \mathcal{J}_m$, generated by the vertex stabilizers S_v and the face stabilizers S_f , for all $v \in V$ and $f \in F$, as defined by Eqs. (10), (11). We partition the Majoranas into $|V|$ subsets $\{J_v\}_{v \in V}$, with each subset indexed by a vertex $v \in V$. If v is an even degree vertex, then $J_v = \{\gamma_{[h]_{\tau}} : [h]_{\tau} \subseteq v\}$, and if v is an odd degree vertex, then $J_v = \{\gamma_{[h]_{\tau}} : [h]_{\tau} \subseteq v\} \cup \{\tilde{\gamma}_v\}$. Similarly, we also partition the N qubits into $|V|$ subsets $\{J'_v\}_{v \in V}$, where J'_v consists of the N_v qubits placed at v for the qubit surface code (see Definition 3.3). Using this, we complete the proof of Corollary 3.8 below.

Proof of Corollary 3.8. Let \mathcal{S}_v be the stabilizer group generated by S_v , and note that $\mathcal{S}_v \subseteq \mathcal{J}_{m, J_v}$ for all v . Let $\mathcal{S}_{\text{maj}} = \langle \{\mathcal{S}_v : v \in V\} \rangle$, and it is a stabilizer group as $\mathcal{S}_{\text{maj}} \subseteq \mathcal{S}(G)$. Define $m_v = |J_v|$ and note that m_v is even. If $\gamma \in \mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$ then $|\text{supp}(\gamma)|$ is even, as γ must commute with S_v . Thus conditions (i)-(iii) of Lemma E.2 are satisfied (with $\ell = |V|$). Our goal is to define a family of surjective, commutation preserving homomorphisms $\mu_v : \mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v} \rightarrow \hat{\mathcal{P}}_{N, J'_v}$, for each vertex v . Fixing a $v \in V$, note that any such μ_v must satisfy $\mu_v(iI) = [I]$ (one argues similarly as for μ_1 in the proof of the converse part of Lemma E.2(c)), and this ensures $\mu_v(\gamma) = \mu_v(\gamma')$ whenever γ, γ' are equivalent up to phase. Thus the two conditions in Lemma E.2(b) are satisfied by μ_v , so given such a family $\{\mu_v\}_{v \in V}$ one concludes that there exists a unique map $\mu : \mathcal{C}(\mathcal{S}_{\text{maj}}) \rightarrow \hat{\mathcal{P}}_N$ that gives rise to $\{\mu_v\}_{v \in V}$. This ensures μ satisfies condition (iv) of Lemma E.2, and then Lemma E.2(c) allows us to conclude that μ is a surjective, commutation preserving homomorphism.

Starting with any half-edge $h \in v$, assign the Majoranas $\gamma_{[h]_{\tau}}, \gamma_{[\tau\rho h]_{\tau}}, \dots, \gamma_{[(\tau\rho)^{\deg(v)-1}h]_{\tau}}$ in J_v the labels $1, \dots, \deg(v)$ respectively, and if v is an odd-degree vertex, assign the Majorana $\tilde{\gamma}_v$ the label $\deg(v) + 1$ (note that this is independent of the labeling scheme discussed in Section 3.2). Also consider the list of sector operators $\mathcal{T}_v = \{q_{[h]_{\rho}}, q_{[\tau\rho h]_{\rho}}, \dots, q_{[(\tau\rho)^{\deg(v)-1}h]_{\rho}}\} \subseteq \mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$, and let $\chi_v : \mathcal{T}_v \rightarrow \hat{\mathcal{P}}_{N, J'_v}$ be the map that assigns each sector operator to an extremal CAL on N_v qubits (as discussed in Section 3.3). Let $\Xi_v \in \mathbb{F}_2^{(\deg(v)+1) \times m_v}$ be the matrix whose last row is all ones, and the top $\deg(v)$ rows are defined by $(\Xi_v)_{ij} = 1$ if and only if the sector operator $q_{[(\tau\rho)^{i-1}h]_{\rho}}$ contains the Majorana with label j . Let \mathcal{V}_v denote the row space of Ξ_v , and since all rows of Ξ_v are binary vectors of even Hamming weight, Lemma C.12 implies that $\mathcal{V}_v \subseteq \{x \in \mathbb{F}_2^{m_v} : \text{Ham}(x) \text{ is even}\}$. Also define $\mathcal{N}_v = \{y \in \mathbb{F}_2^{\deg(v)+1} : y\Xi_v = \vec{0} \in \mathbb{F}_2^{m_v}\}$.

We want to construct μ_v as an extension of χ_v . Since elements of $\mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$ which are equivalent up to phase, must get mapped to the same element of $\hat{\mathcal{P}}_{N, J'_v}$, we may represent any $\gamma \in \mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$ by a row vector $x \in \mathbb{F}_2^{m_v}$ of even Hamming weight, where $x_j = 1$ if and only if γ contains the Majorana with label j . Given any such x , let $y \in \mathbb{F}_2^{\deg(v)+1}$ be a solution of $x = y\Xi_v$.

Then for any γ represented by x we define

$$\mu_v(\gamma) = \prod_{j:y_j=1} \chi_v(q_{[(\tau\rho)^{j-1}h]_\rho}). \quad (77)$$

To show that μ_v is a well defined map, we need to check (i) for all $x \in \mathbb{F}_2^{m_v}$ of even Hamming weight, the equation $x = y\Xi_v$ has a solution y , and (ii) if multiple solutions exist for the same x , then Eq. (77) does not depend on the solution y picked. Observe that $\text{rank}(\Xi_v) = \text{deg}(v) - 1$ if $\text{deg}(v)$ is even, and $\text{rank}(\Xi_v) = \text{deg}(v)$ if $\text{deg}(v)$ is odd. Thus $\text{rank}(\Xi_v) = m_v - 1$ in both cases, and we conclude that $|\mathcal{V}_v| = 2^{m_v-1} = |\{x \in \mathbb{F}_2^{m_v} : \text{Ham}(x) \text{ is even}\}|$. This proves (i). To prove (ii), let $\vec{1} \in \mathbb{F}_2^{\text{deg}(v)}$ be the all-ones row vector, and we consider the $\text{deg}(v)$ odd or even cases separately. First suppose $\text{deg}(v)$ is odd. Then $\mathcal{N}_v = \text{span}\{\vec{1} \ 0\}$, and thus if $y \neq y'$ satisfy $x = y\Xi_v = y'\Xi_v$, then $y - y' = [\vec{1} \ 0]$. Since $\prod_{j=1}^{\text{deg}(v)} \chi_v(q_{[(\tau\rho)^{j-1}h]_\rho}) = [I]$ by Corollary C.11(a), it follows that Eq. (77) gives the same $\mu_v(\gamma)$ whether we choose y or y' . Next suppose $\text{deg}(v)$ is even. Then $\mathcal{N}_v = \text{span}\{\vec{1} \ 0\}, [z \ 1]\}$, where $z \in \mathbb{F}_2^{\text{deg}(v)}$ satisfies $z_j = 1$ if and only if j is even. In this case, both $\prod_{j=1}^{\text{deg}(v)} \chi_v(q_{[(\tau\rho)^{j-1}h]_\rho}) = [I]$ and $\prod_{j=1}^{\text{deg}(v)/2} \chi_v(q_{[(\tau\rho)^{2j-1}h]_\rho}) = [I]$ by Corollary C.11(b), and so again Eq. (77) is well defined. We also see that μ_v is an extension of χ_v . For any fixed j , consider the sector operator $q_{[(\tau\rho)^{j-1}h]_\rho}$ and let x be its representation. Then a solution to $x = y\Xi_v$ is given by y satisfying $y_{j'} = 1$ if and only if $j' = j$, and thus by Eq. (77) we get $\mu_v(q_{[(\tau\rho)^{j-1}h]_\rho}) = \chi_v(q_{[(\tau\rho)^{j-1}h]_\rho})$.

We now show that the map μ_v is a surjective, commutation preserving homomorphism. Let $\gamma, \gamma' \in \mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$ be represented by $x, x' \in \mathbb{F}_2^{m_v}$ respectively, and suppose y, y' satisfy $x = y\Xi_v, x' = y'\Xi_v$. Then we have $(x + x') = (y + y')\Xi_v$, and $\gamma\gamma'$ is represented by $x + x'$. From Eq. (77) we conclude $\mu_v(\gamma\gamma') = \mu_v(\gamma)\mu_v(\gamma')$ proving μ_v is a homomorphism. Since the CAL $\{\chi_v(q_{[h]_\tau}), \dots, \chi_v(q_{[(\tau\rho)^{\text{deg}(v)-1}h]_\rho})\}$ generates $\hat{\mathcal{P}}_{N, J'_v}$ by Corollary C.10, μ_v being a homomorphism also implies it is surjective. Next note that the fact $x = y\Xi_v$ always has a solution for any $x \in \mathbb{F}_2^{m_v}$ of even Hamming weight, implies $\mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v} = \langle iI, S_v, q_{[h]_\rho}, \dots, q_{[(\tau\rho)^{\text{deg}(v)-1}h]_\rho} \rangle$, and from Eq. (77) we have $\mu_v(iI) = \mu_v(S_v) = [I]$. Thus we see that μ_v is commutation preserving on a generating set of $\mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$, and by the same argument as in the paragraph below the proof of Lemma E.1 we conclude that μ_v is commutation preserving.

As mentioned in the first paragraph of the proof, this finishes the construction of μ whose restriction to $\mathcal{C}(\mathcal{S}_v) \cap \mathcal{J}_{m, J_v}$ is μ_v for each v . Now after identifying \mathcal{J}_m with $\mathcal{P}_{m/2}$ using the Jordan-Wigner map (which is a commutation preserving isomorphism), we can apply Lemma E.1 and conclude the existence of a surjective homomorphism $\kappa : \mathcal{C}(\mathcal{S}_{\text{maj}}) \rightarrow \mathcal{P}_N$ such that $\hat{\kappa} = \mu$, and by Lemma E.1(f) we conclude that the stabilizer codes $\mathcal{S}(G)$ and $\kappa(\mathcal{S}(G))$ encode the same number of logical qubits. It remains to show that $\mu(\mathcal{S}(G))$ equals the stabilizer group of the qubit surface code. By Definition 3.3 and Eq. 14, for each $f \in F$ there is a stabilizer of the qubit surface code given by $\prod_{[h]_\rho \subseteq f} \chi_v([h]_\rho)(q_{[h]_\rho})$, where for each sector $[h]_\rho$, $v([h]_\rho)$ is the unique vertex containing the sector. But by Eq. (77) this is equal to $\mu(S_f)$, and the conclusion follows as $\mu(S_v) = [I]$ for all $v \in V$. \square

F A more general family of cyclic qubit stabilizer codes

In this section we show that the cyclic toric code introduced in Section 3.3 are part of a much larger 2-parameter code family of cyclic qubit stabilizer codes, which in turn are special cases of a 4-parameter cyclic code family. We will prove a few facts about these code families. The results here also provide a completely different proof for the number of encoded qubits for the cyclic toric code (see Theorem 3.9).

F.1 A four parameter cyclic code family

We begin by choosing integers N, p, q, r satisfying the conditions (i) $p, q, r \geq 1$, (ii) $r, q, p + r$ all distinct, and (iii) $\max\{q, p + r\} + 1 \leq N$, which ensures $N \geq 4$. We will always assume in this

section that N, p, q, r satisfies these conditions. Now consider a Pauli string $a_0 a_1 \dots a_{N-1}$ of length N such that

$$a_k = \begin{cases} Z & \text{if } k \in \{0, q\}, \\ X & \text{if } k \in \{r, p+r\}, \\ I_2 & \text{otherwise.} \end{cases} \quad (78)$$

Let S_i denote the Pauli string in Eq. (78) cyclically shifted by i indices to the right. Thus S_0 is the unshifted Pauli string. Let $\mathcal{S} = \{S_i : 0 \leq i \leq N-1\}$ denote the set of all cyclically shifted Pauli strings, and $\langle \mathcal{S} \rangle$ be the subgroup generated by \mathcal{S} . It will be convenient to associate subsets of $\mathcal{N} := \{0, \dots, N-1\}$ with subsets of \mathcal{S} , so we introduce the map $\Gamma : 2^{\mathcal{N}} \rightarrow 2^{\mathcal{S}}$ defined as $\Gamma(\mathcal{X}) = \{S_x : x \in \mathcal{X}\}$, for all $\mathcal{X} \in 2^{\mathcal{N}}$.

If all the cyclic shifts of this Pauli string commute with one another, they can be used to define a stabilizer group with each shifted Pauli string defining a stabilizer up to a phase factor of 1 or -1 (see Lemma 3.1(a) for the equivalent statement about the Majorana group, which also applies for the Pauli group due to identification of the two groups via the Jordan-Wigner transform). The quantum stabilizer code that results will be called a 4-parameter (N, p, q, r) cyclic code. Depending on the values of r, q , and $p+r$, there are three distinct subfamilies of these codes. The explicit form of the Pauli string S_0 corresponding to these three subfamilies are given below:

$$S_0 = \begin{cases} ZI^{\otimes r-1} XI^{\otimes q-r-1} ZI^{\otimes p+r-q-1} XI^{\otimes N-p-r-1} & \text{if } r < q < p+r, & \text{(subfamily A)} \\ ZI^{\otimes r-1} XI^{\otimes p-1} XI^{\otimes q-p-r-1} ZI^{\otimes N-q-1} & \text{if } r < p+r < q, & \text{(subfamily B)} \\ ZI^{\otimes q-1} ZI^{\otimes r-q-1} XI^{\otimes p-1} XI^{\otimes N-p-r-1} & \text{if } q < r < p+r. & \text{(subfamily C)} \end{cases} \quad (79)$$

We would like to understand the conditions on the values of N, p, q, r under which all cyclic shifts of S_0 for each subfamily in Eq. (79) commute with one another, which is necessary to define a valid stabilizer group. The following lemma is important towards achieving this goal.

Lemma F.1. *For any Pauli string defined in Eq. (78), $\langle \mathcal{S} \rangle$ is a commuting subgroup if and only if the set $\{S_0, S_q, S_r, S_{p+r}\}$ is pairwise commuting.*

Proof. In this proof all qubit indices will be evaluated modulo N . If $\langle \mathcal{S} \rangle$ is a commuting subgroup, the set $\{S_0, S_q, S_r, S_{p+r}\}$ is pairwise commuting because it is a subset of $\langle \mathcal{S} \rangle$. For the other direction, assume that $\{S_0, S_q, S_r, S_{p+r}\}$ is a pairwise commuting set. To show that $\langle \mathcal{S} \rangle$ is a commuting subgroup, it suffices to show that \mathcal{S} is pairwise commuting, which is in turn equivalent to showing that S_0 commutes with each element in \mathcal{S} , because of cyclicity. For contradiction, assume that this is not true, so there exists $S_\ell \in \mathcal{S}$ that anticommutes with S_0 . This is only possible if there exists a qubit index $0 \leq n \leq N-1$, such that both S_0 and S_ℓ contain a Pauli different from I_2 (not necessarily the same) at that qubit, and so this implies $\ell \in \{\pm p, \pm q, \pm r, \pm(p+r), \pm(q-r), \pm(p+r-q)\}$. Now for any index ℓ' , it follows using cyclicity that S_0 commutes with $S_{\ell'}$ if and only if S_0 commutes with $S_{-\ell'}$, because the relative shifts in both cases are the same. Also by assumption S_0 commutes with each of S_q, S_r , and S_{p+r} . Thus we deduce that $\ell \in \{p, q-r, p+r-q\}$. Using cyclicity again, it also follows by the same reasoning that (i) S_0 and S_p are commuting if and only if S_r and S_{p+r} are commuting, (ii) S_0 and S_{q-r} are commuting if and only if S_r and S_q are commuting, and (iii) S_0 and S_{p+r-q} are commuting if and only if S_q and S_{p+r} are commuting. But this is a contradiction as S_q, S_r and S_{p+r} are pairwise commuting by assumption. \square

Because of Lemma F.1, given any values of N, p, q, r , one only needs to check whether all six pairs $\{S_0, S_q\}$, $\{S_0, S_r\}$, $\{S_0, S_{p+r}\}$, $\{S_q, S_r\}$, $\{S_q, S_{p+r}\}$ and $\{S_r, S_{p+r}\}$ are commuting or not. Each of these checks has complexity $O(1)$, independent of N , because each Pauli string in $\{S_0, S_q, S_r, S_{p+r}\}$ has bounded support, and so determining whether $\langle \mathcal{S} \rangle$ is a commuting subgroup or not also has complexity $O(1)$ for any input values of N, p, q, r . We will say that a quadruple (N, p, q, r) is *consistent* if and only if $\langle \mathcal{S} \rangle$ is a commuting subgroup. For example, the following special case is of particular interest.

Lemma F.2. *A quadruple (N, p, q, r) satisfying $r < p+r < q$ is consistent if $q = p+2r$, and corresponds to a cyclic code of subfamily B.*

Proof. In this proof all qubit indices will be implicitly assumed to be modulo N . If we prove that the quadruple is consistent then we know from Eq. (79) that the cyclic code belongs to subfamily B. First note that the Pauli string S_0 has the form $ZI^{\otimes r-1}XI^{\otimes p-1}XI^{\otimes r-1}ZI^{\otimes N-p-2r-1}$. Let $[a \ b]$ be the symplectic representation of S_0 and $[a' \ b']$ be the symplectic representation of S_m , for some $0 \leq m \leq N-1$, where $a, a', b, b' \in \mathbb{F}_2^N$. Then S_0 and S_m commute if and only if $(a \cdot b' + a' \cdot b) = 0$, where the left hand side is evaluated over \mathbb{F}_2 . Now note that $a_i = 1$ if and only if $i \in \{r, p+r\}$, $b_i = 1$ if and only if $i \in \{0, p+2r\}$, $a'_i = 1$ if and only if $i \in \{m+r, m+p+r\}$, and $b'_i = 1$ if and only if $i \in \{m, m+p+2r\}$. Then

$$\begin{aligned}
a \cdot b' + a' \cdot b &= (\delta_{r,m} + \delta_{r,m+p+2r} + \delta_{p+r,m} + \delta_{p+r,m+p+2r}) \\
&\quad + (\delta_{m+r,0} + \delta_{m+r,p+2r} + \delta_{m+p+r,0} + \delta_{m+p+r,p+2r}) \\
&= (\delta_{r,m} + \delta_{0,m+p+r} + \delta_{p+r,m} + \delta_{0,m+r}) + (\delta_{m+r,0} + \delta_{m,p+r} + \delta_{m+p+r,0} + \delta_{m,r}) \\
&= (\delta_{r,m} + \delta_{m,r}) + (\delta_{0,m+p+r} + \delta_{m+p+r,0}) + (\delta_{p+r,m} + \delta_{m,p+r}) + (\delta_{0,m+r} + \delta_{m+r,0}) \\
&= 0.
\end{aligned} \tag{80}$$

Since m is arbitrary this proves that S_0 and S_m commute for all $0 \leq m \leq N-1$, and using cyclicity we then conclude that (N, p, q, r) is consistent. \square

Given a consistent quadruple (N, p, q, r) , we would like to know the number of qubits encoded by the corresponding cyclic code. The answer to this question is completely provided by the next theorem, which we state below.

Theorem F.3. *For any 4-parameter (N, p, q, r) cyclic code, the number of encoded qubits satisfies*

$$K = \frac{\gcd(p, N) \gcd(q, N)}{\gcd(\text{lcm}(p, q), N)} \tag{81}$$

and is independent of r .

The proof of this theorem is delayed to first present a few general facts, on which the proof depends on. Below we start by proving Lemma F.4, which is then used to prove Lemma F.5, the latter being the most important result that we will need. We then prove Lemma F.6, which is a structure lemma that tells us exactly which subsets of the stabilizers of the 4-parameter (N, p, q, r) cyclic code are independent. Combined with Lemma F.5, this immediately proves Theorem F.3. In the proofs of Lemma F.6 and Theorem F.3, we use the notation introduced in the paragraph below Eq. (78), that associates subsets of $\{0, \dots, N-1\}$ to subsets of stabilizers for the 4-parameter (N, p, q, r) cyclic code.

Lemma F.4. *Let $a, b, m \geq 1$ be positive integers, and α, β, γ be real numbers. Then*

$$(a) \quad \min(\gamma - \min(\alpha, \gamma), \gamma - \min(\beta, \gamma)) = \gamma - \min(\gamma, \alpha + \beta - \min(\alpha, \beta)), \tag{82}$$

$$(b) \quad \gcd\left(\frac{m}{\gcd(a, m)}, \frac{m}{\gcd(b, m)}\right) = \frac{m}{\gcd(\text{lcm}(a, b), m)}. \tag{83}$$

Proof. (a) Without loss of generality assume that $\alpha \geq \beta$. Then $\alpha + \beta - \min(\alpha, \beta) = \alpha$, so the expression on the right-hand side of Eq. (82) simplifies to $\gamma - \min(\gamma, \alpha)$. First consider the case $\gamma \leq \alpha$. We have $\gamma - \min(\gamma, \alpha) = 0$, while $\min(\gamma - \min(\alpha, \gamma), \gamma - \min(\beta, \gamma)) = \min(0, \gamma - \min(\beta, \gamma)) = 0$, since $\gamma - \min(\beta, \gamma) \geq 0$. Next consider the case $\gamma > \alpha \geq \beta$. Then $\gamma - \min(\gamma, \alpha) = \gamma - \alpha$, and $\min(\gamma - \min(\alpha, \gamma), \gamma - \min(\beta, \gamma)) = \min(\gamma - \alpha, \gamma - \beta) = \gamma - \alpha$. This proves Eq. (82) in both cases.

(b) Let $\{p_i\}_{i \geq 1}$ be the enumeration of all the primes in ascending order. Then there exists an n such that a, b, m have the following unique representations

$$a = \prod_{i=1}^n p_i^{\alpha_i}, \quad b = \prod_{i=1}^n p_i^{\beta_i}, \quad m = \prod_{i=1}^n p_i^{\gamma_i}, \tag{84}$$

where $\alpha_i, \beta_i, \gamma_i \in \{0, 1, 2, \dots\}$ for all $1 \leq i \leq n$. Using Eq. (84) it then easily follows that

$$\begin{aligned} \gcd\left(\frac{m}{\gcd(a, m)}, \frac{m}{\gcd(b, m)}\right) &= \gcd\left(\prod_{i=1}^n p_i^{\gamma_i - \min(\alpha_i, \gamma_i)}, \prod_{i=1}^n p_i^{\gamma_i - \min(\beta_i, \gamma_i)}\right) \\ &= \prod_{i=1}^n p_i^{\min(\gamma_i - \min(\alpha_i, \gamma_i), \gamma_i - \min(\beta_i, \gamma_i))}, \end{aligned} \quad (85)$$

and

$$\frac{m}{\gcd(\text{lcm}(a, b), m)} = \frac{\prod_{i=1}^n p_i^{\gamma_i}}{\gcd\left(\prod_{i=1}^n p_i^{\alpha_i + \beta_i - \min(\alpha_i, \beta_i)}, \prod_{i=1}^n p_i^{\gamma_i}\right)} = \prod_{i=1}^n p_i^{\gamma_i - \min(\gamma_i, \alpha_i + \beta_i - \min(\alpha_i, \beta_i))}, \quad (86)$$

and thus by (a) these two expressions are equal. \square

Lemma F.5. *Let $a, b, m \geq 1$ be positive integers, and let “+” denote addition modulo m . Then the set of integers $\mathcal{G} = \{k_1 a + k_2 b : k_1, k_2 \in \mathbb{Z}\}$ is a cyclic subgroup of $(\mathbb{Z}/m\mathbb{Z}, +)$, under +. The order of \mathcal{G} satisfies*

$$|\mathcal{G}| = \frac{m \gcd(\text{lcm}(a, b), m)}{\gcd(a, m) \gcd(b, m)}. \quad (87)$$

The smallest positive integer that generates \mathcal{G} is $\gcd(a, m) \gcd(b, m) / \gcd(\text{lcm}(a, b), m)$.

Proof. \mathcal{G} is closed under +, and for every $x \in \mathcal{G}$, $-x \bmod m \in \mathcal{G}$, and so each element has an additive inverse. Thus \mathcal{G} is a subgroup of $(\mathbb{Z}/m\mathbb{Z}, +)$ since $\mathcal{G} \subseteq \mathbb{Z}/m\mathbb{Z}$, and because $(\mathbb{Z}/m\mathbb{Z}, +)$ is a cyclic group, it follows that \mathcal{G} is cyclic.

We now derive the expression for $|\mathcal{G}|$ in Eq. (87), which we break up into a few steps.

- (i) **Step 1:** We first construct two subsets of \mathcal{G} : $\mathcal{G}_a = \{k_1 a \bmod m : k_1 \in \mathbb{Z}\}$, and $\mathcal{G}_b = \{k_2 b \bmod m : k_2 \in \mathbb{Z}\}$. Using the same argument as for \mathcal{G} , we have that \mathcal{G}_a and \mathcal{G}_b are cyclic subgroups of \mathcal{G} under +, hence also of $(\mathbb{Z}/m\mathbb{Z}, +)$. Since $\text{lcm}(a, m)$ is the smallest positive multiple of a divisible by m , and noting that $\text{lcm}(a, m)/a = m/\gcd(a, m)$, it follows that $\mathcal{G}_a = \{k_1 a \bmod m : 0 \leq k_1 \leq m/\gcd(a, m) - 1\}$. Moreover for any two distinct integers $0 \leq k_1, k'_1 \leq m/\gcd(a, m) - 1$, we have $k_1 a \not\equiv k'_1 a \pmod{m}$ because otherwise $|k_1 - k'_1|a$ is a positive multiple of a strictly smaller than $\text{lcm}(a, m)$ that is also divisible by m . Thus $|\mathcal{G}_a| = m/\gcd(a, m)$, and similarly $|\mathcal{G}_b| = m/\gcd(b, m)$.
- (ii) **Step 2:** Our next goal is to show that $\mathcal{G}_a = \{k_1 \gcd(a, m) : 0 \leq k_1 \leq m/\gcd(a, m) - 1\}$. If $\gcd(a, m) = m$ then the result is clearly true, so assume that $\gcd(a, m) < m$. By Bézout’s identity $\gcd(a, m) = xa + ym$ for some integers x, y , i.e. $\gcd(a, m) \equiv xa \pmod{m}$, and so $\gcd(a, m) \in \mathcal{G}_a$. But this implies that $\{k_1 \gcd(a, m) : 0 \leq k_1 \leq m/\gcd(a, m) - 1\} \subseteq \mathcal{G}_a$, since \mathcal{G}_a forms a group under +. But all the elements in this subset are distinct and the number of elements in it equal $m/\gcd(a, m)$, which is the order of \mathcal{G}_a , and so we have in fact proved that $\mathcal{G}_a = \{k_1 \gcd(a, m) : 0 \leq k_1 \leq m/\gcd(a, m) - 1\}$, and analogously $\mathcal{G}_b = \{k_2 \gcd(b, m) : 0 \leq k_2 \leq m/\gcd(b, m) - 1\}$. This also shows that $\gcd(a, m)$ and $\gcd(b, m)$ are the smallest positive integers that cyclically generate \mathcal{G}_a and \mathcal{G}_b respectively.
- (iii) **Step 3:** We now want to locate the smallest non-zero integer p in \mathcal{G}_a that is also in \mathcal{G}_b . For this we consider two subsets: $\mathcal{G}_a \cap \mathcal{G}_b$ and $\mathcal{G}' := \{k_3 \text{lcm}(a, b) \bmod m : k_3 \in \mathbb{Z}\}$. First note that $\mathcal{G}_a \cap \mathcal{G}_b$ forms a subgroup of both \mathcal{G}_a and \mathcal{G}_b under +, because it is the intersection of two subgroups; hence Lagrange’s theorem implies $|\mathcal{G}_a \cap \mathcal{G}_b| \leq \gcd(|\mathcal{G}_a|, |\mathcal{G}_b|)$. Also since each element in $\mathcal{G}_a \cap \mathcal{G}_b$ is a multiple of both a and b modulo m , it follows that $\mathcal{G}' \subseteq \mathcal{G}_a \cap \mathcal{G}_b$, and so $|\mathcal{G}_a \cap \mathcal{G}_b| \geq |\mathcal{G}'|$. Next, reasoning similarly as in Steps 1 and 2 for \mathcal{G}_a and \mathcal{G}_b , we conclude that \mathcal{G}' is a cyclic group under +, $\mathcal{G}' = \{k_3 \gcd(\text{lcm}(a, b), m) : 0 \leq k_3 \leq m/\gcd(\text{lcm}(a, b), m) - 1\}$, and $|\mathcal{G}'| = m/\gcd(\text{lcm}(a, b), m)$. But applying Lemma F.4(b) gives that $m/\gcd(\text{lcm}(a, b), m) = \gcd(m/\gcd(a, m), m/\gcd(b, m)) = \gcd(|\mathcal{G}_a|, |\mathcal{G}_b|)$. We have thus proved that $\mathcal{G}' = \mathcal{G}_a \cap \mathcal{G}_b$, and it follows that $p = \gcd(\text{lcm}(a, b), m)$.

- (iv) **Step 4:** We now give an equivalent characterization of \mathcal{G} that will help us count the number of elements in it. Let $q = p/\gcd(a, m)$, where p is from Step 3, and notice that q is an integer because

$$\begin{aligned} \gcd(\text{lcm}(a, b), m) &= \gcd\left(\gcd(a, m)\frac{a}{\gcd(a, m)}\frac{b}{\gcd(a, b)}, \gcd(a, m)\frac{m}{\gcd(a, m)}\right) \\ &= \gcd(a, m)\gcd\left(\frac{a}{\gcd(a, m)}\frac{b}{\gcd(a, b)}, \frac{m}{\gcd(a, m)}\right), \end{aligned} \quad (88)$$

and moreover m is a multiple of q , because p divides m . Now construct the set

$$\mathcal{G}'' = \bigcup_{k_1=0}^{q-1} (k_1 \gcd(a, m) + \mathcal{G}_b). \quad (89)$$

We want to prove that $\mathcal{G}'' = \mathcal{G}$. Clearly we have that $\mathcal{G}'' \subseteq \mathcal{G}$, so we only need to show that $\mathcal{G} \subseteq \mathcal{G}''$. If $r \in \mathcal{G}$ then there exists $k_1, k_2 \in \mathbb{Z}$, such that $r = k_1 a + k_2 b$. But $k_1 a + k_2 b = (k_1 a \bmod m) + (k_2 b \bmod m)$, and so $r = x + y$, for some $x \in \mathcal{G}_a$ and $y \in \mathcal{G}_b$. Note that any element $x \in \mathcal{G}_a$ can be expressed as $x = x' + z$ for some $z \in \mathcal{G}_a \cap \mathcal{G}_b$ and $x' \in \{k_1 \gcd(a, m) : 0 \leq k_1 \leq q-1\}$, and thus we have $r = x' + y + z = x' + y'$ for some $y' \in \mathcal{G}_b$, since both $y, z \in \mathcal{G}_b$. This proves $\mathcal{G} \subseteq \mathcal{G}''$.

- (v) **Step 5:** We now want to show that the union appearing in Eq. (89) is a disjoint union. Note that this would imply $|\mathcal{G}| = \sum_{k_1=0}^{q-1} |k_1 \gcd(a, m) + \mathcal{G}_b| = \sum_{k_1=0}^{q-1} |\mathcal{G}_b| = q|\mathcal{G}_b|$ by Step 4, thus proving Eq. (87). So for the sake of contradiction, assume that there exists distinct integers $0 \leq k_1 < k'_1 \leq q-1$ such that $(k_1 \gcd(a, m) + \mathcal{G}_b) \cap (k'_1 \gcd(a, m) + \mathcal{G}_b) \neq \emptyset$. Then there exists $b, b' \in \mathcal{G}_b$ such that $k_1 \gcd(a, m) + b = k'_1 \gcd(a, m) + b'$, and so $(k'_1 - k_1) \gcd(a, m) \equiv (b - b') \pmod{m}$ which implies $(k'_1 - k_1) \gcd(a, m) \in \mathcal{G}_b$. But $k'_1 - k_1 < q$ and so $(k'_1 - k_1) \gcd(a, m) < p$, and since p is the smallest non-zero element in $\mathcal{G}_a \cap \mathcal{G}_b$ by Step 3, this implies $k'_1 = k_1$ which is a contradiction.

For the last part, let s be the smallest non-zero integer in \mathcal{G} ; so $\{ks : 0 \leq k \leq \lceil m/s \rceil - 1\} \subseteq \mathcal{G}$ because \mathcal{G} is a group. Then s divides m because otherwise $t = s\lceil m/s \rceil \bmod m \in \mathcal{G}$, $0 \neq t < s$. Suppose $u \in \mathcal{G}$ and $u \notin \{ks : 0 \leq k \leq m/s - 1\}$. Then $0 \neq u - s\lfloor u/s \rfloor \in \mathcal{G}$ and $u - s\lfloor u/s \rfloor < s$ which is a contradiction. This proves that $\mathcal{G} = \{ks : 0 \leq k \leq m/s - 1\}$, and so $m/s = |\mathcal{G}|$, or equivalently $s = \gcd(a, m)\gcd(b, m)/\gcd(\text{lcm}(a, b), m)$. Thus s is the smallest positive integer that cyclically generates \mathcal{G} . \square

Lemma F.6. Let “+” denote addition modulo N , $\mathcal{G} = \{k_1 p + k_2 q : k_1, k_2 \in \mathbb{Z}\}$, $\mathcal{G}_p = \{kp + 0 : k \in \mathbb{Z}\}$, and $\mathcal{G}_q = \{kq + 0 : k \in \mathbb{Z}\}$. If $\mathcal{A} \subseteq \mathcal{N}$ and $s \in \mathbb{Z}$, denote $\mathcal{A} + s := \{x + s : x \in \mathcal{A}\}$. Then for any quadruple (N, p, q, r) (not necessarily consistent), and for any non-empty $\mathcal{H} \subseteq \mathcal{N}$, the following holds:

- (a) $\Gamma(\mathcal{H})$ multiplies to a X -type Pauli string if and only if \mathcal{H} is a union of the cosets of \mathcal{G}_q in $(\mathbb{Z}/N\mathbb{Z}, +)$.
- (b) $\Gamma(\mathcal{H})$ multiplies to a Z -type Pauli string if and only if \mathcal{H} is a union of the cosets of \mathcal{G}_p in $(\mathbb{Z}/N\mathbb{Z}, +)$.
- (c) $\Gamma(\mathcal{H})$ multiplies to I (up to phase factors) if and only if \mathcal{H} is a union of the cosets of \mathcal{G} in $(\mathbb{Z}/N\mathbb{Z}, +)$.

Proof. In this proof all qubit indices and cyclic shifts will be implicitly assumed to be modulo N , and products of Pauli strings in \mathcal{S} will be considered modulo phase factors. Note that we have already proved \mathcal{G} , \mathcal{G}_p , and \mathcal{G}_q are subgroups of $(\mathbb{Z}/N\mathbb{Z}, +)$ under $+$ by Lemma F.5, so cosets of \mathcal{G} , \mathcal{G}_p , and \mathcal{G}_q are well-defined. It will also be useful to remember the following fact: if $\mathcal{T} \subseteq \mathcal{S}$, and \mathcal{T}' is obtained by cyclically shifting each Pauli string in \mathcal{T} by some integer k , then $\prod \mathcal{T}'$ is obtained by cyclically shifting $\prod \mathcal{T}$ by k .

We next make the following claims, which we prove at the end of the proof: (i) $\prod \Gamma(\mathcal{G}_q)$ is a X -type Pauli string, and $\prod \Gamma(\mathcal{G}_p)$ is a Z -type Pauli string, (ii) if $\prod \Gamma(\mathcal{H}')$ is a X -type Pauli string for any

$\mathcal{H}' \subseteq \mathcal{N}$ such that $x \in \mathcal{H}'$, then $\mathcal{G}_q + x \subseteq \mathcal{H}'$, (iii) if $\prod \Gamma(\mathcal{H}')$ is a Z -type Pauli string for any $\mathcal{H}' \subseteq \mathcal{N}$ such that $x \in \mathcal{H}'$, then $\mathcal{G}_p + x \subseteq \mathcal{H}'$, and (iv) if $\prod \Gamma(\mathcal{H}')$ equals I (up to phase factors) for any $\mathcal{H}' \subseteq \mathcal{N}$ such that $x \in \mathcal{H}'$, then $\mathcal{G} + x \subseteq \mathcal{H}'$.

- (a) Suppose \mathcal{H} is a union of the cosets of \mathcal{G}_q in $(\mathbb{Z}/N\mathbb{Z}, +)$. By claim (i) and the fact mentioned at the beginning of the proof, if \mathcal{H}' is a coset of \mathcal{G}_q in $(\mathbb{Z}/N\mathbb{Z}, +)$, then $\prod \Gamma(\mathcal{H}')$ is a X -type Pauli string, since $\mathcal{H}' = \mathcal{G}_q + s$ for some $s \in \mathcal{N}$. Thus $\prod \Gamma(\mathcal{H})$ is also a X -type Pauli string. For the converse, suppose for contradiction that \mathcal{H} is non-empty, $\prod \Gamma(\mathcal{H})$ is a X -type Pauli string, and \mathcal{H} is not a union of the cosets of \mathcal{G}_q . Then there exists an integer x belonging to some coset \mathcal{H}' of \mathcal{G}_q such that $x \in \mathcal{H}$ and $\mathcal{H}' \not\subseteq \mathcal{H}$. So $\mathcal{H}' = \mathcal{G}_q + x$, which gives a contradiction by claim (ii).
- (b) This follows from (a) by performing Clifford transformations on every qubit interchanging X and Z , and using cyclicity.
- (c) Assume \mathcal{H} is a union of the cosets of \mathcal{G} in $(\mathbb{Z}/N\mathbb{Z}, +)$. Since \mathcal{G}_q (resp. \mathcal{G}_p) is a subgroup of \mathcal{G} , it implies \mathcal{G} is a union of the cosets of \mathcal{G}_q (resp. \mathcal{G}_p) in $(\mathbb{Z}/N\mathbb{Z}, +)$. Thus by (a) and (b), $\prod \Gamma(\mathcal{H})$ is both a X -type and Z -type Pauli string, and so it must equal I up to phase factors. To prove the converse, suppose for contradiction that \mathcal{H} is non-empty, $\prod \Gamma(\mathcal{H})$ equals I (up to phase factors), and \mathcal{H} is not a union of the cosets of \mathcal{G} . Then similar to the proof of the converse in (a), there exists an integer x belonging to some coset \mathcal{H}' of \mathcal{G} such that $x \in \mathcal{H}$ and $\mathcal{H}' \not\subseteq \mathcal{H}$. This implies $\mathcal{H}' = \mathcal{G} + x$, but this contradicts claim (iv).

We now prove the claims made above.

- (i) We observe that for any qubit index $n \in \mathcal{N}$, there are exactly two Pauli strings S_n and S_{n-q} containing Z at index n , exactly two other Pauli strings S_{n-r} and S_{n-p-r} containing X at index n , while all others have I_2 at index n (note that S_n, S_{n-q}, S_{n-r} , and S_{n-p-r} are distinct by assumptions on p, q, r). Since the relative shift of S_n and S_{n-q} is q , and the relative shift of S_{n-r} and S_{n-p-r} is p , it follows from the group structure of \mathcal{G}_q and \mathcal{G}_p that $S_n \in \Gamma(\mathcal{G}_q)$ if and only if $S_{n-q} \in \Gamma(\mathcal{G}_q)$, and $S_{n-r} \in \Gamma(\mathcal{G}_p)$ if and only if $S_{n-p-r} \in \Gamma(\mathcal{G}_p)$. Thus $\prod \Gamma(\mathcal{G}_q)$ has either X or I_2 at index n , and $\prod \Gamma(\mathcal{G}_p)$ has either Z or I_2 at index n .
- (ii) By cyclicity, it suffices to prove this for the special case $x = 0$. So suppose $0 \in \mathcal{H}' \subseteq \mathcal{N}$, $\prod \Gamma(\mathcal{H}')$ is a X -type Pauli string, and for contradiction also assume $\mathcal{G}_q \not\subseteq \mathcal{H}'$. Noting that $\mathcal{G}_q = \{kq \bmod N : 0 \leq k \leq N/\gcd(q, N) - 1\}$, then there exists $0 < k' \leq N/\gcd(q, N) - 1$, such that $k'q \bmod N \notin \mathcal{H}'$ and $k''q \bmod N \in \mathcal{H}'$ for all $0 \leq k'' < k'$. Thus the Pauli string $S_{(k'-1)q} \in \Gamma(\mathcal{H}')$, which contains Z at qubit index $k'q$. The only other Pauli string that has Z at the same index is $S_{k'q}$, but $S_{k'q} \notin \Gamma(\mathcal{H}')$. Thus $\prod \Gamma(\mathcal{H}')$ is not X -type, giving a contradiction.
- (iii) This follows from claim (ii) by cyclicity, i.e. cyclically shifting every element of \mathcal{S} by r units to the left, and then performing Clifford transformations on every qubit swapping X and Z .
- (iv) Let \mathcal{H}' be as in the claim. Since $\prod \Gamma(\mathcal{H}')$ is a X -type Pauli string, we conclude by claim (ii) that $\mathcal{G}_q \subseteq \mathcal{H}'$. Next fix any $q' \in \mathcal{G}_q$, and consider the set $\mathcal{G}_p + q' = \{kp + q' : 0 \leq k \leq N/\gcd(p, N) - 1\}$. Since $\prod \Gamma(\mathcal{H}')$ is a Z -type Pauli string, by claim (iii) we have $\mathcal{G}_p + q' \subseteq \mathcal{H}'$. As q' is arbitrary, this proves $\mathcal{G} = \bigcup_{q' \in \mathcal{G}_q} (\mathcal{G}_p + q') \subseteq \mathcal{H}'$. \square

Proof of Theorem F.3. Let \mathcal{G} be defined as in Lemma F.6. For each coset \mathcal{H} of \mathcal{G} in $(\mathbb{Z}/N\mathbb{Z}, +)$, we can then remove exactly one stabilizer from $\Gamma(\mathcal{H})$ and take the union of the resulting sets, to get a set of stabilizers \mathcal{H}' that still generates $\langle \mathcal{S} \rangle$ by Lemma F.6(c). Since the number of cosets of \mathcal{G} is $N/|\mathcal{G}|$, it follows by Lemma F.5 that $|\mathcal{H}'| = N - \gcd(p, N) \gcd(q, N) / \gcd(\text{lcm}(p, q), N)$. Moreover $\Gamma(\mathcal{H}')$ is independent as it has no subset that multiplies to I , again by Lemma F.6(c), and so the number of independent generators of $\langle \mathcal{S} \rangle$ equals $|\mathcal{H}'|$. The proof is complete noting that $K = N - |\mathcal{H}'|$. \square

Combining the results of Lemma F.5, Lemma F.6(c), and Theorem F.3 we also have the following corollary, which allows us to extract an independent subset of stabilizers that generate $\langle \mathcal{S} \rangle$ from the knowledge of N and K alone.

Corollary F.7. *If a 4-parameter (N, p, q, r) cyclic code encodes K qubits, then $K \geq 1$ and divides N . Moreover a non-empty subset of \mathcal{S} multiplies to I (up to phase factors) if and only if it is a non-empty union of the subsets of \mathcal{S} corresponding to the cosets of \mathcal{G} in $(\mathbb{Z}/N\mathbb{Z}, +)$, where $\mathcal{G} = \{kK : 0 \leq k \leq N/K - 1\}$.*

Proof. The second part is a mere restatement of Lemma F.6(c), combining the result proved in the last paragraph of the proof of Lemma F.5, and the expression of K in Theorem F.3. That $K \geq 1$ and K divides N follows because $\gcd(p, N)\gcd(q, N)/\gcd(\text{lcm}(p, q), N)$ divides N , the proof of which is already contained in the proof of Lemma F.5. \square

As the final result of this subsection, we prove in Lemma F.9 an interesting property of the subgroup $\langle \mathcal{S} \rangle$ for any quadruple (N, p, q, r) , which finds use in the next subsection. The lemma studies the membership of the Pauli strings $X^{\otimes N}$ and $Z^{\otimes N}$ in $\langle \mathcal{S} \rangle$, where membership is decided up to phase factors. Thus we will say for instance that $X^{\otimes N}$ belongs to the subgroup $\langle \mathcal{S} \rangle$ if and only if $\{\pm X^{\otimes N}, \pm iX^{\otimes N}\} \cap \langle \mathcal{S} \rangle \neq \emptyset$. For the proof, we need the following fact (Lemma F.8) that we first prove. Lemma F.9 is stated and proved after that.

Lemma F.8. *Let $1 \leq a, m \in \mathbb{Z}$, $\mathcal{M} = \{x \in \mathbb{Z} : 0 \leq x \leq m - 1\}$, and “+” denote addition modulo m . For any $\mathcal{A} \subseteq \mathcal{M}$, denote $\mathcal{A} + a = \{x + a : x \in \mathcal{A}\}$. A partition $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$ satisfying $\mathcal{M}_2 = \mathcal{M}_1 + a$ exists, if and only if $m/\gcd(a, m) = \text{lcm}(a, m)/a \equiv 0 \pmod{2}$.*

Proof. Let $\mathcal{G}_a = \{ka + 0 : k \in \mathbb{Z}\}$ denote an orbit of the function that adds a modulo m , i.e. $f(x) = x + a$. Each orbit $\mathcal{G}_a + x$ has size $\text{lcm}(a, m)/a$, and for $x = 0, 1, \dots, \gcd(a, m) - 1$ they are all disjoint. We may write $\mathcal{M} = \bigsqcup_{x=0}^{\gcd(a, m)-1} (\mathcal{G}_a + x)$.

Suppose $\text{lcm}(a, m)/a \equiv 0 \pmod{2}$ so the orbits have even size. Create the sets \mathcal{M}_1 and \mathcal{M}_2 as follows. For each $\mathcal{G}_a + x$, place $x + ka$ in \mathcal{M}_1 if k is even and $x + ka$ in \mathcal{M}_2 if k is odd. Because the orbits have even size, \mathcal{M}_1 and \mathcal{M}_2 thus defined are disjoint. Moreover, $\mathcal{M}_2 = \mathcal{M}_1 + a$.

Now suppose $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$ where $\mathcal{M}_2 = \mathcal{M}_1 + a$. If $x \in \mathcal{M}_1$, then $x + ka$ is in \mathcal{M}_1 if k is even and in \mathcal{M}_2 if k is odd. Notice $x \equiv x + (\text{lcm}(a, m)/a)a \pmod{m}$. If $\text{lcm}(a, m)/a$ were odd then x would be in both \mathcal{M}_1 and \mathcal{M}_2 , a contradiction with those sets being disjoint. So $\text{lcm}(a, m)/a$ is even. \square

Lemma F.9. *For any quadruple (N, p, q, r) , the following holds:*

- (a) $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if $\gcd(p, N)/\gcd(q, \gcd(p, N)) \equiv 0 \pmod{2}$.
- (b) $X^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if $\gcd(q, N)/\gcd(p, \gcd(q, N)) \equiv 0 \pmod{2}$.

Proof. We only prove (a), as (b) follows from (a) by cyclicity, i.e. cyclically shifting every element of \mathcal{S} by r units to the left, and then performing Clifford transformations on every qubit swapping X and Z . For the proof, we introduce some notation. Let $m = \gcd(p, N)$, $\mathcal{M} = \{x \in \mathbb{Z} : 0 \leq x \leq m - 1\}$, “+” denote addition modulo N , and “+ _{m} ” denote addition modulo m . For any $\mathcal{A} \subseteq \mathbb{Z}$ and $a \in \mathbb{Z}$, denote $\mathcal{A} +_m a = \{x +_m a : x \in \mathcal{A}\}$ and $\mathcal{A} + a = \{x + a : x \in \mathcal{A}\}$. To prove (a), it suffices to show $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if there exists a partition $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$ with $\mathcal{M}_2 = \mathcal{M}_1 +_m q$, because by Lemma F.8 the latter is true if and only if $m/\gcd(q, m) \equiv 0 \pmod{2}$. In the proof, we consider two Pauli strings equal if and only if they are the same up to phase factors. Let $\mathcal{G} = \{k_1p + k_2q : k_1, k_2 \in \mathbb{Z}\}$, $\mathcal{G}_p = \{kp + 0 : k \in \mathbb{Z}\}$, and recall that \mathcal{G}_p has m distinct cosets $\{\mathcal{G}_p + x\}_{x=0}^{m-1}$. Notice when $q \equiv 0 \pmod{m}$, we have $q \in \mathcal{G}_p$ (since $m \in \mathcal{G}_p$ by Step 2 of the proof of Lemma F.5) implying $\mathcal{G} = \mathcal{G}_p$.

Let $x \in \mathcal{M}$. Observe that $x = x +_m q$ if and only if $q \equiv 0 \pmod{m}$, so the cosets $\mathcal{G}_p + x$ and $\mathcal{G}_p + (x +_m q)$ are different if and only if $q \not\equiv 0 \pmod{m}$. We claim $\prod \Gamma(\mathcal{G}_p + x)$ is a Z -type Pauli string with Z only at qubit indices $(\mathcal{G}_p + x) \sqcup (\mathcal{G}_p + (x +_m q))$ if $q \not\equiv 0 \pmod{m}$, and $\prod \Gamma(\mathcal{G}_p + x) = I$ if $q \equiv 0 \pmod{m}$. The case $q \equiv 0 \pmod{m}$ follows by Lemma F.6(c) as $\mathcal{G} = \mathcal{G}_p$. So now assume $q \not\equiv 0 \pmod{m}$. Lemma F.6(b) implies $\prod \Gamma(\mathcal{G}_p + x)$ is a Z -type Pauli string. For each $y \in \mathcal{G}_p + x$, the qubit indices where the Pauli string S_y contributes Z are y and $y + q$, where $y + q \in \mathcal{G}_p + (x +_m q)$. Thus we conclude using the disjointness of $\mathcal{G}_p + x$ and $\mathcal{G}_p + (x +_m q)$, that $\prod \Gamma(\mathcal{G}_p + x)$ has Z at qubit indices $\{y : y \in \mathcal{G}_p + x\} \sqcup \{y + q : y \in \mathcal{G}_p + x\} = (\mathcal{G}_p + x) \sqcup (\mathcal{G}_p + (x +_m q))$.

Assume that there exists a partition $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$ such that $\mathcal{M}_2 = \mathcal{M}_1 +_m q$. Lemma F.8 implies that $m/\gcd(q, m) \equiv 0 \pmod{2}$, and so $q \not\equiv 0 \pmod{m}$. Let $\mathcal{H} = \bigsqcup_{x \in \mathcal{M}_1} (\mathcal{G}_p + x)$. By the claim, if $x \in \mathcal{M}_1$, then $\prod \Gamma(\mathcal{G}_p + x)$ has Z at qubit indices $(\mathcal{G}_p + x) \sqcup (\mathcal{G}_p + (x +_m q))$. Moreover for distinct $x, x' \in \mathcal{M}_1$, the conditions on the partitions $\mathcal{M}_1, \mathcal{M}_2$ imply the cosets $\mathcal{G}_p + x, \mathcal{G}_p + x', \mathcal{G}_p + (x +_m q)$ and $\mathcal{G}_p + (x' +_m q)$ are distinct. This means that there is no cancellation in the Pauli Z s of $\prod \Gamma(\mathcal{G}_p + x)$ and $\prod \Gamma(\mathcal{G}_p + x')$. Since $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$, we conclude $\prod \Gamma(\mathcal{H})$ has Z at qubit indices $\bigsqcup_{x \in \mathcal{M}} (\mathcal{G}_p + x) = \mathcal{N}$.

Now assume $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$. By Lemma F.6(b), $Z^{\otimes N} = \prod \Gamma(\mathcal{H})$, where $\mathcal{H} = \bigsqcup_{x \in \mathcal{M}_1} (\mathcal{G}_p + x)$ with $\emptyset \neq \mathcal{M}_1 \subseteq \mathcal{M}$. Then $q \not\equiv 0 \pmod{m}$ by claim above, because otherwise $\prod \Gamma(\mathcal{G}_p + x) = I$ for each $x \in \mathcal{M}_1$, contradicting $\prod \Gamma(\mathcal{H}) = Z^{\otimes N}$. Define $\mathcal{M}_2 := \mathcal{M}_1 +_m q \subseteq \mathcal{M}$. The claim also implies that the qubit indices where $\prod \Gamma(\mathcal{H})$ contains Z is a subset of $\bigsqcup_{x \in \mathcal{M}_1 \cup \mathcal{M}_2} (\mathcal{G}_p + x)$. Thus we already have $\mathcal{M} = \mathcal{M}_1 \cup \mathcal{M}_2$ using $\prod \Gamma(\mathcal{H}) = Z^{\otimes N}$. We want to show \mathcal{M}_1 and \mathcal{M}_2 are disjoint. For contradiction, suppose there exists $x \in \mathcal{M}_1 \cap \mathcal{M}_2$. Then there exists $x' \in \mathcal{M}_1$ satisfying $x' +_m q = x$. The Pauli string $\prod \Gamma(\mathcal{G}_p + x')$ has Z at qubit indices $(\mathcal{G}_p + x') \sqcup (\mathcal{G}_p + (x' +_m q))$, while $\prod \Gamma(\mathcal{G}_p + x)$ has Z at qubit indices $(\mathcal{G}_p + x) \sqcup (\mathcal{G}_p + (x +_m q))$. Thus there is cancellation of Pauli Z s at the qubit indices in the coset $\mathcal{G}_p + x$ in the product $\prod \Gamma(\mathcal{H})$. This is a contradiction to $Z^{\otimes N} = \prod \Gamma(\mathcal{H})$, so we have proved $\mathcal{M} = \mathcal{M}_1 \sqcup \mathcal{M}_2$. \square

F.2 A two parameter cyclic code family

Choose two non-negative integers $s, t \geq 0$. Setting $r = 1$, $p = s + 1$, $q = s + 3$, and $N = s + t + 4$ in the 4-parameter (N, p, q, r) cyclic code family of the last subsection gives rise to a 2-parameter (s, t) cyclic code family whose stabilizers are cyclic shifts of $ZXI^{\otimes s}XZI^{\otimes t}$, up to phase factors. Notice that in this case $q = p + 2r$, and $r < p + r < q$, and so by Lemma F.2 the quadruple (N, p, q, r) is consistent. We have computed the $\llbracket N, K, D \rrbracket$ values for this 2-parameter family for all $0 \leq s, t \leq 9$, which we provide in Table 4.

$s \setminus t$	0	1	2	3	4	5	6	7	8	9
0	4,1,2	5,1,3	6,1,2	7,1,3	8,1,3	9,1,3	10,1,3	11,1,3	12,1,3	13,1,3
1	—	6,2,2	7,1,3	8,2,2	9,1,3	10,2,3	11,1,3	12,2,3	13,1,3	14,2,3
2	—	—	8,1,2	9,1,3	10,1,2	11,1,3	12,1,3	13,1,3	14,1,4	15,1,3
3	—	—	—	10,2,2	11,1,3	12,2,2	13,1,5	14,2,3	15,1,3	16,2,3
4	—	—	—	—	12,1,2	13,1,3	14,1,2	15,1,3	16,1,4	17,1,3
5	—	—	—	—	—	14,2,2	15,1,3	16,2,2	17,1,5	18,2,3
6	—	—	—	—	—	—	16,1,2	17,1,3	18,1,2	19,1,5
7	—	—	—	—	—	—	—	18,2,2	19,1,3	20,2,2
8	—	—	—	—	—	—	—	—	20,1,2	21,1,3
9	—	—	—	—	—	—	—	—	—	22,2,2

Table 4: The code parameters $\llbracket N, K, D \rrbracket$ of cyclic quantum codes with stabilizer group generated by $ZXI^{\otimes s}XZI^{\otimes t}$ and its cyclic shifts. Interchanging s and t does not change the code parameters (see Lemma F.10), so we show just the cases $t \geq s$.

One should note that Table 4 is symmetric about the diagonal – so interchanging s and t does not change the code parameters $\llbracket N, K, D \rrbracket$. This is proved in the next lemma.

Lemma F.10. *For any 2-parameter (s, t) cyclic code, the $\llbracket N, K, D \rrbracket$ code parameters are invariant upon interchange of s and t .*

Proof. The cyclic code generated by cyclic shifts of $ZXI^{\otimes s}XZI^{\otimes t}$ is the same as that generated by cyclic shifts of $XZI^{\otimes t}ZXI^{\otimes s}$. Now we perform a Clifford transformation on every qubit that interchanges X and Z (note that Clifford transformations do not change commutativity of Pauli strings). This results in a new cyclic code generated by cyclic shifts of $ZXI^{\otimes t}XZI^{\otimes s}$. Clifford transformations are group isomorphisms of the Pauli group, thus N and K are unchanged. Moreover single qubit Clifford transformations don't change the distance of the code, and hence

the same is true for their compositions. Thus D is also unchanged under interchange of s and t . \square

The number of encoded qubits for the (s, t) cyclic code family is explained by the following theorem.

Theorem F.11. *For any 2-parameter (s, t) cyclic code, the number of encoded qubits satisfies*

$$K = \begin{cases} 2 & \text{if } s \text{ and } t \text{ are odd,} \\ 1 & \text{otherwise.} \end{cases} \quad (90)$$

If $K = 1$, the only subset of \mathcal{S} that multiplies to I up to phase factors is \mathcal{S} . If $K = 2$, there are two disjoint subsets $\mathcal{S}_{\text{even}}$ and $\mathcal{S} \setminus \mathcal{S}_{\text{even}}$ of \mathcal{S} that multiplies to I up to phase factors, where $\mathcal{S}_{\text{even}} = \{S_{2n} : 0 \leq n \leq N/2 - 1\}$.

Proof. In this proof we will use the fact that for any integers $a, b, c \geq 1$, if $\gcd(a, b) = 1$ then $\gcd(ab, c) = \gcd(a, c) \gcd(b, c)$, which easily follows by considering the prime factorizations of a, b, c . Also the variables x and y appearing in this proof are always non-negative integers. We start by noting that when $s+1$ is odd, $\gcd(s+1, s+3) = 1$ which implies that $\text{lcm}(s+1, s+3) = (s+1)(s+3)$, while if $s+1$ is even, then $\gcd(s+1, s+3) = 2$ and so $\text{lcm}(s+1, s+3) = (s+1)(s+3)/2$. Applying Theorem F.3 then gives

$$K = \begin{cases} \gcd(s+1, N) \gcd(s+3, N) / \gcd((s+1)(s+3), N) & \text{if } s+1 \text{ is odd,} \\ \gcd(s+1, N) \gcd(s+3, N) / \gcd\left(\frac{(s+1)(s+3)}{2}, N\right) & \text{otherwise.} \end{cases} \quad (91)$$

In the first case ($s+1$ odd), by the fact mentioned above, we have $\gcd((s+1)(s+3), N) = \gcd(s+1, N) \gcd(s+3, N)$, and so $K = 1$. Now assume that $s+1 = 2x$, i.e. is even, and so $(s+1)(s+3)/2 = 2x(x+1)$. Then there are two subcases which we consider separately below.

- (i) t is odd: In this case $N = s + t + 4$ is even, and so suppose that $N = 2y$. Then we have $\gcd(s+1, N) = \gcd(2x, 2y) = 2 \gcd(x, y)$, $\gcd(s+3, N) = \gcd(2x+2, 2y) = 2 \gcd(x+1, y)$, and $\gcd((s+1)(s+3)/2, N) = \gcd(2x(x+1), 2y) = 2 \gcd(x(x+1), y)$. But $\gcd(x, x+1) = 1$ and so $\gcd(x(x+1), y) = \gcd(x, y) \gcd((x+1), y)$. Plugging these into Eq. (91) gives $K = 2$.
- (ii) t is even: In this case N is odd, and so $\gcd(2x, N) = \gcd(x, N)$, $\gcd(2(x+1), N) = \gcd(x+1, N)$, and $\gcd(2x(x+1), N) = \gcd(x(x+1), N) = \gcd(x, N) \gcd(x+1, N)$. Eq. (91) then gives $K = 1$.

We have thus proved Eq. (90). The last part of the theorem now follows directly from Corollary F.7. \square

Theorem F.11 suggests that the case when s and t are both odd is special. We note another curious property that is also true for this case in the next lemma, which will find an use in the proof of Theorem F.13.

Lemma F.12. *For any 2-parameter (s, t) cyclic code with both s and t odd, let $\mathcal{S}_{\text{even}} = \{S_{2n} : 0 \leq n \leq N/2 - 1\}$ and $\mathcal{S}_{\text{odd}} = \mathcal{S} \setminus \mathcal{S}_{\text{even}}$. Fix any qubit index $0 \leq n \leq N - 1$. Then*

- (a) *If n is even, $\mathcal{S}_{\text{even}}$ contains both stabilizers that contain Z at index n , and \mathcal{S}_{odd} contains both stabilizers that contain X at index n .*
- (b) *If n is odd, $\mathcal{S}_{\text{even}}$ contains both stabilizers that contain X at index n , and \mathcal{S}_{odd} contains both stabilizers that contain Z at index n .*
- (c) *For all subsets $\mathcal{S}' \subset \mathcal{S}_{\text{even}}$ and $\mathcal{S}'' \subset \mathcal{S}_{\text{odd}}$, the products $\prod \mathcal{S}'$ and $\prod \mathcal{S}''$ do not contain Y at any qubit index.*

Proof. All qubit indexes will be assumed to be modulo N in this proof. Since s, t are odd, first note that $N = s + t + 4$ is even, $s+3$ is even, and $s+2$ is odd. Now for any qubit index $0 \leq n \leq N - 1$, the two stabilizers containing Z at index n are S_n and S_{n-s-3} , and the two stabilizers containing

X at index n are S_{n-1} and S_{n-s-2} (note that these four stabilizers are distinct). If n is even, we have $(n-s-3) \bmod N$ is even, $(n-1) \bmod N$ is odd, and $(n-s-2) \bmod N$ is odd; while if n is odd, we have $(n-s-3) \bmod N$ is odd, $(n-1) \bmod N$ is even, and $(n-s-2) \bmod N$ is even. We have thus proved (a) and (b). For (c), notice that in order to obtain Y at any fixed qubit index n , one needs to at least multiply a stabilizer containing X at index n , and another stabilizer containing Z at index n . But by (a) and (b) this is impossible for subsets of $\mathcal{S}_{\text{even}}$ or \mathcal{S}_{odd} . \square

The next theorem states the conditions that determine the membership of the Pauli strings $X^{\otimes N}$, $Y^{\otimes N}$ and $Z^{\otimes N}$ in the stabilizer group $\langle \mathcal{S} \rangle$ for the (s, t) cyclic code family. As with Lemma F.9, we will only determine membership upto phase factors.

Theorem F.13. *For any 2-parameter (s, t) cyclic code, we have the following:*

- (a) $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if $s \equiv 3 \pmod{4}$ and $t \equiv 1 \pmod{4}$.
- (b) $X^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if $s \equiv 1 \pmod{4}$ and $t \equiv 3 \pmod{4}$.
- (c) At most one of $X^{\otimes N}$, $Y^{\otimes N}$, or $Z^{\otimes N}$ can belong to $\langle \mathcal{S} \rangle$.
- (d) $Y^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if $s \equiv 0 \pmod{2}$ and $t \equiv 0 \pmod{2}$.

Proof. (a) Define $\alpha = (s+3) \bmod \gcd(s+1, N)$. Note that $\gcd(s+3, \gcd(s+1, N)) = \gcd(\alpha, \gcd(s+1, N))$, and its useful to also observe that $\gcd(s+1, N) / \gcd(s+3, \gcd(s+1, N)) \equiv 0 \pmod{2}$ implies $\alpha \neq 0$. By Lemma F.9(a), then $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ if and only if $\gcd(s+1, N) / \gcd(\alpha, \gcd(s+1, N)) \equiv 0 \pmod{2}$. We proceed to check the various cases. The variables x and y appearing in this proof are always non-negative integers.

- (i) At least one of s or t is even: First assume that both s, t are even. Then N is even, while $s+1$ is odd, and so $\gcd(s+1, N)$ is odd. Now assume exactly one of s, t is even. Then N is odd, so again $\gcd(s+1, N)$ is odd. Now if $\gcd(s+1, N) = 1$ we have $\alpha = 0$, while if $\gcd(s+1, N) \geq 3$ then $\alpha = 2$. In the latter case this implies that $\gcd(\alpha, \gcd(s+1, N)) = \gcd(2, \gcd(s+1, N)) = 1$, and so $\gcd(s+1, N) / \gcd(\alpha, \gcd(s+1, N)) = \gcd(s+1, N) \equiv 1 \pmod{2}$. Thus $Z^{\otimes N}$ does not belong to $\langle \mathcal{S} \rangle$ in these cases.
- (ii) $s, t \equiv 1 \pmod{4}$; $s, t \equiv 3 \pmod{4}$; $s \equiv 1 \pmod{4}$ and $t \equiv 3 \pmod{4}$: We handle these three cases together. So assume that $s = 4x + 1, t = 4y + 1$ in the first case, $s = 4x + 3, t = 4y + 3$ in the second, and $s = 4x + 1, t = 4y + 3$ in the third. The values of N are $4x + 4y + 6$, $4x + 4y + 10$, and $4x + 4y + 8$ for the three cases respectively. It then follows that

$$\gcd(s+1, N) = \begin{cases} 2 \gcd(2x+1, 2y+2) & \text{if } s = 4x+1, t = 4y+1, \\ 2 \gcd(2x+2, 2y+3) & \text{if } s = 4x+3, t = 4y+3, \\ 2 \gcd(2x+1, 2y+3) & \text{if } s = 4x+1, t = 4y+3. \end{cases} \quad (92)$$

In each of the above cases $\gcd(s+1, N)$ is twice an odd number, so either $\gcd(s+1, N) = 2$ or $\gcd(s+1, N) \geq 6$. If $\gcd(s+1, N) = 2$ we have $\alpha = (s+3) \bmod 2 = 0$, since $s+3$ is even in all three cases; so $Z^{\otimes N}$ does not belong to $\langle \mathcal{S} \rangle$. If $\gcd(s+1, N) \geq 6$, then first we have $\alpha = (s+3) \bmod \gcd(s+1, N) = 2$, and so $\gcd(\alpha, \gcd(s+1, N)) = \gcd(2, \gcd(s+1, N)) = 2$ in all three cases. It follows that $\gcd(s+1, N) / \gcd(\alpha, \gcd(s+1, N)) = \gcd(s+1, N) / 2 \equiv 1 \pmod{2}$ in each case, and $Z^{\otimes N}$ does not belong to $\langle \mathcal{S} \rangle$.

- (iii) $s \equiv 3 \pmod{4}$ and $t \equiv 1 \pmod{4}$: Finally suppose that $s = 4x + 3, t = 4y + 1$; so $N = 4(x+y+2)$ and $\gcd(s+1, N) = \gcd(4x+4, 4x+4y+8) = \gcd(4x+4, 4y+4) = 4 \gcd(x+1, y+1)$ is a multiple of 4. This first implies that $\alpha = 2$, and so $\gcd(\alpha, \gcd(s+1, N)) = \gcd(2, 4 \gcd(x+1, y+1)) = 2$. This in turn implies that $\gcd(s+1, N) / \gcd(\alpha, \gcd(s+1, N)) = 4 \gcd(x+1, y+1) / 2 = 2 \gcd(x+1, y+1) \equiv 0 \pmod{2}$. Thus $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$.

- (b) Let \mathcal{S}' be the set of stabilizers obtained by performing the Clifford transformation that interchanges X and Z at every qubit, for each stabilizer in \mathcal{S} . Thus it suffices to show that $Z^{\otimes N}$

belongs to $\langle \mathcal{S}' \rangle$ if and only if $s \equiv 1 \pmod{4}$ and $t \equiv 3 \pmod{4}$. Now the stabilizers in \mathcal{S}' are generated by cyclic shifts of the Pauli string $ZXI^{\otimes t}XZI^{\otimes s}$. The result then follows from (a).

- (c) For the sake of contradiction assume that any two of $X^{\otimes N}$, $Y^{\otimes N}$, or $Z^{\otimes N}$ belong to $\langle \mathcal{S} \rangle$. Then their product also belongs to $\langle \mathcal{S} \rangle$, and so in fact all three $X^{\otimes N}$, $Y^{\otimes N}$, and $Z^{\otimes N}$ belong to $\langle \mathcal{S} \rangle$. But by (a) and (b) $X^{\otimes N}$ and $Z^{\otimes N}$ cannot belong to $\langle \mathcal{S} \rangle$ simultaneously, which is a contradiction.
- (d) We analyze the different cases separately.

- (i) s, t are even: In this case $N = s + t + 4$ is even. Then for the subset $\mathcal{S}_{\text{even}} = \{S_{2n} : 0 \leq n \leq N/2 - 1\}$ it is easily calculated that $\prod \mathcal{S}_{\text{even}} = \prod (\mathcal{S} \setminus \mathcal{S}_{\text{even}}) = Y^{\otimes N}$ up to phase factors.
- (ii) Exactly one of s, t is even: In this case $N = s + t + 4$ is odd, and so the Pauli strings $X^{\otimes N}$, $Y^{\otimes N}$, and $Z^{\otimes N}$ are pairwise anticommuting. Noticing that each stabilizer in \mathcal{S} also commutes with each of $X^{\otimes N}$, $Y^{\otimes N}$, and $Z^{\otimes N}$, we then conclude that none of the Pauli strings $X^{\otimes N}$, $Y^{\otimes N}$, or $Z^{\otimes N}$ belong to $\langle \mathcal{S} \rangle$.
- (iii) s, t are odd: If $s \equiv 3 \pmod{4}$ and $t \equiv 1 \pmod{4}$, then $Z^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ by (a); while if $s \equiv 1 \pmod{4}$ and $t \equiv 3 \pmod{4}$, then $X^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$ by (b). In both cases $Y^{\otimes N}$ does not belong to $\langle \mathcal{S} \rangle$ by (c). We are now left with two cases: either $s, t \equiv 1 \pmod{4}$, or $s, t \equiv 3 \pmod{4}$, both satisfying $N = s + t + 4 \equiv 2 \pmod{4}$. We handle these cases together and show below that $Y^{\otimes N}$ does not belong to $\langle \mathcal{S} \rangle$.

The proof is by contradiction, so let us suppose that $Y^{\otimes N}$ belongs to $\langle \mathcal{S} \rangle$, and $\prod \mathcal{S}' = Y^{\otimes N}$ up to phase factors, for some subset $\mathcal{S}' \subseteq \mathcal{S}$. Since both s and t are odd, consider the subsets $\mathcal{S}_{\text{even}}$ and \mathcal{S}_{odd} defined in Lemma F.12, and let $\mathcal{S}'_{\text{even}} = \mathcal{S}' \cap \mathcal{S}_{\text{even}}$, and $\mathcal{S}'_{\text{odd}} = \mathcal{S}' \cap \mathcal{S}_{\text{odd}}$. It must then be true that the products $\prod \mathcal{S}'_{\text{even}}$ and $\prod \mathcal{S}'_{\text{odd}}$ have the property that for any fixed qubit index n , exactly one of them contains X and the other one contains Z at index n (this follows from Lemma F.12(c) because otherwise one cannot get Y at index n). Thus both $\prod \mathcal{S}'_{\text{even}}$ and $\prod \mathcal{S}'_{\text{odd}}$ don't contain I_2 at any qubit index. Moreover any two distinct Pauli strings $S_i, S_j \in \mathcal{S}'_{\text{even}}$ must have disjoint supports, because otherwise by Lemma F.12(a),(b) the product $\prod \mathcal{S}'_{\text{even}}$ will contain I_2 at every qubit index in the intersection of the supports of S_i and S_j . Since each stabilizer is supported at exactly four qubit indices this implies that $N \equiv 0 \pmod{4}$, which is a contradiction.

□

Finally, we can find the distance of the two-parameter cyclic toric codes when s is close to t , the first three diagonals of Table 4.

Theorem F.14. *Consider a two-parameter (s, t) cyclic toric code. If $t = s$ or $t = s + 2$, its code distance is two. If $t = s + 1$, then its code distance is three.*

Proof. Letting D indicate the code distance, in all cases $D > 1$ because no Pauli with weight one commutes with all stabilizers.

In the $t = s$ case, we note that both $ZI^{\otimes s+1}XI^{\otimes s+1}$ and $Y^{\otimes s+2}I^{\otimes s+2}$ commute with all stabilizers. They also anticommute with each other. Thus, they are nontrivial logical operators and $D = 2$.

In the $t = s + 1$ case, we note $YI^{\otimes s+1}XXI^{\otimes s+1}$ and $X^{\otimes N}$ are nontrivial logical operators and therefore $D \leq 3$. We must also show there is no weight-two Pauli commuting with all stabilizers. A weight-two Pauli has the form $T = PI^{\otimes a}QI^{\otimes N-a-2}$ for single-qubit Paulis $P, Q \in \{X, Y, Z\}$. If $a < s + 1$ or $N - a - 2 = 2s - a + 3 < s + 1$, then T clearly must anticommute with some stabilizer. This leaves two cases $a = s + 1$ and $a = s + 2$. In the former case, commuting with all stabilizers requires that $Q = X$ and also that $P \otimes Q$ commutes with $Z \otimes Z$ and with $Z \otimes X$, which is clearly impossible. In the latter case, the analogous argument shows $P = X$ and $P \otimes Q$ commutes with $Z \otimes Z$ and with $X \otimes Z$, again a contradiction.

In the $t = s + 2$ case, we note that $XI^{\otimes s+2}XI^{\otimes s+2}$ and $Y^{\otimes s+2}ZXI^{\otimes s+2}$ are a pair of anticommuting logical operators, implying $D = 2$. □

G Embedding of the medial graph

The goal here is to prove some properties of medial graphs claimed in Section 3.4. See Definition 3.4 for the definition of a medial rotation system and, by extension, a medial graph.

Lemma G.1. *The rotation system $R = (H, \lambda, \rho, \tau)$ and its medial rotation system $\tilde{R} = (\tilde{H}, \tilde{\lambda}, \tilde{\rho}, \tilde{\tau})$ embed in the same manifold.*

Proof. We show that both rotation systems have the same orientability and the same genus.

For orientability, we show R is orientable if and only if \tilde{R} is. Start by assuming R is orientable. Therefore, we can partition H into two sets $H_{\pm 1}$ such that λ, ρ, τ applied to any element of H_k takes it to an element of H_{-k} . Define two sets $\tilde{H}_{\pm 1} \subseteq \tilde{H} = H \times \{-1, 1\}$ such that $(h, j) \in \tilde{H}_k$ if and only if $h \in H_{jk}$. These two sets clearly partition \tilde{H} . By definition, if $(h, j) \in \tilde{H}_k$ then $\tilde{\tau}(h, j) = (h, -j) \in \tilde{H}_{-k}$. Also, by using the orientability of R , $\tilde{\lambda}(h, j) = (\rho(h), j) \in \tilde{H}_{-k}$ and $\tilde{\rho}(h, j)$ is either $(\lambda(h), j)$ or $(\tau(h), j)$ both of which are in \tilde{H}_{-k} . We conclude that \tilde{R} is orientable. The other direction – if \tilde{R} is orientable, then so is R – follows the same argument in reverse.

The genus question boils down to counting the numbers of vertices, edges, and faces. Let V, E, F and $\tilde{V}, \tilde{E}, \tilde{F}$ be the sets of vertices, edges, faces for R and \tilde{R} . Then, we note

$$|\tilde{V}| = |E|, \quad (93)$$

$$|\tilde{E}| = \sum_{v \in \tilde{V}} \deg(v) = 2|E|, \quad (94)$$

$$|\tilde{F}| = |V| + |F|. \quad (95)$$

Therefore, $\tilde{\chi} = |\tilde{V}| - |\tilde{E}| + |\tilde{F}| = \chi$ and thus the genera are the same. \square

Lemma G.2. *Let $R' = (H', \lambda', \rho', \tau')$ be a rotation system. Then, $R' = \tilde{R}$ is the medial rotation system of some rotation system $R = (H, \lambda, \rho, \tau)$ if and only if R' is checkerboardable and 4-valent.*

Proof. We begin by showing a medial rotation system is 4-valent and checkerboardable. The degree of a vertex is the size of an orbit of $\tilde{\rho}\tilde{\tau}$. Clearly, any orbit must have even size L because $\tilde{\rho}\tilde{\tau}$ applied to $(h, j) \in \tilde{H} = H \times \{-1, 1\}$ flips the sign of j . Applying $(\tilde{\rho}\tilde{\tau})^L$ to (h, j) gives $((\tau\lambda)^{L/2}h, j)$ if $j = 1$ and $((\lambda\tau)^{L/2}h, j)$ if $j = -1$. By definition, $(\lambda\tau)$ has order two, so $L = 4$ is the size of any orbit of $\tilde{\rho}\tilde{\tau}$ and the degree of any vertex in the medial graph.

Checkerboardability demands that there is a partition of \tilde{H} into two sets $\tilde{H}_{\pm 1}$ such that τ applied to any element $(h, j) \in \tilde{H}_k$ gives an element of \tilde{H}_{-k} and that applying ρ or λ to (h, j) gives an element of \tilde{H}_k . This is clearly achieved by putting (h, j) for any h into the set \tilde{H}_j .

Now we must show the other direction, that a 4-valent, checkerboardable rotation system R' is the medial graph of another rotation system. Because R' is checkerboardable there is a partition $H' = H'_{-1} \sqcup H'_1$ where τ' moves elements from H'_k to H'_{-k} while λ' and ρ' preserve the sets. We let $H = \{(h, \tau'(h)) : h \in H'_1\}$ and define permutations of H called ρ, λ, τ so that

$$\lambda(h, \tau'(h)) = (\tau' \rho' \tau'(h), \rho' \tau'(h)), \quad (96)$$

$$\rho(h, \tau'(h)) = (\lambda'(h), \tau' \lambda'(h)), \quad (97)$$

$$\tau(h, \tau'(h)) = (\rho'(h), \tau' \rho'(h)). \quad (98)$$

Each of these are clearly involutions. To be a legitimate rotation system, λ and τ must also commute. They do so because R' is 4-valent, i.e. $(\rho' \tau')^4 = 1$.

We claim the medial rotation system of $R = (H, \lambda, \rho, \tau)$ is R' . We must therefore establish a one-to-one map $\phi : \tilde{H} = H \times \{-1, 1\} \rightarrow H'$. We do this by defining, for all $h \in H'$ and $j \in \{-1, 1\}$,

$$\phi((h, \tau'(h)), j) = \begin{cases} h, & j = 1 \\ \tau'(h), & j = -1 \end{cases}, \quad (99)$$

$$\phi^{-1}(h) = \begin{cases} ((h, \tau'(h)), 1), & h \in H'_1 \\ ((\tau'(h), h), -1), & h \in H'_{-1} \end{cases}. \quad (100)$$

Then one can easily show $\phi(\tilde{\lambda}(\phi^{-1}(h))) = \lambda'(h)$ for all h and likewise for λ replaced with ρ or τ . This completes the proof. \square

H Using GAP to find normal subgroups

To elucidate the process of finding hyperbolic codes, we give an example of finding low-index normal subgroups in GAP. In particular, we use the LINS package [51], version 0.5. For the sake of illustration, we focus on the index 160 normal subgroup S that defines the $\llbracket 20, 5, 4 \rrbracket$ code in Fig. 18(a) and the first entry of Table 3.

```

LoadPackage("LINS");
F := FreeGroup("l", "r", "t"); //l = λ, r = ρ, t = τ
AssignGeneratorVariables(F); //makes l, r, t into variable names
G := F/[l2, r2, t2, (l * r)5, (l * t)2, (r * t)4]; //quotient group, face-degree 5, vertex-degree 4
H := LowIndexNormalSubgroupsSearchForAll(G, 200); //LINS graph of normal subgroups, index ≤ 200
L := List(H); //convert LINS graph into a list of LINS nodes
S := Grp(L[9]); //S happens to be the 9th entry in the list
IsNormal(G, S); //check it is a normal subgroup; returns true
GeneratorsOfGroup(S); //display (overcomplete) relations of the group

```

Reducing overcomplete generating sets to independent ones was done by hand and verified by drawing the codes as those in Fig. 18 are drawn.