Topological Approaches Around Eastin–Knill: A Survey

Kepha Sher
UC Berkeley

The Eastin-Knill theorem [1] forbids any local error correcting code (EEC) to exhibit a universal set of transversal logical operators. The Bravyi-Konig theorem [2] restricts the available gates for topological error correction based on dimensionality. These theorems necessitates the implementation of various non-unitary methods based on a general desiderata aimed at minimizing experimental complexity [3]. We compare the mainstream approaches of gauge fixing, state injection, and lattice surgery. We summarize the experimental advancements of these techniques, as all of them has been recently implemented. We then shift our focus to Bombin's 3D gauge color code [4], and argue why it is one of the most natural approaches to achieve a universal transversal gate set without state injection techniques. Finally, we give a new presentation of the gauge color codes that clears up some confusion in the original and subsequent [5, 6] approaches.

1 Introduction

In the theory of fault tolerant quantum computation, quantum information is often spread out in some non-local way, assuming that errors have local structure (i.e. affects $O(1)$ physical qubits). Implementations of logical gates should then preserve locality as to mitigate the damage caused by the noise. Thus, local logical gates are highly desirable in the design of quantum codes, with transversal gates acting on each physical qubit being the most natural.

Topological codes provides the notion of spatial locality to physical qubits. We say that an operator acts geometrically locally on a lattice of qubits if there exists a hypercube of constant size that bounds the support of said operator. It is clear that this is a stronger condition than locality. All stabilizers are geometrically local stabilizers in the topological codes we consider.

There exists many classification/no-go theorems that guide the study of topological codes. In addition to Eastin-Knill, a frequently cited theorem is Bravyi-Konig, which states that

Theorem 1.1 *If a unitary operator U implementable by a constant-depth quantum circuit that preserves the codespace C of a topological stabilizer code on a D-dimensional lattice, then the restriction of U onto C implements an encoded gate from the set in the D-th level Clifford hierarchy $\mathcal{P}_D$.*

The Clifford hierarchy is defined recursively by $\mathcal{P}_1 = G_n$, the n-qubit Pauli group, and $U \in \mathcal{P}_m$ iff $UG_nU^\dagger \subseteq \mathcal{P}_{m-1}$ for $m \geq 2$. We will show that color codes saturate this theorem by implementing generalized phase gates

$$R_m = \begin{pmatrix} 1 & 0 \\ 0 & \exp(2\pi i/2^m) \end{pmatrix} \in \mathcal{P}_m \backslash \mathcal{P}_{m-1}. \tag{1.1}$$

The result was extended to subsystem codes by Pastaswki and Yoshida [7]. Other no-go theorems concern code distance and energy barrier. More specifically, Pastawski and Yoshida found that

Theorem 1.2 (Code distance) *If a stabilizer code with geometrically local stabilizers in D-dimensions admits a locality preserving implementation of a logical gate $U \in \mathcal{P}_m \backslash \mathcal{P}_{m-1}$ for $m \geq 2$, then its code distance is upper bounded by $d \leq L^{D+1-m}$.*

Theorem 1.3 (Self correction) *If a D-dimensional stabilizer Hamiltonian, consisting of geometrically local terms with bounded norms, has a macroscopic energy barrier, the set of logical gates, admitting a locality-preserving implementation, is restricted to $\mathcal{P}_{D-1}$. For subsystem codes, the restriction is lifted to $\mathcal{P}_D$.*

These theorems motivate the direction of our exposition. The immediate consequence of the Bravyi–König theorem is that if we want a transversal T gate ($T \in \mathcal{P}_3$) through topological codes, we need to go into 3 dimensions. An important point to note is that Theorem 1.1 is only prohibitive – it does not state which gates can actually be implemented transversally. It turns out that color codes are the family of codes that can implement arbitrary R_m gates as we move up in dimension.

Similarly, the code distance theorem draws an important link between dimensionality and code parameters.

The self correction theorem motivate us to look at subsystem/gauge codes. More specifically, energy barrier is measured as errors that trigger syndrome detection (i.e. excitations). If the energy barrier is constant compared to lattice size, the code would not be a good candidate for quantum memory, as thermal noise will always affect the system regardless of size. Gauge color codes saturate the code distance bound given by Theorem 1.2. Logical operators have support $O(L)$, where L is the lattice size. Combined with the previous theorems, we see that 3D gauge color codes are very natural candidates to consider since it implements a universal gate set, has macroscopic energy barrier, and have scalable code distance.

The rest of the paper is organized as follows: we first present background information of surface and color codes, using a novel framework of relative homology (mathematical background will be relegated to Appendix A). Then, we examine the three main methods: state injection, lattice surgery, and gauge fixing. Finally, we present the construction of the 3D gauge color codes, which we hope will clear up inconsistencies in the literature, and illustrate Theorems 1.1–1.3.

2 Surface Codes

This section relies on knowledge of algebraic topology, for example [8], and generally follows from [9]. See Appendix A for more information.

2.1 Toric Code

First, consider a cellulation $\mathcal{L}$ of the torus into an $L \times L$ lattice, with boundary identified (this is the toric code, later we show how to get the surface code). Qubits are placed on edges, or elements of $C_1 = C_1(\mathcal{L}, \mathbb{F}_2)$. Consider the chain complex

$$C_2 \xrightarrow{\partial_2} C_1 \xrightarrow{\partial_1} C_0 \qquad C^0 \xrightarrow{\partial_1^*} C^1 \xrightarrow{\partial_2^*} C^2 \tag{2.1}$$

We can think of coboundary maps as boundary maps on the dual lattice, but as star operators when viewed on the direct lattice through Poincare duality (see the rotated surface-code lattice in Figure 8(b)). We consider qubit excitations as strings. Let

$$c = \sum_i c_i e_i, \qquad |c\rangle = \bigotimes_i |c_i\rangle, \qquad c_i \in \mathbb{F}_2 \tag{2.2}$$

which lives in the one-chain group (i.e. edges), and where $|c_i\rangle_X$ denotes the X-eigenbasis state with $X_i |c_i\rangle_X = (-1)^{c_i} |c_i\rangle_X$. We suppress the subscript X below; in this basis $Z_a |c\rangle = |c + a\rangle$ for $a \in C_1$. Define

$$X_c = \bigotimes_i X_i^{c_i} \qquad Z_c = \bigotimes_i Z_i^{c_i} \tag{2.3}$$

Note

$$X_c X_{c'} = X_{c+c'}, \qquad Z_c Z_{c'} = Z_{c+c'}. \tag{2.4}$$

where addition is defined through the (Abelian) group C_1. This gives a homomorphism from C_1 to G_n, the n qubit Paulis.

Define the stabilizers as

$$X_v = X_{\partial^* v^*} = \prod_{e: v \in \partial_1 e} X_e, \qquad Z_f = Z_{\partial f} = \prod_{e \in \partial_2 f} Z_e, \tag{2.5}$$

where $v^* \in C^0$ denote the corresponding element in the dual lattice. We also note that the Toric code has CSS structure with

$$H = \begin{pmatrix} H(C_x) & 0 \\ 0 & H(C_z) \end{pmatrix} \qquad H(C_x)H(C_z)^T = 0 \tag{2.6}$$

with $C_x^\perp \subseteq C_z$. The ground state can be seen directly from this structure, but we can also project directly by taking some $|c\rangle \in C_1$

$$\prod_f \frac{1+Z_f}{2} \prod_v \frac{1+X_v}{2} |c\rangle \qquad c \in C_1 \tag{2.7}$$

It is not hard to see that $X_v |c\rangle = (-1)^{[\partial_1 c]\cdot v} |c\rangle$. Therefore The first projector has the affect of sending $c \in C_1$ to $z \in Z_1$. We now use the group projection formula $\prod_g \frac{1+g}{2} = 2^{-|G|} \sum_{g \in G} g$ and the definition of X_f to get

$$\prod_f \frac{1+Z_f}{2} \propto \sum_{c_2 \in C_2} Z_{\partial_2 c_2} = \sum_{b \in B_1} Z_b \tag{2.8}$$

So that an element $z \in Z_1$ gets sent to a superposition of all representatives in the quotient $Z_1/B_1 := H_1$:

$$|\bar{z}\rangle := \sum_{b \in B_1} X_b |z\rangle \qquad z \in Z_1 \tag{2.9}$$

which agrees with CSS theory. Since

$$\prod_f X_f = \prod_v Z_v = 1 \tag{2.10}$$

we have that the number of encoded qubits is

$$k = |E| - |V| - |F| + 2 = |E| - (|V| + |F| - 2) = 2 - \chi = 2g \tag{2.11}$$

for a genus g torus where χ is the Euler characteristic (see Appendix A).

2.2 String operators

Operators of the Toric code are strings. That is to say, since qubits are placed on edges, we define a general operator to have the form

$$A = i^\alpha Z_c X_{c^*} \qquad \alpha \in \{0,1,2,3\}, c \in C_1(\mathcal{L}), c^* \in C_1(\mathcal{L}^*), \tag{2.12}$$

which means the Z operators act on the direct lattice, while the X operators act on the dual lattice. We identify the edges with their dual, since they physically represent the same qubit, making this assumption hold without loss of generality. At this point it may be easier to disregard the lattice, and think about operators as loops on a manifold.

From coding theory we know logical operators are in $N(S)/S$, where $N(S) = N_{G_n}(S)$ is the normalizer of the stabilizer group. It is easy to check that

$$\begin{aligned} [Z_c, X_v] = 0 \ \text{ for all } v &\iff c \in Z_1(\mathcal{L}), \\ [X_{c^*}, Z_f] = 0 \ \text{ for all } f &\iff c^* \in Z_1(\mathcal{L}^*). \end{aligned} \tag{2.13}$$

(once again, this is because endpoints are the problem. Theorem 1.2 from the introduction asserts this must be true). This implies $A \in N(S) \iff (c, c^*) \in Z_1(\mathcal{L}) \times Z_1(\mathcal{L}^*)$. Additionally, if $A \in S$ we would have $A \in \prod_{f_i} Z_{\partial f_i} \prod_{v_i} X_{\partial^* v_i^*} = Z_{\partial_2 c_2} X_{\partial_1^* c_0^*}$ where $c_2 = \sum_i f_i, c_0^* = \sum_i v_i^*$. This holds because of (2.4). Hence we see that

$$\begin{aligned} A \propto i^\alpha Z_{z+b} X_{z^*+b^*} \qquad & b \in B_1(\mathcal{L}),\ b^* \in B_1(\mathcal{L}^*), \\ & z \in Z_1(\mathcal{L}),\ z^* \in Z_1(\mathcal{L}^*). \end{aligned} \tag{2.14}$$

And so

$$N(S)/S \cong H_1(\mathcal{L}; \mathbb{F}_2) \oplus H_1(\mathcal{L}^*; \mathbb{F}_2), \tag{2.15}$$

We can identify $H_1(\mathcal{L}^*)$ as the cohomology group $H^1(\mathcal{L})$, which is isomorphic to $H_1(\mathcal{L})$ since we are working over $\mathbb{F}_2$. The code distance is this the support size of the minimal element in H_1, which is clearly $O(L)$.

2.3 Boundary

We can think about boundary through relative homology. Instead of cellulating a torus, consider a general cellulation of a compact surface with boundary $\mathcal{L}$. In the example of the surface code, we take a 2-sphere S^2, to which we remove the interior of two disjoint disks. The boundary is a union of two subcomplexes

$$\partial\mathcal{L} = \Sigma_R \cup \Sigma_S \tag{2.16}$$

for rough and smooth boundaries, respectively. They can meet only on vertices. Rough boundaries are those where Z strings may end, which means (physically) that the X checks at the endpoints are absent. Smooth boundaries are those where X strings may end, so the endpoint Z checks are absent.

For any subcomplex $A \subset X$, the relative chain groups are

$$C_i(X, A) := C_i(X)/C_i(A), \tag{2.17}$$

with induced boundary map $\partial_i^{\mathrm{rel}}[c] = [\partial_i c]$. One then defines relative cycles and boundaries as

$$\begin{aligned} Z_i(X, A) &= \{[c] \in C_i(X, A) : \partial_i c \in C_{i-1}(A)\}, \\ B_i(X, A) &= \{[\partial_{i+1} c] : c \in C_{i+1}(X)\}, \end{aligned} \tag{2.18}$$

A Z string supported on $c \in C_1$ can end on a rough boundary without causing an excitation when $\partial c \in C_0(\Sigma_R)$, i.e. when $[c]$ is a relative cycle in $C_1(X, \Sigma_R)$. Two Z strings are now equivalent if their relative homology class differ by a relative boundary, so Z logicals are now

$$\begin{aligned} H_1(\mathcal{L}, \Sigma_R; \mathbb{F}_2), \qquad C_2(\mathcal{L}, \Sigma_R) \xrightarrow{\partial_2^{\mathrm{rel}}} \\ C_1(\mathcal{L}, \Sigma_R) \xrightarrow{\partial_1^{\mathrm{rel}}} C_0(\mathcal{L}, \Sigma_R). \end{aligned} \tag{2.19}$$

Analogously, X logicals are represented by the dual relative homology group

$$H_1(\mathcal{L}^*, \Sigma_S^*; \mathbb{F}_2). \tag{2.20}$$

which is the relative cohomology group $H^1(\mathcal{L}, \Sigma_S)$ by Poincare duality. The rough and smooth boundaries should therefore not be combined into a single quotient complex for both operator types: Σ_R is used for Z strings, and Σ_S^* is used for X strings. Note that this whole setup can also be applied to surface code defects [10], as well as braiding, but we will not go this route.

2.4 Higher dimensions

One can also define higher dimensional general Toric code by cellulating any compact oriented manifold and choosing a chain complex at any dimension

$$C_{i+1} \xrightarrow{\partial_{i+1}} C_i \xrightarrow{\partial_i} C_{i-1} \tag{2.21}$$

where qubits live on the i-cells. Similarly the Z logicals to live in H_i, and X logicals to live in $H^i \cong H_{D-i}$ by Poincare duality. Viewed in the direct lattice, this would look like all i-simplicies that intersects transversely with a $(D-i)$-surface.

3 Color Codes

Color codes are a family of topological codes that support convenient transversal implementation originally proposed by Héctor Bombín. For example, 2D color codes support the full Clifford group [11], and higher dimensional color codes support Clifford hierarchy of respective dimensions. We first describe the most general formulation of color codes [4, 12, 5].

3.1 General Color Codes

There are two fundamental objects in color code theory. One generally considers a triangulation of a D-dimensional compact oriented manifold, with the conditions:

- The triangulation generates a homogeneous simplicial D-complex $\mathcal{L}$
- The one-skeleton is a $D+1$-valent graph that is $D+1$ colorable, meaning there is a function

$$\mathrm{color} : C_0 \to \mathbb{Z}_{D+1} \qquad \mathrm{color}(\delta) := \bigsqcup_{v \in C_0(\delta)} \mathrm{color}(v) \tag{3.1}$$

  such that adjacent vertices have different colors.

One then takes the dual of this complex to form a *D-colex*. One important fact is

Theorem 3.1 (Color bijection) *Given a k-cell $\delta \in C_k(\mathcal{L})$ with* $\mathrm{color}(\delta) = \kappa$*, under duality we have*

$$\delta^* \in C_k(\mathcal{L}^*) \cong C_{n-k}(\mathcal{L}), \qquad \mathrm{color}(\delta^*) = \mathbb{Z}_{D+1} \backslash \kappa \tag{3.2}$$

Furthermore, there is a bijection between the subcells of δ and the subsets of $\mathrm{color}(\delta)$*. Similarly, there is a bijection between the supercells of δ^* and the supersets of* $\mathrm{color}(\delta^*)$*.*

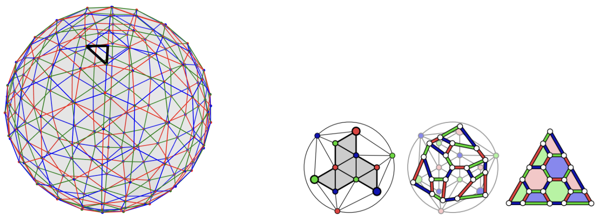
Now we use parameters (d, e) to indicate the dimensions of the stabilizers:

$$C_D(d, e) := \langle X(\delta), Z(\sigma) \mid \delta \in C_{d-1}(\mathcal{L}), \sigma \in C_{e-1}(\mathcal{L}) \rangle \tag{3.3}$$

where $X(\delta) = \bigotimes_{q \in \mathcal{Q}(\delta)} X_q$, where $\mathcal{Q}(\delta)$ is the set of qubits incident to δ. Similarly for $Z(\sigma)$. One should think of these as star operators, and after taking the dual X becomes plaquette stabilizers on $D-(d-1)$ cells and Z becomes plaquettes on $D-(e-1)$ cells. (refer to Theorem 3.1, and note the dual need not be a triangulation).

3.2 Conventional Color Codes

A subset of color codes are known as conventional color codes [13] characterized by $D - d = e$. These codes are known to be locally unitarily isomorphic to tensor product of copies of toric codes (up to ancillas)



(a) Primal Lattice. This shows S^2 with the interior of a 2-simplex/triangle removed (outlined in black). Vertices are placed on triangles, stabilizers are on the vertices. We then flatten this out. Note we obviously used a larger mesh here to illustrate the action better.

(b) Dual Colex. The flattened out triangulation. We then take the dual, which gives the 2-colex on which qubits are placed at vertices, and stabilizers are on the cells.

Figure 1. Primal and dual pictures for the color-code construction.

Shape	Dual V	Dual E	Dual F	Face sizes	Count
Cube	8	12	6	$(4,4,4,4,4,4)$	4

Table 1. Cell types and shapes for the dual colex in Bombin's model for $t = 0$.

Theorem 3.2 *Given color code $C_D(d, D-d)$, there exists a local Clifford unitary U such that*

$$U[C(d,D-d)\otimes\mathcal{S}_1]U^\dagger = \bigotimes_{N\in\mathcal{P}_d(\mathbb{Z}_D)} TC_{D-d}(\mathcal{L}_N)\otimes\mathcal{S}_2 \quad (3.4)$$

where TC_k denotes a k dimensional toric code, $\mathcal{S}_1, \mathcal{S}_2$ are ancillary qubits, $\mathcal{P}_d$ denotes the size d subsets of $\mathbb{Z}_D$, and $\mathcal{L}_N$ is a $D-d+1$ complex obtained from $\mathcal{L}$ known as the shrunk lattice when we "collapse" a specfic color. See the appendix of [12] for the precise definition.

Thus we can understand the logical structure of conventional color codes from toric codes. For the rest of this paper we focus on a specific setup that allows for transversal logical gates. The procedure is as follows:

- Triangulate a d-sphere ($d = 3$ for gauge color codes) such that the conditions in Section 3.1 are satisfied.
- Remove the interior of one of the top-simplicies, leaving us with a (closed) boundary $\partial\mathcal{L}$. Stabilizers associated with the boundary are not activated (this is analogous to the boundary treatment described in surface codes).
- Build the color code as described in Section 3.1, but use $\mathcal{L}_0 := \mathcal{L}\backslash\partial\mathcal{L}$ to define chain groups of all levels (i.e. look at $C_D(d,e)$ defined on $\mathcal{L}_0$. Note the top-chains remain the same.
- Dualize to get $\mathcal{L}_0^*$, and analyze its shapes to understand stabilizer weights, experimental feasibility, etc. Tables 1-4 does some analysis on the model of Bombin [4].

Note that the specific triangulation only affects last step, and does not affect the properties of the color codes themselves. Figure 2 summarizes the construction. This reconciles the seemingly inconsistent approaches of Bombin [4], Brown [6], and Kubica [5].

Shape	Dual V	Dual E	Dual F	Face sizes	Count
Cube	8	12	6	$(4,4,4,4,4,4)$	8
Hexagonal prism	12	18	8	$(4,4,4,4,4,4,6,6)$	4
Six-square-five-hex cell	18	27	11	$(4,4,4,4,4,4,6,6,6,6,6)$	4

Table 2. Cell types and shapes for the dual colex in Bombin's model for $t = 1$.

Shape	(V, E, F)	Face sizes	Count
Cube	$(8, 12, 6)$	$(4,4,4,4,4,4)$	12
Hexagonal prism	$(12, 18, 8)$	$(4,4,4,4,4,4,6,6)$	12
Six-square-five-hex cell	$(18, 27, 11)$	$(4,4,4,4,4,4,6,6,6,6,6)$	12
Truncated octahedron	$(24, 36, 14)$	$(4,4,4,4,4,4,6,6,6,6,6,6,6,6)$	4

Table 3. Cell types and shapes for the dual colex in Bombin's model for $t = 2$.

4 Magic State Injection and Distillation

We define a general magic state as

$$|A_\theta\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\theta}|1\rangle) \quad (4.1)$$

and preparation of logical magic states can implement phase gates $\Lambda = \begin{pmatrix}1 & 0\\ 0 & e^{i\theta}\end{pmatrix}$ as follows:

- Let $|\psi\rangle = a|0\rangle + b|1\rangle$. Prepare $|\Psi_0\rangle = |\psi\rangle\otimes|A_\theta\rangle$
- Measure $\mathcal{S}_1 = \sigma_z\otimes\sigma_z$. With probability $1/2$ we get the syndrome $+1$, similarly -1:

$$\begin{aligned}\left|\Psi_1^+\right\rangle &= (a|00\rangle + be^{i\theta}|11\rangle),\\ \left|\Psi_1^-\right\rangle &= (ae^{i\theta}|00\rangle + b|11\rangle).\end{aligned} \quad (4.2)$$

- Apply $CNOT_{1\to 2}$ to get

$$\begin{aligned}\left|\Psi_2^+\right\rangle &= (a|0\rangle + be^{i\theta}|1\rangle),\\ \left|\Psi_2^-\right\rangle &= (ae^{i\theta}|0\rangle + b|1\rangle).\end{aligned} \quad (4.3)$$

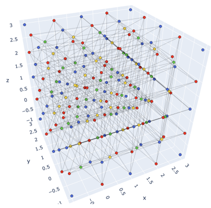
This has the affect of applying $\Lambda(e^{i\theta})$ or $\Lambda(e^{-i\theta})$. Notable is the implementation of the T-gate, which has circuit (see Figure 3).

4.1 Injection

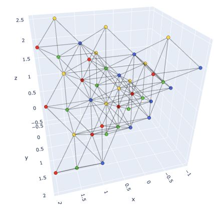
We can inject magic states to surface [16] and color [17, 18] codes. The idea is analogous: we first prepare the magic state on each physical qubit. Then entangle other physical qubits such that a transversally implemented operator on the set of entangled qubits have the same effect as a logical operator, and finally project into the codespace by rounds of error correction depending on the code distance. [16] gives the example of a 3×3 suface code, where the central qubit is entangled with the ones above and below it

$$\alpha|0\rangle + \beta|1\rangle \to \alpha|000\rangle + \beta|111\rangle \propto \alpha|0\rangle_L + \beta|1\rangle_L \quad (4.4)$$

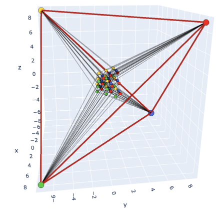
the reason it is proportional to the logical qubits is because it transform in the same way under logical operations. It is important to note that the logical magic state is not protected until it is projected into the codespace (stabilizers turned on), so the process is inherently noisy. Hence the need for distillation, which "purifies" the magic states. Figure 4 summarizes the surface code injection picture.



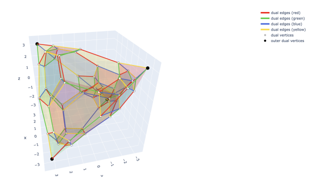
(a) Original BCC lattice. Vertices are 4 colorable.



(b) We remove vertices as given in [4]. Bombin labels this set as M_t.



(c) The set Δ_3 as described in the original paper. Here the 4 boundary vertices are reintroduced. The (interior) tetrahedral removed is the one in bold red.



(d) The rendered dual colex for $t = 1$ for readability. Shapes are collected in Tables 1–3 for $t = 0, 1, 2$

Figure 2. Bombin's construction of the 3D gauge color codes [4]. Parts (a)-(c) are rendered for $t = 2$, (d) for $t = 1$. The original paper did not graph the dual or classify its shapes, as we do in Tables 1–4. The construction follows from section III.A. The interactive model is placed in the references [14].

t	Rel. V	Rel. E	Rel. F	Rel. T	Dual V	Dual E	Dual F	Dual 3-cells
0	4	18	28	15	15	28	18	4
1	16	80	128	65	65	128	80	16
2	40	214	348	175	175	348	214	40

Table 4. The first four column stand for relative primal. These numbers are computed from counting the dual shapes directly. The number of qubits as a function of t is $f(t) = 1 + 4t + 6t^2 + 4t^3$.

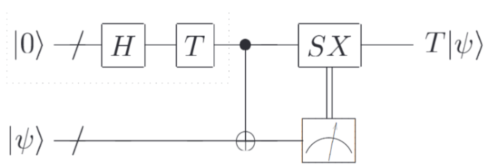
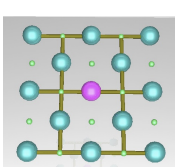
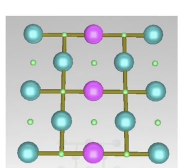


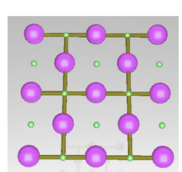
Figure 3. A circuit to produce the T gate [15]



(a)



(b)



(c)

Figure 4. State injection for surface code. The pink qubit in (a) is prepared in the (physical) magic state. CNOT gates entangle it with the qubits above and below it (b). It is then projected back into the codespace (c).

4.2 Distillation

Bravyi and Kitaev [19] proposed a scheme to distill magic states. More specifically, given noisy magic states of error rate ε, there is a threshold ε_0 such that when $\varepsilon < \varepsilon_0$ the output error , after k rounds of distillation

$$\varepsilon_{\text{out}}^{(H)} \approx \frac{1}{5}(5\varepsilon)^{2^k}, \qquad n_H \approx 5^k, \tag{4.5}$$

$$\varepsilon_{\text{out}}^{(T)} \approx \frac{1}{\sqrt{35}}(\sqrt{35}\varepsilon)^{3^k}, \qquad n_T \approx 15^k, \tag{4.6}$$

for the preparation of H- and T-type magic states, respectively (related to the T-gate as $|T_{gate}\rangle = e^{i\pi/8} H S^\dagger |H\rangle$).

$$\begin{gathered} |H\rangle = \cos(\pi/8)\,|0\rangle + \sin(\pi/8)\,|1\rangle\,, \\ |T\rangle = \cos\beta\,|0\rangle + e^{i\pi/4}\sin\beta\,|1\rangle \qquad \cos(2\beta) = \frac{1}{\sqrt{3}}. \end{gathered} \tag{4.7}$$

They use the perfect code and quantum Reed-Muller codes [20, 21] for distillation. The framework is as follows:

- We want to go from the noisy ρ to the clean ρ_{out}, and we define metrics $\varepsilon = 1 - \langle T|\,\rho\,|T\rangle$ and $\varepsilon_{out} = 1 - \langle T|\,\rho_{out}\,|T\rangle$.
- Write the magic state $|T\rangle$ as an eigenstate of T, say $|T_0\rangle := |T\rangle\,, |T_1\rangle$ is the other eigenstate.
- Pass through a depolarizing channel $D(\eta)$ that diagonalizes ρ into $(1-\varepsilon)\,|T_0\rangle\langle T_0| + \varepsilon\,|T_1\rangle\langle T_1|$. The input density matrix becomes $\rho_{in} = \rho^{\otimes c} \to \Pi\rho\Pi^\dagger = \rho_s$, after projection into the codespace where c is the block length of the code used.
- ρ_s involves a sum of $\sum f(\varepsilon)\,|T_0^L\rangle\langle T_0^L| + g(\varepsilon)\,|T_1^L\rangle\langle T_1^L|$. A trivial syndrome is achieved with probability $p_s = \operatorname{tr}\rho_s$.
- The normalized output state is $\rho_s/p_s := (1 - \varepsilon_{out})\,|T_0\rangle\langle T_0| + \varepsilon_{out}\,|T_1\rangle\langle T_1|$. We finally decode so that we retrieve the nonlogical version of the magic state.

- Compare ε_{out} with ε. If ε_0 is below some threshold then we can repeat this process and get doubly exponential refinement, each time with probability p_s.

If the circuit has L non-Clifford gates we expect $\varepsilon_{out} \sim 1/(L \log L)$. In general, if we have a $[[n_2, k_2, d_2]]$ code, we expect error to scale as $\varepsilon_{out} \approx O(\varepsilon^{d_2^r})$ after r rounds of distillation, in other words $r \sim \log_{d_2} \log(1/\varepsilon_{out})$. Due to the contatenating structure of this procedure, if the code encodes k_2 qubits we expect $n \approx \left(\frac{n_2}{k_2}\right)^r$ noisy states to be expended, i.e.

$$
\begin{aligned}
n &\approx \left(\frac{n_2}{k_2}\right)^{\log_{d_2} \log(1/\epsilon_{\text{out}})} \\
&= \left(\log\left(1/\epsilon_{\text{out}}\right)\right)^{\log_{d_2}(n_2/k_2)} \\
&:= O\left(\log\left(1/\epsilon_{\text{out}}\right)^{\gamma}\right) \\
&\implies n \sim (\log L)^{\gamma}, \qquad \gamma = \log_{d_2}\left(\frac{n_2}{k_2}\right)
\end{aligned} \tag{4.8}
$$

Magic states can be thought of as a separate procedure from the usual transversal gates we have talked about earlier. It is non-unitary since it consumes many ancillary qubits not in the original Hilbert space, therefore bypassing the Eastin-Knill theorem. It is traditionally considered an expensive procedure to implement universal quantum computation. For example, Gidney and Ekerå estimates that at a physical gate error rate of 10^{-3}, factoring an RSA-2048 integer requires about 20 million noisy physical qubits. Their distillation analysis requires about 3 billion distilled CCZ states with total distillation error of 6.4%. [22] Figure 5 summarizes different distillation protocols from [3].

5 Lattice Surgery

Lattice [16] is a non-unitary operation on surface codes that allows many gates to be implemented in a more efficient manner. We won't spend much ink here since it does not directly deal with the Eastin-Knill constraint.

There are two main operations associated with surgery: merging and splitting, and each has a smooth/rough variant depending on the boudnary type. Merging takes two sheets (two logical qubits) and prepares a row of qubits down the *join* in $|0\rangle$ for smooth, $|+\rangle$ for rough. It then measures all the stabilizers, thus projecting the qubits into a new surface codespace. We track what happens to logical operators and logical information as the two surfaces combine to form one surface.

In a rough merge of $|\psi\rangle = \alpha |0\rangle_L + \beta |1\rangle_L$, we get

$$
|\psi\rangle M |\phi\rangle = \alpha |\phi\rangle + (-1)^M \beta |\bar{\phi}\rangle = \alpha' |\psi\rangle + (-1)^M \beta' |\bar{\psi}\rangle \tag{5.1}
$$

Rough Merging has the effect of applying $X_L \otimes X_L$ on both sheets (because the stabilizers run down the join), so M denotes the measured value, which is why the state would be in $\frac{1}{\sqrt{2}}(|\psi\rangle |\phi\rangle + (-1)^M |\bar{\psi}\rangle |\bar{\phi}\rangle)$.

For splitting, we measure out the join, as in Figure 6. For smooth split, we measure out in X basis, Z basis for rough. We will have to update the logical operator that gets "split up" across the join, either by physical error correction or by updating the Pauli frame. The logical information transform as

$$
\alpha |0\rangle_L + \beta |1\rangle_L \to \alpha |00\rangle_L + \beta |11\rangle_L \tag{5.2}
$$

These two operations are inverses of each other, for example for a smooth split and merge

$$
\begin{aligned}
\alpha |0\rangle_L + \beta |1\rangle_L &\to \alpha |00\rangle_L + \beta |11\rangle_L \\
&= \frac{1}{\sqrt{2}} |+\rangle \left(\alpha |0\rangle_L + \beta |1\rangle_L\right) \\
&\quad + \frac{1}{\sqrt{2}} |-\rangle \left(\alpha |0\rangle + \beta |1\rangle\right) \\
&\xrightarrow{M} \frac{1}{\sqrt{2}}(\alpha |0\rangle + \beta |1\rangle) + \frac{1}{\sqrt{2}}(\alpha |0\rangle + \beta |1\rangle) \\
&\propto \alpha |0\rangle + \beta |1\rangle .
\end{aligned} \tag{5.3}
$$

One of the advantages of using lattice surgery is the reduction of physical entanglement in CNOT. Figure 7 shows the corresponding CNOT layout. The trick is to produce an intermediary state $|INT\rangle := |+\rangle_L$, and perform, for control $|C\rangle$ and target $|T\rangle$, $|C\rangle\, M\, |INT\rangle \xrightarrow{\text{update frame, split}} |C'\, INT'\rangle \to |C'(INT'\, M\, T)\rangle$. See Appendix B for more information. Note that some of these operation can be done in parallel so that the total complexity is linear in lattice size $O(d)$, where as a transversal gate would take $O(d^2)$, see [16] for more details. Another related technique is *code deformation*, the idea being that since a transversal Hadamard rotates teh lattice by 90 degrees (rough boundaries gets mapped to smooth boundaries, vice versa), we use merge and split operations to manually reset the lattice configuration.

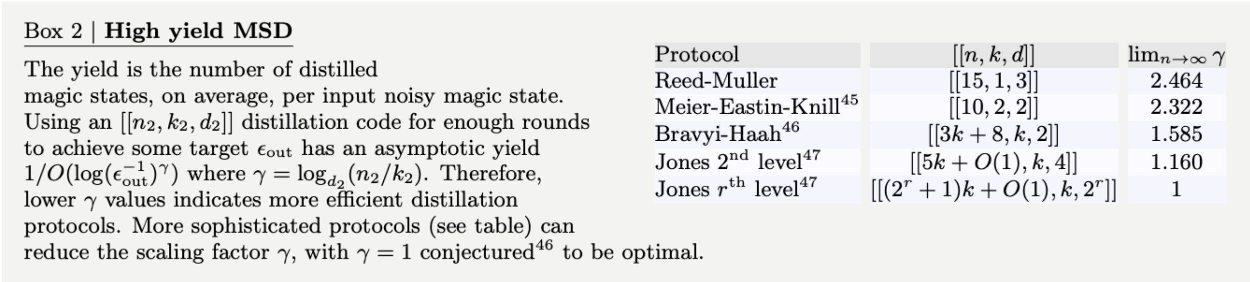
All these approaches are non-unitary, so in that sense they bypass the Eastin-Knill theorem. Lastly, a recent experiment demonstrates lattice surgery [23], shown in Figure 8; more specifically they perform a split on a rotated 3×3 grid that splits the surface code into two repetition codes. They also make bell states using the smooth split.

6 Gauge Fixing

In this last section, we discuss the general procedure of subsystem codes, then transition to a construction of Bombin's 3D gauge color code [4], and explain how this procedure generalizes many other gauge fixing variants.

6.1 Subsystem Codes

A *subsystem stabilizer code* is defined as a stabilizer group $\mathcal{S}$ and a gauge group $\mathcal{G}$ such that $\mathcal{S} = Z(\mathcal{G}) \cap \mathcal{G}$. The gauge group acts on additional gauge logical qubits that is outside the codespace. The idea is that it might be easier to measure gauge qubits and infer the measurement of $\mathcal{S}$ than measuring $\mathcal{S}$ directly. We define *bare logical operators* as elements of $Z(\mathcal{G})/\mathcal{S}$. They fix the gauge space. Define *dressed logical operators* as elements of $Z(\mathcal{S})/\mathcal{G}$. They permute the gauge space, and can be realized as some bare operator times an element in $\mathcal{G}$. Note both

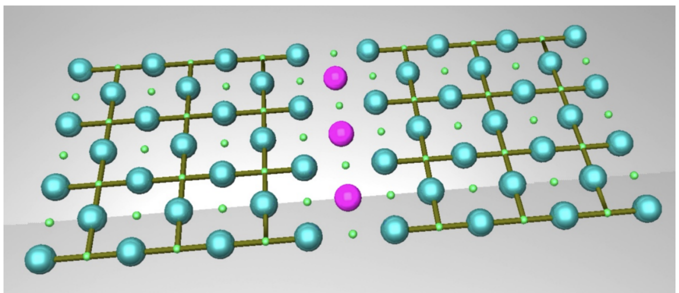


Box 2 | **High yield MSD**

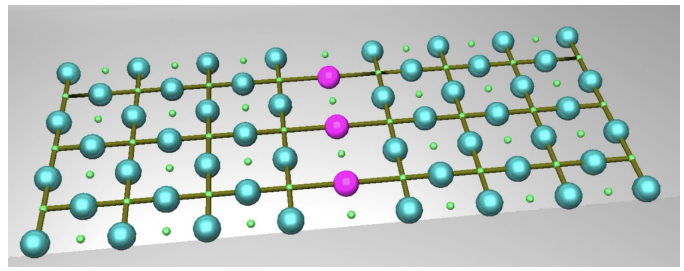
The yield is the number of distilled magic states, on average, per input noisy magic state. Using an $[[n_2, k_2, d_2]]$ distillation code for enough rounds to achieve some target ϵ_{out} has an asymptotic yield $1/O(\log(\epsilon_{\text{out}}^{-1})^\gamma)$ where $\gamma = \log_{d_2}(n_2/k_2)$. Therefore, lower γ values indicates more efficient distillation protocols. More sophisticated protocols (see table) can reduce the scaling factor γ, with $\gamma = 1$ conjectured[46] to be optimal.

Protocol	$[[n, k, d]]$	$\lim_{n\to\infty} \gamma$
Reed-Muller	$[[15, 1, 3]]$	2.464
Meier-Eastin-Knill[45]	$[[10, 2, 2]]$	2.322
Bravyi-Haah[46]	$[[3k + 8, k, 2]]$	1.585
Jones 2^{nd} level[47]	$[[5k + O(1), k, 4]]$	1.160
Jones r^{th} level[47]	$[[(2^r + 1)k + O(1), k, 2^r]]$	1

Figure 5. High-yield magic-state distillation comparison. The exponent $\gamma = \log_{d_2}(n_2/k_2)$ is the scaling exponent appearing in (4.8). Taken from Box 2 of [3].



(a) Rough merge. The pink qubits are prepared in $|0\rangle$ and stabilized into the (new) codespace with two sheets merged.



(b) Smooth split. The pink qubits are measured out in the X basis. The Z logical operator is updated to be well defined on each sheet.

Figure 6. Basic lattice-surgery operations used in merge and split procedures.

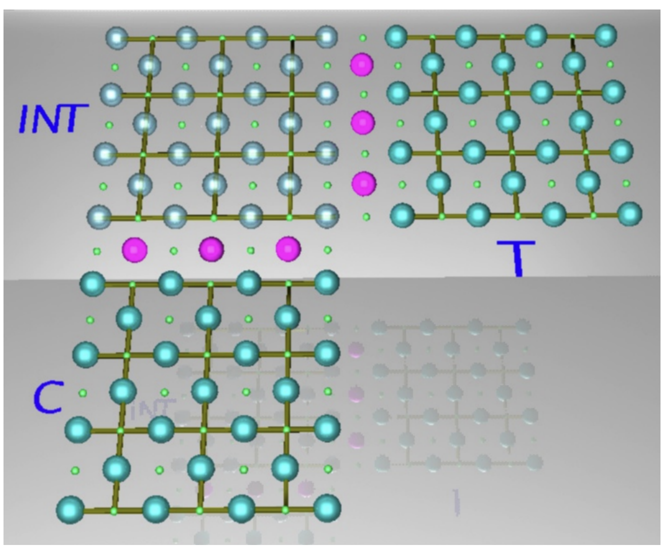


Figure 7. Pictorial demonstration of how CNOT is executed using lattice surgery. In practice we would have many of these intermediate sheets arranged between the computational logical qubits, so that only about 25% of the surface is used for computation [16]. The affect is such that the save in complexity $O(d^2) \to O(d)$ is balanced out by the additional intermediary qubits.

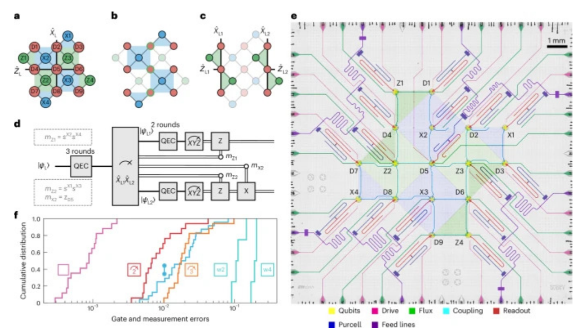


Figure 8. Experimental setup for the lattice surgery experiment. Parts (a)-(c) demonstrates the rotated 3×3 lattice. Part (d) implements the lattice surgery circuit (as discussed we keep track of the projective $X_L \otimes X_L$ and stabilizers using Pauli frame updates. (f) shows the error rates for single-qubit gates (pink), two-qubit CZ gates (cyan), readout with two-state (red) and three-state (orange) discrimination and cycle-average syndrome elements for weight-two and weight-four stabilizers (w2 and w4). [23]

leaves the codespace intact as $\mathcal{S} \subset \mathcal{G} \iff Z(\mathcal{G}) \subset Z(\mathcal{S})$. We can also prove that

$$Z(\mathcal{G})/\mathcal{S} \cong Z(\mathcal{S})/\mathcal{G} \tag{6.1}$$

using the second isomorphism theorem from algebra. In general, gauge qubits need not mutually commute, so measurement can be destructive. [24] gives the condition that

Theorem 6.1 *We can extract a single syndrome bit corresponding to $S \in \mathcal{S}$ using gauge measurements if and only if the following decomposition holds:*

$$S = K_m \dots K_2 K_1 \tag{6.2}$$

such that $[K_j, K_{j-1} \dots K_1] = 0$, where K_j's denote the gauge operators. One then perform sequential measurement of $K_1 \dots K_m$ obtaining $\gamma_1, \dots, \gamma_m \in \{\pm 1\}$, and deduce the measurement of S as $\sigma = \gamma_1 \dots \gamma_m$.

In our case it is much simpler, as our gauge group has CSS form, $\mathcal{G} = \langle Z(G_a), X(H_b) \rangle \subset \mathcal{P}_n$. Thus we can measure one operator type at a time and not worry about measurement issues.

Gauge fixing is the technique to realize subsystem codes. By measuring qubits in the gauge group we can switch between codes while preserving all logical information. In other words, some gauge qubit degrees of freedom are now "fixed" by forcing them into the stabilizer group. Formally speaking, suppose we have two subsystem codes $(\mathcal{S}_1, \mathcal{G}_1)$ and $(\mathcal{S}_2, \mathcal{G}_2)$. The *gauge fixing condition* is

$$\mathcal{S}_1 \subseteq \mathcal{S}_2 \subseteq \mathcal{G}_1, \qquad \mathcal{G}_2 \propto Z(\mathcal{S}_2) \cap \mathcal{G}_1 \tag{6.3}$$

where $\propto$ means we mod out by irrelevant phases. There is a useful equivalence that states if both codes share the same set of bare logicals (that is, $Z(\mathcal{G}_1)/S_1 \cong Z(\mathcal{G}_2)/\mathcal{S}_2$ and a set $\mathcal{L} = \langle g_1 s_1 \dots g_m s_m \rangle = \{\prod_i (g_i s_i)^{a_i} \mid a_i \in \{0,1\}, g_i \in Z(\mathcal{G}_1), s_i \in \mathcal{S}_1\}$) then

$$\mathcal{S}_1 \subseteq \mathcal{S}_2 \qquad \text{or} \qquad \mathcal{G}_2 \subseteq \mathcal{G}_1 \tag{6.4}$$

is equivalent to condition in (6.3). See Appendix C for a rigorous proof of the theorems here. Lastly, we define two (general) logical states:

$$|\bar{0}\rangle |g_X\rangle = \sum_{X(G) \in \mathcal{G}} X(G) |\mathbf{0}\rangle \qquad |\bar{1}\rangle |g_X\rangle = \bar{X} |\bar{0}\rangle |g_X\rangle \tag{6.5}$$

$$|\bar{0}\rangle |g_Z\rangle = \sum_{X(S) \in \mathcal{S}} X(S) |\mathbf{0}\rangle \qquad |\bar{1}\rangle |g_Z\rangle = \bar{X} |\bar{0}\rangle |g_X\rangle \tag{6.6}$$

where $|g_X\rangle$ means it is a $+1$ eigenstate of all X type operators in $\mathcal{G}$, and $|g_Z\rangle$ means it is $+1$ eigenstate of all Z type operators in $\mathcal{G}$. $|\mathbf{0}\rangle$ denotes the set of physical qubits. These two states will make the gauge fixing procedure clearer.

6.2 Experimental Results

Before we begin the presentation of Bombin's 3D gauge color code, we first note that recent experiments have realized color-code scaling, magic-state injection, and lattice-surgery operations; see Figure 9 [25]. That experiment demonstrates a distance-scaling factor $\Lambda_{3\to5} := \varepsilon_3/\varepsilon_5 \approx 0.0171/0.011 \approx 1.56$ and below-threshold performance. Separately, code switching from the $[[15, 1, 3]]$ quantum Reed–Muller code ($QRM(4) \cong C_3(1, 2)$, see [5]) to the $[[7, 1, 3]]$ Steane code ($QRM(3) \cong C_2(1, 1)$, see [5]) has been used to prepare high-fidelity logical magic states [26]; this is closely related to dimensional jumping in color codes [27] and earlier Steane/Reed–Muller code switching [20, 21].

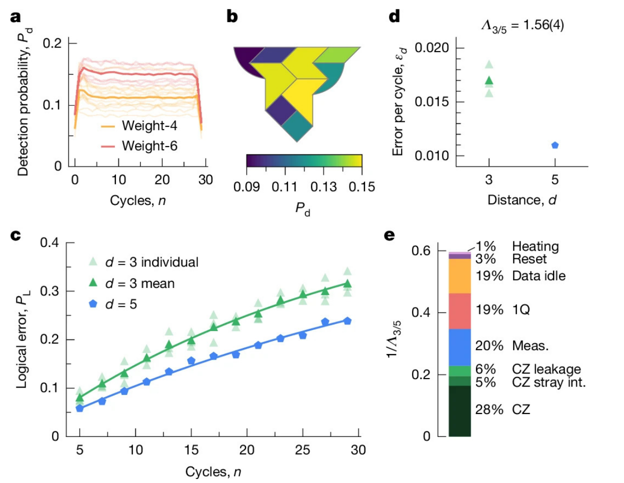


Figure 9. Experimental demonstration of 2D color codes. (b) shows the color code lattice chosen so that measurements share the same diagonal. The tones reflect error rate. Taken from [25]

6.3 Gauge Color Codes

We resume the discussion from Section 3. The reader should have the primal lattice $\leftrightarrow$ dual colex picture in Figure 1 in mind. Bombin's construction uses a specfic triangulation created using a 3D BCC lattice, removing tetrahedral to form M_t (where t indicates how fine the mesh is), add back the boundary vertices to form Δ_d, and finally take the dual. We generated interactive models to visualize this process, linked in the references.

Gauge color codes are defined as the subsystem code with $d + e \leq D$

$$\mathcal{S} = C_D(d, e), \qquad \mathcal{G} = C_D(\bar{e}, \bar{d}). \tag{6.7}$$

The theorem we need to prove that this is a valid subsystem code is

$$Z(C_D(d, e)) \propto \langle X_{\mathcal{Q}}, Z_{\mathcal{Q}} \rangle \cdot C_D(\bar{e}, \bar{d}) \tag{6.8}$$

where $\bar{x} = D - x$, and $\mathcal{Q}$ is the set of physical qubits. The proof of this is rather involved and will be left to Appendix D. Now the idea is to code switch from

$$C_3(1, 1) \leftrightarrow C_3(1, 2) \tag{6.9}$$

Next we discuss how each gate is transversal. In this setup they are all bare logical operators. There are many technical theorems needed to prove properties, which will be mentioned but not proven. See [5] for more details.

- $X, Z, CNOT$. *Even Support.* We consider the primal lattice $\mathcal{L}_0$. For any k-simplex $\delta \subset C_k(\mathcal{L}_0)$, with $0 \le k < d$, the set of supported qubits satisfy $|\mathcal{Q}(\delta)| \equiv 0 \mod 2$. This means if we define $\bar{X} = X_{\mathcal{Q}}, \bar{Z} = Z_{\mathcal{Q}}$ they are in $N(\mathcal{S})$. In addition, we set $|\mathcal{Q}| \equiv 1 \mod 2$ so that $\bar{X}$ and $\bar{Z}$ anticommute (this depends on how we choose the lattice). $CNOT$ is automatically transversal due to the CSS strcture, since $\bar{X}, \bar{Z}$ has the right transformation under it.
- H. The Hadamard gate is transversal in $C(0,0)$ because the code is self-dual (i.e. remains the same under $X(G) \leftrightarrow Z(G)$ substitutions. Therefore it preserves $\mathcal{G}$. Since this is the only gate not in $C(1,2)$ we can think of the gauge fixing procedure as follows. First, every codeword in $C(1,2)$ is a codeword in $C(1,1)$. This is due to the fact $\boxed{C(d_1, e_1) \subset C(d_2, e_2) \iff d_1 \le d_2, e_1 \le e_2}$. So if we start from a $C(1,2)$ state $|\psi\rangle |g\rangle$ and apply $H_{\mathcal{Q}}$, we get $\bar{H} |\psi\rangle |g'\rangle$ which leaves the codespace of $C(1,2)$ but still lives in $C(1,1)$. We then *fix* the gauge by re-projecting the state into $C(1,2)$:

$$|\psi\rangle |g'\rangle \xrightarrow{H_{\mathcal{Q}}} \bar{H} |\psi\rangle |g'\rangle \xrightarrow{\text{gauge fixing}} \bar{H} |\psi\rangle |g\rangle \tag{6.10}$$

- R_n. The phase gate has some theorems governing its transversality properties. The main result is $\boxed{R_n \text{ can be implemented transversally if } D \ge n\bar{e}}$. For $D = 3, n = 3, e = 2$ we see that $C_3(1,2)$ can implement the T gate transversally. The idea is to pick a set $T \subset \mathcal{Q}$ and apply the gate $R = R_n^k(T) R_n^{-k}(T^c)$ such that $k(|T| - |T^c|) \equiv 1 \mod 2^n$. We need T to satisfy

$$\forall X(G) \in \mathcal{G} : |T \cap G| \equiv |T^c \cap G| \mod 2^n \tag{6.11}$$

Then

$$\begin{aligned} R|\bar{0}\rangle |g_X\rangle &= \sum_{X(G) \in \mathcal{G}} R_n^k(T) R_n^{-k}(T^c) X(G)|\mathbf{0}\rangle \\ &= \sum_{X(G) \in \mathcal{G}} e^{\frac{2\pi i k}{2^n}|T \cap G|} e^{-\frac{2\pi i k}{2^n}|T^c \cap G|} \\ &\quad \cdot X(G)|\mathbf{0}\rangle \\ &= \sum_{X(G) \in \mathcal{G}} X(G)|\mathbf{0}\rangle = |\bar{0}\rangle |g_X\rangle, \end{aligned} \tag{6.12}$$

$$\begin{aligned} R|\bar{1}\rangle |g_X\rangle &= R_n^k(T) R_n^{-k}(T^c) \bar{X}|\bar{0}\rangle |g_X\rangle \\ &= e^{\frac{2\pi i k}{2^n}|T|} e^{-\frac{2\pi i k}{2^n}|T^c|} \bar{X} R|\bar{0}\rangle |g_X\rangle \\ &= e^{\frac{2\pi i}{2^n}} \bar{X}|\bar{0}\rangle |g_X\rangle = e^{\frac{2\pi i}{2^n}} |\bar{1}\rangle |g_X\rangle \end{aligned} \tag{6.13}$$

How to find such a set T comes from the bipartiteness of the dual 1-skeleton of the color code lattice; see Figure 10. We will not persue the general case here, but notice in two dimensions the primal lattice is always bipartite. These theorems imply that the $C_3(1,2)$ color code has both the S and T gate transversally, so the entire subsystem code implements the full universal gate set. This also implies that the 2D color code $C_2(1,1)$ implements the entire Clifford group, although $C_3(1,2)$ does not have transversal H.

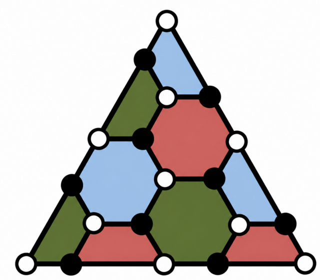


Figure 10. An example of the set T in Bombin's construction of the 2D color code. One can easily check that the graph is bipartite.

The results in this section are general, meaning that as long as we select a suitable initial triangulation of S^d, we are guaranteed to be able to implement a universal transversal gate set.

We note that one of the major advantages of this approach is we can produce magic states without injection in a fault tolerant manner (see the experiment section), which is a lot less noisy than injecting a physical T state. Indeed, [25] demonstrates state of the art performance.

Finally, the only non-unitary procedure in gauge fixing is measurement, so it bypasses the Eastin-Knill theorem.

7 Conclusion and Outlook

In this paper we examined the primarily *theoretical* aspects behind the methods of topological codes, namely magic state injection/distillation, lattice surgery, and gauge fixing. We also highlighted recent experimental procedures that verifies their efficacy. Another reason this paper is expository is because the theory of surface and color codes has been completed around a decade ago (canonical papers are published 2013-2016), while recent advancements focus on decoding and improving thresholds [28, 29, 30, 31]. Another direction has been dynamical codes. The Floquet codes are seen as a continuation of the color codes [32, 33], and have the property of significantly reduced stabilizer weights. We hope that the paper can serve as a guide to inform researchers about the background and advances on topological quantum error correction, and guide them towards more recent works.

A Algebraic Topology

This appendix provides the background in algebraic topology necessary for understanding topological codes. All vector spaces are taken over $\mathbb{F}_2$. We obviously cannot present all background, only those absolutely necessary. For more information, please see [8].

A.1 Chains, cochains, and boundary maps

Let us intuit the concept of a cell complex $\mathcal{L}$. In the two-dimensional case, its 0-cells are vertices, its 1-cells are edges, and its 2-cells are faces. Denote the $\mathbb{F}_2$-vector space generated by the i-cells of $\mathcal{L}$ is denoted

$$C_i(\mathcal{L}).$$

An element of $C_i(\mathcal{L})$ is called an i-chain. These can be thought of as higher dimensional analogs of the 2D shapes.

The boundary maps are homomorphisms/linear map between the chain vector spaces (I sometimes used vector space and groups interchangeably, although they are really $\mathbb{F}_2$-vector spaces)

$$\partial_i : C_i(\mathcal{L}) \longrightarrow C_{i-1}(\mathcal{L}), \tag{A.1}$$

where ∂_i sends each i-cell to the sum of its boundary $(i-1)$-cells. These maps satisfy the chain-complex condition

$$\partial_i \partial_{i+1} = 0. \tag{A.2}$$

The dual vector space

$$C^i(\mathcal{L}) := \mathrm{Hom}_{\mathbb{F}_2}(C_i(\mathcal{L}), \mathbb{F}_2)$$

is called the vector space of i-cochains. Linear algebra tells us that they are (non-canonically) isomorphic to the corresponding chain vector spaces.

The coboundary maps are the dual maps of the boundary maps:

$$\partial_i^* : C^{i-1}(\mathcal{L}) \longrightarrow C^i(\mathcal{L}). \tag{A.3}$$

They are defined by the relation

$$(\partial_i^* \varphi)(c) = \varphi(\partial_i c),$$

for $\varphi \in C^{i-1}(\mathcal{L})$ and $c \in C_i(\mathcal{L})$. Since $\partial_i \partial_{i+1} = 0$, the dual maps satisfy

$$\partial_{i+1}^* \partial_i^* = 0 \tag{A.4}$$

A.2 Cycles, boundaries, homology, and cohomology

An i-cycle is an i-chain with no boundary, meaning it is the kernel of the boundary map

$$Z_i(\mathcal{L}) := \ker \partial_i. \tag{A.5}$$

An i-boundary is an i-chain that is the boundary of an $(i+1)$-chain. The space of i-boundaries is

$$B_i(\mathcal{L}) := \mathrm{im}\, \partial_{i+1}. \tag{A.6}$$

(A.2) implies

$$B_i(\mathcal{L}) \subseteq Z_i(\mathcal{L}).$$

The homology groups are defined by their quotient, for each dimension/level as

$$H_i(\mathcal{L}) := Z_i(\mathcal{L})/B_i(\mathcal{L}). \tag{A.7}$$

Two cycles represent the same homology class (i.e. are homologous) iff they differ by a boundary.

Dually, an i-cocycle is an element of

$$Z^i(\mathcal{L}) := \ker \partial_{i+1}^*. \tag{A.8}$$

An i-coboundary is an element of

$$B^i(\mathcal{L}) := \mathrm{im}\, \partial_i^*. \tag{A.9}$$

The ith cohomology group is

$$H^i(\mathcal{L}) := Z^i(\mathcal{L})/B^i(\mathcal{L}). \tag{A.10}$$

For surface codes with qubits on edges, a primal Z string is represented by a 1-chain in $C_1(\mathcal{L})$, while a dual X string is represented by a 1-cochain in $C^1(\mathcal{L})$, equivalently by a 1-chain on the dual cellulation $\mathcal{L}^*$.

A.3 Euler characteristic and encoded qubits

For a two-dimensional cell complex $\mathcal{L}$, the Euler characteristic is

$$\chi(\mathcal{L}) := |V| - |E| + |F|. \tag{A.11}$$

Equivalently, in terms of Betti numbers,

$$\chi(\mathcal{L}) = \dim H_0(\mathcal{L}) - \dim H_1(\mathcal{L}) + \dim H_2(\mathcal{L}). \tag{A.12}$$

where the ith Betti number is the dimension of the ith homology group considered as a $\mathbb{F}_2$ vector space.

If $\mathcal{L}$ torus of genus g, then

$$\dim H_0(X) = 1, \qquad \dim H_1(X) = 2g, \\ \dim H_2(X) = 1.$$

Hence

$$\chi(\mathcal{L}) = 2 - 2g \implies k = \dim H_1(X) = 2g = 2 - \chi(\mathcal{L}). \tag{A.13}$$

This is why, for example, a toric code encodes $k = 2$ qubits.

A.4 Relative homology and boundaries

For surfaces codes, ordinary homology is not always the right language because some strings are allowed to end on certain boundary types. Hence we present relative homology.

Let $A \subseteq \mathcal{L}$ be a subcomplex, usually a union of boundary cells. The relative chain group is

$$C_i(\mathcal{L}, A) := C_i(\mathcal{L})/C_i(A). \tag{A.14}$$

Thus two i-chains in $\mathcal{L}$ are identified when they differ by an i-chain that belongs entirely inside A.

The boundary map descends to a quotient map:

$$\partial_i : C_i(X,A) \longrightarrow C_{i-1}(X,A). \tag{A.15}$$

This implies that an i-chain $c \in C_i(X)$ is a relative cycle when its boundary lies in A

$$c \in Z_i(X,A) \qquad \Longleftrightarrow \qquad \partial_i c \in C_{i-1}(A). \tag{A.16}$$

Thus a relative 1-cycle may have endpoints, but only on A. In surface code language this means X or Z type string operators can have excitations on their corresponding boundaries.

Relative boundaries are chains of the form

$$c = \partial_{i+1} b + a, \qquad b \in C_{i+1}(X), \quad a \in C_i(A). \tag{A.17}$$

The relative homology group is

$$H_i(\mathcal{L}, A) := Z_i(\mathcal{L}, A)/B_i(\mathcal{L}, A). \tag{A.18}$$

A.5 Poincare duality

Poincare duality gives a bijection between relative homology on the primal lattice and relative homology on the dual lattice. For the purposes of surface codes, we have

$$H_1(\mathcal{L}, \Sigma_R) \cong H^1(\mathcal{L}, \Sigma_S). \tag{A.19}$$

We can build a sort of dictionary as

$$\text{primal relative } Z \text{ strings modulo deformation} \quad \longleftrightarrow$$
$$\text{dual relative } X \text{ strings modulo deformation.}$$

More specifically:

- A direct-lattice Z string is represented by a primal 1-chain.
- It commutes with the relevant X-type checks iff it is a relative cycle.
- If the string is allowed to end on a rough boundary R, its representative class lives in $H_1(\mathcal{L}, \Sigma_R)$.
- A dual-lattice X string is represented by a dual 1-chain, or equivalently a primal 1-cochain.
- It is allowed to end on the smooth boundary Σ_S, so its class is in $H^1(\mathcal{L}, \Sigma_S)$.

B Lattice surgery

Here we elaborate on the merge procedure. Let us again consider rough merging, with the starting states being $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|\phi\rangle = \alpha'|0\rangle + \beta'|1\rangle$. As described before, we apply $X_L \otimes X_L$ to this state with syndrome M, to get

$$\begin{aligned}
&(\alpha|0\rangle + \beta|1\rangle)(\alpha'|0\rangle + \beta'|1\rangle) \\
&\quad + (-1)^M(\alpha|1\rangle + \beta|0\rangle)(\beta'|0\rangle + \alpha'|1\rangle) \\
&= (\alpha\alpha' + (-1)^M\beta\beta')|00\rangle + (\beta\beta' + (-1)^M\alpha\alpha')|11\rangle \\
&\quad + (\alpha\beta' + (-1)^M\alpha'\beta)|01\rangle + (\alpha'\beta + (-1)^M\alpha\beta')|10\rangle .
\end{aligned} \tag{B.1}$$

We want the new logical ground state to be the even parity state under $Z_L \otimes Z_L$, so we define the new logical states to be

$$\begin{aligned}
|0\rangle_L &\to \frac{1}{\sqrt{2}}(|00\rangle_L + (-1)^M|11\rangle_L) \\
|1\rangle_L &\to \frac{1}{\sqrt{2}}(|01\rangle_L + (-1)^M|10\rangle_L)
\end{aligned}$$

We recover rule (5.1) by considering the two cases $M = 0, 1$.

B.1 CNOT Implementation

Here we give a brief rundown of how CNOT is implemented, see Figure 7. Write $|C\rangle = \alpha|0\rangle + \beta|1\rangle = \bar{\alpha}|+\rangle + \bar{\beta}|-\rangle$, $|T\rangle = \alpha'|0\rangle + \beta'|1\rangle$, and $|INT\rangle = |+\rangle$. Then we have that

$$|C\rangle\, M\, |INT\rangle = \bar{\alpha}|+\rangle + (-1)^M\bar{\beta}|-\rangle \tag{B.2}$$

$$= \begin{cases} \alpha|0\rangle + \beta|1\rangle & M = 0 \\ \beta|0\rangle + \alpha|1\rangle & M = 1 \end{cases} \tag{B.3}$$

Therefore if we update the Pauli frame by applying an X_L, we get the state $|C\rangle\, M\, |INT\rangle = \alpha|0\rangle + \beta|1\rangle$. If we perform a smooth split we get $|C'\, INT'\rangle = \alpha|00\rangle + \beta|11\rangle$. Finally, if we do a rough merge between $|INT'\rangle$ and $|T\rangle$, we get

$$|C'(INT'\, M\, T)\rangle = \alpha|0\rangle(|0\rangle\, M\, |T\rangle) + \beta|1\rangle(|1\rangle\, M\, |T\rangle) \tag{B.4}$$

$$= \alpha|0\rangle\, M\, |T\rangle + (-1)^{M'}\beta|1\rangle \otimes |\bar{T}\rangle \tag{B.5}$$

We again fix the pauli frame such that $M' = 0$ so the final logical state is

$$\alpha|0\rangle|T\rangle + \beta|1\rangle|\bar{T}\rangle$$

C Subsystem Codes Details

In this part of the appendix we prove two facts left unproved in [4, 5].

Theorem C.1

$$Z(\mathcal{G})/Z(\mathcal{G}) \cap \mathcal{G} \cong Z(Z(\mathcal{G}) \cap \mathcal{G})/\mathcal{G}$$

where $\mathcal{S} = Z(\mathcal{G}) \cap G$.

Proof. We use the symplectic representation of Pauli operators, ignoring overall phases:

$$X^{\otimes a} Z^{\otimes b} \longleftrightarrow (a \mid b) \in \mathbb{F}_2^{2n},$$

where $a, b \in \mathbb{F}_2^n$ More specifically, let

$$P = X^{\otimes a} Z^{\otimes b}, \qquad P' = X^{\otimes a'} Z^{\otimes b'}.$$

Then, using $ZX = -XZ$ for each qubit, we get

$$P'P = X^{\otimes a'} Z^{\otimes b'} X^{\otimes a} Z^{\otimes b} \tag{C.1}$$

$$= (-1)^{a\cdot b'} X^{\otimes(a'+a)} Z^{\otimes(b'+b)}, \tag{C.2}$$

while

$$PP' = X^{\otimes a} Z^{\otimes b} X^{\otimes a'} Z^{\otimes b'} \tag{C.3}$$
$$= (-1)^{a'\cdot b} X^{\otimes(a'+a)} Z^{\otimes(b'+b)}. \tag{C.4}$$

Therefore P and P' commute if and only if the two signs agree

$$(-1)^{a\cdot b'} = (-1)^{a'\cdot b}.$$

Equivalently,

$$a \cdot b' + a' \cdot b = 0 \pmod 2.$$

This motivates the symplectic inner product

$$\big\langle (a \mid b), (a' \mid b') \big\rangle_s = a \cdot b' + a' \cdot b \in \mathbb{F}_2.$$

In particular, if A is X-type and B is Z-type, then the inner product is $a \cdot b \mod 2$.

This immediately implies $Z(\mathcal{G}) \leftrightarrow G^\perp$, where the r.h.s. is the orthogonal space of the vector space spanned by G. Therefore we can write

$$Z(\mathcal{G}) \cap \mathcal{G} = G^\perp \cap G \tag{C.5}$$
$$\implies Z(Z(\mathcal{G}) \cap \mathcal{G}) = (G^\perp \cap G)^\perp \tag{C.6}$$
$$= G + G^\perp = \mathcal{G}Z(\mathcal{G}) \tag{C.7}$$

where we have identified the addition of vector spaces with multiplcation in the operator space. This implies

$$Z(Z(\mathcal{G}) \cap \mathcal{G})/\mathcal{G} \cong \mathcal{G}Z(\mathcal{G})/\mathcal{G} \cong Z(\mathcal{G})/Z(\mathcal{G}) \cap \mathcal{G}$$

by the second isomorphism [34]

We now an equivalent condition of the gauge fixing conditions, which sheds light on how a codeword from a larger code can be seen as a valid codeword for a smaller one, as in the case of $C_3(1,1) \subset C_3(1,2)$. The idea is since they have a shared representative set of bare logical operators, actions on logical information are the same for each code. By representative we mean a fixed coset representative for each class in $Z(\mathcal{G}_1)/\mathcal{S}_1 \cong Z(\mathcal{G}_2)/\mathcal{S}_2$. Note that this is very natural to consider since ultimately the physical action is defined on a concrete set of operators, not as a homology class.

Theorem C.2 *Suppose that two subsystem codes share a bare logical operator set $\mathcal{L}$ such that $\mathcal{L}\mathcal{S}_1 \propto Z(\mathcal{G}_1), \mathcal{L}\mathcal{S}_2 \propto Z(\mathcal{G}_2)$. Then the gauge fixing condition is equivalent to $\mathcal{S}_1 \subseteq \mathcal{S}_2$.*

Proof. The goal is to prove

$$\mathcal{S} \propto Z(\mathcal{L}\mathcal{G}) \qquad \mathcal{G} \propto Z(\mathcal{L}\mathcal{S}) \tag{C.8}$$

from which the result follows as

$$\mathcal{S}_1 \subseteq \mathcal{S}_2 \implies \mathcal{G}_2 \propto Z(\mathcal{L}\mathcal{S}_2) \subseteq Z(\mathcal{L}\mathcal{S}_1) \propto \mathcal{G}_1$$
$$\implies \mathcal{S}_2 \propto Z(\mathcal{G}_2) \cap \mathcal{G}_2 \subseteq \mathcal{G}_2 \subseteq \mathcal{G}_1$$
$$\implies Z(\mathcal{S}_2) \cap \mathcal{G}_1 \propto Z(\mathcal{S}_2) \cap Z(\mathcal{L}\mathcal{S}_1)$$
$$= Z(\mathcal{L}\mathcal{S}_1\mathcal{S}_2) = Z(\mathcal{L}\mathcal{S}_2) \propto \mathcal{G}_2.$$

Let's first prove $\mathcal{S} \propto Z(\mathcal{L}\mathcal{G})$. $\mathcal{S} \subseteq Z(\mathcal{L}\mathcal{G})$ is clear from definitions. Take $p \in Z(\mathcal{L}\mathcal{G})$. Then $\mathcal{L} \subseteq \mathcal{L}\mathcal{G} \implies Z(\mathcal{L}\mathcal{G}) \subseteq Z(\mathcal{L})$ and thus $p \in Z(\mathcal{L}) \implies p \in Z(\mathcal{G}) \propto \mathcal{L}\mathcal{S}$. Noe write $p = \ell s$ for $\ell \in \mathcal{L}, s \in \mathcal{S}$. Since $\ell s \in Z(\mathcal{L})$ and $s \in Z(\mathcal{L})$ by definition, $\ell \in Z(\mathcal{L})$. But this means the coset $[\ell\mathcal{S}] \in Z(\mathcal{G})/\mathcal{S}$ commutes with all other bare logical operators, which must mean $\ell \in \mathcal{S}$. This means $p \in \mathcal{S}$, as desired. The other equation in (C.8) is similar to this, so we skip it.

For the converse direction, we use the symplectic representation described earlier, we have $\mathcal{G}_2 = \mathcal{G}_1 \cap \mathcal{S}_2^\perp \implies Z(\mathcal{G}_2) = S_2 + G_1^\perp = \mathcal{S}_2 Z(\mathcal{G}_1) = Z(\mathcal{G}_1)\mathcal{S}_2$. This implies bare logical operators of both codes can be chosen such that the representative set is shared. Lastly we prove, for the sake of sanity, that

Theorem C.3 *The gauge fixing conditions does indeed give a valid subsystem code, that is*

$$\mathcal{S}_2 = Z(\mathcal{G}_2) \cap \mathcal{G}_2$$

Proof. Using the symplectic representation described earlier, we have

$$Z(\mathcal{G}_2) \cap \mathcal{G}_2 = G_2^\perp \cap G_2.$$

Since $G_2 = S_2^\perp \cap G_1$, we have

$$G_2^\perp = (S_2^\perp \cap G_1)^\perp = S_2 + G_1^\perp.$$

Therefore

$$Z(\mathcal{G}_2) \cap \mathcal{G}_2 = (S_2 + G_1^\perp) \cap G_2.$$

Because $S_2 \subseteq G_2$, we have

$$(S_2 + G_1^\perp) \cap G_2 = S_2 + (G_1^\perp \cap G_2).$$

But $G_2 \subseteq G_1$ by the previous theorem, so

$$G_1^\perp \cap G_2 \subseteq G_1^\perp \cap G_1 = S_1.$$

Since $S_1 \subseteq S_2$, it follows that

$$S_2 + (G_1^\perp \cap G_2) = S_2.$$

Thus

$$Z(\mathcal{G}_2) \cap \mathcal{G}_2 = S_2,$$

as claimed.

D Color Codes Details

Here we prove

Theorem D.1

$$Z(C(d,e)) = \langle X_\mathcal{Q}, Z_\mathcal{Q} \rangle \cdot C(\bar{e}, \bar{d})$$

Proof. The idea is to first write the color code as a parity check matrix using the symplectic representation. For $1 \le k < D$, define the incidence matrix/parity check matrix to be

$$\begin{aligned} H_k &: \mathbb{F}_2^{|\mathcal{Q}|} \to \mathbb{F}_2^{C_{k-1}}, \\ (H_k a)_\delta &= \sum_{q \in \mathcal{Q}(\delta)} a_q \mod 2, \qquad \delta \in C_{k-1}(\mathcal{L}) \end{aligned}$$

In other words, we note each physical qubit to be 1 if it is incident to a particular δ. Equivalently, the rows of H_k are the characteristic vectors

$$\text{rows of } H_k = \mathbf{1}_{\mathcal{Q}(\delta)} \in \mathbb{F}_2^{|\mathcal{Q}|}.$$

We then define the rows to be

$$R_k := \text{row}\,(H_k) = \text{span}_{\mathbb{F}_2}\left\{\mathbf{1}_\delta : \delta \in \Delta_{k-1}\right\}.$$

so that

$$C(d,e) \propto R_d \oplus R_e \subseteq \mathbb{F}_2^{|\mathcal{Q}|} \oplus \mathbb{F}_2^{|\mathcal{Q}|},$$

where R_d is the X part and R_e is the Z part (i.e. we are writing the color codes in terms of their CSS structure). A Pauli operator $X_A Z_B \leftrightarrow (a,b)$ centralizes $C(d,e)$ iff

$$a \in R_e^\perp, \quad b \in R_d^\perp.$$

Therefore

$$Z(C(d,e)) \propto R_e^\perp \oplus R_d^\perp.$$

We will be done if we can show

$$R_k^\perp = R_{\bar{k}} + \langle \mathbf{1}_\mathcal{Q} \rangle, \quad 1 \le k < D,$$

We first show

$$R_{\bar{k}} + \langle \mathbf{1}_\mathcal{Q} \rangle \subseteq R_k^\perp.$$

Take $\alpha \in \Delta_{k-1}$ and $\beta \in \Delta_{\bar{k}-1}$. Then if we consider the smallest simplex τ containing the support of both α and β, i.e. $\mathcal{Q}(\tau) = \mathcal{Q}(\alpha) \cap \mathcal{Q}(\beta)$, we know from the even support lemma that $|\mathcal{Q}(\tau)| \equiv 0 \mod 2$ (see [5], the existence of such τ is guaranteed by the intersection lemma (2) in the same paper). The same lemma also implies $\mathbf{1}_Q \cdot R_k \equiv 0 \mod 2$. Hence we see that $R_{\bar{k}} + \langle \mathbf{1}_\mathcal{Q} \rangle \subseteq R_k^\perp$. Now we show they have equal dimension. For a moment let's assume we're working with the conventional color code $C(k, \bar{k})$. Then the number of encoded qubit satisfies the equation

$$1 = k = n - \dim H_k - \dim H_{\bar{k}}$$

We choose a lattice, as we did above, with $|\mathbf{1}_\mathcal{Q}| \equiv 0 \mod 2$ so that $\mathbf{1}_\mathcal{Q} \notin R_{\bar{k}}$, and so dimensionally

$$\dim(R_{\bar{k}} + \langle \mathbf{1}_\mathcal{Q} \rangle) = \dim H_{\bar{k}} + 1$$

Using $\dim(R_k^\perp) = n - \dim H_k$ we have

$$n - \dim H_k = \dim H_{\bar{k}} + 1 \implies \dim R_k^\perp = \dim(R_{\bar{k}} + \langle \mathbf{1}_\mathcal{Q} \rangle)$$

as desired.

References

Generative AI was used as assistance in generating the graphs for the interactive displays for color codes, and for formatting LaTeX.

[1] Bryan Eastin and Emanuel Knill. "Restrictions on Transversal Encoded Quantum Gate Sets". In: *Physical Review Letters* 102.11 (2009), p. 110502. DOI: 10.1103/PhysRevLett.102.110502.

[2] Sergey Bravyi and Robert König. "Classification of Topologically Protected Gates for Local Stabilizer Codes". In: *Physical Review Letters* 110.17 (2013), p. 170503. DOI: 10.1103/PhysRevLett.110.170503. arXiv: 1206.1609 [quant-ph].

[3] Earl T. Campbell, Barbara M. Terhal, and Christophe Vuillot. "Roads towards fault-tolerant universal quantum computation". In: *Nature* 549.7671 (2017), pp. 172–179. DOI: 10.1038/nature23460. arXiv: 1612.07330 [quant-ph].

[4] Héctor Bombín. "Gauge Color Codes: Optimal Transversal Gates and Gauge Fixing in Topological Stabilizer Codes". In: *New Journal of Physics* 17.8 (2015), p. 083002. DOI: 10.1088/1367-2630/17/8/083002. arXiv: 1311.0879 [quant-ph].

[5] Aleksander Kubica and Michael E. Beverland. "Universal transversal gates with color codes: A simplified approach". In: *Physical Review A* 91.3 (2015), p. 032330. DOI: 10.1103/PhysRevA.91.032330. arXiv: 1410.0069 [quant-ph].

[6] Benjamin J. Brown, Naomi H. Nickerson, and Dan E. Browne. "Fault-tolerant error correction with the gauge color code". In: *Nature Communications* 7 (2016), p. 12302. DOI: 10.1038/ncomms12302. arXiv: 1503.08217 [quant-ph].

[7] Fernando Pastawski and Beni Yoshida. "Fault-tolerant logical gates in quantum error-correcting codes". In: *Physical Review A* 91.1 (2015), p. 012305. DOI: 10.1103/PhysRevA.91.012305. arXiv: 1408.1720 [quant-ph].

[8] Allen Hatcher. *Algebraic Topology*. Cambridge University Press, 2002.

[9] Héctor Bombín. "An Introduction to Topological Quantum Codes". In: (2013). arXiv: 1311.0277 [quant-ph].

[10] Austin G. Fowler et al. "Surface codes: Towards practical large-scale quantum computation". In: *Physical Review A* 86.3 (2012), p. 032324. DOI: 10.1103/PhysRevA.86.032324. arXiv: 1208.0928 [quant-ph].

[11] Hector Bombin and Miguel A. Martin-Delgado. "Topological Quantum Distillation". In: *Physical Review Letters* 97.18 (2006), p. 180501. DOI: 10.1103/PhysRevLett.97.180501. arXiv: quant-ph/0605138.

[12] Aleksander Kubica, Beni Yoshida, and Fernando Pastawski. "Unfolding the Color Code". In: *New Journal of Physics* 17.8 (2015), p. 083026. DOI: 10.1088/1367-2630/17/8/083026. arXiv: 1503.02065 [quant-ph].

[13] Hector Bombin and Miguel A. Martin-Delgado. "Topological Computation without Braiding". In: *Physical Review Letters* 98.16 (2007), p. 160502. DOI: 10.1103/PhysRevLett.98.160502. arXiv: quant-ph/0610024.

[14] Kepha Sher. *Interactive Visualizations for Topological Quantum Codes*. https://github.com/KephaSher/kephasher.github.io. GitHub repository. 2026.

[15] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. 10th Anniversary Edition. Cambridge University Press, 2010.

[16] Clare Horsman et al. "Surface Code Quantum Computing by Lattice Surgery". In: *New Journal of Physics* 14.12 (2012), p. 123011. DOI: 10.1088/1367-2630/14/12/123011. arXiv: 1111.4022 [quant-ph].

[17] Jiaxuan Zhang, Yu-Chun Wu, and Guo-Ping Guo. "Facilitating practical fault-tolerant quantum computing based on color codes". In: *Physical Review Research* 6.3 (2024), p. 033086. DOI: 10.1103/PhysRevResearch.6.033086. arXiv: 2309.05222 [quant-ph].

[18] Andrew J. Landahl and Ciarán Ryan-Anderson. "Quantum Computing by Color-Code Lattice Surgery". In: *arXiv preprint arXiv:1407.5103* (2014). arXiv: 1407.5103 [quant-ph].

[19] Sergey Bravyi and Alexei Kitaev. "Universal Quantum Computation with Ideal Clifford Gates and Noisy Ancillas". In: *Physical Review A* 71.2 (2005), p. 022316. DOI: 10.1103/PhysRevA.71.022316. arXiv: quant-ph/0403025.

[20] Andrew M. Steane. "Multiple-Particle Interference and Quantum Error Correction". In: *Proceedings of the Royal Society A* 452.1954 (1996), pp. 2551–2577. DOI: 10.1098/rspa.1996.0136. arXiv: quant-ph/9601029.

[21] Jonas T. Anderson, Guillaume Duclos-Cianci, and David Poulin. "Fault-Tolerant Conversion between the Steane and Reed-Muller Quantum Codes". In: *Physical Review Letters* 113.8 (2014), p. 080501. DOI: 10.1103/PhysRevLett.113.080501. arXiv: 1403.2635 [quant-ph].

[22] Craig Gidney and Martin Ekerå. "How to factor 2048 bit RSA integers in 8 hours using 20 million noisy qubits". In: *Quantum* 5 (2021), p. 433. DOI: 10.22331/q-2021-04-15-433. arXiv: 1905.09749 [quant-ph].

[23] Ilya Besedin et al. "Lattice surgery realized on two distance-three repetition codes with superconducting qubits". In: *Nature Physics* 22 (2026), pp. 189–194. DOI: 10.1038/s41567-025-03090-6.

[24] Martin Suchara, Sergey Bravyi, and Barbara M. Terhal. "Constructions and Noise Threshold of Topological Subsystem Codes". In: *Journal of Physics A: Mathematical and Theoretical* 44.15 (2011), p. 155301. DOI: 10.1088/1751-8113/44/15/155301. arXiv: 1012.0425 [quant-ph].

[25] Nathan Lacroix et al. "Scaling and logic in the colour code on a superconducting quantum processor". In: *Nature* 645.8081 (2025), pp. 614–619. DOI: 10.1038/s41586-025-09061-4. arXiv: 2412.14256 [quant-ph].

[26] Lucas Daguerre et al. "Experimental Demonstration of High-Fidelity Logical Magic States from Code Switching". In: *Physical Review X* 15.4 (2025), p. 041008. DOI: 10.1103/dck4-x9c2. arXiv: 2506.14169 [quant-ph].

[27] Héctor Bombín. "Dimensional Jump in Quantum Error Correction". In: *New Journal of Physics* 18.4 (2016), p. 043038. DOI: 10.1088/1367-2630/18/4/043038. arXiv: 1412.5079 [quant-ph].

[28] Seok-Hyung Lee, Andrew Li, and Stephen D. Bartlett. "Color code decoder with improved scaling for correcting circuit-level noise". In: *Quantum* 9 (2025), p. 1609. DOI: 10.22331/q-2025-01-27-1609. arXiv: 2404.07482 [quant-ph].

[29] Francesco Pio Barone et al. *Color Code Thresholds under Circuit-Level Noise beyond the Pauli Framework*. 2025. DOI: 10.48550/arXiv.2511.05719. arXiv: 2511.05719 [quant-ph].

[30] Yugo Takada and Keisuke Fujii. "Improving threshold for fault-tolerant color code quantum computing by flagged weight optimization". In: *PRX Quantum* 5.3 (2024), p. 030352. DOI: 10.1103/PRXQuantum.5.030352. arXiv: 2402.13958 [quant-ph].

[31] Nicolas Delfosse. "Decoding color codes by projection onto surface codes". In: *Physical Review A* 89.1 (2014), p. 012317. DOI: 10.1103/PhysRevA.89.012317. arXiv: 1308.6207 [quant-ph].

[32] Alex Townsend-Teague, Julio Magdalena de la Fuente, and Markus Kesselring. "Floquetifying the Colour Code". In: *Electronic Proceedings in Theoretical Computer Science* 384 (2023), pp. 265–303. DOI: 10.4204/EPTCS.384.14. arXiv: 2307.11136 [quant-ph].

[33] Matthew B. Hastings and Jeongwan Haah. "Dynamically Generated Logical Qubits". In: *Quantum* 5 (2021), p. 564. DOI: 10.22331/q-2021-10-19-564. arXiv: 2107.02194 [quant-ph].

[34] Wikipedia contributors. *Isomorphism theorems*. Accessed May 10, 2026. 2026. URL: https://en.wikipedia.org/wiki/Isomorphism_theorems%5C#Theorem_B_(groups).