Projecting three-dimensional color codes onto three-dimensional toric codes

Arun B. Aloshious[*] and Pradeep Kiran Sarvepalli[†]

Department of Electrical Engineering, Indian Institute of Technology Madras, Chennai 600 036, India

(Received 10 May 2018; published 2 July 2018)

Toric codes and color codes are two important classes of topological codes. Kubica *et al.* [A. Kubica *et al.*, New J. Phys. **17**, 083026 (2015)] showed that any D-dimensional color code can be mapped to a finite number of toric codes in D dimensions. We propose an alternate map of three-dimensional (3D) color codes to 3D toric codes with a view to decoding 3D color codes. Our approach builds on Delfosse's result [N. Delfosse, Phys. Rev. A **89**, 012317 (2014)] for 2D color codes and exploits the topological properties of these codes. Our result reduces the decoding of 3D color codes to that of 3D toric codes. Bit-flip errors are decoded by projecting on one set of 3D toric codes, while phase-flip errors are decoded by projecting onto another set of 3D toric codes. We use these projections to study the performance of a class of 3D color codes called stacked codes.

DOI: 10.1103/PhysRevA.98.012302

I. INTRODUCTION

Three-dimensional (3D) color codes [1], like toric codes [2,3], are topological quantum codes defined on 3D lattices. One important advantage of the 3D color codes is that they can have a transversal non-Clifford gate [1], unlike the toric codes. To achieve universal fault-tolerant quantum computation with toric codes, we need to use magic-state distillation, which can lead to large overheads. Avoiding such techniques through 3D color codes along with code switching opens up the possibility of reduced overheads. However, it must be noted that the stabilizers of the 3D color codes are of comparatively larger weight than toric codes, making them somewhat impractical. Nonetheless, it is hoped, in the final analysis, that the benefits due to reduced overheads could offset these limitations. Further, recent work in asymmetric error models [4–6] indicates that asymmetric codes could be more useful for specific channels. All of these reasons motivate us to study the 3D color codes and understand their structure more thoroughly.

Although somewhat different at first sight, toric codes and color codes are very closely related. Assuming translation and scale invariance, Yoshida showed that many classes of codes can be classified using the geometry of logical operators [7]. These results imply that 2D color codes are equivalent to 2D toric codes. Independently, Bombin *et al.* also showed that local translation-invariant stabilizer codes, including color codes, can be mapped to a finite number of copies of the toric code on the square lattice [8].

Approaching these relations between toric codes and color codes from a different vantage point, Delfosse proposed yet another mapping between color codes and toric codes [9]. Motivated by the problem of efficiently decoding 2D color codes, and exploiting the CSS structure of the color codes, he showed that 2D color codes can be projected onto copies of toric codes. Unlike the previous results [7,8], this does not explicitly address the problem of local equivalence.

It is natural to study whether similar results hold for codes in higher spatial dimensions. The central result of our paper is a mapping from 3D color codes to 3D toric codes. By exploiting the topological properties of color codes, we establish a mapping between 3D color codes and 3D toric codes. This mapping generalizes the work of Delfosse. The results in [9] are proved using the machinery of chain complexes derived from hypergraphs. We take a somewhat simpler approach and do not explicitly make use of chain complexes based on hypergraphs.

The closest work related to ours in 3D is that of Kubica *et al.* [10], who showed, among other things, that the 3D color code can be mapped to three copies of 3D toric codes. There are some important differences between our work and that of [10]. Our results give a different mapping from the color code to the toric codes. They show that 3D color codes are locally equivalent to 3D toric codes. Our map, on the other hand, does not establish a unitary equivalence. We project the X errors and Z errors onto different sets of toric codes, unlike [10] which employs just one set of toric codes. These projections are local in that local errors on the color code are projected to local errors on the toric codes. Furthermore, our map also preserves the CSS structure of the color code.

Our projection maps, like [9], are also motivated using the problem of decoding 3D color codes. A consequence of our results is that decoding 3D color codes can be reduced to the decoding of 3D toric codes. We apply our results to study the performance of stacked (color) codes. These are a different class of 3D color codes [11]. Here we study the performance of these codes. These codes are studied because they are amenable to a two-dimensional architecture for fault-tolerant quantum computation. Other codes similar to stacked codes were proposed by Jones *et al.* [12] and Bravyi *et al.* [13]. We obtained a threshold of 13.3% for bit-flip errors, but we did not observe a threshold for phase-flip errors in stacked codes. Our results have immediate application in the decoding of gauge color codes as well.

[*] aloshious.sp@gmail.com

[†] pradeep@ee.iitm.ac.in

©2018 American Physical Society

In this context, we note that a gauge color code in 3D was studied in [14]. One of the implications of this study is that the same decoder used therein can also be used to decode the phase-flip errors of the 3D color code [15]. For the bit-flip errors of the 3D color code, the decoder proposed in [14] could be used, but it will be suboptimal as the stabilizers of gauge color code are a subset of the stabilizers of the underlying 3D color code. The stacked codes that we study are different from the codes studied in [14].

The paper is structured as follows. In Sec. II, we give a brief review of 3D toric codes and color codes. In the next section, we present the central result of the paper showing how to project a 3D color code onto a collection of 3D toric codes and propose a decoding scheme for color codes. We then conclude with a brief discussion and outlook for further research. We assume that the reader is familiar with stabilizer codes [16,17].

II. PRELIMINARIES

A. 3D toric codes

We briefly review 3D topological codes. A 3D toric code is defined over a cell complex (denoted Γ) in 3D. We assume that qubits are placed on the edges of the complex. For each vertex v and face f, we define stabilizer generators as follows:

$$A_v = \prod_{e\in\iota(v)} X_e \quad\text{and}\quad B_f = \prod_{e\in\partial(f)} Z_e, \tag{1}$$

where $\iota(v)$ is the set of edges incident on v and $\partial(f)$ is the set of edges that constitute the boundary of f. When there are periodic boundary conditions, the stabilizer generators A_v and B_f are constrained as follows:

$$\prod_v A_v = I \quad\text{and}\quad \prod_{f\in\partial(\nu)} B_f = I, \tag{2}$$

where ν is any 3-cell and $\partial(\nu)$ is the collection of faces that form the boundary of ν. If Γ has boundaries, then the constraints have to be modified accordingly. Additional constraints could be present depending on the cell complex.

Sometimes it is useful to define the (3D) toric codes using the dual complex. We denote the dual of Γ by Γ^*. Qubits are placed on the faces of the dual complex. The stabilizer generators for a 3-cell ν and an edge e in the dual complex are defined as

$$A_\nu = \prod_{f\in\partial(\nu)} X_f \quad\text{and}\quad B_e = \prod_{f:e\in\partial(f)} Z_f, \tag{3}$$

where $\partial(\nu)$ is the boundary of ν and $\partial(f)$ the boundary of f.

Phase-flip errors on the toric code are detected by the operators A_ν. Phase-flip errors can be visualized as paths or strings on the lattice. The nonzero syndromes always occur in pairs. The bit-flip errors, on the other hand, are detected by the operators B_e. They are better visualized in the dual complex. Since qubits are associated to faces in the dual complex, an X error can be viewed as a surface obtained by the union of faces (with errors) and the (nonzero) syndrome as the boundary of the surface. Also note that since the boundary of each face is a cycle of trivial homology, the syndrome of X errors is a collection of cycles of trivial homology in Γ^*.

B. 3D color codes

Consider a complex with 4-valent vertices and 3-cells that are 4-colorable. Such colored complexes are called 3-colexes [1]. A 3D color code is a topological stabilizer code constructed from a 3-colex. The stabilizer generators of the color code are given as

$$B_\nu^X = \prod_{v\in\nu} X_v \quad\text{and}\quad B_f^Z = \prod_{v\in f} Z_v, \tag{4}$$

where ν is a 3-cell and f a face. It turns out that for each 3-cell ν, we can define a (dependent) Z stabilizer as $B_\nu^Z = \prod_{v\in\nu} Z_v$. A 3-colex complex defines a stabilizer code with the parameters $[[v,3h_1]]$, where h_1 is the first Betti number of the complex [1,18].

We can also define the color code in terms of the dual complex. Now qubits correspond to 3-cells, X-stabilizer generators to vertices, and Z-stabilizer generators to edges of Γ^*,

$$B_v^X = \prod_{\nu:v\in\nu} X_\nu \quad\text{and}\quad B_e^Z = \prod_{\nu:e\in\nu} Z_\nu. \tag{5}$$

We quickly review some relevant colorability properties of 3-colexes. The edges of such a 3-colex can also be 4-colored: the outgoing edges of every 3-cell can be colored with the same color as the 3-cell. We can color the faces based on the colors of the 3-cells. A face is adjacent to exactly two 3-cells. A face adjacent to 3-cells colored c and c' is colored cc'. This means that the 3-colex is 6-face-colorable. In view of the colorability of the 3-colex, we refer to a c-colored 3-cell as c-cell without explicitly mentioning that it is a 3-cell. Likewise, we can unambiguously refer to the cc'-colored faces as cc'-cells or cc'-faces, c-colored edges as c-edges, and c-colored vertices as c-vertices. We denote the i-dimensional cells of a complex Γ as $\mathsf{C}_i(\Gamma)$ and the i-dimensional cells of color c as $\mathsf{C}_i^c(\Gamma)$.

III. PROJECTING A 3D COLOR CODE ONTO 3D TORIC CODES

In this section, we state and prove the central result of the paper, namely, 3D color codes can be projected onto a finite collection of 3D toric codes. A more precise statement will be given later. First, we give an intuitive explanation and then proceed to prove it rigorously.

A. Intuitive explanation through the decoding problem

The main intuition behind the projection of color codes onto toric codes is that any such mapping should preserve the error-correcting capabilities of the color code and enable decoding. From the point of view of a decoder, the information available to it is simply the syndrome information. In a topological code, this syndrome information can be represented by the cell complex. Our main goal is to preserve the syndrome information on the 3-colex while translating it into a different cell complex.

In a 3-colex, qubits reside on the vertices, while the checks correspond to faces and volumes. If we look at the dual complex, the qubits correspond to 3-cells which are tetrahedrons; the X-type checks to vertices and the Z-type checks to edges. Due to this correspondence, we often refer to the boundary of a qubit or a collection of qubits wherein we mean the boundary

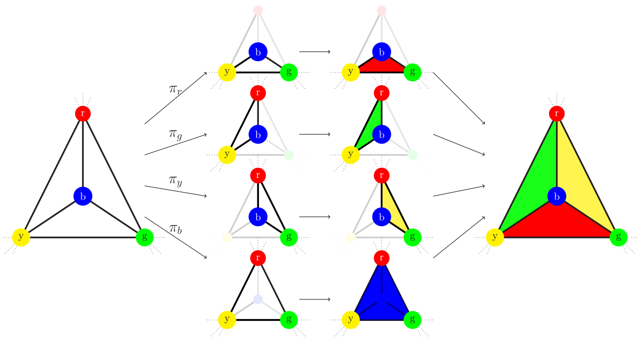

FIG. 1. We can recover the boundary of a tetrahedron from the edges by reconstructing the faces and then combining the faces. From each projection π_c, we recover a face. The faces from all the projections give the boundary of the tetrahedron.

of the 3-cells which correspond to those qubits in the dual complex.

We address the bit-flip and phase-flip errors separately. Suppose that an X error occurs. Since the qubits correspond to volumes, error correction is equivalent to (i) identifying the boundary which *encloses* the qubits in error and (ii) specifying whether the erroneous qubits lie inside or outside the boundary. The second step is necessary because the qubits with errors have the same boundary as the qubits without errors. In the case of bit-flip errors, the syndrome information is present on the edges; this is clearly not the boundary of a volume. The question then arises of how do we recover the boundary of the erroneous qubits when we appear to be in possession of some partial information about the boundary.

To see how we might solve this problem, let us assume that there is just one bit-flip error; see Fig. 1 for illustration. This causes all six edges of the tetrahedron to carry nonzero syndromes. While these edges are contained in the boundary of the tetrahedron, it is not the surface we are looking for. One way to recover the boundary of the tetrahedron is as follows. Imagine we deleted one vertex of the tetrahedron; then we would also be deleting three of the four faces of the tetrahedron and we would end with just one face. All of the edges of this remaining face carry nonzero syndromes. These edges are precisely the boundary of that face. Similarly, by deleting other vertices of the tetrahedron (separately), we would be able to recover all the faces of the tetrahedron. Since the union of these faces constitutes the boundary of the erroneous qubit, we are able to recover the boundary of the error. However, error correction is not complete. Both the single qubit in error and the collection of the qubits without error have the same boundary. To complete error correction, we also need to choose which of these sets of qubits are in error. We can decide on the volume which contains a fewer number of qubits. In the present case, we would choose the qubit in error completing the error correction. Let us identify the key ideas in the previous procedure:

(i) We construct a collection of complexes obtained by deleting c-vertices of tetrahedrons.

(ii) Then, in each complex, from the edges carrying nonzero syndrome, we recover part of the boundary of the erroneous qubits.

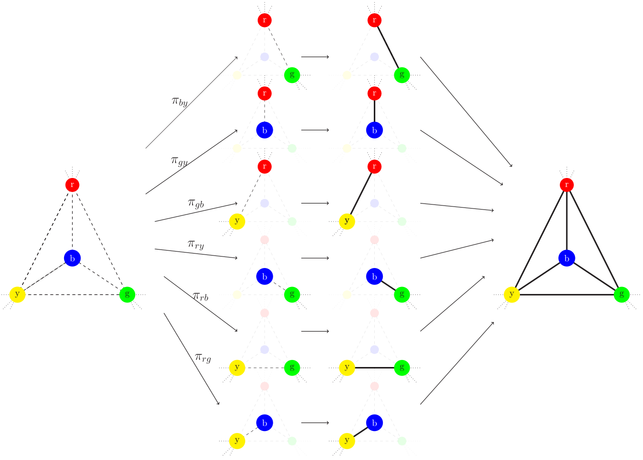

FIG. 2. We can recover the boundary of a tetrahedron from the vertices by first recovering the edges in the boundary of the tetrahedron. From each projection $\pi_{cc'}$, we recover an edge. With the edges recovered, we can proceed, as illustrated in Fig. 1, to recover the boundary of the tetrahedron.

(iii) Then we combine the boundary pieces found in (ii) to recover the boundary of the erroneous qubits.

(iv) Finally, we decide whether the interior or the exterior set of qubits enclosed by the boundary are in error.

Step (ii) is key to making the connection with the 3D toric codes. This step is identical to the correction of the X-type errors in 3D toric codes.

A similar idea will lead us to the procedure for decoding the phase-flip errors; see Fig. 2. In this case, the syndromes that detect the phase-flip errors correspond to the vertices in the dual complex. Suppose now that there is a single phase-flip error. The vertices of the erroneous tetrahedron will carry the syndrome information about the error. Now we seem to have even less information about the boundary of the erroneous tetrahedron than before. However, we can recover the boundary by the following procedure. Delete any pair of vertices of the tetrahedron. We will be left with one edge and two vertices. We can first identify the edge as a piece of the boundary that we are looking for. Deleting all six possible pairs of vertices, we are able to recover the six edges which are in the boundary of the tetrahedron. Now the problem is identical to the one we solved for correcting bit-flip errors. We summarize the key steps:

(i) We construct new complexes from the original complex by deleting pairs of vertices of each tetrahedron.

(ii) Then we recover the edges which are in the boundary of the erroneous qubits.

(iii) At this point, the problem is the same as the problem of decoding bit-flip errors, which can be solved using the previous procedure.

In correcting the Z-type errors, the connection to the toric codes happens in (ii). This is precisely the process used to decode Z-type errors in 3D toric codes.

The procedures we outline are heuristic and somewhat imprecise; they need a rigorous justification as to correctness and efficiency. We also need to consider the cases where the boundary recovery procedure can fail. We now turn to address these issues in the next section.

B. 3-colexes, duals, and minors

As we saw in the previous section, our approach to decoding color codes leads us to duals and minors of complexes. So we begin by studying the properties of the 3-colexes and their minors. First we state some properties of the dual of a 3-colex. Since they are immediate from the properties of the 3-colex, we omit the proof. For the rest of the paper, we assume that Γ is a 3-colex.

Lemma 1. Let Γ be a 3-colex. Then the dual complex Γ^* is 4-vertex-colorable, 6-edge-colorable, 4-face-colorable.

Every qubit in the 3-colex is identified with a tetravalent vertex incident on four 3-cells which are 4-colorable. Therefore, in the dual complex, every qubit corresponds to a tetrahedron whose vertices are of different colors. Similarly, the four faces of each tetrahedron are also of different colors. (Follows from Lemma 1.) Let us denote the minor complex of Γ^*, i.e., the complex obtained by deleting all the vertices of color c, by $\Gamma^{*\setminus c}$. We denote this operation as π_c so that

$$\pi_c(\Gamma^*) = \Gamma^{*\setminus c}. \tag{6}$$

The resulting structure $\Gamma^{*\setminus c}$ is a well-defined complex. Clearly, its vertices and edges are a subset of the parent complex Γ^*. The structure of the faces and 3-cells is not so obvious. A face in Γ^* that is incident on a c-vertex will not survive in $\Gamma^{*\setminus c}$. Therefore, only the c-faces of Γ^* which are not incident on a c-vertex will be faces of $\Gamma^{*\setminus c}$. The 3-cells of $\Gamma^{*\setminus c}$ are formed by merging all the tetrahedrons that are incident on a c-vertex.

We can now extend the action of π_c to the individual cells of Γ^*; there is some freedom on how to extend as long as we retain the information needed for error correction. We are primarily interested in extending π_c so that it captures the information about (i) the qubits, (ii) the errors on them, and (iii) the associated syndrome.

A vertex that is not colored c will be mapped to a vertex in $\Gamma^{*\setminus c}$. An edge that is not incident on a c-vertex will be mapped to an edge in $\Gamma^{*\setminus c}$. Edges that are incident on a c-vertex can be thought of as being mapped to the empty set. A c-face in Γ^* will continue to be a face in $\Gamma^{*\setminus c}$ as it is not incident on any c-vertices. A 3-cell has exactly 4 faces and, on the deletion of a c-vertex, just the c-face in its boundary will be left in $\Gamma^{*\setminus c}$. A 3-cell in Γ^* corresponds to a qubit, so we can interpret the c-face in its boundary as the qubit in $\Gamma^{*\setminus c}$. Since every face is shared between two 3-cells, there exist two distinct cells ν_1 and ν_2 such that $\pi_c(\nu_1) = \pi_c(\nu_2)$. The following equations summarize the preceding discussion:

$$\pi_c(v) = v \;\text{ if }\; v \in \mathsf{C}_0(\Gamma^*) \setminus \mathsf{C}_0^c(\Gamma^*), \tag{7a}$$

$$\pi_c(e) = e \;\text{ if }\; e \in \mathsf{C}_1(\Gamma^*) \setminus \mathsf{C}_1^{cc'}(\Gamma^*), \tag{7b}$$

$$\pi_c(f) = f \;\text{ if }\; f \in \mathsf{C}_2^c(\Gamma^*), \tag{7c}$$

$$\pi_c(\nu) = f^c_{\ni\nu} = \partial\nu \cap \mathsf{C}_2^c(\Gamma^*), \;\; \nu \in \mathsf{C}_3(\Gamma^*), \tag{7d}$$

where $f^c_{\ni\nu}$ is the unique c-face in ν. From these relations, we can write the boundary of a 3-cell as

$$\partial(\nu) = \sum_c \pi_c(\nu) = \sum_c f^c_{\ni\nu}, \tag{8}$$

where the summation is carried addition modulo 2. The boundary of a collection of 3-cells can be extended linearly.

Some relevant properties of the minor complexes are considered next. They concern both the structure and coloring properties of the minor complexes.

Lemma 2. Let Γ be a 3-colex and $c \in \{r,b,g,y\}$. Then the minor complex $\Gamma^{*\setminus c}$ has only c-colored faces and dd'-edges, where $d,d' \in \{r,b,g,y\} \setminus \{c\}$.

Proof. Suppose that we delete the vertices colored c in $\Gamma^{*\setminus c}$; this leads to the deletion of the edges and faces that are incident on these vertices. On any c-colored vertex, only the c-colored faces are not incident. Any face colored with $c' \in \{r,b,g,y\} \setminus c$ is incident on some c-vertex. This leads to deletion of all but one face of each tetrahedron incident on any c-colored vertex. The remaining face is colored c. Thus, $\Gamma^{*\setminus c}$ contains only c-colored faces. Since only d,d'-vertices are present in $\Gamma^{*\setminus c}$, the edges connecting them are colored dd'. ■

Lemma 3. The 3-cells of $\Gamma^{*\setminus c}$ can be indexed by the c-vertices of Γ^* and the boundary of a 3-cell $\nu_v \in \mathsf{C}_3(\Gamma^{*\setminus c})$ is the sum of c-faces of tetrahedrons incident on v.

Proof. The deletion of a c-vertex causes all the tetrahedrons incident on it to be combined into one single 3-cell in $\Gamma^{*\setminus c}$. Since the four vertices of each tetrahedron are different colors, two tetrahedrons can be merged only if they are incident on the same c-vertex. Thus two distinct c-vertices lead to distinct 3-cells. Hence, the 3-cells of $\Gamma^{*\setminus c}$ can be indexed by the c-vertices of Γ^*. The deletion of the c-vertex v creates a 3-cell and leaves behind a c-face for every tetrahedron incident on v. These c-faces enclose the 3-cell formed by merging the qubits incident on v; therefore, they must form its boundary. Denote by ν_v such a 3-cell. Then, its boundary $\partial(\nu_v)$ is given by

$$\partial(\nu_v) = \sum_{\nu:v\in\nu} \partial(\nu) \overset{(a)}{=} \sum_{\nu:v\in\nu} f^c_{\ni\nu} + \sum_{\nu:v\in\nu}\sum_{i\neq c} f^i_{\ni\nu} \tag{9}$$

$$\overset{(b)}{=} \sum_{\nu:v\in\nu} f^c_{\ni\nu}, \tag{10}$$

which is precisely the sum of c-faces of tetrahedrons incident on v. Note that (a) follows from Eq. (8), while (b) is due to the fact that every c'-face incident on v is shared between exactly two qubits incident on v causing the second summation to vanish. ■

Let $\Gamma^{*\setminus cc'}$ denote the minor of Γ^* obtained by deleting all vertices colored c and c'. We assume that $c \neq c'$ for the rest of the paper. Denote this operation as $\pi_{cc'}$. Then we have

$$\pi_{cc'}(\Gamma^*) = \Gamma^{*\setminus cc'}. \tag{11}$$

Note that the order of deletion of vertices does not matter, and therefore we have

$$\pi_{cc'}(\Gamma^*) = \pi_{c'c}(\Gamma^*). \tag{12}$$

As in the case of $\Gamma^{*\setminus c}$, the vertices and edges of $\Gamma^{*\setminus cc'}$ are a subset of $\Gamma^{*\setminus c}$ and can be easily identified. The faces and 3-cells are not so obvious. The faces of $\Gamma^{*\setminus cc'}$ are not faces in the parent complexes Γ^* or $\Gamma^{*\setminus c}$. We need to define the faces and 3-cells of $\Gamma^{*\setminus cc'}$. We make one small observation before defining them.

Lemma 4. Let Γ be a 3-colex and $c,c' \in \{r,b,g,y\}$. Then, $\Gamma^{*\setminus cc'}$ has only dd'-colored edges, where $\{d,d'\} = \{r,b,g,y\} \setminus \{c,c'\}$.

Proof. Suppose that we delete all the vertices colored c,c' in Γ^*; then, all edges incident on c-vertices and c'-vertices will be

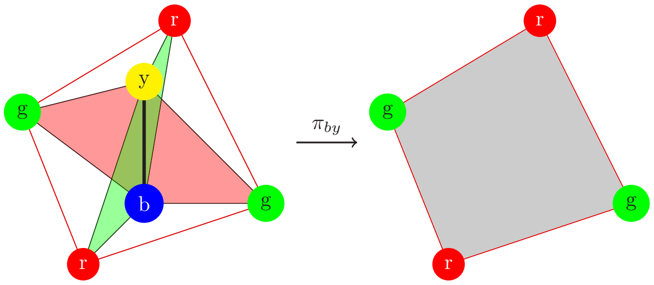

FIG. 3. Consider the by-edge (in bold) and the tetrahedra (qubits) incident on it. Since only r-, g-vertices (i.e., red and green) will survive in $\Gamma^{*\backslash by}$, each of these surviving vertices will have exactly two rg-edges (of the tetrahedra incident on the by-edge) incident on them. Therefore, these rg-edges form a cycle, as shown on the right. These edges form the boundary of a face in $\Gamma^{*\backslash by}$. The faces of other complexes $\Gamma^{*\backslash cc'}$ are similarly formed.

deleted. Thus only edges between d- and d'-colored vertices will remain. These edges are colored dd'. ■

Let e be a cc'-edge e in Γ^* and $\mathcal{V}_e$ be the qubits containing e,

$$\mathcal{V}_e = \{\nu \mid e \in \nu\}. \tag{13}$$

Let $e^{dd'}_{\ni\nu}$ be the unique dd'-edge in ν. For each of the qubits in $\mathcal{V}_e$, exactly one dd'-edge will survive in $\Gamma^{*\backslash cc'}$. No two qubits in $\mathcal{V}_e$ share the same dd'-edge. Further, the surviving dd'-edges will form a cycle in $\Gamma^{*\backslash cc'}$; see Fig. 3 for illustration. This can be seen as follows. Let $\mathsf{V}_{cc'}(\mathcal{V}_e)$ denote the vertices of $\mathcal{V}_e$ that remain in $\Gamma^{*\backslash cc'}$. Every such vertex v is incident on exactly two qubits of $\mathcal{V}_e$. Therefore, two dd'-edges are incident on v. Hence, the dd'-edges of $\mathcal{V}_e$ form a cycle. This cycle is of trivial homology since it is on the boundary of a 3-cell (formed by the qubits in $\mathcal{V}_e$). We can associate a face to this cycle such that it lies entirely in the 3-cell. In other words, to every cc'-edge in Γ^*, we can associate a face in $\Gamma^{*\backslash cc'}$. The boundary of this face is the collection of the dd'-edges belonging to the qubits incident on e, alternatively,

$$\partial f_e = \sum_{\nu\in\mathcal{V}_e} e^{dd'}_{\ni\nu}. \tag{14}$$

The qubits incident on distinct cc'-edges e_1 and e_2 will be disjoint intersecting in either edges or vertices, so the faces associated to them, i.e., f_{e_1} and f_{e_2}, will also be disjoint and intersect in edges or vertices. Therefore, the faces are well defined. The preceding discussion proves the following result.

Lemma 5. Let Γ be a 3-colex and $c,c' \in \{r,b,g,y\}$. Then the faces of the minor complex $\Gamma^{*\backslash cc'}$ are in one-to-one correspondence with cc'-edges of Γ^*.

With faces of $\Gamma^{*\backslash cc'}$ defined, the 3-cells of $\Gamma^{*\backslash cc'}$ can be identified. Let v be a c- or c'-vertex. Consider the cc' edges incident on v. Each of these edges corresponds to a face in $\Gamma^{*\backslash cc'}$, by Lemma 5. We define the volume enclosed by these faces, such that it contains v, to be a 3-cell of $\Gamma^{*\backslash cc'}$. Since every such v leads to a 3-cell in $\Gamma^{*\backslash cc'}$, we have the following:

Lemma 6. Let Γ be a 3-colex and $c,c' \in \{r,b,g,y\}$. Then, 3-cells of the minor complex $\Gamma^{*\backslash cc'}$ are in one-to-one correspondence with vertices in $\mathsf{C}_0^c(\Gamma^*) \cup \mathsf{C}_0^{c'}(\Gamma^*)$.

We can also extend $\pi_{cc'}$ to the cells of Γ^*, as we did for π_c. Then we can write

$$\pi_{cc'}(v) = v \;\text{ if }\; v \in \mathsf{C}_0(\Gamma^*) \setminus \left[\mathsf{C}_0^c(\Gamma^*) \cup \mathsf{C}_0^{c'}(\Gamma^*)\right], \tag{15a}$$

$$\pi_{cc'}(e) = e \;\text{ if }\; e \in \mathsf{C}_1^{dd'}(\Gamma^*);\;\; d,d' \not\in \{c,c'\}, \tag{15b}$$

$$\pi_{cc'}(\nu) = e^{dd'}_{\ni\nu} = \partial[\pi_c(\nu)] \cap \mathsf{C}_1^{dd'}(\Gamma^*);\;\; \nu \in \mathsf{C}_3(\Gamma^*), \tag{15c}$$

where $e^{dd'}_{\in\nu}$ is the unique dd'-edge in ν. In these equations and henceforth, we assume $d,d' \in \{r,b,g,y\} \setminus \{c,c'\}$ and $d \neq d'$. None of the faces of $\Gamma^{*\backslash c}$ or Γ^* will survive in $\Gamma^{*\backslash cc'}$. Since the faces of Γ^* do not carry any information about the qubits and the Z error syndromes, we are not particularly interested in them; we have some freedom as to how to define $\pi_{cc'}$ for faces in $\Gamma^{*\backslash c}$.

We define the edge boundary of a 3-cell in Γ^* as

$$\delta\nu = \sum_{cc'} \pi_{cc'}(\nu) = \sum_{cc'} e^{dd'}_{\ni\nu}. \tag{16}$$

We call it the edge boundary because $\pi_{cc'}(\nu)$ is an edge; see Eq. (15c). We extend δ to multiple 3-cells linearly.

A simple example of 3-colex and related complexes are shown in Fig. 4. We can relate the various complexes and the objects of interest for us as follows:

	Γ	Γ^*	$\Gamma^{*\backslash c}$	$\Gamma^{*\backslash cc'}$
Qubit	vertex	tetrahedron	triangle	edge
Z-check	face	edge	edge	
X-check	3-cell	vertex	vertex	vertex

The preceding lemmas lead to the following corollary.

Corollary 1. Let Γ be a 3-colex with v vertices, $e = 2v$ edges, $f_{cc'}$ cc'-faces, and ν_c c-cells. Let the total number of faces be $f = \sum_{cc'} f_{cc'}$ and 3-cells be $\nu = \sum_c \nu_c$. The following table summarizes the number of cells in $\Gamma^{*\backslash c}$ and $\Gamma^{*\backslash cc'}$, where $c,c',d,d' \in \{r,b,g,y\}$ are distinct:

	Γ	Γ^*	$\Gamma^{*\backslash c}$	$\Gamma^{*\backslash cc'}$
3-cells	ν	v	ν_c	$\nu_c + \nu_{c'}$
Faces	f	$2v$	$v/2$	$f_{cc'}$
Edges	$2v$	f	$f_{dc'} + f_{c'd'} + f_{dd'}$	$f_{dd'}$
Vertices	v	ν	$\nu_{c'} + \nu_d + \nu_{d'}$	$\nu_d + \nu_{d'}$

Proof. The i-cells in dual complex Γ^* are in one-to-one correspondence with the $(3-i)$-cells of Γ. Now suppose that Γ^* is modified so that all vertices colored i are deleted. Then all 3-cells incident on it will be merged to form a new 3-cell. Since all the 3-cells in Γ^* are incident on some i-vertex, they will be part of some new 3-cell and $\Gamma^{*\backslash c}$ will contain c_i 3-cells. On deleting the i-vertices, exactly one face will remain from each 3-cell in Γ^*. Since each of these faces will be shared between two 3-cells, there will be $v/2$-faces. The edges in $\Gamma^{*\backslash i}$ are those that are incident on vertices other than i-vertices.

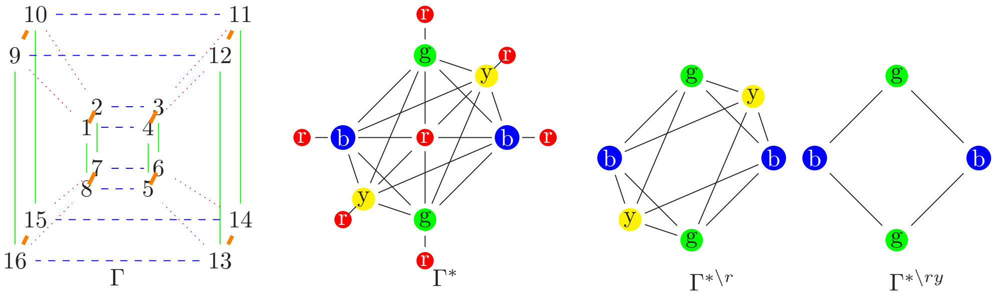

FIG. 4. A 3-colex Γ and its dual Γ^*; i-cells of Γ^* correspond to the $(3-i)$-cells of Γ. The minor complexes $\Gamma^{*\backslash r}$ and $\Gamma^{*\backslash ry}$ are also shown. (In Γ, we use smooth lines for green edges, dotted lines for red edges, dashed lines for blue edges, and thick lines for yellow edges.)

Similarly, in $\Gamma^{*\backslash cc'}$, only the d-vertices, d'-vertices, and dd'-edges will survive. These are precisely $v_d + v_{d'}$-vertices and $f_{dd'}$-edges. The number of faces and 3-cells of $\Gamma^{*\backslash cc'}$ is immediate from Lemmas 5 and 6. ■

Remark 1. The minor complexes defined here are the duals of the shrunk complexes defined in [18]. For example, $\Gamma^{*\backslash b}$ is exactly the dual of the b-shrunk complex and $\Gamma^{*\backslash ry}$ is the dual of the ry-shrunk complex.

C. X-type errors on 3D color codes

Let us now see how to perform error correction on a color code. It is helpful to see the (topological) structure of the errors in the dual of the 3-colex. We analyze the bit-flip and phase-flip errors separately. Suppose that we have X errors on some set of qubits. In the dual colex, the erroneous qubits correspond to a volume. Through π_c, we can associate qubits to faces of the minor complex $\Gamma^{*\backslash c}$. Thus we can project errors from Γ^* to $\Gamma^{*\backslash c}$.

If a qubit v has a bit-flip error, then we place an X error on the image of v in $\Gamma^{*\backslash c}$. In other words,

$$\pi_c(X_v) = X_{\pi_c(v)}, \tag{17}$$

where $\pi_c(X_v)$ gives an X error acting on the qubits in $\Gamma^{*\backslash c}$.

A consequence of Eq. (17), together with linearity of π_c, is that for two adjacent qubits v_1 and v_2 sharing a c-colored face f, we have $\pi_c(X_{v_1}X_{v_2}) = X_{\pi_c(v_1)}X_{\pi_c(v_2)} = I$, where we used the fact that $\pi_c(v_1) = \pi_c(v_2) = f$. In other words, a c-face common to two qubits in error corresponds to an error-free qubit in $\Gamma^{*\backslash c}$.

The syndrome corresponding to bit-flip errors is associated to edges of Γ^*. In $\Gamma^{*\backslash c}$, not all edges are present. But if an edge is present, we associate to that edge the same syndrome as in Γ^*. Syndromes in the minor complex are essentially the restriction of the syndromes in Γ^*. Let s_e be the syndrome on edge e; then,

$$\pi_c(s_e) = s_{\pi_c(e)} = s_e. \tag{18}$$

This is consistent with Eq. (7b).

At this point, we have qubits living on the faces and syndromes on the edges of $\Gamma^{*\backslash c}$ just as we would have in a 3D toric code. But it needs to be shown that indeed we truly have the structure of a 3D toric code and not merely the appearance of it. This we shall take up next.

First let us consider the edge-type checks on $\Gamma^{*\backslash c}$. Consider an edge e in $\Gamma^{*\backslash c}$. Then, e is also present in Γ^*. For each qubit incident on e, there is a c-colored face incident on e. These c-faces exhaust all the faces incident on e in $\Gamma^{*\backslash c}$. Thus, in the 3D toric code associated to $\Gamma^{*\backslash c}$, every face incident on e participates in that check on e as required for the edge-type checks in the 3D toric code. Next we look at the projected syndromes on the minor complex.

Theorem 1. Projection of X errors onto toric codes. Let s be the syndrome for an X error E on the 3D color code defined on a 3-colex Γ and $\pi_c(s)$ the restriction of s on $\Gamma^{*\backslash c}$. Then the error $\pi_c(E)$ in $\Gamma^{*\backslash c}$ produces the syndrome $\pi_c(s)$ in the toric code associated to $\Gamma^{*\backslash c}$.

Proof. We now will show that in $\Gamma^{*\backslash c}$, the syndrome produced by $\pi_c(E)$ is the same as $\pi_c(s)$. Consider any edge in $\Gamma^{*\backslash c}$; by definition, $\pi_c(s_e) = s_e$, where s_e is the syndrome on e with respect to Γ^*. Since Γ is 3-colex, an even number of qubits are incident on e, say $2m$. Then, $s_e = \oplus_{i=1}^{2m} q_i$ where $q_i = 1$ if there is an X error on the ith qubit, and zero otherwise. Each of these qubits (tetrahedrons) incident on e is projected to a qubit in $\Gamma^{*\backslash c}$. But note that two qubits which share a face are mapped to the same qubit in $\Gamma^{*\backslash c}$. Thus there are m qubits (triangles) incident on e with respect to $\Gamma^{*\backslash c}$. These (projected) qubits are in error if and only if one of the parent qubits in Γ^* is in error. Let $r_j = 1$ if there is an error on the projected qubit, and zero otherwise. Then, $r_j = q_{2j-1} \oplus q_{2j}$, where $2j-1$ and $2j$ are the qubits which are projected onto the jth qubit in $\Gamma^{*\backslash c}$. The syndrome on e as computed in the 3D toric code is $\oplus_{j=1}^{m} r_j = \oplus_{j=1}^{m}(q_{2j-1} \oplus q_{2j}) = s_e$. Thus the projected error $\pi_c(E)$ produces the same syndrome as the projected syndrome $\pi_c(s)$.

Corollary 2. Let s be the syndrome for an X error on the 3D color code defined on a 3-colex Γ and $\pi_c(s)$ the restriction of s on $\Gamma^{*\backslash c}$. Then, $\pi_c(s)$ is a valid syndrome for (an X error on) the toric code on $\Gamma^{*\backslash c}$.

Proof. By Theorem 8, $\pi_c(s)$ is the same as the syndrome produced by an X error on $\Gamma^{*\backslash c}$. Therefore, it must be a valid syndrome for an X error for the 3D toric code on $\Gamma^{*\backslash c}$.

Lemma 7. Let v be a c-vertex in Γ^* and ν_v be the 3-cell in $\Gamma^{*\backslash c}$ obtained by merging all the qubits incident on v. Then the X-type stabilizer B_v^X of the color code on Γ^* is mapped to an X-type stabilizer generator of the toric code on $\Gamma^{*\backslash c}$,

$$\pi_c\left(B_v^X\right) = B_{\nu_v}^X \text{ and } \pi_{c'}\left(B_v^X\right) = I \text{ for } c' \neq c. \tag{19}$$

Proof. We have $B_v^X = \prod_{\nu:v\in\nu} X_\nu$. Then,

$$\pi_c(B_v^X) = \pi_c\left(\prod_{\nu:v\in\nu} X_\nu\right) = \prod_{\nu:v\in\nu} X_{\pi_c(\nu)} \tag{20}$$

$$\overset{(a)}{=} \prod_{f\in\partial(\nu_v)} X_f = B_{\nu_v}^X, \tag{21}$$

where (a) follows from Lemma 3. Thus, $\pi_c(B_v^X)$ is exactly the X-type stabilizer generator defined on the 3-cell ν_v.

Now consider a c'-face incident on v. Such a face is in the boundary of two qubits incident on v. This means that for every qubit ν incident on v, there exists another qubit ν' incident on v such that $\pi_{c'}(X_\nu) = \pi_{c'}(X_{\nu'})$. Therefore, $\pi_{c'}(B_v^X) = \prod_{\nu:v\in\nu} X_{\pi_c(\nu)} = I$. ■

Note that the previous lemma implies that for a c'-vertex, $\pi_c(B_v^X) = I$.

Corollary 3. Let $\overline{S}$ be an X-type stabilizer on the color code. Then, $\pi_c(\overline{S})$ is an X-stabilizer on the toric code defined by $\Gamma^{*\backslash c}$. Conversely, for every X-stabilizer S on $\Gamma^{*\backslash c}$, there exists an X-stabilizer generator $\overline{S}$ on Γ^* such that $\pi_c(\overline{S}) = S$ and $\pi_{c'}(\overline{S}) = I$.

Proof. The first statement is a consequence of Lemma 7 as $\{B_v^X\}$ generate all X-stabilizers of the color code. Further, $\{\pi_c(B_v^X)\} = \{B_{\nu_v}^X\}$, where $v \in \mathsf{C}_0^c(\Gamma^*)$. By Lemma 3, $\{B_{\nu_v}^X\}$ generate the X-type stabilizers of the toric code on $\Gamma^{*\backslash c}$. Thus the converse also holds. ■

Let the support of an error E be defined [19] as

$$\text{supp}(E) = \sum_{i:E_i\neq I} i, \tag{22}$$

where i could be a 3-cell, face, or an edge, depending on where the qubits are located. It follows that

$$\text{supp}(EE') = \text{supp}(E) + \text{supp}(E'). \tag{23}$$

We define the boundary of an error in Γ^* to be the boundary of the volume that corresponds to the collection of qubits on which the error acts nontrivially. In other words,

$$\partial E = \sum_{\nu:E_\nu\neq I} \partial\nu = \partial[\text{supp}(E)]. \tag{24}$$

We use the same notation ∂ for the boundary of cells as well as operators. Note that $\partial(EE') = \partial E + \partial E'$.

Lemma 8. X error boundary. Using the same notation as in Theorem 1, the boundary of an error E in Γ^* is

$$\partial E = \sum_c \text{supp}[\pi_c(E)]. \tag{25}$$

Proof. Let E_ν denote the error on the 3-cell corresponding to the νth qubit. Then we can write

$$\partial E \overset{(a)}{=} \sum_{\nu:E_\nu\neq I} \partial\nu \overset{(b)}{=} \sum_{\nu:E_\nu\neq I}\sum_c \pi_c(\nu) \tag{26}$$

$$= \sum_c \sum_{\nu:E_\nu\neq I} \pi_c(\nu) \overset{(c)}{=} \sum_c \sum_{\nu:E_\nu\neq I} \text{supp}[\pi_c(E_\nu)] \tag{27}$$

$$\overset{(d)}{=} \sum_c \text{supp}[\pi_c(E)], \tag{28}$$

where (a) follows from the definition of the boundary of an error; (b) follows from the fact that the boundary of a single qubit is the collection of four faces of the tetrahedron that correspond to the qubit; (c) follows from rearranging the order of summation and the observation that $\pi_c(\nu)$ is the c-face in the boundary of ν [see Eq. (7d)], and if $E_\nu \neq I$, then this is the same as the support of $\pi_c(E_\nu)$; and (d) follows from Eq. (23) and completes the proof. ■

Algorithm 1. Estimating (face) boundary of X-type error

```
Input: A 3-colex $\Gamma$, Syndrome of an $X$ error $E$
Output: $\mathsf{F}$, an estimate of $\partial E$ where $\mathsf{F} \subseteq \mathsf{C}_2(\Gamma^*)$
for each $c \in \{r,b,g,y\}$ do
    for each edge $e$ in $\Gamma^{*\backslash c}$ do          // syndrome projection
        $s_{\pi_c(e)} = s_e$                                // $s_e$ is syndrome on edge $e$
    end for
    Using the projected syndrome on $\Gamma^{*\backslash c}$, estimate the error $\mathsf{F}_c$
  by any 3D toric code decoder for $X$ errors
end for
Return $\mathsf{F} = \bigcup_c \mathsf{F}_c$
```

By Lemma 8, the boundary of an error E can be broken down into four surfaces, each lying in a separate minor complex $\Gamma^{*\backslash c}$. We show that if these surfaces were modified by the support of a stabilizer in the minor complexes, then these modified surfaces form the boundary of an error E' which is equivalent to E up to a stabilizer, i.e., $E' = ES$ for some X-stabilizer S on the color code.

Lemma 9. X error boundary modulo stabilizer. Let E be an error on Γ^*. Let S_c be X-stabilizer generators on $\Gamma^{*\backslash c}$. Then, $\partial E + \sum_c \text{supp}(S_c)$ is the boundary of ES, for some X-stabilizer S on Γ^* i.e.,

$$\partial E + \sum_c \text{supp}(S_c) = \partial(ES). \tag{29}$$

Proof. Since S_c is a stabilizer generator, by Corollary 3, there exists an X-stabilizer $\overline{S}_c$ on Γ^* such that $\pi_c(\overline{S}_c) = S_c$ and $\pi_{c'}(\overline{S}_c) = I$. By Lemma 8, $\partial(\overline{S}_c) = \text{supp}(S_c)$. Then, using $\partial EF = \partial E + \partial F$, we obtain $\partial E\overline{S}_c = \partial E + \partial\overline{S}_c$. Repeating this for all c, we have $\partial(E\prod_c \overline{S}_c) = \partial E + \sum_c \partial\overline{S}_c$. Letting $\prod_c \overline{S}_c = S$, we can write this as $\partial(ES) = \partial E + \sum_c \text{supp}(S_c)$ as claimed. ■

The importance of the previous result is that we can independently estimate the four components of the boundary of an error. With these results in hand, we can estimate the boundary of an X error.

Theorem 2. Estimating the face boundary of X errors. The boundary of an X error E on the dual of the color code can be estimated by Algorithm 1. The algorithm estimates ∂E up to the boundary of an X-stabilizer of the color code, provided $\pi_c(E)$ is estimated up to a stabilizer on $\Gamma^{*\backslash c}$, where $c \in \{r,g,b,y\}$.

Proof. We only sketch the proof as it is a straightforward consequence of the results we have shown thus far. By Lemma 8, the boundary of the error consists of support of $\pi_c(E)$, i.e., the projections of the error on the 3D toric codes on $\Gamma^{*\backslash c}$. By Theorem 1, the syndrome of $\pi_c(E)$ is the restriction of the syndrome on Γ^*. Therefore, $\pi_c(E)$ can be estimated by decoding on $\Gamma^{*\backslash c}$. By Lemma 9, if the estimate for $\pi_c(E)$

is equivalent up to a stabilizer on $\Gamma^{*\backslash c}$, we can obtain the boundary of E up to the boundary of a stabilizer on the color code. ■

If any of component decoders on the minor complexes make a logical error, then Algorithm 1 may fail to produce a valid boundary.

D. Z-type errors on 3D color codes

One can prove results similar to the previous section for the Z-type errors also. As we noted earlier, the syndrome information about Z errors resides on the vertices of Γ^*, while the error corresponds to a volume. To recover the boundary of the error, the objects of interest are the minor complexes $\Gamma^{*\backslash cc'}$. Recall that the edges of $\Gamma^{*\backslash cc'}$ are associated with qubits. So we project the errors on cell ν to $\pi_{cc'}(\nu)$, which is an edge in $\Gamma^{*\backslash cc'}$. We define

$$\pi_{cc'}(Z_\nu) = Z_{\pi_{cc'}(\nu)}. \tag{30}$$

Let the syndrome on $v \in \mathsf{C}_0(\Gamma^*)$ be s_v. Then we define the syndrome on $v \in \Gamma^{*\backslash cc'}$ as

$$\pi_{cc'}(s_v) = s_{\pi_{cc'}(v)} = s_v, \tag{31}$$

so the syndrome on $\pi_{cc'}(\Gamma^*)$ is simply the restriction of the syndrome on Γ^*. To project the color code onto toric codes, this syndrome must be a valid syndrome on the minor complex.

We next show that both Z errors and their associated syndromes can be projected consistently onto the minor complexes $\Gamma^{*\backslash cc'}$.

Theorem 3. Projection of Z errors onto toric codes. Let E be a Z-type error on Γ^* and its associated syndrome s. Then the error $\pi_{cc'}(E)$ in $\Gamma^{*\backslash cc'}$ produces the syndrome $\pi_{cc'}(s)$.

Proof. We only need to show the theorem for vertices $v \in \Gamma^{*\backslash cc'}$. Suppose E produces the syndrome s_v on v; then, $s_v = \bigoplus_{\nu:v\in\nu} q_\nu$, where $q_\nu = 1$, if there is a Z error on ν and zero otherwise. We need to show that $\pi_{cc'}(E)$ produces the syndrome s_v on v. Every qubit incident on v must have a dd'-edge incident on v since v must be a d- or d'-vertex. Two qubits incident on v can share at most one such edge. Then we can partition the qubits incident on v depending on the dd'-edge on which they are incident. Let $\{e_1,\ldots,e_m\}$ be these dd'-edges. Then we can write

$$\{\nu : v \in \nu\} = \cup_i \{\nu : v, e_i \in \nu\}, \tag{32}$$

$$s_v = \bigoplus_{\nu:v\in\nu} q_\nu = \bigoplus_i \bigoplus_{\nu:v,e_i\in\nu} q_\nu = \bigoplus_i r_i, \tag{33}$$

where $r_i = \bigoplus_{i:v,e_i\in v_i} q_\nu$.

The qubits containing e_i are projected onto the dd'-edge e_i in the minor complex $\Gamma^{*\backslash cc'}$ and there is an error on e_i if and only if $r_i = \bigoplus_{\nu:v,e_i\in\nu} q_\nu = 1$. Thus the syndrome on the vertex v as computed with respect to the toric code on $\Gamma^{*\backslash cc'}$ is $\bigoplus_{i=1}^m r_i = s_v = \pi_{cc'}(s_v)$ as required. ■

Corollary 4. Validity of Z syndrome restriction. Let s be the syndrome for a Z error on Γ^*. Then, $\pi_{cc'}(s)$ is a valid syndrome for a Z error on $\Gamma^{*\backslash cc'}$.

Proof. From Theorem 3, we see that $\pi_{cc'}(s)$ coincides with the syndrome produced by a Z error on $\Gamma^{*\backslash cc'}$. Hence, $\pi_{cc'}(s)$ must be a valid syndrome for a Z error on a 3D toric code. ■

Having projected both the error and the syndrome onto the minor complexes, we recover the boundary of the error in steps. To this end, we define the edge boundary of an error E as

$$\delta E = \sum_{\nu:E_\nu\neq I} \delta\nu = \sum_{\nu:E_\nu\neq I}\sum_{cc'} \pi_{cc'}(\nu) \tag{34}$$

$$= \sum_{cc'} \text{supp}[\pi_{cc'}(E)], \tag{35}$$

where Eq. (35) follows from interchanging the order of summation in Eq. (34). We see that the edge boundary of E can be recovered by recovering $\pi_{cc'}(E)$. The next lemma shows an interesting property of the edge boundary that will help us in correcting Z errors.

Lemma 10. Edge boundary corresponds to an X syndrome. Let $E_Z = \prod_{\nu\in\Omega} Z_\nu$ and $E_X = \prod_{\nu\in\Omega} X_\nu$. Then the syndrome of E_X is nonzero on the edges in δE_Z.

Proof. Consider the syndrome of E_X on an edge e. Then,

$$s_e \neq 0 \;\text{ iff }\; |\{\nu \in \Omega \mid e \in \nu\}| \;\text{ is odd.} \tag{36}$$

In other words, s_e is nonzero if and only if the number of qubits in Ω incident on e is odd. From Eq. (34), we see that e will be present in the edge boundary of E_Z if and only if an odd number of qubits are incident on e. Thus the syndrome of E_X is nonzero on the edge boundary of E_Z. ■

We cannot always expect to estimate the edge boundary exactly because the estimates for $\pi_{cc'}(E)$ on the minor complexes could be off. The next lemmas show this will not be problem as long as the estimates for $\pi_{cc'}(E)$ are off by stabilizer elements.

Lemma 11. Suppose S is a Z-stabilizer on $\Gamma^{*\backslash cc'}$; then there exists a Z-stabilizer $\overline{S}$ in Γ^* such that $\pi_{cc'}(\overline{S}) = S$ and $\pi_{xy}(\overline{S}) = I$ for $xy \neq cc'$.

Proof. A Z-stabilizer in $\Gamma^{*\backslash cc'}$ is generated by the face-type stabilizers in $\Gamma^{*\backslash cc'}$. Therefore, it suffices to consider when S is a face-type stabilizer. By Lemma 5, the faces of $\Gamma^{*\backslash cc'}$ are in correspondence with the cc'-edges of Γ^*. So we can let $S = B_{f_e}^Z$ for some face f_e in $\Gamma^{*\backslash cc'}$ and cc'-edge e in Γ^*. The stabilizer of the color code attached to e is given by $B_e^Z = \prod_{\nu:e\in\nu} Z_\nu$. Then,

$$\pi_{cc'}(B_e^Z) = \prod_{\nu:e\in\nu} Z_{\pi_{cc'}(\nu)} \overset{(a)}{=} \prod_{\nu:e\in\nu} Z_{e^{dd'}_{\ni\nu}} \tag{37}$$

$$\overset{(b)}{=} \prod_{t\in\partial(f_e)} Z_t = B_{f_e}^Z = S, \tag{38}$$

where (a) follows from Eq. (15c) and (b) from Eq. (14).

Denote by $\mathcal{V}_e$ the set of qubits incident on e. Every qubit $\nu \in \mathcal{V}_e$ contains a $c'd'$-edge. These edges must also be incident on the c'-vertex of e. Two qubits ν and ν' which have the same $c'd'$-edge must share a face since they already share e. Hence only two qubits $\nu,\nu' \in \mathcal{V}_e$ can share a $c'd'$-edge. For these qubits, we have $\pi_{cd}(\nu) = \pi_{cd}(\nu')$. This implies $\pi_{cd}(B_e^Z) = \prod_{\nu:e\in\nu} Z_{e^{c'd'}_{\ni\nu}} = I$. Similar arguments can be used to show that $\pi_{xy}(\overline{S}) = I$ for other $xy \neq cc'$. We omit the details. ■

Lemma 12. Edge boundary modulo stabilizer. Let E be a Z-type error on the color code and S a Z-stabilizer on $\Gamma^{*\backslash cc'}$. Then, $\delta E + \sum_{e:S_e\neq I} e$ is the edge boundary of $E\overline{S}$ for some Z-stabilizer $\overline{S}$ on Γ^*.

Algorithm 2. Estimating (edge) boundary of Z-type error

```
Input: A 3-colex $\Gamma$, Syndrome of a $Z$ error $E$
Output: $\mathsf{E}$, an estimate of the edge boundary $\delta E$, where $\mathsf{E} \subseteq \mathsf{C}_1(\Gamma^*)$
for each $c,c' \in \{r,b,g,y\}$ do
    for each vertex $v$ in $\Gamma^{*\backslash cc'}$ do          // syndrome projection
        $s_{\pi_{cc'}(v)} = s_v$                           // $s_v$ is syndrome on vertex $v$
    end for
    Estimate the error $\mathsf{E}_{cc'}$ using any 3D toric code decoder for $Z$
  errors on $\Gamma^{*\backslash cc'}$
end for
Return $\mathsf{E} = \sum_{c,c'} \mathsf{E}_{cc'}$
```

Proof. By Lemma 11, there exists a Z-stabilizer $\overline{S}$ on Γ^* such that $\pi_{cc'}(\overline{S}) = S$ and $\pi_{xy}(\overline{S}) = I$ for $xy \neq cc'$. Therefore,

$$I = \pi_{xy}(\overline{S}) = \pi_{xy}\left(\prod_{v:\overline{S}_v \neq I} Z_v\right) = \prod_{v:\overline{S}_v \neq I} Z_{\pi_{xy}(v)}. \tag{39}$$

From this, we infer that for $xy \neq cc'$,

$$\sum_{v:\overline{S}_v \neq I} \pi_{xy}(v) = 0. \tag{40}$$

Therefore, the edge boundary of $\overline{S}$ has support only in $\Gamma^{*\backslash cc'}$. Furthermore, substituting for S and $\overline{S}$ in $S = \pi_{cc'}(\overline{S})$, we obtain

$$\prod_{e:S_e \neq I} Z_e = \pi_{cc'}\left(\prod_{v:\overline{S}_v \neq I} Z_v\right) = \prod_{v:\overline{S}_v \neq I} Z_{\pi_{cc'}(v)}. \tag{41}$$

Equation (41) implies that

$$\sum_{e:S_e \neq I} e = \sum_{v:\overline{S}_v \neq I} \pi_{cc'}(v) \tag{42}$$

$$= \sum_{v:\overline{S}_v \neq I} \pi_{cc'}(v) + \sum_{xy \neq cc'} \sum_{v:\overline{S}_v \neq I} \pi_{xy}(v) \tag{43}$$

$$= \delta \overline{S}. \tag{44}$$

Thus, $\delta\overline{S} = \sum_{e:S_e \neq I} e$ and $\delta E + \sum_{e:S_e \neq I} e = \delta(E\overline{S})$. ■

With these results in hand, we show how to estimate the edge boundary of a Z error from the minor complexes given the syndrome on its vertices.

Theorem 4. Estimating edge boundary of Z errors. Let Γ be a 3-colex and E a Z error on the associated color code. Algorithm 2 estimates δE, the edge boundary of E, up to the boundary of a Z-stabilizer of the color code, provided $\pi_{cc'}(E)$ is estimated up to a stabilizer on $\Gamma^{*\backslash cc'}$.

Proof. By Corollary 4, the restriction of the syndrome of E is a valid syndrome on $\Gamma^{*\backslash cc'}$. By Theorem 3, $\pi_{cc'}(E)$ has the same syndrome as the restriction and $\pi_{cc'}(E)$ can be estimated using a 3D toric decoder on $\Gamma^{*\backslash cc'}$. We can reconstruct the edge boundary from $\pi_{cc'}(E)$ using Eq. (35). By Lemma 12, if the estimates for $\pi_{cc'}(E)$ are up to a stabilizer on $\Gamma^{*\backslash cc'}$, the estimate for the edge boundary of E will differ by the boundary of a Z-stabilizer on the color code. ■

What we have achieved so far is that we have taken a Z error whose syndrome is on vertices and converted it to a valid syndrome on edges for an X error with the same support. We can take this syndrome on edges and recover the face boundary of the error using Theorem 2.

Note that the estimate for the edge boundary returned by Algorithm 2 need not be a valid edge boundary if any of the component decoders fail. So we need a method to check the validity of the edge boundary. Recall that the X syndrome on the toric code is the boundary of a collection of faces. Therefore, it is a union of homologically trivial cycles. We can project the edge boundary E obtained in Algorithm 2 onto each of the minor complexes $\Gamma^{*\backslash c}$. If the edge boundary is valid, then all the homologically nontrivial closed surfaces in $(\Gamma^{*\backslash c})^*$ will intersect with the projected syndrome an even number of times. This test can be carried out in linear time in the number of qubits.

Theorem 5. Estimating face boundary of Z-type errors. Let E be a Z-type error on Γ^* whose edge boundary is estimated using Algorithm 2. If the edge boundary is valid, then we can estimate the face boundary of E up to a Z-stabilizer on the color code using Algorithm 1.

Proof. By Theorem 4, we can estimate the edge boundary of E up to a Z-stabilizer boundary. By Lemma 12, these edges are precisely the syndrome for an X error. By Lemma 10, this X error has the same support as E up to an X-stabilizer. But in a color code, for every X-stabilizer there exists a Z-stabilizer with the same support. Thus the final boundary of E is estimated up to a Z-stabilizer provided all the intermediate estimates from Algorithms 1 and 2 are all up to a stabilizer on the respective 3D toric code decoders. ■

E. Decoding 3D color codes

The projection onto the toric codes allow us to decode the 3D color code. Before we can give the complete decoding algorithm, we need one more component. Following Theorems 2 and 5, we only end up with the boundary of the error. We need to identify the qubits which are in error. The procedure for lifting the boundary to volume is given in Algorithm 3. The main idea behind this algorithm is the fact that the color code is connected and we can partition the qubits into two groups: those inside and those outside of the boundary. The following lemma justifies the procedure in Algorithm 3.

Lemma 13. Lifting the boundary of error. Algorithm 3 will give the smallest collection of 3-cells $\Omega \subseteq \mathsf{C}_3(\Gamma^*)$ such that $\partial\Omega = \mathsf{F}$. If F is not a valid boundary, then the algorithm returns an empty set.

Proof. The algorithm takes as input a collection of faces supposed to enclose a volume. If the faces enclose a volume, we can label all the 3-cells inside and outside the boundary differently. Cells adjacent to each other and enclosed within the same boundary are labeled the same. The algorithm proceeds by labeling a random choice of initial qubit and then proceeds to assign labels to all its adjacent qubits. If two qubits share a face that is not in the boundary F, then they must have the same label because one must cross the boundary to change the label. Two qubits that are adjacent and share a face that is in the boundary must have different labels. The algorithm stops when there are no more qubits to be labeled or when a qubit is assigned contradicting labels, indicating that F is not a boundary. ■

Algorithm 3. Lifting a boundary to a volume

```
Input: Complex $\Gamma^*$, Set of faces $\mathsf{F} \subseteq \mathsf{C}_2(\Gamma^*)$
Output: $\Omega \subseteq \mathsf{C}_3(\Gamma^*)$ such that $\mathsf{F}$ is the boundary of $\Omega$
Set $\Omega = \emptyset; m_\nu = 0$ for all $\nu \in \mathsf{C}_3(\Gamma^*)$            // Initialization
Initialize $\Omega = \{\nu_o\}, m_{\nu_o} = 1$            // For some 3-cell $\nu_o$
while $m_\mu = 0$ for some $\mu \in \mathcal{N}_\nu$ with $m_\nu \neq 0$ do
    for each $\mu \in \mathcal{N}_\nu$ do    // $\mathcal{N}_\nu$ := 3-cells sharing a face with $\nu$
        if $m_\mu = 0$ then
            if $\mu \cap \nu \in \mathsf{F}$ then
                $m_\mu = -m_\nu$        // Qubits on different side of error boundary
            else
                $m_\mu = m_\nu$    // Qubits on same side of error boundary
            end if
            if $m_\mu = 1$ then
                $\Omega = \Omega \cup \{\mu\}$
            end if
        else
            if $\mu \cap \nu \in \mathsf{F}$ and $m_\mu \neq -m_\nu$ then
                $\Omega = \emptyset$; Exit            // F not a valid boundary
            end if
            if $\mu \cap \nu \notin \mathsf{F}$ and $m_\mu \neq m_\nu$ then
                $\Omega = \emptyset$; Exit            // F not a valid boundary
            end if
        end if
    end for
end while
if $|\Omega| > |\mathsf{C}_3 \setminus \Omega|$ then
    $\Omega = \mathsf{C}_3 \setminus \Omega$            // Pick the smaller volume
end if
```

The running time of the algorithm is linear in the number of qubits. The algorithm assumes that all qubits have the same error probability. It can be modified so that it picks the most likely qubits if the error probabilities are not uniform. We now give the decoding procedure for color codes.

Theorem 6. Decoding 3D color codes via 3D toric codes. An error E on a color code can be estimated using Algorithm 4. The estimate will be within a stabilizer on the color code provided the intermediate decoders also estimate within a stabilizer on the respective codes.

Proof. The proof of this theorem is straightforward given our previous results. The decoding is performed separately for X and Z errors and it makes use of the fact that the color code is a CSS code. The algorithm proceeds by estimating the boundary of the X-type errors and Z errors separately. The correctness of these procedures is due to Theorems 2 and 5. Lemma 13 ensures that these boundaries can be lifted to find the qubits that are in error. Decoding failure results if any of the component decoders fail or make logical errors. This will lead to either the failure of the lifting procedure or an invalid edge boundary in line 7 of Algorithm 4. The validity of the edge boundary can be checked by ensuring that the restricted syndrome $\pi_c(\delta E_Z)$ consists of homologically trivial cycles. ■

Remark 2. The decoder could fail if any of the intermediate decoders make a logical error.

The overall running time depends on the running time of the 3D toric code decoders. We can run them independently or we can take advantage of the fact that the errors on component 3D toric codes are correlated.

Algorithm 4. Decoding 3D color codes

```
Input: A 3-colex $\Gamma$ and the syndrome
Output: Error estimate $\hat{E}$
 1: Let $s_X$ be syndrome for $X$ type error $E_X$
 2: Obtain the face boundary $\partial E_X$ from Algorithm 1 with $s_X$ as
    input
 3: Lift the boundary $\partial E_X$ by running Algorithm 3 and obtain $\Omega_X$,
    the support of $E_X$
 4: if $\Omega_X = \emptyset$ and $s_X \neq 0$ then
 5:     Declare decoder failure and exit
 6: end if
 7: Let $s_Z$ be syndrome for $Z$ type error $E_Z$
 8: Estimate the edge boundary $\delta E_Z$ from Algorithm 2 with $s_Z$ as
    input
 9: Check $\pi_c(\delta E_Z)$ consists of homologically trivial cycles only,
    otherwise declare decoding failure and exit.
10: Obtain the face boundary $\partial E_Z$ from Algorithm 1 with $\delta E_Z$ as
    input
11: Lift the boundary $\partial E_Z$ by running Algorithm 3 and obtain $\Omega_Z$,
    the support of $E_Z$
12: if $\Omega_Z = \emptyset$ and $s_Z \neq 0$ then
13:     Declare decoder failure and exit
14: end if
15: Return $\hat{E} = \prod_{\nu\in\Omega_X} X_\nu \prod_{\nu\in\Omega_Z} Z_\nu$
```

The proposed decoder for 3D color codes can also be adapted for decoding the gauge color code. Recall that the gauge operators of the gauge color code are defined on the faces of a 3D cell complex. They are similar to the Z-type operators of the 3D color code. However, they contain an X-type and Z-type gauge operator for each face of the complex. More relevant is the fact that the stabilizers of the gauge color code are defined on the 3-cells of the complex and are similar to the X-type stabilizers of the 3D color code. For each 3-cell, we can define an X-type and Z-type stabilizer, similar to B_v^X in Eq. (4). In other words, decoding the bit flip (or the phase errors) is similar to decoding of the phase-flip errors of the 3D

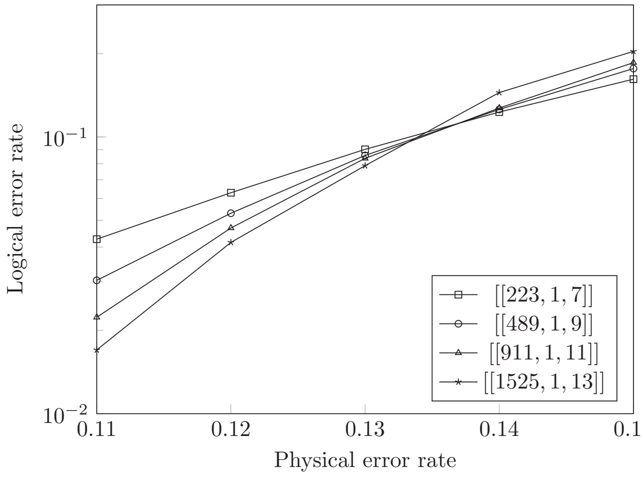

FIG. 5. Performance of the stacked color codes over the bit-flip channel.

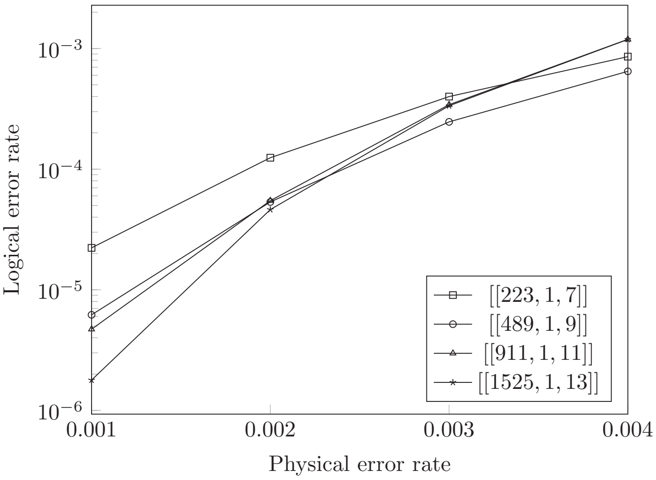

FIG. 6. Performance of the stacked color codes over the phase-flip channel.

color code. Thus we can decode the gauge color code using the same approach used for decoding phase errors in the 3D color code. An alternative approach is to promote all the Z-type gauge operators to be stabilizers and then decode as a 3D color code. This amounts to gauge fixing [20]. Other variations are possible that exploit the structure of subsystem codes.

F. Performance of stacked codes

Recently, a new class of 3D color codes, called stacked codes, was proposed in [11]. These codes have a layered architecture which makes them more amenable to implementation than an arbitrary 3D color code. In this section, we study the performance of these codes using the projection map we proposed. Note that the stacked color code has boundaries. This implies the toric codes onto which it is projected also contain boundaries. It turns out these toric codes encode zero logical qubits and can be decoded by generalizing the ideas of [21]. Taking these into account, we implemented a slightly modified version of the decoder given in Algorithm 4. These details, including the 3D toric code decoding, will be discussed elsewhere.

The performance is shown in Fig. 5 for bit-flip errors and Fig. 6 for phase-flip errors. We obtain a (code) threshold of $\gtrsim 13.3\%$ for bit-flip errors. For phase-flip errors, we did not obtain a clear threshold. The performance of the decoder for phase-flip errors is limited by certain stabilizer generators of the stacked color code whose size grows with the code. One approach to improve the performance would be to exploit the correlations among the various copies of toric codes. Alternatively, it is possible that stacked codes do not have a threshold. This absence of threshold could be due to the growing size of the certain stabilizers. A similar observation was made concerning these stabilizers in [11,12]. The latter also argued that such codes may not have a threshold.

IV. CONCLUSION

In this paper, we have shown how to project 3D color codes onto 3D toric codes. Our work provides an alternative perspective to that of [10] who also proposed a map between color codes and toric codes. Our approach emphasizes the topological properties of color codes. The mapping was motivated by the problem of decoding 3D color codes. The toric codes thus obtained are linearly related to the size of the parent color code. So if the corresponding 3D toric codes can be decoded efficiently, then so can the 3D color codes by projecting them onto 3D toric codes using our map. Using our results, we studied the performance of the stacked color codes. Stacked color code requires the decoding of a 3D toric code encoding zero logical qubits. This map also finds application in the decoding of gauge color codes [14,22,23] by projecting onto 3D toric codes. One direction for research would be to incorporate measurement errors as in [24]. Another avenue for further research is to study the possible use of this map for fault-tolerant quantum computing protocols.

ACKNOWLEDGMENT

The authors would like to thank the anonymous referees for their comments and suggestions. This research was supported by the Centre for Industrial Consultancy & Sponsored Research, IIT Madras.

[1] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **98**, 160502 (2007).

[2] C. Castelnovo and C. Chamon, Phys. Rev. B **78**, 155120 (2008).

[3] A. Hamma, P. Zanardi, and X.-G. Wen, Phys. Rev. B **72**, 035307 (2005).

[4] L. Ioffe and M. Mézard, Phys. Rev. A **75**, 032345 (2007).

[5] P. Brooks and J. Preskill, Phys. Rev. A **87**, 032310 (2013).

[6] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, Proc. R. Soc. London, Ser. A **465**, 1645 (2009).

[7] B. Yoshida, Ann. Phys. **326**, 15 (2011).

[8] H. Bombin, G. Duclos-Cianci, and D. Poulin, New J. Phys. **14**, 073048 (2012).

[9] N. Delfosse, Phys. Rev. A **89**, 012317 (2014).

[10] A. Kubica, B. Yoshida, and F. Pastawski, New J. Phys. **17**, 083026 (2015).

[11] T. Jochym-O'Connor and S. D. Bartlett, Phys. Rev. A **93**, 022323 (2016).

[12] C. Jones, P. Brooks, and J. Harrington, Phys. Rev. A **93**, 052332 (2016).

[13] S. Bravyi and A. Cross, arXiv:1509.03239.

[14] B. J. Brown, N. H. Nickerson, and D. E. Browne, Nat. Commun. **7**, 12302 (2016).

[15] We thank an anonymous referee for pointing this out to us.

[16] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, IEEE Trans. Inf. Theory **44**, 1369 (1998).

[17] D. Gottesman, Ph.D. thesis, California Institute of Technology, 1997, arXiv:quant-ph/9705052.

[18] H. Bombin and M. A. Martin-Delgado, Phys. Rev. B **75**, 075103 (2007).

[19] Usually the support is defined as a set; here it is convenient to define as a linear combination.

[20] We thank an anonymous referee for this observation.

[21] N. Delfosse and N. H. Nickerson, arXiv:1709.06218.

[22] H. Bombin, Phys. Rev. X **5**, 031043 (2015).

[23] H. Bombín, New J. Phys. **17**, 083002 (2015).

[24] A. M. Stephens, arXiv:1402.3037.