

Projecting 3D color codes onto 3D toric codes

Arun B. Alosious* and Pradeep Kiran Sarvepalli†

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

(Dated: October 25, 2016)

Toric codes and color codes are two important classes of topological codes. Kubica, Yoshida, and Pastawski showed that any D -dimensional color code can be mapped to a finite number of toric codes in D -dimensions. In this paper we propose an alternate map of 3D color codes to 3D toric codes with a view to decoding 3D color codes. Our approach builds on Delfosse's result 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.

I. INTRODUCTION

Three dimensional (3D) toric codes [1, 2] and color codes [3] are topological quantum codes defined on 3D lattices. Like their 2D counterparts, they are also Calderbank-Shor-Steane (CSS) codes [4] where the bit flip and phase errors can be corrected independently. Three dimensional topological codes are inherently asymmetrical in their error correcting capabilities for the bit flip errors and the phase flip errors, which is perhaps one of the reasons why they have not received as much attention as their 2D analogues. However, the growing interest in asymmetric error models [5–7] motivates us to study them in closer detail.

Another reason for studying 3D codes, specifically the 3D color codes, comes from the fact that in some ways they are also richer than 2D codes. Certain 3D color codes also possess a transversal non-Clifford gate [3]. This is not possible for 2D codes or 3D toric codes. While the 3D toric code on a cubic lattice has been studied in [1, 2, 8], the general case of an arbitrary lattice has not been investigated as much.

For these codes to be useful for fault tolerant quantum computing it is necessary to develop efficient decoding algorithms. However, there appears to be no previous work on the decoding of 3D color codes. While we do not solve this problem in this paper, we make some progress in the decoding of 3D color codes by reducing it to the decoding of the 3D toric codes. Errors corresponding to chains in the lattice can be easily decoded on a 3D toric code. But efficient decoders are not known for errors corresponding to surfaces except in the case of cubic lattice. In this case a decoder similar to the decoder for the 4D toric code in [9] can be used.

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. The work most similar to ours is that of Kubica, Yoshida, and Pastawski [10] who showed, among other things, that the 3D color code can be mapped to three copies of 3D toric codes. Our results give a different mapping from the color code to the toric codes. Our map also preserves the CSS nature of the color code. We project the

X errors and Z errors onto different sets of toric codes unlike [10] which employs just one set of toric codes. Their map also implies that a 3D color code can be decoded via 3D toric codes. The question of which map is better for decoding is not yet known. This will be investigated in a later work.

Another work related to ours is that of Delfosse who showed that 2D color code can be projected onto surface codes [11] using the machinery of chain complexes derived from hypergraphs. Our results generalize his approach to 3D. We take a somewhat simpler approach and do not explicitly make use of chain complexes based on hypergraphs. In passing we mention that similar mappings were known in 2D [12–14]. Their approaches can also be generalized to 3D.

The paper is structured as follows. In Section II we give a brief review of 3D toric codes and color codes. In the subsequent 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 novel 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 [15, 16].

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:

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

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:

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

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.

* alosious.sp@gmail.com

† pradeep@ee.iitm.ac.in

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

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

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 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, [3]. A 3D color code is a topological stabilizer code constructed from a 3-colex. The stabilizer generators of the color code are given as

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

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, [3, 17].

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^* .

$$B_\nu^X = \prod_{\nu: v \in \nu} X_\nu \text{ and } B_e^Z = \prod_{\nu: e \in \nu} Z_\nu \quad (5)$$

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 $C_i(\Gamma)$ and the i -dimensional cells of color c as $C_i^c(\Gamma)$.

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 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 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 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 the 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 the edges of this remaining face carry nonzero syndromes. These edges are precisely the boundary of that face. Similarly 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.

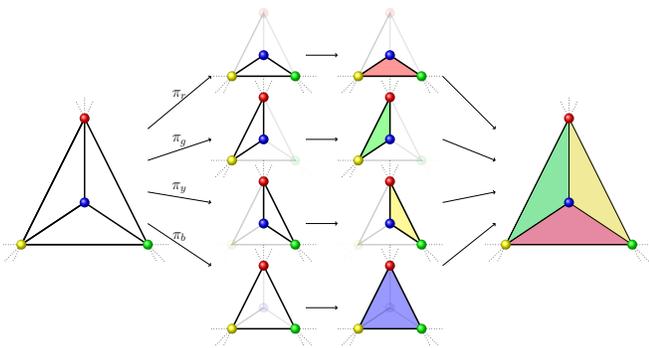


Figure 1: We can recover the boundary of a tetrahedron from the edges by reconstructing the faces and then combining the faces.

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 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.
- (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 lesser 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 piece of the boundary we are looking for. Deleting all the 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 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

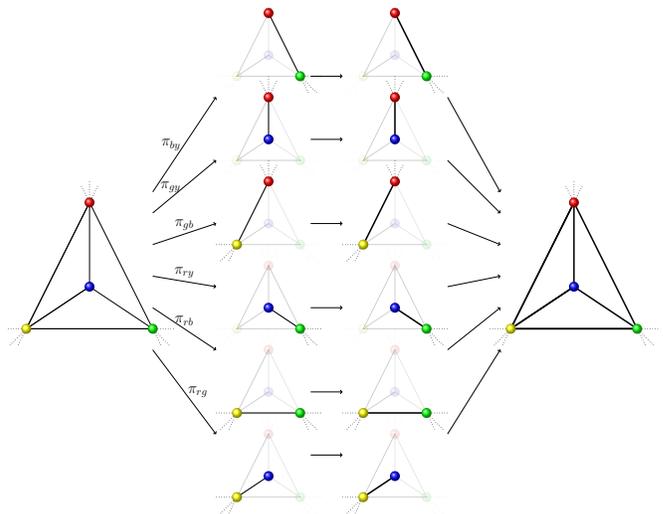


Figure 2: We can recover the boundary of a tetrahedron from the vertices by first recovering the edges in the boundary of the tetrahedron. With the edges recovered we can proceed as illustrated in Fig.1 to recover the boundary of the tetrahedron.

decode Z -type errors in 3D toric codes.

The procedures we outlined 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

$$\pi_c(\Gamma^*) = \Gamma^* \setminus^c \quad (6)$$

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 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.

$$\pi_c(v) = v \text{ if } v \in C_0(\Gamma^*) \setminus C_0^c(\Gamma^*) \quad (7a)$$

$$\pi_c(e) = e \text{ if } e \in C_1(\Gamma^*) \setminus C_1^{cc'}(\Gamma^*) \quad (7b)$$

$$\pi_c(f) = f \text{ if } f \in C_2^c(\Gamma^*) \quad (7c)$$

$$\pi_c(\nu) = f_{\ni \nu}^c = \partial \nu \cap C_2^c(\Gamma^*); \nu \in C_3(\Gamma^*) \quad (7d)$$

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

$$\partial(\nu) = \sum_c \pi_c(\nu) = \sum_c f_{\ni \nu}^c, \quad (8)$$

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' . \square

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 $\nu_v \in C_3(\Gamma^* \setminus c)$ 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

$$\partial(\nu_v) = \sum_{\nu: v \in \nu} \partial(\nu) \stackrel{(a)}{=} \sum_{\nu: v \in \nu} f_{\ni \nu}^c + \sum_{\nu: v \in \nu} \sum_{i \neq c} f_{\ni \nu}^i \quad (9)$$

$$\stackrel{(b)}{=} \sum_{\nu: v \in \nu} f_{\ni \nu}^c, \quad (10)$$

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. \square

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

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

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

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

As in case of $\Gamma^* \setminus c$, the vertices and edges of $\Gamma^{*\setminus cc'}$ are a subset of $\Gamma^* \setminus cc'$ 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 deleted. Thus only edges between d and d' colored vertices will remain. These edges are colored dd' . \square

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

$$\mathcal{V}_e = \{\nu \mid e \in \nu\} \quad (13)$$

Let $e_{\ni \nu}^{dd'}$ be the unique dd' -edge in ν . For each of the qubits in \mathcal{V}_e , exactly one dd' -edge will survive in $\Gamma^{*\setminus cc'}$. No two qubits in \mathcal{V}_e share the same dd' -edge. Further, the surviving dd' edges will form a cycle in $\Gamma^{*\setminus cc'}$, see Fig. 3 for illustration. This can be seen as follows. Let $\mathcal{V}_{cc'}(\mathcal{V}_e)$, denote the vertices of \mathcal{V}_e that remain in $\Gamma^{*\setminus cc'}$. 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 $\Gamma^*\setminus cc'$. The boundary of this face is the collection of the dd' -edges belonging to the qubits incident on e , alternatively,

$$\partial f_e = \sum_{\nu \in \mathcal{V}_e} e_{\ni \nu}^{dd'}. \quad (14)$$

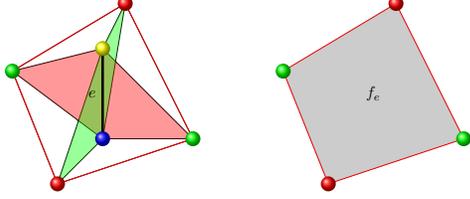


Figure 3: Face in $\Gamma^*\setminus cc'$. Consider the by -edge (in bold) and the qubits incident on it. Since only r, g -vertices (i.e. red and green) will survive in $\Gamma^*\setminus by$, each of these surviving vertices will have exactly two rg -edges incident on them. Therefore these rg -edges form a cycle.

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 $\Gamma^*\setminus cc'$ are in one to one correspondence with cc' edges of Γ^* .*

With faces of $\Gamma^*\setminus cc'$ defined, the 3-cells of $\Gamma^*\setminus cc'$ can be identified. Let v be a c or c' -vertex. Consider the edges incident on v . Each of these edges corresponds to a face in $\Gamma^*\setminus cc'$, by Lemma 5. We define the volume enclosed by these faces, such that it contains v , to be a 3-cell of $\Gamma^*\setminus cc'$. Since every such v leads to a 3-cell in $\Gamma^*\setminus cc'$, 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 $\Gamma^*\setminus cc'$ are in one to one correspondence with vertices in $C_0^c(\Gamma^*) \cup C_0^{c'}(\Gamma^*)$.*

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

$$\pi_{cc'}(v) = v \text{ if } v \in C_0(\Gamma^*) \setminus (C_0^c(\Gamma^*) \cup C_0^{c'}(\Gamma^*)) \quad (15a)$$

$$\pi_{cc'}(e) = e \text{ if } e \in C_1^{dd'}(\Gamma^*); d, d' \notin \{c, c'\} \quad (15b)$$

$$\pi_{cc'}(\nu) = e_{\ni \nu}^{dd'} = \partial(\pi_c(\nu)) \cap C_1^{dd'}(\Gamma^*); \nu \in C_3(\Gamma^*) \quad (15c)$$

where $e_{\ni \nu}^{dd'}$ 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 $\Gamma^*\setminus c$ or Γ^* will survive in $\Gamma^*\setminus cc'$. 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 $\Gamma^*\setminus c$.

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

$$\delta \nu = \sum_{cc'} \pi_{cc'}(\nu) = \sum_{cc'} e_{\ni \nu}^{dd'} \quad (16)$$

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^*	$\Gamma^*\setminus c$	$\Gamma^*\setminus cc'$
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 7. *Let Γ be a 3-colex with v vertices, $e = 2v$ edges, $f_{cc'}$ cc' -faces, 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 $\Gamma^*\setminus c$ and $\Gamma^*\setminus cc'$ where $c, c', d, d' \in \{r, b, g, y\}$ are distinct.*

	Γ	Γ^*	$\Gamma^*\setminus c$	$\Gamma^*\setminus cc'$
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 $\Gamma^*\setminus c$ 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 $\Gamma^*\setminus i$ are those that are incident on vertices other than i -vertices.

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

Remark 1. *The minor complexes defined here are the duals of the shrunk complexes defined in [17]. For example, $\Gamma^*\setminus b$ is the exactly the dual of b -shrunk complex and $\Gamma^*\setminus ry$ is the dual of 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

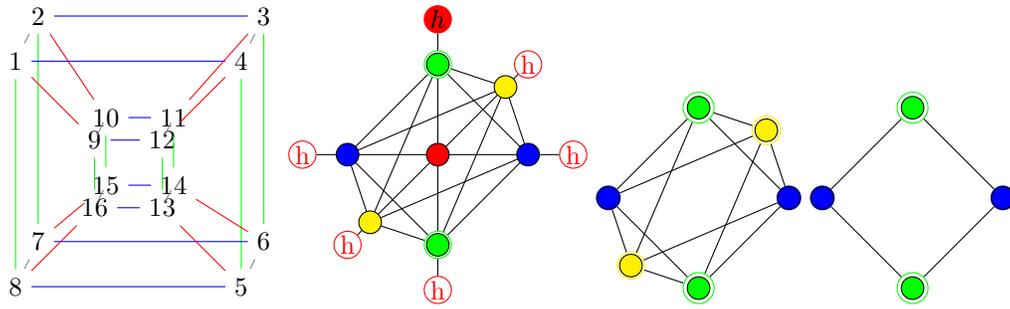


Figure 4: A 3-colex Γ and its dual Γ^* ; i -cells of Γ^* correspond to the $3 - i$ cells of Γ . The minor complexes $\Gamma^{*\setminus r}$ and $\Gamma^{*\setminus ry}$ are also shown.

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 $\Gamma^{*\setminus c}$. Thus we can project errors from Γ^* to $\Gamma^{*\setminus c}$.

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

$$\pi_c(X_\nu) = X_{\pi_c(\nu)}, \quad (17)$$

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

A consequence of Eq. (17), together with linearity of π_c , is that for two adjacent qubits ν_1 and ν_2 sharing a c -colored face f , we have $\pi_c(X_{\nu_1}X_{\nu_2}) = X_{\pi_c(\nu_1)}X_{\pi_c(\nu_2)} = I$, where we used the fact that $\pi_c(\nu_1) = \pi_c(\nu_2) = f$. In other words, a c -face common to two qubits in error corresponds to an error free qubit in $\Gamma^{*\setminus c}$.

The syndrome corresponding to bit flip errors is associated to edges of Γ^* . In $\Gamma^{*\setminus c}$ 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

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

This is consistent with Eq. (7b).

At this point we have qubits living on the faces and syndromes on the edges of $\Gamma^{*\setminus c}$ just as we would have in a 3D toric code. But it needs to be shown that indeed that 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 $\Gamma^{*\setminus c}$. Consider an edge e in $\Gamma^{*\setminus c}$. 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 $\Gamma^{*\setminus c}$. Thus in the 3D toric code associated to $\Gamma^{*\setminus c}$, 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 8 (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 $\Gamma^{*\setminus c}$. Then the error $\pi_c(E)$ in $\Gamma^{*\setminus c}$ produces the syndrome $\pi_c(s)$ in the toric code associated to $\Gamma^{*\setminus c}$.*

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

Corollary 9. *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 $\Gamma^{*\setminus c}$. Then $\pi_c(s)$ is a valid syndrome for (an X error on) the toric code on $\Gamma^{*\setminus c}$.*

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

Lemma 10. *Let v be a c -vertex in Γ^* and ν_v be the 3-cell in $\Gamma^{*\setminus c}$ 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 $\Gamma^{*\setminus c}$.*

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

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

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

$$\stackrel{(a)}{=} \prod_{f \in \partial(\nu_v)} X_f = B_{\nu_v}^X. \quad (21)$$

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 $\pi_{c'}(B_v^X) = \prod_{\nu: v \in \nu} X_{\pi_c(\nu)} = I$. \square

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

Corollary 11. *Let \bar{S} be an X -type stabilizer on the color code. Then $\pi_c(\bar{S})$ is an X -stabilizer on the toric code defined by $\Gamma^{*\setminus c}$. Conversely for every X -stabilizer S on $\Gamma^{*\setminus c}$, there exists an X -stabilizer generator \bar{S} on Γ^* such that $\pi_c(\bar{S}) = S$ and $\pi_{c'}(\bar{S}) = I$.*

Proof. The first statement is a consequence of Lemma 10 as $\{B_v^X\}$ generate all X -stabilizers of the color code. Further, $\{\pi_c(B_v^X)\} = \{B_{\nu_v}^X\}$, where $v \in C_0^c(\Gamma^*)$. By Lemma 3, $\{B_{\nu_v}^X\}$ generate the X -type stabilizers of the toric code on $\Gamma^{*\setminus c}$. Thus the converse also holds. \square

Let the support of an error E be defined[18] as

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

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

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

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,

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

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

Lemma 12 (X error boundary). *Using the same notation as in Theorem 8, the boundary of an error E in Γ^* is*

$$\partial E = \sum_c \text{supp}(\pi_c(E)). \quad (25)$$

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

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

$$= \sum_c \sum_{\nu: E_\nu \neq I} \pi_c(\nu) \stackrel{(c)}{=} \sum_c \sum_{\nu: E_\nu \neq I} \text{supp}(\pi_c(E_\nu)) \quad (27)$$

$$\stackrel{(d)}{=} \sum_c \text{supp}(\pi_c(E)) \quad (28)$$

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 $\pi_c(E_\nu)$; (d) follows from Eq. (23) and completes the proof. \square

By Lemma 12, the boundary of an error E can be broken down into four surfaces each lying in a separate minor complex $\Gamma^{*\setminus c}$. 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 13 (X error boundary modulo stabilizer). *Let E be an error on Γ^* . Let S_c be X -stabilizer generators on $\Gamma^{*\setminus c}$. Then $\partial E + \sum_c \text{supp}(S_c)$ is the boundary of ES , for some X -stabilizer S on Γ^* i.e.*

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

Proof. Since S_c is a stabilizer generator, by Corollary 11, there exists an X -stabilizer \bar{S}_c on Γ^* such that $\pi_c(\bar{S}_c) = S_c$ and $\pi_{c'}(\bar{S}_c) = I$. By Lemma 12, $\partial(\bar{S}_c) = \text{supp}(S_c)$. Then using $\partial EF = \partial E + \partial F$, we obtain $\partial E \bar{S}_c = \partial E + \partial \bar{S}_c$. Repeating this for all c we have $\partial(E \prod_c \bar{S}_c) = \partial E + \sum_c \partial \bar{S}_c$. Letting $\prod_c \bar{S}_c = S$, we can write this as $\partial(ES) = \partial E + \sum_c \text{supp}(S_c)$ as claimed. \square

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

Theorem 14 (Estimating face boundary of X errors). *The boundary an X -error E on the dual of 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 $\Gamma^{*\setminus c}$, 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 12 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 $\Gamma^{*\setminus c}$. By Lemma 8, the syndrome of $\pi_c(E)$ is the restriction of the syndrome on Γ^* . Therefore, $\pi_c(E)$ can be estimated by decoding on $\Gamma^{*\setminus c}$. By Lemma 13, if the estimate for $\pi_c(E)$ is equivalent up to a stabilizer on $\Gamma^{*\setminus c}$, we can obtain the boundary of E up to the boundary of a stabilizer on the color code. \square

Algorithm 1: Estimating (face) boundary of X type error

Input: A 3-colex Γ , Syndrome of an X error E

Output: F , an estimate of ∂E where $F \subseteq C_2(\Gamma^*)$

```

1: for each  $c \in \{r, b, g, y\}$  do
2:   for each edge  $e$  in  $\Gamma^{*\setminus c}$  do // syndrome projection
3:      $s_{\pi_c(e)} = s_e$  //  $s_e$  is syndrome on edge  $e$ 
4:   end for
5:   Using the projected syndrome on  $\Gamma^{*\setminus c}$ , estimate the error  $F_c$ 
   by any 3D toric code decoder for  $X$  errors
6: end for
7: Return  $F = \bigcup_c F_c$ 

```

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 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 $\Gamma^* \setminus cc'$. Recall that the edges of $\Gamma^* \setminus cc'$ are associated with qubits. So we project the errors on cell ν to $\pi_{cc'}(\nu)$ which is an edge in $\Gamma^* \setminus cc'$. We define

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

Let syndrome on $v \in C_0(\Gamma^*)$ be s_v . Then we define the syndrome on $v \in \Gamma^* \setminus cc'$ as

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

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 $\Gamma^* \setminus cc'$.

Theorem 15 (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 $\Gamma^* \setminus cc'$ produces the syndrome $\pi_{cc'}(s)$.*

Proof. We only need to show the theorem for vertices $v \in \Gamma^* \setminus cc'$. 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 $\{e_1, \dots, e_m\}$ be these dd' -edges. Then we can write

$$\{\nu : v \in \nu\} = \bigcup_i \{\nu : v, e_i \in \nu\} \quad (32)$$

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

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

The qubits containing e_i are projected onto the dd' -edge e_i in the minor complex $\Gamma^* \setminus cc'$ 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 $\Gamma^* \setminus cc'$ is $\pi_{cc'}(E) = \bigoplus_{i=1}^m r_i = s_v = \pi_{cc'}(s_v)$ as required. \square

Corollary 16 (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 $\Gamma^* \setminus cc'$.*

Proof. From Theorem 15, we see that $\pi_{cc'}(s)$ coincides with the syndrome produced by a Z error on $\Gamma^* \setminus cc'$. Hence, $\pi_{cc'}(s)$ must be a valid syndrome for a Z error on a 3D toric code. \square

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

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

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

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 17 (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

$$s_e \neq 0 \text{ iff. } |\{\nu \in \Omega \mid e \in \nu\}| \text{ is odd} \quad (36)$$

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 . \square

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 18. *Suppose S is a Z -stabilizer on $\Gamma^* \setminus cc'$, then there exists a Z -stabilizer \bar{S} in Γ^* such that $\pi_{cc'}(\bar{S}) = S$ and $\pi_{cd}(\bar{S}) = I$ for $xy \neq cc'$.*

Proof. A Z -stabilizer in $\Gamma^* \setminus cc'$ generated by the face type stabilizers in $\Gamma^* \setminus cc'$. Therefore, it suffices to consider when S is a face type stabilizer. By Lemma 5, the faces of $\Gamma^* \setminus cc'$ are in correspondence with the cc' -edges of Γ^* . So we can let $S = B_{f_e}^Z$ for some face f_e in $\Gamma^* \setminus cc'$ 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

$$\pi_{cc'}(B_e^Z) = \prod_{\nu: e \in \nu} Z_{\pi_{cc'}(\nu)} \stackrel{(a)}{=} \prod_{\nu: e \in \nu} Z_{e_{\bigoplus \nu}^{dd'}}, \quad (37)$$

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

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 $\pi_{cd}(B_e^Z) = \prod_{\nu: e \in \nu} Z_{e_{\nu}^{c'd'}} = I$. Similar arguments can be used to show that $\pi_{cd}(\bar{S}) = I$ for other $xy \neq cc'$. We omit the details. \square

Lemma 19 (Edge boundary modulo stabilizer). *Let E be a Z -type error on the color code and S a Z -stabilizer on $\Gamma^* \setminus cc'$. Then $\delta E + \sum_{e: S_e \neq I} e$ is the edge boundary of $E\bar{S}$ for some Z -stabilizer \bar{S} on Γ^* .*

Proof. By Lemma 18, there exists a Z stabilizer \bar{S} on Γ^* such that $\pi_{cc'}(\bar{S}) = S$ and $\pi_{xy}(\bar{S}) = I$ for $xy \neq cc'$. Therefore,

$$I = \pi_{xy}(\bar{S}) = \pi_{xy} \left(\prod_{\nu: \bar{S}_\nu \neq I} Z_\nu \right) = \prod_{\nu: \bar{S}_\nu \neq I} Z_{\pi_{xy}(\nu)} \quad (39)$$

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

$$\sum_{\nu: \bar{S}_\nu \neq I} \pi_{xy}(\nu) = 0 \quad (40)$$

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

$$\prod_{e: S_e \neq I} Z_e = \pi_{cc'} \left(\prod_{\nu: \bar{S}_\nu \neq I} Z_\nu \right) = \prod_{\nu: \bar{S}_\nu \neq I} Z_{\pi_{cc'}(\nu)} \quad (41)$$

Eq. (41), implies that

$$\sum_{e: S_e \neq I} e = \sum_{\nu: \bar{S}_\nu \neq I} \pi_{cc'}(\nu) \quad (42)$$

$$= \sum_{\nu: \bar{S}_\nu \neq I} \pi_{cc'}(\nu) + \sum_{xy \neq cc'} \sum_{\nu: \bar{S}_\nu \neq I} \pi_{xy}(\nu) \quad (43)$$

$$= \delta \bar{S} \quad (44)$$

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

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 20 (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 $\Gamma^* \setminus cc'$.*

Proof. By Corollary 16, the restriction of the syndrome of E is a valid syndrome on $\Gamma^* \setminus cc'$. By Theorem 15, $\pi_{cc'}(E)$ has the same syndrome as the restriction and $\pi_{cc'}(E)$ can be estimated using a 3D toric decoder on $\Gamma^* \setminus cc'$. We can reconstruct the edge boundary from $\pi_{cc'}(E)$ using the Eq. (35). By Lemma 19 if the estimates for $\pi_{cc'}(E)$ are upto a stabilizer on $\Gamma^* \setminus cc'$, the estimate for edge boundary of E will differ by the boundary of a Z -stabilizer on the color code. \square

Algorithm 2: Estimating (edge) boundary of Z type error

Input: A 3-colex Γ , Syndrome of a Z error E

Output: E , an estimate of the edge boundary δE , where $E \subseteq C_1(\Gamma^*)$

```

1: for each  $c, c' \in \{r, b, g, y\}$  do
2:   for each vertex  $v$  in  $\Gamma^* \setminus cc'$  do //syndrome projection
3:      $s_{\pi_{cc'}(v)} = s_v$  //  $s_v$  is syndrome on vertex  $v$ 
4:   end for
5:   Estimate the error  $E_{cc'}$  using any 3D toric code decoder for  $Z$ 
   errors on  $\Gamma^* \setminus cc'$ 
6: end for
7: Return  $E = \sum_{c, c'} E_{cc'}$ 

```

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 14.

Note that the estimate for 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 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 $\Gamma^* \setminus c$. If the edge boundary is valid, then all the homologically nontrivial closed surfaces in $(\Gamma^* \setminus c)^*$ 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 21 (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 20 we can estimate the edge boundary of E up to a Z -stabilizer boundary. By Lemma 19 these edges are precisely the syndrome for an X error. By Lemma 17, this X error has the same support as E up to a 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 & 2 are all up to a stabilizer on the respective 3D toric code decoders. \square

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 14 and 21, 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.

Algorithm 3: Lifting a boundary to a volume

Input: Complex Γ^* , Set of faces $F \subseteq C_2(\Gamma^*)$
Output: $\Omega \subseteq C_3(\Gamma^*)$ such that F is the boundary of Ω

- 1: Set $\Omega = \emptyset$; $m_\nu = 0$ for all $\nu \in C_3(\Gamma^*)$ // Initialization
- 2: Initialize $\Omega = \{\nu_o\}$, $m_{\nu_o} = 1$ // For some 3-cell ν_o
- 3: **while** $m_\mu = 0$ for some $\mu \in \mathcal{N}_\nu$ with $m_\nu \neq 0$ **do**
- 4: **for each** $\mu \in \mathcal{N}_\nu$ **do** // $\mathcal{N}_\nu := 3$ -cells sharing a face with ν
- 5: **if** $m_\mu = 0$ **then**
- 6: **if** $\mu \cap \nu \in F$ **then**
- 7: $m_\mu = -m_\nu$ //Qubits on different side of error boundary
- 8: **else**
- 9: $m_\mu = m_\nu$ //Qubits on same side of error boundary
- 10: **end if**
- 11: **if** $m_\mu = 1$ **then**
- 12: $\Omega = \Omega \cup \{\mu\}$
- 13: **end if**
- 14: **else**
- 15: **if** $\mu \cap \nu \in F$ and $m_\mu \neq -m_\nu$ **then**
- 16: $\Omega = \emptyset$; Exit // F not a valid boundary
- 17: **end if**
- 18: **if** $\mu \cap \nu \notin F$ and $m_\mu \neq m_\nu$ **then**
- 19: $\Omega = \emptyset$; Exit // F not a valid boundary
- 20: **end if**
- 21: **end if**
- 22: **end for**
- 23: **end while**
- 24: **if** $|\Omega| > |C_3 \setminus \Omega|$ **then**
- 25: $\Omega = C_3 \setminus \Omega$ //Pick the smaller volume
- 26: **end if**

Lemma 22 (Lifting the boundary of error). *Algorithm 3 will give the smallest collection of 3-cells $\Omega \subseteq C_3(\Gamma^*)$ such that $\partial\Omega = F$. 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 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. \square

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 23 (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 14 and 21. Lemma 22 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 lifting procedure or an invalid edge boundary in line 7. 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. \square

Algorithm 4: Decoding 3D color codes

Input: A 3-colex Γ 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 ∂E_X from Algorithm 1 with s_X as input
- 3: Lift the boundary ∂E_X by running Algorithm 3 and obtain Ω_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 δE_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 ∂E_Z from Algorithm 1 with δE_Z as input
- 11: Lift the boundary ∂E_Z by running Algorithm 3 and obtain Ω_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 $\hat{E} = \prod_{\nu \in \Omega_X} X_\nu \prod_{\nu \in \Omega_Z} Z_\nu$

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.

IV. CONCLUSION

In this paper we have shown how to project 3D color codes onto 3D toric codes. The projection 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 3D toric codes on arbitrary lattices can be decoded efficiently, then so can the 3D color codes by projecting them onto 3D toric codes using our map. 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. At this point there is no data available for performance of the decoders based on our map as well as the map due to [10]. One difficulty to compare the performance of the decoders arising out of these maps, as we mentioned earlier, is that we do not have efficient decoders for 3D toric code on an arbitrary lat-

tice. (Decoders are known only for the cubic lattice.) So an open question for further research is to study the decoding of 3D toric codes. This map could also find application in the decoding of gauge color codes [19–21] by projecting onto 3D toric codes. Another avenue for further research is to study the possible use of this map for fault tolerant quantum computing protocols.

Acknowledgement This research was supported by a grant from Center for Industrial Consultancy and Sponsored Research.

-
- [1] C. Castelnovo and C. Chamon, Phys. Rev. B **78**, 155120 (2008).
 [2] A. Hamma, P. Zanardi, and X.-G. Wen, Phys. Rev. B **72**, 035307 (2005).
 [3] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **98**, 160502 (2007).
 [4] A. R. Calderbank and P. W. Shor, Phys. Rev. A **54**, 1098 (1996).
 [5] L. Ioffe and M. Mézard, Phys. Rev. A **75**, 032345 (2007).
 [6] P. Brooks and J. Preskill, Phys. Rev. A **87**, 032310 (2013).
 [7] P. K. Sarvepalli, A. Klappenecker, and M. Rötteler, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **465**, 1645 (2009).
 [8] S. Bravyi, B. Leemhuis, and B. M. Terhal, Annals of Physics **326**, 839 (2011).
 [9] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Journal of Mathematical Physics **43**, 4452 (2002).
 [10] A. Kubica, B. Yoshida, and F. Pastawski, New Journal of Physics **17**, 083026 (2015).
 [11] N. Delfosse, Physical Review A **89**, 012317 (2014).
 [12] B. Yoshida, Annals of Physics **326**, 15 (2011), january 2011 Special Issue.
 [13] A. Bhagoji and P. Sarvepalli, 2015 IEEE International Symposium on Information Theory (ISIT) , 1109 (2015), arXiv:1503.03009.
 [14] H. Bombin, G. Duclos-Cianci, and D. Poulin, New Journal of Physics **14**, 073048 (2012).
 [15] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. A. Sloane, IEEE Transactions on Information Theory **44**, 1369 (1998).
 [16] D. Gottesman, *Stabilizer Codes and Quantum Error correction*, Phd thesis, California Institute of Technology (1997), arXiv:9705052v1 [arXiv:quant-ph].
 [17] H. Bombin and M. a. Martin-Delgado, Physical Review B - Condensed Matter and Materials Physics **75** (2007), 10.1103/PhysRevB.75.075103, arXiv:0607736 [cond-mat].
 [18] Usually the support is defined as a set, here it is convenient to define as a linear combination.
 [19] B. J. Brown, N. H. Nickerson, and D. E. Browne, , 4 (2015), arXiv:1503.08217.
 [20] H. Bombin, Physical Review X **5**, 1 (2015).
 [21] H. Bombin, New Journal of Physics **17**, 083002 (2015).