Universal transversal gates with color codes: A simplified approach

Aleksander Kubica and Michael E. Beverland

Universal transversal gates with color codes — a simplified approach

Aleksander Kubica and Michael E. Beverland

Institute for Quantum Information & Matter, California Institute of Technology, Pasadena CA 91125, USA

We provide a simplified, yet rigorous presentation of the ideas from Bombín's paper *Gauge Color Codes* [arXiv:1311.0879v3]. Our presentation is self-contained, and assumes only basic concepts from quantum error correction. We provide an explicit construction of a family of color codes in arbitrary dimensions and describe some of their crucial properties. Within this framework, we explicitly show how to transversally implement the generalized phase gate $R_n = \mathrm{diag}(1, e^{2\pi i/2^n})$, which deviates from the method in the aforementioned paper, allowing an arguably simpler proof. We describe how to implement the Hadamard gate H fault-tolerantly using code switching. In three dimensions, this yields, together with the transversal CNOT, a fault-tolerant universal gate set $\{H, \mathrm{CNOT}, R_3\}$ without state-distillation.

I. INTRODUCTION

To build a fully functioning quantum computer, it is necessary to encode quantum information to protect it from noise. In physical systems, one expects noise to act locally. Therefore, *topological codes* [1–4], which naturally protect against local errors, represent our best hope for storing quantum information. However, a quantum computer must also be capable of processing this information. This motivates the search for topological codes allowing the implementation of a set of gates which (i) can operate in the presence of typical noise without corrupting the stored information, and (ii) can perform any computation on the encoded information. A theoretical framework has been developed around these ideas — a gate which is *fault-tolerant* does not propagate typical errors into uncorrectable errors [5, 6], and therefore satisfies (i). A set of gates which is *universal* can generate any unitary on the code space with arbitrary precision [7, 8], and therefore satisfies (ii).

The known methods of implementing a universal, fault-tolerant gate set in topological codes typically require an enormous amount of overhead. For instance, magic state distillation [9] with the two-dimensional toric code requires many additional ancilla qubits [10], whereas computing by braiding non-abelian anyons [1, 11] requires additional time to move anyons around macroscopic loops [12]. These forms of overhead can make quantum processing orders of magnitude less efficient than storage alone in topological codes. This may render such approaches impractical given the experimental difficulty of scaling up quantum hardware [10, 13, 14]. In this paper we focus on a new construction by Bombín [15], for a universal fault-tolerant gate set with topological color codes. This approach avoids the types of overhead mentioned above. However, a lattice of at least three dimensions is required, limiting the construction's practicality, and there may be other sources of overhead (related to gauge fixing).

Following Bombín's construction, we use the simplest form of fault-tolerant gate — the *transversal gate*, which is a code-space preserving unitary composed of separate unitaries applied to each physical qubit. However, according to a no-go theorem by Eastin and Knill [16], for any code which protects against arbitrary single-qubit errors, the set of transversal gates forms a finite group and therefore cannot be universal. Some recent approaches to circumvent this no-go theorem in order to implement a universal gate set with transversal gates have been put forward [17–19].

In Ref. [15], Bombín applies the approach of *gauge fixing* [18, 19] to color codes on a d-dimensional lattice. Color codes were first introduced in two dimensions by Bombín and Martin-Delgado in Ref. [3]. They are *topological stabilizer codes* [1, 20–22], meaning they are defined on a lattice and have macroscopic distance together with geometrically local stabilizer generators. The main new conceptual contribution in Ref. [15] is that gauge fixing allows one to fault-tolerantly switch between a (stabilizer) color code on a d-dimensional lattice, in which CNOT and $R_d = \mathrm{diag}\left(1, \exp\left(\frac{2\pi i}{2^d}\right)\right)$ are transversal, and a different (subsystem) color code on the same lattice, in which H is transversal. Critically, for $d \geq 3$, $\{H, \mathrm{CNOT}, R_d\}$ forms a universal gate set. To the authors' knowledge, this represents the first construction using gauge fixing to achieve a universal gate set in a topological code.

In Ref. [15], Bombín argues that for every $d \geq 2$, there exists a d-dimensional color code with a transversal implementation of $R_d \in \mathcal{P}_d \setminus \mathcal{P}_{d-1}$, which is the main technical contribution therein. Here, $\mathcal{P}_d$ is the dth level of the Clifford hierarchy* [23]. At the same time, for any topological stabilizer code, Bravyi and König [22] showed that the group of logical gates implemented transversally must be contained in $\mathcal{P}_d$. These results have been extended beyond the stabilizer code setting [24, 25]. Color codes are the only family of topological stabilizer codes currently known to saturate the Bravyi-König classification in every dimension $d \geq 2$.

In this paper, we provide a simplified yet rigorous presentation of the ideas in Ref. [15]. The organization is as

---

* The Clifford hierarchy is defined sequentially for $j > 1$ according to $\mathcal{P}_j = \{\text{unitary } U | UPU^\dagger \in \mathcal{P}_{j-1} \ \forall P \in \mathcal{P}_1\}$ with $\mathcal{P}_1$ representing the Pauli group. Note that $\mathcal{P}_2$ is the well-known Clifford group.

follows. First, to build some intuition, we introduce color codes in two dimensions in Section II. We explain how to transversally implement the gate set $\{H, \mathrm{CNOT}, R_2\}$, which generates the Clifford group. Then, we describe the generalization of color codes to d dimensions in Section III. Next, in Section IV we discuss transversal gates in those codes with an emphasis on the phase gate R_n, and show that in certain d-dimensional color codes R_d is transversal. Our construction utilizes the bipartite property of the lattice allowing for a simpler verification than in Ref. [15]. Finally, in Section V we explain how to switch between color codes fault-tolerantly using the technique of gauge fixing. In particular, this allows one to implement a fault-tolerant universal gate set $\{H, \mathrm{CNOT}, R_3\}$ in a color code in three dimensions.

II. COLOR CODE IN TWO DIMENSIONS

In this section, we give an explicit construction of a stabilizer color code in two dimensions [3, 26]. We consider a 3-valent lattice formed as a tiling of a sphere, such that faces of the lattice are colored with three colors, where neighboring faces have distinct colors. Qubits are placed at the vertices of this lattice. To define a color code on this lattice, we associate an X- and a Z-type stabilizer generator with every face. This code encodes no logical qubits. A new code, which encodes a single logical qubit, can be formed through the removal of a single physical qubit. We describe the transversal implementation of the logical gates $\overline{\mathrm{CNOT}}$, $\overline{H}$ and $\overline{R}_2$ in the new code[†].

A. Color code with no encoded qubits

Color codes in two dimensions are CSS stabilizer codes [20, 21], and are therefore specified by their stabilizer group $\mathcal{S}$ generated by X- and Z-type stabilizer generators. The code space is the simultaneous $+1$ eigenspace of every stabilizer generator. In the construction, we use a two-dimensional lattice $\mathcal{L}_0^*$, obtained from a tiling of the 2-sphere, and satisfying the following requirements

- valence — every vertex is 3-valent, meaning it belongs to exactly 3 edges,

- colorability — faces can be colored with 3 colors: red, green and blue, such that every two faces sharing an edge have different colors.

An example of such a tiling of the 2-sphere is presented in Fig. 1(a). From these properties alone, one can show that the total number of vertices in $\mathcal{L}_0^*$ is even. To see this, note that the Euler characteristic is $V - E + F = 2$,

where V, E and F denote the number of vertices, edges and faces in $\mathcal{L}_0^*$, respectively. Since every vertex is 3-valent, we obtain $E = \frac{3}{2}V$, and then $V = 2(F - 2)$, which is even.
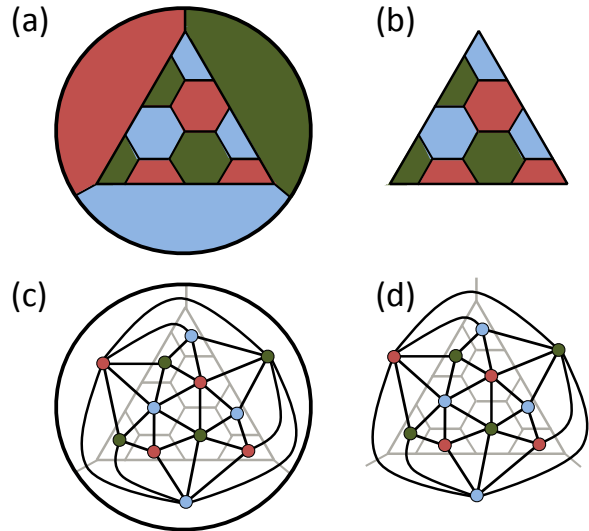

FIG. 1. (Color online) Construction of color codes in two dimensions. In (a) and (b), qubits are placed at vertices, and X- and Z-type stabilizer generators are associated with faces. In (c) and (d) (the dual picture), qubits are placed on faces, and X- and Z-type stabilizer generators are associated with vertices. (a) Take a lattice $\mathcal{L}_0^*$, which is a tiling of the 2-sphere with 3-colorable faces and 3-valent vertices. The surrounding circle is identified with a vertex v. The color code on $\mathcal{L}_0^*$ encodes no logical qubits. (b) To obtain $\mathcal{L}^*$, remove from $\mathcal{L}_0^*$ the vertex v, together with the three edges and three faces containing it. The color code on $\mathcal{L}^*$ encodes one logical qubit. (c) (Dual) lattice $\mathcal{L}_0$ is obtained from $\mathcal{L}_0^*$ by replacing faces, edges and vertices by vertices, edges and faces, respectively. All faces are triangles, and the vertices are 3-colorable. The color code on $\mathcal{L}_0$ encodes no logical qubits. (d) Lattice $\mathcal{L}$ formed from $\mathcal{L}_0$ by removing a single face. No stabilizer generators are associated with those vertices belonging to the boundary of the removed face. The color code on $\mathcal{L}$ encodes one logical qubit.

At every vertex in $\mathcal{L}_0^*$ we place a qubit. We refer to the set of all qubits by Q, whereas by $\mathcal{Q}(\Pi) \subset Q$ we denote the set of vertices of a face Π. Alternatively, we can think of $\mathcal{Q}(\Pi)$ as the set of qubits belonging to Π. To define the color code, it is sufficient to specify X- and Z-type stabilizer generators. For every face Π, we define an X-type stabilizer generator $X(\Pi)$ to be a tensor product of Pauli X operators supported on qubits $\mathcal{Q}(\Pi)$, similarly for Z-type generators. Then, the stabilizer group $\mathcal{S}$ is generated by

$$\mathcal{S} = \langle X(\Pi), Z(\Pi), \text{ for every face } \Pi \text{ in } \mathcal{L}_0^* \rangle. \quad (1)$$

To prove that this specifies a well-defined stabilizer code, we must verify that all the generators of $\mathcal{S}$ commute. It is sufficient to check that for any two faces Π_1

[†] We use a bar to indicate action on logical code space. The absence of a bar indicates action on physical qubits.

and Π_2 in $\mathcal{L}_0^*$, $X(\Pi_1)$ and $Z(\Pi_2)$ commute. First take the case $\Pi_1 \neq \Pi_2$. If Π_1 and Π_2 share no vertices, then $X(\Pi_1)$ and $Z(\Pi_2)$ trivially commute. If they share a vertex, then by 3-valence, they also share an edge. Moreover, due to valence and colorability conditions, Π_1 and Π_2 cannot share two consecutive edges, and thus their intersection has to contain an even number of vertices,

$$|\mathcal{Q}(\Pi_1) \cap \mathcal{Q}(\Pi_2)| \equiv 0 \mod 2. \qquad (2)$$

For the case $\Pi_1 = \Pi_2 = \Pi$, due to 3-colorability and 3-valence, the number of vertices belonging to a face Π is even,

$$|\mathcal{Q}(\Pi)| \equiv 0 \mod 2. \qquad (3)$$

Therefore, we obtain commutation of $X(\Pi_1)$ and $Z(\Pi_2)$ for arbitrary Π_1 and Π_2.

From the construction of the lattice, one obtains that each vertex belongs to exactly three faces, colored with three different colors. Thus, one can express the set of vertices in $\mathcal{L}_0^*$ as the disjoint union‡ of vertices belonging to red faces, and similarly for green and blue [3, 26], namely

$$Q = \bigsqcup_{\Pi_R} \mathcal{Q}(\Pi_R) = \bigsqcup_{\Pi_G} \mathcal{Q}(\Pi_G) = \bigsqcup_{\Pi_B} \mathcal{Q}(\Pi_B), \qquad (4)$$

where $\{\Pi_R\}$, $\{\Pi_G\}$ and $\{\Pi_B\}$ are the sets of all red, green and blue faces, respectively. This implies that not all the stabilizer generators we have defined are independent

$$\prod_{\Pi_R} X(\Pi_R) = \prod_{\Pi_G} X(\Pi_G) = \prod_{\Pi_B} X(\Pi_B), \qquad (5)$$

$$\prod_{\Pi_R} Z(\Pi_R) = \prod_{\Pi_G} Z(\Pi_G) = \prod_{\Pi_B} Z(\Pi_B). \qquad (6)$$

In fact, these are the only conditions [26, 27] which relate the stabilizer generators to one another.

We can now verify that the color code which we have defined on the lattice $\mathcal{L}_0^*$ encodes no logical qubits. As before, using the Euler characteristic we obtain $F - 2 = E - V$, and from 3-valence of vertices — $E = \frac{3}{2}V$. We have placed physical qubits at vertices, thus $|Q| = V$. There are $2F - 4$ independent stabilizer generators, since there are two stabilizer generators for every face and four conditions (5) and (6). The number of logical qubits is equal to the number of physical qubits minus the number of independent stabilizer generators, and we obtain

$$|Q| - (2F - 4) = V - 2(E - V) = 0. \qquad (7)$$

---

‡ We use the *disjoint union* $A \sqcup B$ in place of the union $A \cup B$ of two sets A and B when their intersection is empty, $A \cap B = \emptyset$.

B. Color code with one logical qubit

To obtain a color code with one encoded logical qubit, we can remove one vertex from the lattice $\mathcal{L}_0^*$, together with three edges and three faces it belongs to, obtaining a new lattice $\mathcal{L}^*$ (see Fig. 1b). By removing one vertex, we also discard six stabilizer generators associated with the removed faces, and thus the stabilizer generators no longer have to satisfy (5) and (6). One can check that this new code encodes one logical qubit, since there is one qubit more than independent stabilizer generators. By removing more vertices, one could encode more logical qubits, but we will not analyze that case. Note that the total number of qubits in $\mathcal{L}^*$ is odd, $|Q| \equiv 1 \mod 2$, which plays an important role in our considerations.

On physical grounds, it is of interest to consider stabilizer codes with stabilizer generators which are low-weight and geometrically local. In the construction we have presented, this can be achieved if each face in the lattice $\mathcal{L}^*$ is geometrically local and contains a small number of vertices, as in Fig. 1b. It can be shown that following this construction, the resulting color code has macroscopic distance [3], and therefore is a topological stabilizer code.

Later, when we discuss color codes in d dimensions, we follow a similar construction. We briefly outline the procedure here, deferring detailed discussion to Section III. We start with a tiling of a d-sphere, place qubits at vertices and define (gauge group) generators to be supported on suitable cells. Then, we remove one vertex and all the cells containing it. In particular, we discard generators supported on the removed cells. Such a code encodes only one logical qubit [27].

C. Transversal gates

Consider a stabilizer code encoding one logical qubit, with the stabilizer group $\mathcal{S}$. In this setting, a *transversal gate* $\overline{U}$ on a single logical qubit is implemented as a tensor product of single physical qubit unitaries $U_1 \otimes \ldots \otimes U_{|Q|}$, which preserves the code space. On the other hand, a logical gate on two logical qubits requires two copies of the code, in which case we say that the *overall* code space is the $+1$ eigenspace of the elements in $\mathcal{S} \otimes \mathcal{S}$. A transversal gate on two logical qubits is implemented as a tensor product of two qubit gates on pairs of corresponding qubits in both copies of the code, which preserves the overall code space. Observe that transversal gates are fault-tolerant since they do not spread errors within each copy of the code.

We now show that in the two-dimensional color code described in the previous subsection, one can transversally implement the gate set $\{\overline{H}, \overline{\text{CNOT}}, \overline{R_2}\}$, which generates the (non-universal) Clifford group. The Clifford group, combined with computational basis state preparation and measurement, can be simulated efficiently on a classical computer [28, 29]. For each gate, $\overline{H}$, $\overline{\text{CNOT}}$ and

$\overline{R}_2$, we verify that a particular transversal unitary implements the logical gate by showing that it has the correct action under conjugation on generators of the logical Pauli group, and that the stabilizer group is preserved[§].

This two-dimensional color code is a CSS stabilizer code encoding a single logical qubit with logical Pauli operators $\overline{X} = X(Q)$ and $\overline{Z} = Z(Q)$. In addition it is a *self-dual CSS stabilizer code* — a code with the same support for X- and Z-type stabilizer group elements (for each face, there is an X- and a Z-type generator). This implies that the logical Hadamard gate can be implemented transversally, as under conjugation by $H(Q)$, $\overline{X} \mapsto H(Q)X(Q)H(Q)^\dagger = \overline{Z}$ and similarly $\overline{Z} \mapsto \overline{X}$. Moreover, $X(\Pi) \mapsto Z(\Pi)$, $Z(\Pi) \mapsto X(\Pi)$, and thus $\mathcal{S}$ is preserved.

The logical gate $\overline{\mathrm{CNOT}}$ can be implemented transversally between two identical copies of this color code by applying a physical gate CNOT to every pair of corresponding qubits in the first and the second copy. This can be verified by checking that under conjugation by $\overline{\mathrm{CNOT}}$, $\overline{X}\,\overline{I} \mapsto \overline{X}\,\overline{X}$, $\overline{I}\,\overline{X} \mapsto \overline{I}\,\overline{X}$, $\overline{Z}\,\overline{I} \mapsto \overline{Z}\,\overline{I}$, $\overline{I}\,\overline{Z} \mapsto \overline{Z}\,\overline{Z}$ and $\mathcal{S} \otimes \mathcal{S}$ is preserved[¶].

To show that $\overline{R}_2$ can be implemented transversally, we use the fact that the set of vertices in $\mathcal{L}^*$ is bipartite (see Fig. 2(a)). In other words, Q can be split into two subsets, T and $T^c := Q \setminus T$, such that vertices in T are connected only to vertices in T^c and vice versa. To prove this, first note that every face in $\mathcal{L}_0^*$ has an even number of edges. Moreover, every cycle in $\mathcal{L}_0^*$ (as a tiling of the 2-sphere) is contractible. This implies that every cycle in $\mathcal{L}_0^*$ is a boundary of faces and is therefore even. Using the following lemma

Lemma 1 (Graph Bipartition). *A graph containing only even cycles is bipartite [30].*

we see that $\mathcal{L}_0^*$ must be bipartite, and so is the lattice $\mathcal{L}^*$ due to its construction from $\mathcal{L}_0^*$.

Now, we can show that $R = R_2^k(T)R_2^{-k}(T^c)$ implements $\overline{R}_2$, for some choice of integer k. We use the relations $R_2 X R_2^\dagger = iXZ$ and $R_2 Z R_2^\dagger = Z$. Since $|Q| \equiv 1 \mod 2$, then $|T| - |T^c| = 2|T| - |Q| \equiv \pm 1 \mod 4$, and picking $k = |T| - |T^c| \mod 4$ ensures that $k(|T| - |T^c|) \equiv 1 \mod 4$. With this choice of k, the action by conjugation of $R = R_2^k(T)R_2^{-k}(T^c)$ on the logical $\overline{X}$ and $\overline{Z}$ is

$$R\overline{X}\,R^\dagger = i^{k(|T|-|T^c|)}\overline{X}\,\overline{Z} = i\overline{X}\,\overline{Z}, \qquad (8)$$

$$R\overline{Z}\,R^\dagger = \overline{Z}. \qquad (9)$$

Furthermore, as every face Π in the lattice $\mathcal{L}^*$ has an equal number of vertices in T and T^c, under the action

of R the stabilizer generators $X(\Pi)$ and $Z(\Pi)$ become:

$$RX(\Pi)R^\dagger = i^{k(|T\cap\Pi|-|T^c\cap\Pi|)}X(\Pi)Z(\Pi) \qquad (10)$$

$$= X(\Pi)Z(\Pi) \in \mathcal{S}, \qquad (11)$$

$$RZ(\Pi)R^\dagger = Z(\Pi), \qquad (12)$$

implying that the stabilizer group $\mathcal{S}$ is preserved. This completes the verification that R implements $\overline{R}_2$.
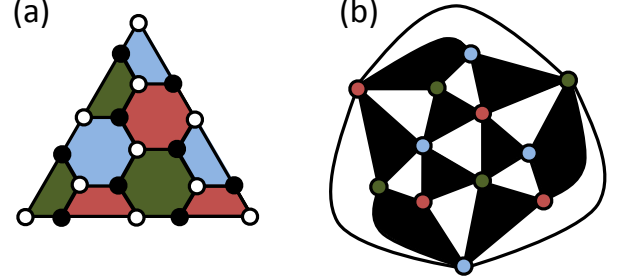

FIG. 2. (Color online) (a) The set of vertices of $\mathcal{L}^*$, the lattice used to define the color code, is bipartite — it can be split into two subsets: T (hollow circles), and its compliment T^c (filled circles). Vertices in T are only connected to vertices in T^c and vice versa. The logical gate $\overline{R}_2$ can implemented by applying R_2^k to qubits in T, and R_2^{-k} to qubits in T^c, where $k \equiv |T| - |T^c| \mod 4$. (b) The dual lattice $\mathcal{L}$. Faces are bipartite.

D. Dual lattice picture

We can alternatively express the construction of color codes in the dual lattice picture, which we will use extensively for $d > 2$ dimensions. We use a two-dimensional (dual) lattice $\mathcal{L}_0$, obtained from a tiling of the 2-sphere, and satisfying the following requirements

- all faces are triangles,

- vertices are 3-colorable, meaning two vertices belonging to the same edge are colored with different colors.

See Fig. 1(c) for a simple example. Note that these conditions are equivalent to the conditions of 3-valence of vertices and 3-colorability of faces required for the tiling $\mathcal{L}_0^*$ of the 2-sphere, where lattices $\mathcal{L}_0^*$ and $\mathcal{L}_0$ are dual to one another.

A qubit is placed on every face of $\mathcal{L}_0$, and an X- and a Z-type stabilizer generator is associated with every vertex, meaning they are supported on qubits corresponding to faces containing that vertex. The resulting color code is exactly the same as that described in Section II A, and therefore has zero logical qubits. To encode a single logical qubit, one should remove a face from $\mathcal{L}_0$, together with stabilizer generators associated with the vertices belonging to the removed face, see Fig. 1(d).

The bipartition of vertices in $\mathcal{L}^*$ corresponds to a bipartition of faces in $\mathcal{L}$, meaning that faces in $\mathcal{L}$ can be

---

[§] Preservation of the stabilizer group is a sufficient (but not necessary) condition that implies preservation of the code.

[¶] Notice that generators of $\mathcal{S} \otimes \mathcal{S}$ are mapped under conjugation to a different generators, namely $X(\Pi) \otimes I(\Pi) \mapsto X(\Pi) \otimes X(\Pi)$, $Z(\Pi) \otimes I(\Pi) \mapsto Z(\Pi) \otimes I(\Pi)$, $I(\Pi) \otimes X(\Pi) \mapsto I(\Pi) \otimes X(\Pi)$ and $I(\Pi) \otimes Z(\Pi) \mapsto Z(\Pi) \otimes Z(\Pi)$.

split into two sets, T and its compliment T^c, such that faces in T share an edge only with faces in T^c and vice-versa. See Fig. 2(b).

III. COLOR CODE IN HIGHER DIMENSIONS

Here we present a construction of color codes on d-dimensional lattices. In higher dimensions it is easier to describe the construction in the language of the dual lattice. The majority of this section is devoted to defining dual lattices satisfying certain conditions and analyzing their properties. The discussion is a generalization of that already presented for two dimensions. The basic idea of how to construct the dual lattice $\mathcal{L}$ is to first tile a d-sphere with d-simplices to form a lattice $\mathcal{L}_0$. We require that every vertex in $\mathcal{L}_0$ can be assigned one of $d+1$ distinct colors and two vertices belonging to the same edge have different colors. The lattice $\mathcal{L}$, used to define the color code, is formed by removing one d-simplex from $\mathcal{L}_0$, which results in a color code encoding one logical qubit [15, 27].

A. Simplicial complexes and colorability

A d-simplex δ is a d-dimensional polytope which is a convex hull of its $d+1$ affinely independent vertices $v_0, v_1, \ldots, v_d$, namely

$$\delta = \left\{ \sum_{i=0}^{d} t_i v_i \,\middle|\, 0 \le t_i \wedge \sum_{i=0}^{d} t_i = 1 \right\}. \qquad (13)$$

In particular, 0-simplices are vertices, 1-simplices are edges, 2-simplices are triangles, 3-simplices are tetrahedra and so on.

A convex hull of a subset of vertices of size $k+1 \le d+1$ is a k-simplex σ, which we call a k-face of δ, and $\sigma \subset \delta$. For example, the faces of a 3-simplex (a tetrahedron) are: four 0-simplices, six 1-simplices, four 2-simplices and a single 3-simplex. More generally, δ contains $\binom{d+1}{k+1}$ k-faces, since every k-face is uniquely determined by the choice of $k+1$ vertices spanning it. By $\Delta_k(\delta)$ we call the set of all k-faces of δ, namely

$$\Delta_k(\delta) = \{\sigma \subset \delta | \sigma \text{ is a } k\text{-simplex}\}. \qquad (14)$$

Instead of having only one simplex, we can consider a collection of them. Moreover, we can create new objects, called simplicial complexes [31], by gluing simplices along their proper faces of matching dimension. We restrict ourselves to simplicial complexes containing finitely many simplices. We will define a d-dimensional color code on a lattice $\mathcal{L}$ obtained by gluing together d-simplices. The technical name for such a lattice is a homogeneous simplicial d-complex.

Although $\mathcal{L}$ is formally a collection of simplices, by the same symbol we also denote the union of these simplices

as a topological space. Notice that $\mathcal{L}$ is a manifold with a boundary, which we can think of as being embedded in real space. We denote by $\partial\mathcal{L}$ the set of simplices belonging to the boundary of $\mathcal{L}$, where the boundary of $\mathcal{L}$ is the set of points in the closure of $\mathcal{L}$ not belonging to the interior of $\mathcal{L}$. Moreover, by $\Delta'_k(\mathcal{L})$ we understand the set of all k-simplices belonging to $\mathcal{L} \setminus \partial\mathcal{L}$. Note that $\Delta'_d(\mathcal{L}) = \Delta_d(\mathcal{L})$.

We say that a simplicial d-complex $\mathcal{L}$ is $(d+1)$-colorable if we can introduce a function

$$\text{color} : \Delta_0(\mathcal{L}) \to \mathbb{Z}_{d+1}, \qquad (15)$$

where $\mathbb{Z}_{d+1} = \{0, 1, \ldots, d\}$ is a set of $d+1$ colors, and two vertices belonging to the same edge have different colors. Moreover, by $\text{color}(\delta)$ we understand the set of colors assigned to all the vertices of a simplex δ, namely

$$\text{color}(\delta) = \bigsqcup_{v \in \Delta_0(\delta)} \text{color}(v). \qquad (16)$$

An example of a 3-colorable, homogeneous, simplicial 2-complex is the lattice $\mathcal{L}$ shown in Fig. 1(d). Note in particular that it is composed of nineteen 2-simplices (triangles). The exact shape of objects in $\mathcal{L}$ is not important due to its topological nature — the lattice is not rigid and can be smoothly deformed. In this example, $\Delta'_0(\mathcal{L})$ consists of the set of 9 vertices (the three vertices in the boundary are excluded). $\Delta'_1(\mathcal{L})$ is the set of 27 edges, (the three edges in the boundary are excluded). $\Delta'_2(\mathcal{L})$ is the set of all 19 triangular faces.

B. Definition of color code

Here we define color codes on a d-dimensional lattice $\mathcal{L}$, which must satisfy the following conditions

Condition 1. *$\mathcal{L}$ is a homogeneous simplicial d-complex obtained as a triangulation of the interior of a d-simplex.*

Condition 2. *$\mathcal{L}$ is $(d+1)$-colorable.*

One can obtain such a lattice $\mathcal{L}$ from any $(d+1)$-colorable tiling of the d-sphere with d-simplices, followed by the removal of one d-simplex. In $d = 2$ dimensions, this is precisely the procedure described in Section II D. An explicit construction of a family of lattices satisfying these conditions is outlined in Appendix 1.

Qubits are placed on each and every d-simplex of $\mathcal{L}$, and thus the set of all qubits Q is equal to $\Delta_d(\mathcal{L})$. This motivates the next definition, namely for a simplex $\delta \subset \mathcal{L} \setminus \partial\mathcal{L}$ we define

$$\mathcal{Q}(\delta) = \{\sigma \in \Delta_d(\mathcal{L}) | \sigma \supset \delta\}. \qquad (17)$$

In other words, $\mathcal{Q}(\delta)$ can be thought of as the set of qubits placed on d-simplices containing δ. We say that qubits $\mathcal{Q}(\delta)$ are supported on δ. By saying that an operator is

supported on δ we mean that it is supported on the set $\mathcal{Q}(\delta)$, for example $X(\delta) := X(\mathcal{Q}(\delta))$.

A color code is a CSS subsystem code [32, 33]. Recall that a CSS subsystem code is specified by its gauge group $\mathcal{G}$. Each X-type gauge group generator $X(G^x)$ consists of Pauli X operators applied to qubits G^x; similarly for Z-type generators. The stabilizer group $\mathcal{S} \subset \mathcal{G}$ is the group generated by all Pauli operators $X(S^x)$ and $Z(S^z)$ contained in $\mathcal{G}$, which commute with every element of $\mathcal{G}$. Note that $-I \notin \mathcal{S}$. The codewords are $+1$ eigenvectors of all elements of $\mathcal{S}$.

We define a d-dimensional color code [15] on the lattice $\mathcal{L}$, where $d = \dim \mathcal{L}$, as the CSS subsystem code with X- and Z-type gauge generators supported on $(d - 2 - z)$- and $(d - 2 - x)$-simplices in $\mathcal{L}$,

$$\mathcal{G} = \langle X(\delta), Z(\sigma) | \forall \delta \in \Delta'_{d-2-z}(\mathcal{L}), \sigma \in \Delta'_{d-2-x}(\mathcal{L}) \rangle, \quad (18)$$

where $x + z \leq d - 2$. The X- and Z-type generators of the stabilizer group $\mathcal{S}$ are supported on x- and z-simplices, namely

$$\mathcal{S} = \langle X(\delta), Z(\sigma) | \forall \delta \in \Delta'_x(\mathcal{L}), \sigma \in \Delta'_z(\mathcal{L}) \rangle. \quad (19)$$

We refer to this code by $CC_{\mathcal{L}}(x, z)$. When context makes the lattice unambiguous, we sometimes use $CC_d(x, z)$ to emphasize the dimensionality of the lattice, $\dim \mathcal{L} = d$. Note that the generators of the gauge and stabilizer groups are supported on simplices which do not belong to $\partial \mathcal{L}$, the boundary of the lattice $\mathcal{L}$.

To illustrate the language introduced in this section, we revisit the two-dimensional color code described in Sections II B and II D. We begin with the lattice $\mathcal{L}$ shown in Fig. 1d. Qubits are placed on 2-simplices (triangular faces). Since $x + z \leq \dim \mathcal{L} - 2 = 0$, there is only one color code on the two-dimensional lattice $\mathcal{L}$, namely $CC_{\mathcal{L}}(0, 0)$, which is a stabilizer code. Stabilizer generators are associated with 0-simplices (vertices). Note that no stabilizer generators are assigned to the three vertices belonging to the boundary of $\mathcal{L}$.

C. Properties of the lattice

Here we present some properties of any $(d+1)$-colorable homogeneous simplicial d-complex $\mathcal{L}$. We use these properties to verify that $CC_{\mathcal{L}}(x, z)$ is a valid code, and later that there is a transversal implementation of $\overline{R}_n$. We start with the following lemmas.

Lemma 2 (Intersection). *Let δ and σ be two simplices in $\mathcal{L} \setminus \partial \mathcal{L}$. If $\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma) \neq \emptyset$, then $\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma) = \mathcal{Q}(\tau)$, where τ is the smallest simplex containing both δ and σ.*

Proof. If $\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma) \neq \emptyset$, then there exists $\epsilon \in \Delta_d(\mathcal{L})$ such that $\epsilon \supset \delta, \sigma$. Let $C = \text{color}(\delta) \cup \text{color}(\sigma)$ and set τ to be the unique $(|C| - 1)$-simplex in ϵ, colored with the set of colors C. Since $\tau \supset \delta$, then $\mathcal{Q}(\tau) \subset \mathcal{Q}(\delta)$; similarly $\mathcal{Q}(\tau) \subset \mathcal{Q}(\sigma)$, and therefore $\mathcal{Q}(\tau) \subset \mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma)$. As τ

is the smallest simplex containing δ and σ, then $\mathcal{Q}(\tau) \supset \mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma)$ and thus $\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma) = \mathcal{Q}(\tau)$. $\qquad\square$

Lemma 3 (Disjoint Union). *Let $\mathcal{L}$ be a simplicial d-complex which is $(d + 1)$-colorable. Then, for a simplex $\delta \subset \mathcal{L} \setminus \partial \mathcal{L}$ and a chosen set of colors C, such that $\text{color}(\delta) \subset C \subset \mathbb{Z}_{d+1}$, there exists a partition of the set of qubits supported on δ into a disjoint union of sets of qubits supported on $(|C| - 1)$-simplices containing δ, namely*

$$\mathcal{Q}(\delta) = \bigsqcup_{\substack{\sigma \supset \delta \\ \sigma \in \Delta'_{|C|-1}(\mathcal{L}) \\ \text{color}(\sigma) = C}} \mathcal{Q}(\sigma). \quad (20)$$

Proof. First note, that two different k-simplices δ_1 and δ_2 in $\mathcal{L} \setminus \partial \mathcal{L}$ colored with the same colors, $\text{color}(\delta_1) = \text{color}(\delta_2)$, cannot belong to the same l-simplex, $l \geq k$, thus do not share a qubit, $\mathcal{Q}(\delta_1) \cap \mathcal{Q}(\delta_2) = \emptyset$. Moreover, if $\mathcal{Q}(\epsilon) \subset \mathcal{Q}(\delta)$, where $\epsilon \in \Delta_d(\mathcal{L})$, then $\epsilon \supset \delta$ and there exists a unique simplex $\sigma \subset \epsilon$ colored with colors C. Since $\text{color}(\sigma) = C \supset \text{color}(\delta)$, then $\sigma \supset \delta$, which finishes the proof of the (Disjoint Union) Lemma 3. $\qquad\square$

As a corollary of the (Disjoint Union) Lemma 3 we obtain the following

Lemma 4 (Even Support). *Let δ be a k-simplex not belonging to the boundary of the lattice, $\delta \subset \Delta'_k(\mathcal{L})$, with $0 \leq k < d$. Then*

$$|\mathcal{Q}(\delta)| \equiv 0 \mod 2. \quad (21)$$

Before we prove the (Even Support) Lemma 4, we explain its consequences. For $CC_d(x, z)$ to be a subsystem code, where $x + z \leq d - 2$, the stabilizer generators have to commute with each other, as well as with the gauge group generators. Notice that for two arbitrary X- and Z-type stabilizer generators to commute, the intersection of their supports has to contain even number of elements. By definition, X- and Z-type stabilizer generators be supported on $\delta \subset \Delta'_x(\mathcal{L})$ and $\sigma \subset \Delta'_z(\mathcal{L})$, respectively. If the intersection $\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma)$ is non-empty, then due to the (Intersection) Lemma 2 there exists a simplex τ such that $\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma) = \mathcal{Q}(\tau)$. Moreover, since δ is spanned by $x + 1$ vertices and σ by $z + 1$ vertices, then τ is spanned by at most $x + z + 2 \leq d$ vertices. Thus, τ is a k-simplex with $k < d$, and the (Even Support) Lemma 4 applies, $|\mathcal{Q}(\delta) \cap \mathcal{Q}(\sigma)| = |\mathcal{Q}(\tau)| \equiv 0 \mod 2$, showing that $X(\delta)$ and $Z(\sigma)$ commute. The commutation of stabilizer generators with the gauge generators follows similarly.

Proof. The set of qubits supported on any k-simplex δ in $\mathcal{L} \setminus \partial \mathcal{L}$ with $k < d$ can be decomposed as a disjoint union of qubits supported on $(d - 1)$-simplices σ containing δ and colored with a chosen set of d colors, $C \supset \text{color}(\delta)$. Note that any $(d - 1)$-simplex $\sigma \in \Delta'_{d-1}(\mathcal{L})$ separates

two d-simplices, and then $|\mathcal{Q}(\sigma)| = 2$, which immediately yields

$$|\mathcal{Q}(\delta)| = \sum_{\substack{\sigma \supset \delta \\ \sigma \in \Delta'_{d-1}(\mathcal{L}) \\ \text{color}(\sigma) = C}} |\mathcal{Q}(\sigma)| \equiv 0 \mod 2, \qquad (22)$$

showing the (Even Support) Lemma 4. $\qquad\square$

The property needed for the transversal implementation of the gate $\overline{R}_n$, presented in Section IV, can be encapsulated in the following lemma

Lemma 5 (Bipartition of Qubits)**.** *The set of d-simplices in $\mathcal{L}$, $\Delta_d(\mathcal{L})$, is bipartite.*

Let us first explain the (Bipartition of Qubits) Lemma 5 — the d-simplices in $\mathcal{L}$ can be split into two disjoint sets, where d-simplices in the first set share $(d-1)$-faces only with d-simplices from the second set, and vice versa.

Proof. First, construct a graph $G = (V, E)$ with the set of vertices $V = \Delta_d(\mathcal{L})$ and the set of edges $E = \Delta'_{d-1}(\mathcal{L})$. Two vertices $v, w \in V$ are connected by an edge $e \in E$ iff d-simplices corresponding to v and w share a $(d-1)$-face corresponding to e. Since for all $\delta \in \Delta'_{d-2}(\mathcal{L})$ the (Even Support) Lemma 4 gives $|\mathcal{Q}(\delta)| \equiv 0 \mod 2$, and every cycle in $\mathcal{L}$ is contractible, we obtain that every cycle in the graph G is even. Using the (Graph Bipartition) Lemma 1 we immediately obtain that G is bipartite. This shows that the set of d-simplices in $\mathcal{L}$, which is equal to the set of qubits, $\Delta_d(\mathcal{L}) = Q$, is bipartite. $\qquad\square$

IV. TRANSVERSAL GATES IN COLOR CODES

As mentioned in the introduction, transversal gates are fault-tolerant. In this section, we first review some relevant features of a class of CSS subsystem codes, which includes the color codes defined in Section III. Then, we examine transversal gates of codes in this class. We show that $\overline{\text{CNOT}}$ is transversal in any such code and under certain additional conditions the Hadamard and $\overline{R}_n$ can be transversal, too. Finally, we show that the additional conditions are satisfied by certain color codes.

A. Subsystem codes

A CSS subsystem code [32, 33] is specified by its gauge group $\mathcal{G}$, which is a subgroup of the Pauli group on physical qubits Q. Each X-type gauge group generator $X(G^x)$ consists of Pauli X operators applied to qubits G^x; similarly for Z-type generators. The stabilizer group $\mathcal{S} \subseteq \mathcal{G}$ is the group generated by all Pauli operators $X(S^x)$ and $Z(S^z)$ contained in $\mathcal{G}$, which commute with every element of $\mathcal{G}$. (Note that a stabilizer code is a special case of a subsystem code, for which $\mathcal{G} = \mathcal{S}$). The codewords

are the $+1$ eigenvectors of all elements of $\mathcal{S}$. We say that two codewords are equivalent if they differ by application of a linear combination of elements of $\mathcal{G} \setminus \mathcal{S}$. This allows one to decompose the subspace of codewords into a tensor product of two spaces: *logical* qubits and *gauge* qubits. Elements of $\mathcal{G} \setminus \mathcal{S}$ have no effect on the state of the logical qubits, but may change that of the gauge qubits.

For a subsystem code, we say a unitary implements a *logical gate* if it preserves the space of all codewords, and has an action on the logical qubits which is independent of any action on the gauge qubits. A logical gate $\overline{U}$ can be implemented on the logical qubits $|\psi\rangle$ as a *bare* gate U_{bare} which leaves gauge qubits $|g\rangle$ unchanged, $U_{\text{bare}} : |\psi\rangle|g\rangle \mapsto (\overline{U}|\psi\rangle)|g\rangle$, or more generally as a *dressed* gate U_{dressed}, which can affect the gauge qubits too, $U_{\text{dressed}} : |\psi\rangle|g\rangle \mapsto (\overline{U}|\psi\rangle)|g'\rangle$.

Consider the class of CSS subsystem codes which

- encode one logical qubit,

- have bare logical $\overline{X}$ and $\overline{Z}$ implemented by $X(Q)$ and $Z(Q)$.

Note that these codes are defined on an odd number of physical qubits, $|Q| \equiv 1 \mod 2$, since $\overline{X}$ and $\overline{Z}$ anticommute.

We can define a pair of inequivalent (and not normalized) codewords, which are representatives of logical $|\overline{0}\rangle$ and $|\overline{1}\rangle$, namely

$$|\overline{0}\rangle|g_X\rangle = \sum_{X(G) \in \mathcal{G}} X(G)|\mathbf{0}\rangle, \qquad (23)$$

$$|\overline{1}\rangle|g_X\rangle = \overline{X}|\overline{0}\rangle|g_X\rangle, \qquad (24)$$

where $|\mathbf{0}\rangle$ is a state with every physical qubit set to $|0\rangle$, and $|g_X\rangle$ is a fixed state of the gauge qubits. One can verify that the states $|\overline{0}\rangle|g_X\rangle$ and $|\overline{1}\rangle|g_X\rangle$ are $+1$ eigenstates of $\mathcal{S}$, and satisfy $\overline{Z}|\overline{0}\rangle|g_X\rangle = |\overline{0}\rangle|g_X\rangle$, $\overline{Z}|\overline{1}\rangle|g_X\rangle = -|\overline{1}\rangle|g_X\rangle$. They are also $+1$ eigenstates of every X-type generator of $\mathcal{G}$. All equivalent codewords can be generated from $|\overline{0}\rangle|g_X\rangle, |\overline{1}\rangle|g_X\rangle$ by application of a linear combination of elements from $\mathcal{G} \setminus \mathcal{S}$. An alternative pair of representatives of logical $|\overline{0}\rangle$ and $|\overline{1}\rangle$ is

$$|\overline{0}\rangle|g_Z\rangle = \sum_{X(S) \in \mathcal{S}} X(S)|\mathbf{0}\rangle, \qquad (25)$$

$$|\overline{1}\rangle|g_Z\rangle = \overline{X}|\overline{0}\rangle|g_Z\rangle, \qquad (26)$$

which are $+1$ eigenstates of all Z-type generators of $\mathcal{G}$.

B. Transversal gates in subsystem codes

Consider a CSS subsystem code with one logical qubit, and $\overline{X}$ and $\overline{Z}$ implemented by $X(Q)$ and $Z(Q)$. To check that a physical unitary U implements a dressed logical gate $\overline{U}$ in such a code, one can verify its action on $|\overline{0}\rangle|g\rangle$, and $|\overline{1}\rangle|g\rangle$ for every state $|g\rangle$ of the gauge qubits. Alternatively, it is sufficient to verify that U has the correct

action by conjugation on $\overline{X}$ and $\overline{Z}$, and that it preserves[**] the gauge group $\mathcal{G}$.

The logical gate $\overline{\text{CNOT}}$ can be implemented transversally between two identical copies of this class of CSS subsystem codes by applying a physical gate CNOT to every pair of corresponding qubits in the first and the second copy. This can be verified by checking that under conjugation by $\overline{\text{CNOT}}$, $\overline{XI} \mapsto \overline{XX}$, $\overline{IX} \mapsto \overline{IX}$, $\overline{ZI} \mapsto \overline{ZI}$, $\overline{IZ} \mapsto \overline{ZZ}$ and $\mathcal{G} \otimes \mathcal{G}$ is preserved[††].

If the CSS subsystem code is also *self-dual*, namely it has X- and Z-type gauge group generators supported on the same sets of qubits, $\mathcal{G} = \langle X(G_i), Z(G_i) \rangle$, then a dressed logical Hadamard gate can be implemented transversally as $\overline{H} = H(Q)$. To see this, observe that under conjugation by $H(Q)$, $\overline{X} \mapsto \overline{Z}$, $\overline{Z} \mapsto \overline{X}$, $X(G) \mapsto Z(G)$ and $Z(G) \mapsto X(G)$, and thus $\mathcal{G}$ is preserved.

The last logical gate we analyze is $\overline{R}_n = \text{diag}\left(1, e^{\frac{2\pi i}{2^n}}\right)$, for an integer $n > 0$. We aim to implement $\overline{R}_n$ transversally as a bare logical gate by applying the same single-qubit unitary to some subset $T \subset Q$ of the physical qubits, and applying that unitary's inverse to the rest of the qubits $T^c := Q \setminus T$. Specifically, we now prove that $\overline{R}_n$ is implemented by $R = R_n^k(T) R_n^{-k}(T^c)$, for some suitably chosen $k \in \{1, 2, \ldots, 2^n - 1\}$, provided that T and $\mathcal{G}$ satisfy

$$\forall X(G) \in \mathcal{G} : |T \cap G| \equiv |T^c \cap G| \mod 2^n. \quad (27)$$

First, pick k such that

$$k(|T| - |T^c|) \equiv 1 \mod 2^n. \quad (28)$$

The existence of k is guaranteed by Bezout's lemma, since $|Q|$ is odd, $|T| - |T^c| = 2|T| - |Q| \equiv 1 \mod 2$, and thus $\gcd(|T| - |T^c|, 2^n) = 1$. Noting that $R_n^{\pm k}|0\rangle = |0\rangle$ and $R_n^{\pm k} X = e^{\pm \frac{2\pi i k}{2^n}} X R_n^{\mp k}$, we obtain

$$R|\overline{0}\rangle|g_X\rangle = \sum_{X(G) \in \mathcal{G}} R_n^k(T) R_n^{-k}(T^c) X(G)|\mathbf{0}\rangle \quad (29)$$

$$= \sum_{X(G) \in \mathcal{G}} e^{\frac{2\pi i k}{2^n}|T \cap G|} e^{-\frac{2\pi i k}{2^n}|T^c \cap G|} X(G)|\mathbf{0}\rangle \quad (30)$$

$$= \sum_{X(G) \in \mathcal{G}} X(G)|\mathbf{0}\rangle = |\overline{0}\rangle|g_X\rangle, \quad (31)$$

$$R|\overline{1}\rangle|g_X\rangle = R_n^k(T) R_n^{-k}(T^c) X(Q)|\overline{0}\rangle|g_X\rangle \quad (32)$$

$$= e^{\frac{2\pi i k}{2^n}|T|} e^{-\frac{2\pi i k}{2^n}|T^c|} X(Q) R|\overline{0}\rangle|g_X\rangle \quad (33)$$

$$= e^{\frac{2\pi i}{2^n}} X(Q)|\overline{0}\rangle|g_X\rangle = e^{\frac{2\pi i}{2^n}}|\overline{1}\rangle|g_X\rangle, \quad (34)$$

[**] Note that preservation of the gauge group under the action of a physical unitary U is a sufficient, but not a necessary condition for U to implement a dressed logical gate.

[††] Notice, that generators of $\mathcal{G} \otimes \mathcal{G}$ are mapped under conjugation to another set of generators, namely $X(G) \otimes I(G) \mapsto X(G) \otimes X(G)$, $Z(G) \otimes I(G) \mapsto Z(G) \otimes I(G)$, $I(G) \otimes X(G) \mapsto I(G) \otimes X(G)$ and $I(G) \otimes Z(G) \mapsto Z(G) \otimes Z(G)$.

which shows that R correctly implements logical $\overline{R}_n$ when the gauge qubits are in the state $|g_X\rangle$. It remains to show that R implements $\overline{R}_n$ for arbitrary states of the gauge qubits. However, all other states of the gauge qubits can be reached from $|g_X\rangle$ by application of linear combinations of Z-type operators from $\mathcal{G} \setminus \mathcal{S}$, which all commute with R (since it is diagonal in the Z-basis). Therefore for any state $|g\rangle$ of the gauge qubits, it must be that $R : |\overline{0}\rangle|g\rangle \mapsto |\overline{0}\rangle|g\rangle, |\overline{1}\rangle|g\rangle \mapsto e^{\frac{2\pi i}{2^n}}|\overline{1}\rangle|g\rangle$, verifying that R implements the bare logical gate $\overline{R}_n$.

It may not be obvious that there exists a set $T \subset Q$ satisfying (27) for a given code. Later we will find such a T for color codes in d dimensions, with $n \leq d$. Condition (27) can be inferred from the following condition

$$\left| T \cap \bigcap_{i=1}^m G_i \right| \equiv \left| T^c \cap \bigcap_{i=1}^m G_i \right| \mod 2^{n-m+1}, \quad (35)$$

where $m = 1, \ldots, n$ and $\{X(G_1), \ldots, X(G_m)\}$ is any subset of the X-type generators of the gauge group $\mathcal{G}$. To see the implication (35) $\implies$ (27) notice, that for any $X(G) \in \mathcal{G}$, we can write $X(G)$ as a product of generators, namely $X(G) = \prod_{i=1}^m X(G_i)$. Then

$$G = G_1 \veebar G_2 \veebar \ldots \veebar G_m, \quad (36)$$

where we used the symmetric difference of sets, $A \veebar B := (A \setminus B) \cup (B \setminus A)$. Using the Inclusion-Exclusion Principle for symmetric difference[‡‡] we obtain

$$|T \cap G| = |T \cap (G_1 \veebar G_2 \veebar \ldots \veebar G_m)| \quad (37)$$

$$= \sum_i |T \cap G_i| - 2 \sum_{i \neq j} |T \cap (G_i \cap G_j)| +$$

$$4 \sum_{i \neq j \neq k} |T \cap (G_i \cap G_j \cap G_k)| - \ldots$$

$$+ (-2)^{m-1}|T \cap (G_1 \cap G_2 \cap \ldots \cap G_m)|, (38)$$

and a similar expression for $|T^c \cap G|$. Clearly, if condition (35) holds, then $|T \cap G| - |T^c \cap G| \equiv 0 \mod 2^n$, showing (27). Moreover, condition (35) is easier to verify than condition (27), since we only need to check it for the X-type generators of $\mathcal{G}$, rather than for every X-type element of $\mathcal{G}$.

We can summarize the discussion of the implementation of transversal $\overline{R}_n$ in the following lemma

Lemma 6 (Sufficient Condition). *Consider a CSS subsystem code encoding one logical qubit. Let the code be defined on a set of physical qubits Q, where $|Q|$ is odd and with bare logical operators $\overline{X} = X(Q)$ and $\overline{Z} = Z(Q)$. If there exists $T \subset Q$, such that for any $m = 1, \ldots, n$:*

$$\left| T \cap \bigcap_{i=1}^m G_i \right| \equiv \left| T^c \cap \bigcap_{i=1}^m G_i \right| \mod 2^{n-m+1}, \quad (39)$$

[‡‡] For sets $A_1, A_2, \ldots, A_m$, we have $|A_1 \veebar A_2 \veebar \ldots \veebar A_m| = \sum_i |A_i| - 2 \sum_{i \neq j} |A_i \cap A_j| + \ldots + (-2)^{m-1}|A_1 \cap A_2 \cap \ldots \cap A_m|$.

for every subset $\{X(G_1), \ldots, X(G_m)\}$ of the X-type gauge generators of the code, then

$$R = R_n^k(T) R_n^{-k}(T^c) \qquad (40)$$

implements logical $\overline{R}_n$, where k is a solution to $k(|T| - |T^c|) \equiv 1 \mod 2^n$ and $T^c = Q \setminus T$.

C. Transversal implementation of $\overline{R}_n$ in color code

Here we show how to implement the logical gate $\overline{R}_n$ transversely in the color code $CC_{\mathcal{L}}(x, z)$, for any integer $n \leq d/(d-1-z)$, where $d = \dim(\mathcal{L})$. One applies $R = R_n^k(T) R_n^{-k}(T^c)$ for some integer k, where T and its compliment $T^c = Q \setminus T$ correspond to the bipartite decomposition of qubits Q specified in the (Bipartition of Qubits) Lemma 5. We make use of the following property

Lemma 7 (Property of T). *For any m-simplex σ in $\mathcal{L} \setminus \partial \mathcal{L}$ with $m < d$*

$$|T \cap \mathcal{Q}(\sigma)| = |T^c \cap \mathcal{Q}(\sigma)|. \qquad (41)$$

Proof. By the choice of the set T, every $(d-1)$-simplex δ has one qubit in T, and one qubit in $T^c = Q \setminus T$, which is equivalent to $|T \cap \mathcal{Q}(\delta)| = |T^c \cap \mathcal{Q}(\delta)|$. Using the (Disjoint Union) Lemma 3, we can decompose the set of qubits $\mathcal{Q}(\sigma)$ supported on an m-simplex σ, where $m < d$, as a disjoint union of qubits supported on $(d-1)$-simplices colored with a chosen set of d colors $C \supset \mathrm{color}(\sigma)$, and then we immediately obtain

$$|T \cap \mathcal{Q}(\sigma)| - |T^c \cap \mathcal{Q}(\sigma)| = \qquad (42)$$

$$\sum_{\substack{\delta \supset \sigma \\ \delta \in \Delta'_{d-1}(\mathcal{L}) \\ \mathrm{color}(\delta) = C}} |T \cap \mathcal{Q}(\delta)| - |T^c \cap \mathcal{Q}(\delta)| = 0, \qquad (43)$$

which shows the (Property of T) Lemma 7.

$\square$

Note that (39) in the (Sufficient Condition) Lemma 6 follows form the (Property of T) Lemma 7. To see this, observe first that every stabilizer generator $X(\delta_i)$ is supported on a $(d-2-z)$-simplex δ_i, thus $G_i = \mathcal{Q}(\delta_i)$ and for $m = 1, \ldots, n$ we obtain

$$\bigcap_{i=1}^{m} \mathcal{Q}(\delta_i) = \emptyset \quad \text{or} \quad \bigcap_{i=1}^{m} \mathcal{Q}(\delta_i) = \mathcal{Q}(\tau), \qquad (44)$$

where τ is a simplex colored with colors $C = \bigcup_{i=1}^{m} \mathrm{color}(\delta_i)$, such that $\tau \supset \delta_1, \ldots, \delta_m$. The case of an empty intersection is trivial. Since $|\mathrm{color}(\delta_i)| = d-1-z$, then obviously $|C| \leq m(d-1-z) \leq n(d-1-z)$, and for τ to be at most a $(d-1)$-simplex, we need $n \leq d/(d-1-z)$.

Using the (Property of T) Lemma 7 we obtain that for any $m = 1, \ldots, n$:

$$\left| T \cap \bigcap_{i=1}^{m} \mathcal{Q}(\delta_i) \right| - \left| T^c \cap \bigcap_{i=1}^{m} \mathcal{Q}(\delta_i) \right| = \qquad (45)$$

$$|T \cap \mathcal{Q}(\tau)| - |T^c \cap \mathcal{Q}(\tau)| = 0, \qquad (46)$$

which implies (39). The (Sufficient Condition) Lemma 6 implies that R implements the logical $\overline{R}_n$. In particular, one can implement $\overline{R}_d$ using the code $CC_d(0, d-2)$, since $z = d-2$, and thus $\lfloor d/(d-1-z) \rfloor = d$.

V. UNIVERSAL TRANSVERSAL GATES WITH COLOR CODES

A finite set of gates which is universal can be used to implement any logical unitary, with arbitrary precision. In particular, due to the Solovay-Kitaev [7, 8] theorem, the number of applied gates scales poly-logarithmically with the precision of approximation. Note that the set $\{\overline{H}, \overline{\mathrm{CNOT}}, \overline{R}_n\}$ is universal for any integer $n > 2$.

In this section, we show how to achieve a universal transversal gate set with color codes by using the technique of *gauge fixing* to switch between different codes. This technique allows one to take advantage of the transversally implementable gates for different color codes. We first illustrate the method with a simple example of two 15-qubit codes [18, 19]. Then, we define a partial order between color codes. One can switch between color codes which are comparable with respect to the partial order to implement a universal gate set in three or higher dimensions.

A. Switching between codes using gauge fixing

First, let us define matrices H_1 and H_2 given by

$$H_1 = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{pmatrix}, \qquad (47)$$

$$H_2 = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}. \qquad (48)$$

These matrices have a property that each row in H_1 is orthogonal to every row in H_1 and H_2 (modulo 2). Moreover, for a binary matrix M, we define M^X to be a matrix obtained from M by the following substitutions, $0 \mapsto I$ and $1 \mapsto X$. Similarly for M^Z, we substitute $0 \mapsto I$ and $1 \mapsto Z$. Let $\mathscr{C}_A$ be the stabilizer code with the stabilizer group $\mathcal{S}_A$ generated by rows of H_1^X, H_1^Z and H_2^Z, which we denote by

$$\mathcal{S}_A = \langle H_1^X, H_1^Z, H_2^Z \rangle. \qquad (49)$$

Let $\mathscr{C}_B$ be the subsystem code with the stabilizer group $\mathcal{S}_B$ and the gauge group $\mathcal{G}_B$ chosen as follows

$$\mathcal{S}_B = \langle H_1^X, H_1^Z \rangle, \quad \mathcal{G}_B = \langle H_1^X, H_2^X, H_1^Z, H_2^Z \rangle. \quad (50)$$

We can consider both codes $\mathscr{C}_A$ and $\mathscr{C}_B$ to be defined on the same 15 physical qubits. One can check that $\mathscr{C}_A$ represents the $[[15, 1, 3]]$ quantum Reed-Muller (stabilizer) code [19, 34, 35] and $\mathscr{C}_B$ is a $[[15, 1, 3]]$ (subsystem) code, which can be thought of as the $[[15, 7, 3]]$ Hamming code, with six of the seven logical qubits treated as gauge qubits. Note also that $\mathcal{S}_B \subset \mathcal{G}_A = \mathcal{S}_A$ and $\mathcal{G}_B$ has X- and Z-type generators supported on the same qubits (i.e. $\mathscr{C}_B$ is a self-dual subsystem code).

Since the X-type generators of $\mathcal{S}_B$ coincide with the X-type generators of $\mathcal{S}_A$, the codewords of $\mathscr{C}_A$ and $\mathscr{C}_B$ are the same when the latter has a gauge state $|g_Z\rangle$. In other words, codewords $|\bar{0}\rangle$, $|\bar{1}\rangle$ for $\mathscr{C}_A$ are the same as codewords $|\bar{0}\rangle|g_Z\rangle$, $|\bar{1}\rangle|g_Z\rangle$ for $\mathscr{C}_B$, as defined in Eqs. (25) and (26). On the other hand the codewords $|\bar{0}\rangle|g_X\rangle$, $|\bar{1}\rangle|g_X\rangle$ for $\mathscr{C}_B$ (as defined in Eqs. (23) and (24)), are not valid codewords for $\mathscr{C}_A$.

Now we show that $R_3^{\otimes 15}$ implements $\overline{R}_3$ transversally in $\mathscr{C}_A$. Consider any three of the four X-type generators for $\mathcal{G}_A$, and specify their support on subsets of qubits G_1, G_2, G_3, which correspond to rows of H_1. One can verify that $|G_a| = 8 \equiv 0 \mod 2^3$, $|G_a \cap G_b| = 4 \equiv 0 \mod 2^2$, and $|G_a \cap G_b \cap G_c| = 2 \equiv 0 \mod 2$, where $\{a, b, c\} = \{1, 2, 3\}$. Therefore by the (Sufficient Condition) Lemma 6, and by setting T to be an empty set, $T = \emptyset$, we see that $R_3^{\otimes 15}$ implements $\overline{R}_3$ transversally in the code $\mathscr{C}_A$. In contrast for the code $\mathscr{C}_B$, the extra X-type generators in $\mathcal{G}_B \setminus \mathcal{G}_A$ do not satisfy these conditions, and thus one cannot show that $\overline{R}_3$ is implemented transversally in $\mathscr{C}_B$.

It is straightforward to verify that $\overline{H}$ is implemented transversally by $H^{\otimes 15}$ in $\mathscr{C}_B$. It swaps X and Z on any physical qubit, and therefore acts on the representative states as $H^{\otimes 15} : |\psi\rangle|g_Z\rangle \mapsto (\overline{H}|\psi\rangle)|g_X\rangle$. Since the state of the gauge qubits has changed, $H^{\otimes 15}$ is a dressed implementation of $\overline{H}$ in $\mathscr{C}_A$. Clearly, $H^{\otimes 15}$ does not implement $\overline{H}$ in $\mathscr{C}_A$, since it takes the state $|\psi\rangle|g_Z\rangle \in \mathscr{C}_A$ to $(\overline{H}|\psi\rangle)|g_X\rangle \notin \mathscr{C}_A$.

To implement $\overline{H}$ fault-tolerantly in $\mathscr{C}_A$, we use the technique of *gauge fixing*. First, one should apply $H^{\otimes 15}$, resulting in mapping $|\psi\rangle|g_Z\rangle$ to $(\overline{H}|\psi\rangle)|g_X\rangle$, which is a codeword of $\mathscr{C}_B$, but not of $\mathscr{C}_A$. Then, to switch from code $\mathscr{C}_B$ to $\mathscr{C}_A$, one should sequentially measure each of the six Z-type stabilizer generators generated by rows of H_2^Z, i.e. those in $\mathcal{S}_A \setminus \mathcal{S}_B$. Note that it is possible to fault-tolerantly measure the stabilizer generators [8]. If the measurement reveals that a particular Z-type generator is not satisfied, then one should apply an X-type Pauli operator which commutes with all generators in H_2^Z and H_1^Z, except for the violated stabilizer generator (with which it must anti-commute). Such an X-type Pauli operator always exists. Following this, the Z-type generator will no longer be violated. Therefore, after this procedure is carried out for all six generators in H_2^Z, the

state will have changed from $(\overline{H}|\psi\rangle)|g_X\rangle$ to $(\overline{H}|\psi\rangle)|g_Z\rangle$, as required. Specifically, we use the term *gauge fixing* to refer to the process of measuring and setting the gauge qubits to a desired state, without affecting the logical qubits.

To recap, in the $[[15, 1, 3]]$ Reed-Muller code $\mathscr{C}_A$, one can implement $\overline{H}$ fault-tolerantly with the following procedure

$$|\psi\rangle|g_Z\rangle \xmapsto{H^{\otimes 15}} (\overline{H}|\psi\rangle)|g_X\rangle \xmapsto{\text{gauge fixing}} (\overline{H}|\psi\rangle)|g_Z\rangle. \quad (51)$$

In combination with the transversal gates of $\mathscr{C}_A$, this allows one to implement a fault-tolerant universal gate set $\{\overline{H}, \overline{\text{CNOT}}, \overline{R}_3\}$. We will repeat essentially the same procedure for color codes later.

B. Partial order of color codes

Given a d-dimensional lattice $\mathcal{L}$, $\dim \mathcal{L} = d$, satisfying Conditions 1 and 2 in Section III B, we can catalog all color codes defined on $\mathcal{L}$. Namely, a pair of integers $x, z \geq 0$, such that $x + z \leq d - 2$, corresponds to a color code, denoted as $CC_{\mathcal{L}}(x, z)$, with X- and Z-type gauge generators supported on $(d - 2 - z)$- and $(d - 2 - x)$- simplices. Note that the X- and Z-type stabilizer generators of $CC_{\mathcal{L}}(x, z)$ are supported on x-simplices and z-simplices, respectively. In two dimensions, $d = 2$, there is only one color code, $CC_2(0, 0)$ — a stabilizer code, with both X- and Z-type stabilizer generators supported on 0-simplices, whereas in three dimensions, $d = 3$, there are three color codes, $CC_3(1, 0)$, $CC_3(0, 1)$ — stabilizer codes, and $CC_3(0, 0)$ — a subsystem code.

One can define a partial order for subsystem color codes defined on the same lattice $\mathcal{L}$ if each codeword of code $\mathscr{C}$ is also a codeword of the other code $\mathscr{C}'$. In particular, we say that $\mathscr{C} \succ \mathscr{C}'$ holds if

- $\mathscr{C}$ and $\mathscr{C}'$ encode the same number of logical qubits, with identical bare logical Pauli operators,

- the gauge group $\mathcal{G}$ of $\mathscr{C}$ is contained in the gauge group $\mathcal{G}'$ of $\mathscr{C}'$, $\mathcal{G} \subset \mathcal{G}'$.

Note that $\mathcal{G} \subset \mathcal{G}'$ implies $\mathcal{S}' \subset \mathcal{S}$, thus any codeword of $\mathscr{C}$ is also a codeword of $\mathscr{C}'$, and since the bare Pauli operators for the logical qubit are the same in both codes, it actually represents the same logical codeword in both codes. Observe, that the partial order we have just defined can be succinctly expressed as

$$CC_{\mathcal{L}}(x, z) \succ CC_{\mathcal{L}}(x', z') \iff x \geq x' \wedge z \geq z', \quad (52)$$

as illustrated in Fig. 3. This follows from the observation that due to the (Disjoint Union) Lemma 3 the X-type gauge generators of $CC_{\mathcal{L}}(x, z)$, which are supported on $(d - 2 - z)$-simplices, can be expressed as the product of the X-type gauge generators of $CC_{\mathcal{L}}(x', z')$ supported on $(d - 2 - z')$-simplices, since $z \geq z'$. Similarly for Z-type gauge generators. We represent the family of color codes in Fig. 3, and show their partial order using arrows.
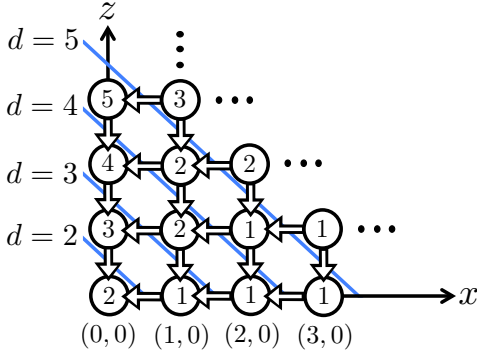
FIG. 3. (Color online) Family of color codes. For a given lattice $\mathcal{L}$, only color codes below the dth diagonal line can be realized, where $d = \dim \mathcal{L}$ and the point (x, z) corresponds to the color code $CC_{\mathcal{L}}(x, z)$. This constraint holds, since x and z have to satisfy $x + z \leq d - 2$. An arrow from code $\mathscr{C}$ to $\mathscr{C}'$ indicates partial order between them, $\mathscr{C} \succ \mathscr{C}'$. The number placed at (x, z) indicates the maximum gate $\overline{R}_n$ which can be implemented transversally with the stabilizer color code $CC_d(x, z)$, with $d = x + z + 2$, resulting in $n = \lfloor d/(d - 1 - z) \rfloor$.

C. Universal fault-tolerant gate set in color codes

Here we apply the techniques just discussed to color codes defined on the same lattice $\mathcal{L}$. One can switch back and forth between two codes which are *comparable*, $CC_{\mathcal{L}}(x, z) \succ CC_{\mathcal{L}}(x', z')$, as follows

- $CC_{\mathcal{L}}(x, z) \mapsto CC_{\mathcal{L}}(x', z')$: one does nothing, since codewords of $CC_{\mathcal{L}}(x, z)$ are codewords of $CC_{\mathcal{L}}(x', z')$,

- $CC_{\mathcal{L}}(x', z') \mapsto CC_{\mathcal{L}}(x, z)$: one can view the codewords of $CC_{\mathcal{L}}(x, z)$ as those for $CC_{\mathcal{L}}(x', z')$ with the additional gauge qubits present in $CC_{\mathcal{L}}(x, z)$ set to a particular state. To switch, one fixes the state of the additional gauge qubits to the appropriate state.

Given a three-dimensional lattice $\mathcal{L}$, $\dim \mathcal{L} = 3$, one can implement a universal gate set starting with the code $CC_{\mathcal{L}}(0, 1)$. As explained earlier, one can transversally perform the logical $\overline{\text{CNOT}}$ and $\overline{R}_3$ on that code. To form a universal gate set, it suffices to also implement logical $\overline{H}$. This gate cannot be implemented transversally in $CC_{\mathcal{L}}(0, 1)$, but can be in $CC_{\mathcal{L}}(0, 0)$. Note that $CC_3(0, 0) \prec CC_3(0, 1)$, therefore any codeword in $CC_3(0, 1)$ is a valid codeword in $CC_3(0, 0)$. In particular, we can think of $|\psi\rangle \in CC_3(0, 1)$ as $|\psi\rangle|g\rangle \in CC_3(0, 0)$,

where $|g\rangle$ is a state of the gauge qubits of $CC_3(0, 0)$. By applying $H(Q)$ we perform the logical $\overline{H}$ on the logical qubits of $CC_3(0, 0)$, which also changes the state of the gauge qubits, namely

$$H(Q) \left(|\psi\rangle|g\rangle \right) = \left(\overline{H}|\psi\rangle \right) |g'\rangle. \tag{53}$$

Note that the resulting codeword $\left(\overline{H}|\psi\rangle \right) |g'\rangle \in CC_3(0, 0)$ is not a valid codeword of $CC_3(0, 1)$, since the gauge qubits are in the state $|g'\rangle \neq |g\rangle$. To return to $CC_3(0, 1)$, one needs to *fix the gauge qubits* to the correct state, namely $|g'\rangle \mapsto |g\rangle$, and we obtain a codeword $\overline{H}|\psi\rangle|g\rangle \in CC_3(0, 1)$. Since $CC_3(0, 1)$ is a stabilizer code, it is possible to measure and correct the violated stabilizers in a fault-tolerant way, just as in Section V A. Therefore, to fix the gauge, one should first measure all Z-type stabilizer generators supported on 1-simplices, and then apply the appropriate X-type Pauli operators in order to correct any violated stabilizer generators. After this, assuming no errors have occurred, all the stabilizer generators for $CC_3(0, 1)$ are satisfied.

To summarize, we can perform the logical $\overline{H}$ on $CC_3(0, 1)$ by first applying $H(Q)$ and subsequently fixing the gauge to return to the code space of $CC_3(0, 1)$,

$$|\psi\rangle|g\rangle \xmapsto{H(Q)} \left(\overline{H}|\psi\rangle \right) |g'\rangle \xmapsto{\text{gauge fixing}} \left(\overline{H}|\psi\rangle \right) |g\rangle. \tag{54}$$

Since $\overline{\text{CNOT}}$ and $\overline{R}_3$ can be performed transversally in $CC_3(0, 1)$, one can fault-tolerantly implement a universal gate-set $\{\overline{H}, \overline{\text{CNOT}}, \overline{R}_3\}$ in $CC_3(0, 1)$. This procedure can be directly generalized to fault-tolerantly implement the universal gate set $\{\overline{H}, \overline{\text{CNOT}}, \overline{R}_d\}$ with the code $CC_d(0, d - 2)$ in d dimensions.

VI. ACKNOWLEDGEMENTS

[1] A. Y. Kitaev, Ann. Phys. **303**, 2 (2003).
[2] M. A. Levin and X.-G. Wen, Phys. Rev. B **71**, 045110 (2005).
[3] H. Bombín and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).
[4] P. H. Bonderson, S. Das Sarma, M. H. Freedman, and C. Nayak, (2010), arXiv:1003.2856.
[5] P. Shor, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE Comput. Soc. Press, 1996) pp. 56–65.

[6] J. Preskill, Proc. Roy. Soc. Lond. **454**, 385 (1998).
[7] A. Y. Kitaev, Russ. Math. Surv. **52**, 1191 (1997).
[8] M. Nielsen and I. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, 2010).
[9] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[10] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, Phys. Rev. A **86**, 032324 (2012).
[11] C. Nayak, S. H. Simon, A. Stern, M. Freedman, and S. Das Sarma, Rev. Mod. Phys. **80**, 1083 (2008).
[12] D. Beckman, D. Gottesman, M. Nielsen, and J. Preskill, Phys. Rev. A **64**, 052309 (2001).
[13] M. H. Devoret and R. J. Schoelkopf, Science **339**, 1169 (2013).
[14] D. Wecker, B. Bauer, B. K. Clark, M. B. Hastings, and M. Troyer, Phys. Rev. A **90**, 022305 (2014).
[15] H. Bombín, (2013), arXiv:1311.0879.
[16] B. Eastin and E. Knill, Phys. Rev. Lett. **102**, 110502 (2009).
[17] T. Jochym-O'Connor and R. Laflamme, Phys. Rev. Lett. **112**, 010505 (2014).
[18] A. Paetznick and B. W. Reichardt, Phys. Rev. Lett. **111**, 090505 (2013).
[19] J. T. Anderson, G. Duclos-Cianci, and D. Poulin, Phys. Rev. Lett. **113**, 080501 (2014).
[20] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).
[21] A. Calderbank, E. Rains, P. Shor, and N. Sloane, Phys. Rev. Lett. **78**, 405 (1997).
[22] S. Bravyi and R. König, Phys. Rev. Lett. **110**, 170503 (2013).
[23] D. Gottesman and I. L. Chuang, Nature **402**, 390 (1999).
[24] F. Pastawski and B. Yoshida, (2014), arXiv:1408.1720.
[25] M. E. Beverland, R. König, F. Pastawski, J. Preskill, and S. Sijher, (2014), arXiv:1409.3898.
[26] H. Bombín, in *Topological Codes*, edited by D. A. Lidar and T. A. Brun (Cambridge University Press, 2013).
[27] H. Bombín and M. Martin-Delgado, Phys. Rev. B **75**, 075103 (2007).
[28] D. Gottesman, (1998), arXiv:quant-ph/9807006.
[29] S. Aaronson and D. Gottesman, Phys. Rev. A **70**, 052328 (2004).
[30] R. J. Wilson, *Introduction to Graph Theory* (Longman, 1996).
[31] A. Hatcher, *Algebraic Topology* (Cambridge University Press, 2002).
[32] D. Poulin, Phys. Rev. Lett. **95**, 230504 (2005).
[33] D. Bacon, Phys. Rev. A **73**, 012340 (2006).
[34] F. MacWilliams and N. Sloane, *The theory of error-correcting codes* (North-Holland, 1977).
[35] A. Steane, IEEE Trans. Info. Th. **45**, 1701 (1999).
[36] D. Nigg, M. Mller, E. A. Martinez, P. Schindler, M. Hennrich, T. Monz, M. A. Martin-Delgado, and R. Blatt, Science **345**, 302 (2014).

Appendix: Examples of color codes

1. Construction of a lattice in d dimensions

A recipe to obtain a lattice $\mathcal{L}$ satisfying Conditions 1 and 2 in Section III B required to define color codes in d dimensions is as follows (see Fig. 4 for an example in $d = 2$).
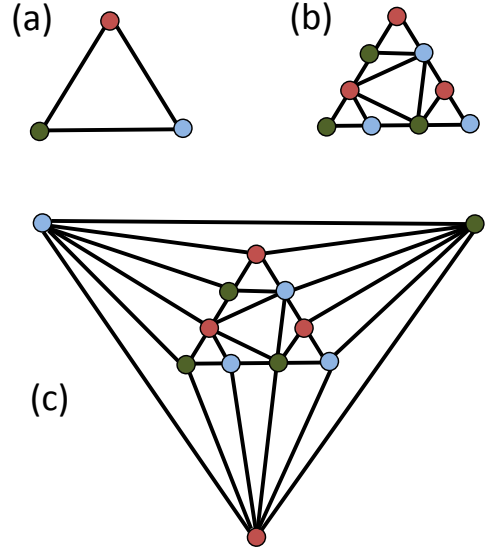

FIG. 4. (Color online) Construction of a lattice for a color code in 2D. (a) Take a 2-simplex δ, with vertices colored in red, green and blue. (b) Divide δ into "smaller" simplices with matching colors. This is a 3-colorable homogeneous simplicial 2-complex $\mathcal{K}$. (c) Place $\mathcal{K}$ inside a 2-simplex τ and attach 2-simplices between τ and $\mathcal{K}$. The resulting homogeneous simplicial 2-complex $\mathcal{L}$ is 3-colorable, and thus we can define a color code on the lattice $\mathcal{L}$.

1. Start with a d-simplex δ, with vertices which are colored with $d + 1$ colors $\mathbb{Z}_{d+1}$.

2. Construct a homogeneous simplicial d-complex $\mathcal{K}$ from δ by dividing k-faces of δ into k-simplices. We also require that the coloring is preserved, i.e. every k-face $\sigma \subset \delta$ colored with $C = \mathrm{color}(\sigma)$ is divided into k-simplices colored with C and the whole complex $\mathcal{K}$ is $(d + 1)$-colorable.

3. Place the d-complex $\mathcal{K}$ inside a d-simplex τ colored with $\mathbb{Z}_{d+1}$.

4. For every k-face $\rho \subsetneq \tau$ and for every $(d - k - 1)$-simplex $\omega \subset \mathcal{K}$ obtained from a $(d-k-1)$-face $\sigma \subset \delta$ with complementary colors, $\mathrm{color}(\omega) = \mathbb{Z}_{d+1} \setminus \mathrm{color}(\rho)$, attach a d-simplex spanned by ρ and ω.

5. Choose $\mathcal{L}$ to be the collection of all d-simplices added in Step 4, together with simplices belonging to $\mathcal{K}$ and τ. This can be used to define a color code on the lattice $\mathcal{L}$ as specified in Section III.

Note that in the above recipe, step 2 is not fully specified. Any homogeneous simplicial d-complex $\mathcal{K}$ obtained from a d-simplex δ will work, as long as $\mathcal{K}$ is $(d + 1)$-colorable. Such lattices always exist — below we give an explicit example of a family of lattices in any dimension

$d \geq 2$. Following steps 3–5, we always obtain a lattice on which we can define a color code in d dimensions.

Our explicit family of lattices results in (fractal) color codes in d dimensions, for which there is an explicit recipe for $\mathcal{K}$. The resulting codes neither have spatially local generators nor have macroscopic distance, and therefore do not result in topological color codes. The prescription is as follows.

1. The first member is defined on the lattice $\mathcal{L}_1$, obtained from the recipe by setting $\mathcal{K}$ to be a d-simplex.

2. The $i + 1$ member of the family is defined on the lattice $\mathcal{L}_{i+1}$, obtained from the recipe by setting $\mathcal{K} = \mathcal{L}_i$.

The first three members of the family of the two-dimensional (fractal) color codes given by this construction are illustrated in Fig. 5.

In Ref. [15], a systematic construction in two and three dimensions for families of color codes with spatially local generators is presented. In two dimensions, $\mathcal{K}$ is chosen to be part of a triangular lattice (as in Fig. 4), whereas in three dimensions, $\mathcal{K}$ is part of a BCC lattice. Bombín's constructions result in topological color codes.
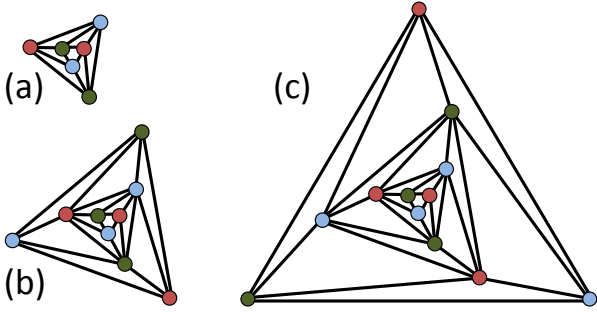

FIG. 5. (Color online) The family of (fractal) color codes in two dimensions. The first three members of the family — two-dimensional color codes encoding one logical qubits using (a) 7, (b) 13 and (c) 19 physical qubits.

2. Quantum Reed-Muller codes as color codes

There exists a family of codes known as the quantum Reed-Muller codes [19, 34, 35]. Here we are concerned with the subfamily of quantum Reed-Muller codes with members labeled uniquely by an integer $m \geq 3$ with parameters $[[2^m - 1, 1, 3]]$, i.e. encoding one logical qubit into $2^m - 1$ physical qubits, with a distance of three. We denote by $\text{QRM}(m)$ the mth member of this subfamily. These codes are defined in terms of matrices M_i satisfying the recursion relations

$$M_1 = (1), \quad M_{i+1} = \begin{pmatrix} M_i & 0 & M_i \\ 0 \ldots 0 & 1 & 1 \ldots 1 \end{pmatrix}. \qquad \text{(A.1)}$$

Note that the set of columns of M_m is the set of all non-zero binary vectors of length m. By $M_m^\perp$ we denote a matrix dual to M_m, namely a matrix with rows being a basis of the kernel of M_m. Clearly, $M_m(M_m^\perp)^T = 0$. We can define $\text{QRM}(m)$ as the stabilizer code with the stabilizer group $\mathcal{S}_m$ generated by rows of M_m and $M_m^\perp$ with 0's and 1's replaced by I's and X's or Z's, namely

$$\mathcal{S}_m = \langle M_m^X, (M_m^\perp)^Z \rangle. \qquad \text{(A.2)}$$

We now show that $\text{QRM}(m)$ is the same as the stabilizer color code $CC_{m-1}(0, m - 3)$ obtained from the construction described in Appendix 1 by taking the simplicial complex $\mathcal{K}$ to be a $(m - 1)$-simplex δ, $\mathcal{K} = \delta$. In other words, $\text{QRM}(m)$ is equal to the first member of the (fractal) color code family in $m-1$ dimensions (see Fig. 5 (a) for $m = 3$ case). In particular, $\text{QRM}(3)$ is Steane's 7-qubit code and $\text{QRM}(4)$ is the 15-qubit Reed-Muller code.
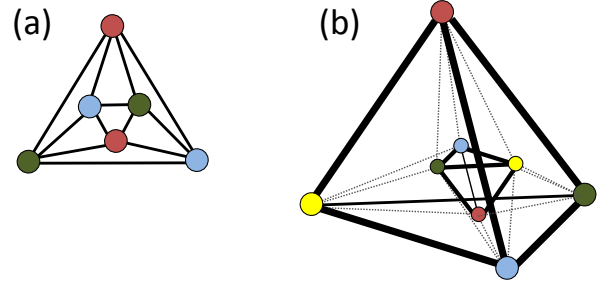

FIG. 6. (Color online) Quantum Reed-Muller code $\text{QRM}(m)$ as a special case of a (stabilizer) color code $CC_{m-1}(0, m - 3)$ for (a) $m = 3$ — Steane's 7-qubit code, and (b) $m = 4$ — the 15-qubit Reed-Muller code. Steane's code with all the possible transversal gates have recently been implemented experimentally [36].

To prove this equivalence, it is sufficient to show that there is a one-to-one identification of physical qubits of $\text{QRM}(m)$ with those of $CC_{m-1}(0, m - 3)$ such that the logical Pauli operators $\overline{X}$ and $\overline{Z}$ are identical, and that the X-type stabilizer generators are identical. Note that this completely specifies the stabilizer group $\mathcal{S}$, since the Z-type generator matrix is a dual to the X-type generator matrix. In particular, we show that the X-type generator matrix M_m' for $CC_{m-1}(0, m - 3)$ is the same as M_m for $\text{QRM}(m)$ up to a permutation of columns, which corresponds to relabelling of physical qubits.

Using the construction described in Appendix 1, and taking the simplicial complex $\mathcal{K} = \delta$, where δ is an $(m-1)$-simplex, results in a lattice $\mathcal{L}$, with $\dim \mathcal{L} = m-1$. The total number of $(m - 1)$-simplices in $\mathcal{L}$ is $2^m - 1$. This is because we attach $(m - 1)$-simplices between every $(k - 1)$-face $\rho \subset \tau$, for every $1 \leq k \leq m - 1$, and the $(m - k - 1)$-face $\sigma \subset \delta$ colored with the complementary colors, $\text{color}(\sigma) = \mathbb{Z}_{d+1} \setminus \text{color}(\rho)$. We can pick a subset of k vertices of τ in $\binom{m}{k}$ different ways and thus the number of newly attached $(m - 1)$-simplices is

$\binom{m}{1} + \binom{m}{2} + \ldots + \binom{m}{m-1} = 2^m - 2$. Therefore, including a qubit placed at δ, there are exactly $2^m - 1$ physical qubits in $CC_{\mathcal{L}}(0, m-3)$. On the other hand, there are exactly m vertices in $\mathcal{L} \setminus \partial\mathcal{L}$, and thus there are m distinct X-type stabilizer generators in $CC_{\mathcal{L}}(0, m-3)$. The weight of a column in M'_m, corresponding to a qubit supported on a $(m-1)$-simplex π, is given by the number of X-type stabilizer generators supported on that qubit, i.e. the number of vertices belonging to π but not to $\partial\mathcal{L}$.

There are exactly $\binom{m}{m-k}$ $(m-1)$-simplices containing k vertices not belonging to $\partial\mathcal{L}$ and each of them contains a different set of k vertices. Thus, there are $\binom{m}{k}$ different columns of weight k in M'_m and the only way this can occur is if the columns of M'_m are the set of all non-zero binary vectors of length m. Thus, up to a relabeling of physical qubits, M'_m and M_m are identical. Also note that the logical operators of both codes are $\overline{X} = X(Q)$ and $\overline{Z} = Z(Q)$. Therefore the codes are the same.