Finding the disjointness of stabilizer codes is NP-complete

John Bostanci[1] and Aleksander Kubica[2, 1, 3, 4]

[1]*Institute for Quantum Computing, University of Waterloo, Waterloo, ON N2L 3G1, Canada*
[2]*Perimeter Institute for Theoretical Physics, Waterloo, ON N2L 2Y5, Canada*
[3]*AWS Center for Quantum Computing, Pasadena, CA 91125, USA*
[4]*California Institute of Technology, Pasadena, CA 91125, USA*

The disjointness of a stabilizer code is a quantity used to constrain the level of the logical Clifford hierarchy attainable by transversal gates and constant-depth quantum circuits. We show that for any positive integer constant c, the problem of calculating the c-disjointness, or even approximating it to within a constant multiplicative factor, is NP-complete. We provide bounds on the disjointness for various code families, including the CSS codes, concatenated codes and hypergraph product codes. We also describe numerical methods of finding the disjointness, which can be readily used to rule out the existence of any transversal gate implementing some non-Clifford logical operation in small stabilizer codes. Our results indicate that finding fault-tolerant logical gates for generic quantum error-correcting codes is a computationally challenging task.

Designing fault-tolerant schemes is an essential step toward scalable universal quantum computation [1–4]. To protect quantum information from the detrimental effects of noise one typically encodes it into an quantum error-correcting code. In addition to reliably storing quantum information, one also seeks to perform fault-tolerant logical operations on the encoded information.

One of the simplest ways to realize fault-tolerant logical operations is via transversal gates, which act independently on individual physical qubits and thus do not spread errors in an uncontrollable way. Recently, many works have been devoted to transversal gates implementing non-Clifford logical operations in topological quantum codes [5–11] and the consequent universal quantum computation schemes [12–17]. Transversal gates also prove useful for magic state distillation [18–20], as they form the backbone of many distillation protocols; see Ref. [21] and the references therein.

Logical operations implemented via transversal gates are somewhat limited. Namely, the computational universality of transversal gates is incommensurate with the capability of the underlying code to correct errors, as exemplified by the Eastin-Knill theorem [22, 23], and its approximate versions [24–26]. More generally, bounded-spread logical operators, which propagate errors in a benign way and include constant-depth quantum circuits and locality-preserving operators, are also computationally limited [27–31].

Although systematic approaches to finding transversal logical gates for generic quantum error-correcting codes are not known, for stabilizer codes [32] we can rule out the possibility of implementing certain logical operations. Namely, if M is the level of the logical Clifford hierarchy [33] attainable by transversal logical gates, then the following upper bound holds

$$M \leq \lfloor \log_\Delta (d_\uparrow / d_\downarrow) \rfloor + 2, \tag{1}$$

given the min-distance $d_\downarrow > 1$, max-distance $d_\uparrow$ and disjointness Δ of the stabilizer code [30]. The disjointness, roughly speaking, captures the maximal number of mostly non-overlapping representatives of any given non-trivial logical Pauli operator. Until now, however, the problem of finding the disjointness as well as its computational hardness have not been explored.

In our work we focus on the problem of finding the disjointness of stabilizer codes, which serves as a proxy to understanding what are the admissible fault-tolerant logical gates. First, in Section II we show that for any positive integer c it is NP-complete to calculate the c-disjointness, as well as to approximate it to within any constant multiplicative factor. Our result thus indicates that finding fault-tolerant logical gates that can be implemented with generic quantum error-correcting codes is a computationally challenging task. Then, in Section III we discuss numerical methods of finding the disjointness, which we illustrate with the example of the $[[14, 3, 3]]$ stabilizer code [34]. We also provide a strengthening of the bound in Eq. (1), which subsequently rules out the existence of any transversal logical non-Clifford gate in the aforementioned $[[14, 3, 3]]$ stabilizer code. Lastly, in Section IV we provide bounds on the disjointness for various code families, including the CSS codes [35, 36], concatenated codes [37] and hypergraph product codes [38].

I. PRELIMINARIES

In this section, we briefly discuss basic constructions of stabilizer codes, as well as the notions of code distance and disjointness. We also comment on certain graph-theory problems and their computational complexity.

A. Stabilizer code constructions

Stabilizer codes are an important class of quantum error-correcting codes. A stabilizer code is defined by its stabilizer group $\mathcal{S}$, i.e., an Abelian subgroup of the Pauli group that does not contain $-I$. In what follows we identify the stabilizer code with its stabilizer group. The code space of the stabilizer code $\mathcal{S}$ is the simultaneous $(+1)$-eigenspace of all of the stabilizer operators. We

denote by $[\![n, k, d]\!]$ a stabilizer code that encodes k logical qubits into n physical qubits and has code distance d. To specify the $[\![n, k, d]\!]$ stabilizer code, we can provide a binary matrix of size $(n-k) \times 2n$, whose rows correspond to independent stabilizer generators of $\mathcal{S}$. For concreteness, we identify a bit string $(b_1, \ldots, b_{2n}) \in \{0, 1\}^{2n}$ with the following Pauli operator $\bigotimes_{i=1}^n X_i^{b_i} Z_i^{b_{i+n}}$, where P_i denotes a Pauli $P \in \{X, Z\}$ operator acting on qubit $i \in [n] = \{1, \ldots, n\}$. We say that an operator is of X- or Z-type iff it is a tensor product of either Pauli X or Pauli Z operators (and the identity operators).

For any stabilizer code, logical Pauli operators, which are the elements of the the normalizer of the stabilizer group in the Pauli group, can always be implemented as tensor products of single-qubit Pauli operators. We write $\overline{L}$ to represent a logical Pauli operator itself, as well as the set of its equivalent representatives. Also, we write $\mathcal{L}$ to denote the set of all non-trivial logical Pauli operators.

CSS codes.—A stabilizer code is a CSS code iff there exists a choice of stabilizer generators such that every generator is either a Pauli X- or Z-type operator. Given a CSS code with code parameters $[\![n, k, d]\!]$, we can always choose its logical Pauli operators in a way that for every $i \in [k]$ the logical Pauli $\overline{X}_i$ and $\overline{Z}_i$ operators are implemented via Pauli X- and Z-type operators, respectively. We refer to such a set of logical operators $\{\overline{X}_i, \overline{Z}_i\}_{i \in [k]}$ as a standard logical basis. Then, for every logical Pauli operator $\overline{L}$ we can find its decomposition in a standard logical basis, i.e., $\overline{L} = \overline{L}^X \overline{L}^Z$, where $\overline{L}^X$ and $\overline{L}^Z$ are X- and Z-type logical Pauli operators, respectively.

Concatenated stabilizer codes.—Given two stabilizer codes $\mathcal{S}_1$ and $\mathcal{S}_2$ with parameters $[\![n_1, k_1, d_1]\!]$ and $[\![n_2, 1, d_2]\!]$, respectively, we can concatenate them to obtain a new stabilizer code $\mathcal{S}_1 \lhd \mathcal{S}_2$. To construct the concatenated code $\mathcal{S}_1 \lhd \mathcal{S}_2$, we first encode the logical information into the stabilizer code $\mathcal{S}_1$, then we encode each and every qubit of the stabilizer code $\mathcal{S}_1$ into the stabilizer code $\mathcal{S}_2$. The concatenated code $\mathcal{S}_1 \lhd \mathcal{S}_2$ is a stabilizer code with parameters $[\![n_1 n_2, k_1, d]\!]$, where $d \geq d_1 d_2$.

Hypergraph product codes.—Given two full-rank binary matrices H_1 and H_2 of size $m_1 \times n_1$ and $m_2 \times n_2$, respectively, the corresponding hypergraph product code is specified by the following binary matrix

$$\left(\begin{array}{cc|cc} H_1 \otimes I_{m_2} & I_{m_1} \otimes H_2 & 0_{m_1 m_2, n_1 m_2 + n_2 m_1} \\ \hline 0_{n_1 n_2, n_1 m_2 + n_2 m_1} & I_{n_1} \otimes H_2^{\mathrm{T}} & H_1^{\mathrm{T}} \otimes I_{n_2} \end{array} \right), \quad (2)$$

where H_*^{T} denotes the transpose of H_*, $0_{a,b}$ and I_a are the zero matrix and the identity matrix of size $a \times b$ and $a \times a$, respectively. Note that hypergraph product codes are CSS codes.

B. Distance and disjointness

Let $\overline{L} \in \mathcal{L}$ be any non-trivial logical Pauli operator for the stabilizer code $\mathcal{S}$. Following Ref. [30], we define the distance $d(\overline{L})$ to be the size of the support of the smallest representative of $\overline{L}$, i.e.,

$$d(\overline{L}) = \min_{L \in \overline{L}} |\operatorname{supp} L|, \quad (3)$$

and introduce the notions of the min-distance and max-distance as follows

$$d_\downarrow = \min_{\overline{L} \in \mathcal{L}} d(\overline{L}), \quad d_\uparrow = \max_{\overline{L} \in \mathcal{L}} d(\overline{L}). \quad (4)$$

Note that the min-distance is the same as the standard stabilizer code distance.

Let $\mathcal{A} \subseteq \overline{L}$ be a subset of representatives of $\overline{L}$ or, more generally, a multiset that allows for multiple instances for each of the representatives of $\overline{L}$. We say that $\mathcal{A}$ is a c-disjoint collection of representatives of $\overline{L}$, where c is a positive integer, iff for every qubit there are at most c elements of $\mathcal{A}$ that are supported on that qubit. We then define the c-disjointness $\Delta_c(\overline{L})$ to be the size of the largest c-disjoint collection for $\overline{L}$ divided by c, i.e.,

$$\Delta_c(\overline{L}) = c^{-1} \max_{\mathcal{A} \subseteq \overline{L}} \{|\mathcal{A}| : \text{ at most } c \text{ elements } L \in \mathcal{A} \quad (5)$$

$$\text{are supported on any qubit}\}. (6)$$

Subsequently, the disjointness $\Delta(\mathcal{S})$ of the stabilizer code $\mathcal{S}$ is defined as follows

$$\Delta(\mathcal{S}) = \sup_{c \geq 1} \min_{\overline{L} \in \mathcal{L}} \Delta_c(\overline{L}). \quad (7)$$

Note that the disjointness defined here, which allows representatives of $\overline{L}$ to be selected multiple times, is greater or equal to the disjointness defined in Ref. [30]. Thus, by using our definition in Eq. (1) we may obtain a tighter bound on the level of the logical Clifford hierarchy attainable by transversal logical gates. At the same time, it is not obvious that the supremum in Eq. (7) can be admitted for some finite positive integer c. We establish this fact in Theorem 5 and Proposition 6.

Finally, for any positive integer c we introduce the following decision problem based on the c-disjointness.

> **c-DISJOINTNESS**
> **Input**: A full-rank binary matrix of size $(n - k) \times 2n$ specifying a stabilizer code $\mathcal{S}$, a string of $2n$ bits representing a logical Pauli operator $\overline{L}$ and a positive integer a.
> **Question**: Is the size of the largest c-disjoint collection of representatives of $\overline{L}$ greater or equal to a, i.e., $c\Delta_c(\overline{L}) \geq a$?

C. Computational complexity and graph theory

NP is the complexity class of problems that can be solved in polynomial time using a non-deterministic Turing machine, which can perform multiple operations at the same time in parallel at every time step, and accepts if any one of the parallel operations leads to an accepting
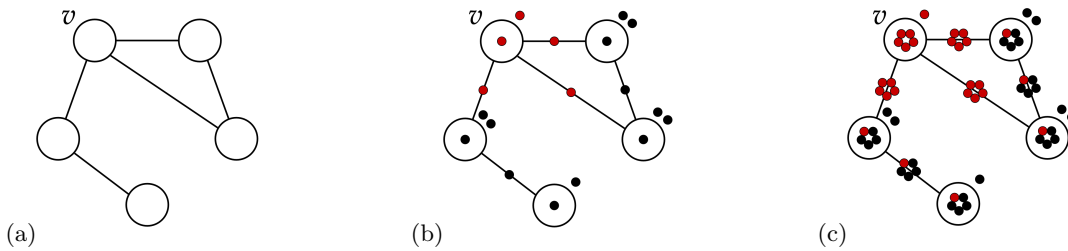
FIG. 1. (a) A graph $G = (V, E)$ can be used to define a CSS stabilizer code $\mathcal{S}_c^G$. In (b) and (c), we illustrate the construction for $c = 1$ and $c = 2$, respectively. Qubits (black and red dots) are placed at vertices and on edges of the graph G. We depict in red the support of the representative $X(v)$ of the logical Pauli operator $\overline{L}_c^G$ that is associated with the vertex $v \in V$.

state. A problem is NP-hard if every problem in NP can be reduced to it in polynomial time, and a problem is NP-complete if it is both NP-hard and in NP.

Let $G = (V, E)$ be a graph with the sets of vertices V and edges E. We say that a subset of vertices $V' \subseteq V$ is an independent set for the graph G iff no two vertices in V' are joined by an edge in E. Moreover, we say that a collection $\mathcal{A}$ comprising subsets of the vertices of G is an independent collection for G iff any two different $A, A' \in \mathcal{A}$ are disjoint and no two vertices belonging to, respectively, A and A' are joined by an edge in E. Note that an independent set is a special case of an independent collection. We denote the size of a maximum independent set for G by $\alpha(G)$ and refer to it as the independence number of G. Then, the following decision problem is NP-complete [39].

INDEPENDENT SET
Input: A graph $G = (V, E)$ and a positive integer a.
Question: Is the independence number of G greater or equal to a, i.e., $\alpha(G) \geq a$?

Furthermore, for generic graphs and a positive real number ϵ the problem of approximating $\alpha(G)$ up to a multiplicative factor of $|V|^{1-\epsilon}$ is NP-hard [40, 41]. We use this result in our reduction in the following section.

II. HARDNESS OF FINDING AND APPROXIMATING THE c-DISJOINTNESS

We start this section by constructing a CSS stabilizer code $\mathcal{S}_c^G$ and a logical Pauli operator $\overline{L}_c^G$ for any finite graph G and positive integer c. Then, we show that the c-disjointness $\Delta_c\left(\overline{L}_c^G\right)$ can be related to the independence number $\alpha(G)$ of the graph G. This, in turn, allows us to prove our main theorem by reducing the problem of finding the maximum independent set for G to the problem of finding the c-disjointness $\Delta_c\left(\overline{L}_c^G\right)$.

A. Constructing the CSS stabilizer code

Let $G = (V, E)$ be a graph and c be a positive integer constant. To define a CSS stabilizer code $\mathcal{S}_c^G$ associated with the graph G, we first place $\binom{|V|}{c-1}$ qubits on every vertex $v \in V$ and on every edge $e \in E$; see Fig. 1 for an illustrative example. We label each qubit by a pair (v, ν) or (e, ν), where $\nu \subseteq [|V|]$ is a subset of $c - 1$ integers from $[|V|] = \{1, \ldots, |V|\}$. Additionally, we place one or two qubits at each vertex v depending on whether $\binom{|V|}{c-1}(\deg v + 1)$ is odd or even respectively, where $\deg v$ denotes the degree of v. We label those additional qubits by a pair (v, i), where $i \in [2]$. We denote by P_q a Pauli operator $P = X, Z$ supported on qubit with label q. We denote by $Q(v)$ and $Q(e)$ all the qubits placed at the vertex v and on the edge e, i.e., all the qubits with labels $(v, *)$ and $(e, *)$, respectively. We also denote by Q the set of all the qubits, i.e.,

$$Q = \bigcup_{v \in V} Q(v) \cup \bigcup_{e \in E} Q(e). \tag{8}$$

The stabilizer group $\mathcal{S}_c^G$ associated with G and c is

$$\mathcal{S}_c^G = \left\langle X(u)X(v), \prod_{q \in Q} Z_q \,\middle|\, u, v \in V \right\rangle, \tag{9}$$

where $X(v)$ denotes a Pauli X operator associated with a vertex $v \in V$, which we define as follows

$$X(v) = \prod_{q \in Q(v)} X_q \prod_{e \ni v} \left(\prod_{q \in Q(e)} X_q \right). \tag{10}$$

Note that for any $v \in V$ the Pauli X-type operator $X(v)$ is supported on the even number of qubits. Thus, $X(v)$ and $\prod_{q \in Q} Z_q$ commute and $\mathcal{S}_c^G$ is an Abelian subgroup of the Pauli group satisfying $-I \notin \mathcal{S}_c^G$. Moreover, for any $u, v \in V$ the Pauli X operators $X(u)$ and $X(v)$ are the representatives of the same non-trivial logical Pauli operator, which we denote by $\overline{L}_c^G$. The last statement follows from the fact that $X(u)$ and $X(v)$ commute with the all stabilizer operators, are not contained in the stabilizer group $\mathcal{S}_c^G$, and their product forms a stabilizer operator

$X(u)X(v)$. For convenience, for any subset of vertices $A \subseteq V$ we define

$$X(A) = \prod_{v \in A} X(v). \qquad (11)$$

Note that if $|A| \equiv 0 \mod 2$, then $X(A)$ is a stabilizer operator; otherwise, $X(A)$ is a representative of the logical Pauli operator $\overline{L}_c^G$.

We remark that if the graph G has at least $c + 2$ vertices, i.e., $|V| \geq c + 2$, then the stabilizer code $\mathcal{S}_c^G$ associated with G and c is error-detecting, i.e., its min-distance satisfies $d_\downarrow > 1$. This, in turn, implies that the disjointness of $\mathcal{S}_c^G$ is greater than one, i.e., $\Delta(\mathcal{S}_c^G) > 1$; see Lemma 2(ii) in Ref. [30]. To establish the claim that $d_\downarrow > 1$, it suffices to check that every single-qubit Pauli operator anticommutes with some stabilizer operator from $\mathcal{S}_c^G$. Since the stabilizer operator $\prod_{q \in Q} Z_q$ is supported on every qubit, any single-qubit Pauli X or Y operator anticommutes with it. For any vertex $v \in V$ a Pauli Z operator on qubit (v, i) or (v, ν) anticommutes with a stabilizer operator $X(u)X(v)$ for any vertex $u \in V \setminus \{v\}$ or $u \in V \setminus (\{v\} \cup \nu)$, respectively. Lastly, for any edge $e \in E$ a Pauli Z operator on qubit (e, ν) anticommutes with a stabilizer operator $X(u)X(v)$ for any vertex $u \in V \setminus (\{w, w'\} \cup \nu)$, where $w, w' \in V$ are two vertices incident to e. Note that we use $|V| \geq c + 2$ to guarantee that the set $V \setminus (\{w, w'\} \cup \nu)$ is non-empty.

B. Relating the c-disjointness to the independence number

Let $\mathcal{S}_c^G$ and $\overline{L}_c^G$ be a stabilizer code and a logical Pauli operator, which are associated with the graph $G = (V, E)$ and the positive integer c. We abuse terminology and say that a collection $\mathcal{A}$ comprising subsets of the vertices of G is c-disjoint iff $\{X(A) \mid A \in \mathcal{A}\}$ is a c-disjoint collection for the logical operator $\overline{L}_c^G$ of the stabilizer code $\mathcal{S}_c^G$. We refer to $X(A)$ as a representative from $\mathcal{A}$. We remark that $\mathcal{A}$ is allowed to be a multiset. In what follows, we establish three technical lemmas.

Lemma 1. *Let $G = (V, E)$ be a graph, $V' \subseteq V$ be an independent set and c be a positive integer. Then, a collection $\mathcal{A} = \{\{v\} \mid v \in V'\}$ is c-disjoint. Moreover, if $|V'|(c+1) \equiv 1 \mod 2$, then $\mathcal{A} \cup \{V'\}$ is c-disjoint, too.*

Proof. If $c = 1$, then $|V'|(c+1) \equiv 0 \mod 2$ and one can straightforwardly show that $\mathcal{A}$ is c-disjoint. In what follows we assume that $c \geq 2$. Let $v \in V$ be a vertex, $e \in E$ be an edge incident to two vertices $w, w' \in V$, $\nu \subset V$ be a subset of $c-1$ vertices and $i \in [2]$. First note that any qubit (v, i) supports one or zero representatives from $\mathcal{A}$, depending on whether v belongs to V' or not. Let $u \in V'$ be a vertex and $X(u)$ be a corresponding representative of $\overline{L}_c^G$. If $X(u)$ is supported on the qubit (v, ν), then $u \in (\nu \cup \{v\}) \cap V'$. Thus, the qubit (v, ν) supports at most $|\nu| + 1 = c$ representatives from $\mathcal{A}$. Similarly, $X(u)$

is supported on qubit (e, ν), then $u \in (\nu \cup \{w, w'\}) \cap V'$. Since V' is an independent set, at most one of w and w' can belong to V'. We then conclude that qubit (e, ν) supports at most $|\nu| + 1 = c$ representatives from $\mathcal{A}$. Thus, $\mathcal{A}$ is c-disjont.

Now, assume that $|V'|(c+1) \equiv 1 \mod 2$. Then, $X(V')$ is a representative of $\overline{L}_c^G$. It remains to show that no qubit supports more than c representatives from $\mathcal{A} \cup \{V'\}$. Since $\mathcal{A}$ is c-disjoint, thus a potential problem may only arise for qubits which already support c representatives from $\mathcal{A}$. This, however, cannot happen for any qubit (v, i), as $c \geq 2$. If the qubit (v, ν) supports c representatives from $\mathcal{A}$, then $\nu \cup \{v\} \subseteq V'$, and subsequently the qubit (v, ν) does not support $X(V')$, as $|\nu \cup \{v\}| \equiv 0 \mod 2$. Similarly, if the qubit (e, ν) supports c representatives from $\mathcal{A}$, then it does not support $X(V')$. We conclude that $\mathcal{A} \cup \{V'\}$ is c-disjoint. $\square$

Lemma 2. *Let $G = (V, E)$ be a graph and c be a positive integer. Let $\mathcal{A}$ be a collection of subsets of the vertices of G that is c-disjoint and satisfies $|\mathcal{A}| \geq 9c^3/2$. Then, there exists an independent collection $\mathcal{A}' \subseteq \mathcal{A}$, such that $|\mathcal{A}'| \geq |\mathcal{A}| - 3c(c-1)$.*

Proof. If $c = 1$, then one can straightforwardly show that $\mathcal{A}$ is an independent collection and, subsequently, one can set $\mathcal{A}' = \mathcal{A}$. In what follows we assume that $c \geq 2$ and provide an explicit construction of $\mathcal{A}'$.

We start by finding the subsets of vertices $L_1, \ldots, L_{3(c-1)} \in \mathcal{A}$ and vertices $v_1, \ldots, v_{3(c-1)}$, such that $v_i \in L_i$ and $v_i \notin L_j$ for any two different $i, j \in [3(c-1)]$. Let $\mathcal{A}_1 = \mathcal{A}$. For $i = 1, \ldots, 3(c-1)$ one proceeds inductively to find L_i and v_i. Namely,

1. choose L_i to be the smallest subset of vertices in $\mathcal{A}_i$, i.e., $L_i = \arg\min_{L \in \mathcal{A}_i} |L|$,

2. choose v_i to be any vertex in $L_i \setminus \bigcup_{j=1}^{i-1} L_i$,

3. define $\mathcal{A}_i'$ to be the collection of the elements of $\mathcal{A}_i$ that do not contain v_i, i.e., $\mathcal{A}_i' = \{L \in \mathcal{A}_i \mid L \not\ni v_i\}$,

4. define $\mathcal{A}_{i+1}$ be the collection of the elements of $\mathcal{A}_i'$ that are not fully contained within $\bigcup_{j=1}^{i} L_j$, i.e., $\mathcal{A}_{i+1} = \{L \in \mathcal{A}_i' \mid L \setminus \bigcup_{j=1}^{i} L_j \neq \emptyset\}$.

It suffices to show that $|\mathcal{A}_i| \geq 1$ for any $i \in [3(c-1)]$, as this would imply that L_i and v_i can be found in steps 1 and 2.

By construction, we have the following chain of inclusions $\mathcal{A}_i \supseteq \mathcal{A}_i' \supseteq \mathcal{A}_{i+1}$. We also have

$$|\mathcal{A}_{i+1}| \geq |\mathcal{A}_i'| - ic \geq |\mathcal{A}_i| - (i+1)c. \qquad (12)$$

To show Eq. (12), first note that for every $L \in \mathcal{A}$ containing the vertex v_i, the corresponding operator $X(L)$ is supported on qubit $(v_i, 1)$. Since $\mathcal{A}$ is c-disjoint, there are at most c elements of $\mathcal{A}$ that include v_i. Thus,

we obtain $|\mathcal{A}'_i| \geq |\mathcal{A}_i| - c$. By construction, we have $|L_1| \leq \ldots \leq |L_i|$, which leads to

$$\left|\bigcup_{j=1}^{i} L_j\right| \leq \sum_{j=1}^{i} |L_j| \leq i|L_i|. \tag{13}$$

This, in turn, allows us to upper bound the number of elements of $\mathcal{A}'_i$ that are contained in $\bigcup_{j=1}^{i} L_j$. By definition of L_i in step 1, for any $L \in \mathcal{A}'_i$ we have $|L| \geq |L_i|$. Since $\mathcal{A}$ is c-disjoint, so is $\mathcal{A}'_i$ and we conclude that there are at most

$$c\left|\bigcup_{j=1}^{i} L_j\right|/|L_i| \leq ic \tag{14}$$

elements of $\mathcal{A}'_i$ that are contained in $\bigcup_{j=1}^{i} L_i$. This leads to $|\mathcal{A}_{i+1}| \geq |\mathcal{A}'_i| - ic$, which, in turn, establishes Eq. (12). By recursively using Eq. (12) we obtain

$$|\mathcal{A}_i| \geq |\mathcal{A}_1| - (i(i+1)/2 - 1)c \tag{15}$$
$$\geq 9c^3/2 - (9c^2 - 15c + 4)c/2 > 1 \tag{16}$$

for any $i \in [3(c-1)]$, where the last inequality holds for any positive integer c.

Now, let us consider

$$\mathcal{A}' = \{L \in \mathcal{A} \mid \forall i \in [3(c-1)] : v_i \notin L\}. \tag{17}$$

Since $\mathcal{A}$ is c-disjoint, we obtain $|\mathcal{A}'| \geq |\mathcal{A}| - 3c(c-1)$. To show that $\mathcal{A}'$ is an independent collection, we assume the contrapositive. Thus, there exist $L, L' \in \mathcal{A}'$, such that either (i) L and L' overlap and $v \in L \cap L'$ is the shared vertex, or (ii) L and L' are disjoint, and $u \in L$ and $w \in L'$ are two vertices incident to the same edge $e \in E$. Let us assume that the proposition (i) holds and

$$\mu = \{i \in [3(c-1)] \mid v \in L_i\}. \tag{18}$$

Note that $|\mu| \leq c - 2$; otherwise, qubit $(v, 1)$ would support at least $c+1$ representatives corresponding to L, L' and L_i for $i \in \mu$. Thus, we can find a subset of indices $\nu \subset [3(c-1)] \setminus \mu$ of cardinality $c-1$, such that $v \notin L_i$ for $i \in \nu$. This leads to a contradiction, as the qubit (v, ν) supports $c+1$ representatives corresponding to L, L' and L_i for $i \in \nu$. Now, let us assume that the proposition (ii) holds and

$$\mu = \{i \in [3(c-1)] \mid \{u, w\} \cap L_i \neq \emptyset\}. \tag{19}$$

Note that $|\mu| \leq 2c-4$; otherwise, one of the qubits $(u, 1)$ and $(w, 1)$ would support at least $c+1$ representatives among the ones that correspond to L, L' and L_i for $i \in \mu$. Thus, we can find a subset of $c-1$ indices $\nu \subset [3(c-1)] \setminus \mu$, such that $u, w \notin L_i$ for $i \in \nu$. This leads to a contradiction, as the qubit (e, ν) supports $c+1$ representatives corresponding to L, L' and L_i for $i \in \nu$. $\square$

Lemma 3. *Let $G = (V, E)$ be a graph with no isolated vertices, such that $\alpha(G) \geq 9c^3/2$, where c is a positive integer. Then, the c-disjointness for the logical operator $\overline{L}_c^G$ of the stabilizer code $\mathcal{S}_c^G$ is given by*

$$\Delta_c\left(\overline{L}_c^G\right) = (\alpha(G) + b)/c, \tag{20}$$

where $b = \alpha(G)(c+1) \mod 2$.

Proof. Let $\mathcal{A}$ be a c-disjoint collection for $\overline{L}_c^G$ of largest possible size, i.e., $|\mathcal{A}| = c\Delta_c\left(\overline{L}_c^G\right)$. Lemma 1 implies that a maximum independent set $V' \subseteq V$ for G leads to a c-disjoint collection for $\overline{L}_c^G$ of size $\alpha(G) + b$, and thus, by definition of $\mathcal{A}$, we have $|\mathcal{A}| \geq \alpha(G) + b$. Let $\mathcal{A}' \subseteq \mathcal{A}$ be an independent collection within $\mathcal{A}$ of largest possible size. Note that $|\mathcal{A}'| \leq \alpha(G)$, as the size of any independent collection cannot be greater than the independence number of G. Thus, we arrive at

$$|\mathcal{A}| \geq \alpha(G) + b \geq \alpha(G) \geq |\mathcal{A}'|. \tag{21}$$

Note that $|\mathcal{A}| = |\mathcal{A}'|$ can hold only if $b = 0$, and in that case we recover Eq. (20). In what follows, we thus consider the case $\mathcal{A} \setminus \mathcal{A}' \neq \emptyset$ and choose $L \in \mathcal{A} \setminus \mathcal{A}'$.

By definition, $\mathcal{A}'$ is the largest independent collection within $\mathcal{A}$. Thus, Lemma 2 implies that

$$|\mathcal{A}'| \geq |\mathcal{A}| - 3c(c-1) \geq 9c^3/2 - 3c(c-1) \geq c+1, \tag{22}$$

where the last inequality holds for any positive integer c. Subsequently, let $L_1, \ldots, L_{c+1} \in \mathcal{A}'$ be $c+1$ different subsets of vertices and select vertices $v_i \in L_i$ for $i \in [c+1]$. We can rule out the possibility that $L \cap \bigcup \mathcal{A}' = \emptyset$, where for brevity we write $\bigcup \mathcal{A}' = \bigcup_{L' \in \mathcal{A}'} L'$. Namely, assume $L \cap \bigcup \mathcal{A}' = \emptyset$. In that case, there must exists an edge $e \in E$ incident to $u \in L$ and $v \in \bigcup \mathcal{A}'$; otherwise, $\mathcal{A}' \cup \{L\} \subseteq \mathcal{A}$ would be an independent collection larger than $\mathcal{A}'$, leading to a contradiction. Without loss of generality, let $v_c = v$. Then, qubit $(e, \{v_i\}_{i \in [c-1]})$ supports $c+1$ different logical operators $X(L), X(L_1), \ldots, X(L_c)$, which is in a contradiction with $\mathcal{A}$ being c-disjoint. We thus conclude that $L \cap \bigcup \mathcal{A}' \neq \emptyset$.

We can show that $L \supseteq \bigcup \mathcal{A}'$ by assuming the contrapositive. Since $L \cap \bigcup \mathcal{A}' \neq \emptyset$, there exists a vertex $v \in \bigcup \mathcal{A}' \setminus L$. Without loss of generality, let $v_c \in L \cap \bigcup \mathcal{A}'$ and $v_{c+1} = v$. Then, one of the qubits $(v_c, \{v_i\}_{i \in [c-1]})$ or $(v_{c+1}, \{v_i\}_{i \in [c-1]})$ is guaranteed to support $X(L)$, on top of other c different operators, either $X(L_1), \ldots, X(L_c)$ or $X(L_1), \ldots, X(L_{c-1}), X(L_{c+1})$, respectively. This is in a contradiction with $\mathcal{A}$ being c-disjoint. We thus conclude that $L \supseteq \bigcup \mathcal{A}'$.

Note that c has to be even; otherwise, the qubit $(v_c, \{v_i\}_{i \in [c-1]})$ would support $c+1$ different logical operators $X(L), X(L_1), \ldots, X(L_c)$, leading to a contradiction. If c is even, then $b = \alpha(G) \mod 2$ and $\alpha(G) + b$ is even. We also obtain that $\mathcal{A} \setminus \mathcal{A}' = \{L\}$; otherwise, there would exist at least two subsets of vertices $L, K \in \mathcal{A} \setminus \mathcal{A}'$ satisfying $L, K \supseteq \bigcup \mathcal{A}'$ and the qubit $(v_{c-1}, \{v_i\}_{i \in [c-1]})$ would support at least $c+1$ different operators $X(L), X(K), X(L_1), \ldots, X(L_{c-1})$, leading to a contradiction. We thus conclude that $|\mathcal{A}| = |\mathcal{A}'| + 1$.

There are two cases to consider: (i) $L = \bigcup \mathcal{A}'$ and (ii) $L \setminus \bigcup \mathcal{A}' \neq \emptyset$. Recall that for any subset of vertices $L' \subseteq V$ the Pauli operator $X(L')$ is a representative of $\overline{L}_c^G$ iff $|L'| \equiv 1 \mod 2$. First, assume that the proposition (i) holds. Then, $|\mathcal{A}'| \equiv 1 \mod 2$; otherwise,

$$|L| = |\bigcup \mathcal{A}'| = \sum_{L' \in \mathcal{A}'} |L'| \equiv |\mathcal{A}'| \equiv 0 \mod 2, \quad (23)$$

leading to a contradiction with $X(L)$ being a representative of $\overline{L}_c^G$. Using $|\mathcal{A}| = |\mathcal{A}'| + 1$, Eq. (21) and the fact that $\alpha(G) + b$ is even, we finally recover Eq. (20). Now, assume that the proposition (ii) holds. Then, $\mathcal{A}' \cup \{L \setminus \bigcup \mathcal{A}'\}$ is an independent collection; otherwise, there would exist an edge $e \in E$ incident to vertices $u \in L \setminus \bigcup \mathcal{A}'$ and $v \in \bigcup \mathcal{A}'$, we could set $v_c = v$ and the qubit $(e, \{v_i\}_{i \in [c-1]})$ would support $c+1$ different logical operators $X(L), X(L_1), \ldots, X(L_c)$, leading to a contradiction. Since the size of any independent collection is at most $\alpha(G)$, thus $|\mathcal{A}'| + 1 \leq \alpha(G)$. Using $|\mathcal{A}| = |\mathcal{A}'| + 1$ and Eq. (21), we conclude that $b = 0$ and subsequently recover Eq. (20). $\square$

C. Putting things together

Now, we are ready to establish the main result of our work, which asserts that it is NP-complete to calculate (or even approximate) the c-disjointness for any positive integer constant c.

Theorem 4. *For any positive integer constant c the decision problem c-DISJOINTNESS is NP-complete. Furthermore, for any $\epsilon > 0$ it is NP-hard to approximate the c-disjointness to within a multiplicative factor of $n^{(1-\epsilon)/(c+1)}$, where n is the number of physical qubits of the stabilizer code.*

Proof. First note that INDEPENDENT SET is NP-complete, even if we restrict our attention to the graphs with no isolated vertices and the independence number greater or equal to $9c^3/2$, which we require in our proof. We can show that c-DISJOINTNESS is NP-hard by reducing INDEPENDENT SET to it.

Let $G = (V, E)$ be a graph satisfying $\alpha(G) \geq 9c^3/2$. We construct a new graph $G' = (V', E')$ as follows: for every vertex $v \in V$ we introduce two vertices $v_1, v_2 \in V'$, and for every edge $(u, v) \in E$ we introduce four edges $(u_1, v_1), (u_1, v_2), (u_2, v_1), (u_2, v_2) \in E'$. One can show that

$$\alpha(G') = 2\alpha(G). \quad (24)$$

Following Section II A, we construct the stabilizer code $\mathcal{S}_c^{G'}$ and the logical operator $\overline{L}_c^{G'}$. Since $\alpha(G') \equiv 0 \mod 2$, then from Lemma 3 we obtain

$$c\Delta_c\left(\overline{L}_c^{G'}\right) = \alpha(G') = 2\alpha(G) \quad (25)$$

and, subsequently, $\alpha(G) \geq a$ iff $c\Delta_c\left(\overline{L}_c^{G'}\right) \geq 2a$. We conclude that the output of the c-DISJOINTNESS problem for the instance $\mathcal{S}_c^{G'}$, $\overline{L}_c^{G'}$ and $2a$ is the output of INDEPENDENT SET for G and a, which, in turn, establishes that c-DISJOINTNESS is NP-hard.

Now we turn to approximation algorithms. Given the graph $G = (V, E)$, the number of physical qubits in the stabilizer code $\mathcal{S}_c^G$ is

$$n \leq |V|\left(\binom{|V|}{c-1} + 2\right) + |E|\binom{|V|}{c-1} < |V|^{c+1}, \quad (26)$$

where we use $|E| \leq \binom{|V|}{2}$. Depending on the parity of $\alpha(G)(c+1)$, the size of the largest c-disjoint collection for the logical operator $\overline{L}_c^G$ is equal to either $\alpha(G)$ or $\alpha(G) + 1$. Thus, for any $\epsilon > 0$ the ability to approximate the c-disjointness $\Delta_c\left(\overline{L}_c^G\right)$ to within a multiplicative factor of $n^{(1-\epsilon)/(c+1)}$ implies that we can approximate $\alpha(G)$ to within a multiplicative factor of $n^{(1-\epsilon)/(c+1)} < |V|^{1-\epsilon}$. Because approximating $\alpha(G)$ is NP-hard, thus approximating the c-disjointness $\Delta_c\left(\overline{L}_c^G\right)$ is also NP-hard.

Finally, we show that c-DISJOINTNESS is in NP. Let us consider a polynomial time verifier that takes as its witness a collection $\mathcal{A}$, which supposedly comprises representatives of the logical operator $\overline{L}$ specified by its representative $L \in \overline{L}$. The verifier needs to check the following three conditions: (i) the size of $\mathcal{A}$ is greater than ca (ii) every qubit supports at most c operators from $\mathcal{A}$, and (iii) every element of $\mathcal{A}$ is a representative of $\overline{L}$. Conditions (i) and (ii) can be easily checked in time polynomial in n. Also, condition (iii) can be stated equivalently as follows: for every $P \in \mathcal{A}$ the operator PL belongs to the stabilizer group $\mathcal{S}$ specifying the stabilizer code. Let M be the binary matrix of size $(n-k) \times 2n$ that describes $\mathcal{S}$. In order to check the last condition, for every $P \in \mathcal{A}$ the verifier appends to M a row corresponding to a string of $2n$ bits representing PL and computes its rank over the field $\mathbb{F}_2$. The rank is equal to $n-k$ iff P is a representative of $\overline{L}$. Since finding the rank of a binary matrix can be done in time polynomial in its size, thus the witness can be verified in time polynomial in n, and, subsequently, c-DISJOINTNESS is in NP. $\square$

III. DISJOINTNESS IN PRACTICE

In this section we express the problem of calculating the disjointness as a linear program with exponentially many variables, which allows us to find the disjointness in exponential time. We use this linear program to find the disjointness of the $[\![14, 3, 3]\!]$ stabilizer code, and ultimately rule out the existence of a transversal logical non-Clifford gate.

A. A linear program for disjointness

Let c be a positive integer and $\overline{L}$ be a non-trivial logical Pauli operator for an $[\![n,k,d]\!]$ stabilizer code. We can formulate an optimization problem whose optimal value is the c-disjointness $\Delta_c(\overline{L})$. Namely, for every representative L of $\overline{L}$ we introduce a variable x_L, which admits non-negative integer values and represents the number of times the representative L appears in a collection $\mathcal{A} \subseteq \overline{L}$. We want to maximize the size of $\mathcal{A}$ divided by c, i.e., $\sum_{L \in \overline{L}} x_L/c$, subject to $\mathcal{A}$ being c-disjoint. Note that this constraint can be equivalently phrased as follows: for every qubit $i \in [n]$ we have $\sum_{L \in \overline{L}:\mathrm{supp}\,L \ni i} x_L/c \leq 1$. Thus, we conclude that $\Delta_c(\overline{L})$ can be found as a solution to an integer linear program with 2^{n-k} variables.

We can relax this integer linear program by removing the integrality constraint of each variable and allowing $x'_L = x_L/c$ to be a non-negative real number for every $L \in \overline{L}$. We subsequently arrive at the following linear program

$$\text{maximize} \quad \Delta = \sum_{L \in \overline{L}} x'_L, \tag{27}$$

$$\text{subject to} \quad \forall i \in [n]: \sum_{L \in \overline{L}:\mathrm{supp}\,L \ni i} x'_L \leq 1, \tag{28}$$

$$\forall L \in \overline{L} : x'_L \geq 0. \tag{29}$$

We now state the following theorem.

Theorem 5. *For any $[\![n,k,d]\!]$ stabilizer code and non-trivial logical Pauli operator $\overline{L} \in \mathcal{L}$, the optimal value Δ^* attained by the linear program in Eqs. (27)-(29) is equal to the supremum of the c-disjointness for $\overline{L}$, i.e.,*

$$\Delta^* = \sup_{c \in \mathbb{Z}_+} \Delta_c(\overline{L}). \tag{30}$$

Moreover, the supremum of the c-disjointness for $\overline{L}$ is attained at some positive integer $c^* = 2^{\mathrm{poly}(n2^{n-k})}$, i.e.,

$$\sup_{c \in \mathbb{Z}_+} \Delta_c(\overline{L}) = \Delta_{c^*}(\overline{L}). \tag{31}$$

Proof. Let c be any positive integer and $\mathcal{A}$ be a c-disjoint collection for $\overline{L}$ of the largest possible size, i.e., $|\mathcal{A}| = c\Delta_c(\overline{L})$. If x_L is the number of times a representative $L \in \overline{L}$ appears in $\mathcal{A}$, then for every qubit $i \in [n]$ we have $\sum_{L \in \overline{L}:\mathrm{supp}\,L \ni i} x_L/c \leq 1$, as $\mathcal{A}$ is c-disjoint. Moreover,

$$\Delta^* \geq \sum_{L \in \overline{L}} x_L/c = \Delta_c(\overline{L}), \tag{32}$$

leading to

$$\Delta^* \geq \sup_{c \in \mathbb{Z}_+} \Delta_c(\overline{L}). \tag{33}$$

Note that there exists some finite optimal solution to the linear program in Eqs. (27)-(29), since every variable

x'_L appears in at least one constraint and, subsequently, every feasible solution is bounded, i.e., $x'_L \leq 1$ for all $L \in \overline{L}$. Moreover, we can specify the linear program in Eqs. (27)-(29) in matrix form $\max\{\mathbf{c}^{\mathrm{T}}\mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n, \mathbf{A}\mathbf{x} = \mathbf{b}, \mathbf{x} \geq \mathbf{0}\}$, where $\mathbf{A}$ is a binary matrix of size $2^{n-k} \times n$ that captures the support of every representative, and $\mathbf{b}$ and $\mathbf{c}$ are vectors of all ones of length n and 2^{n-k}, respectively. The existence of a finite optimal solution and the fact that $\mathbf{A}$, $\mathbf{b}$ and $\mathbf{c}$ have rational entries implies that there exists a rational optimal solution $\{x_L^*\}_{L \in \overline{L}}$ whose bit size is polynomially bounded in terms of the bit sizes of $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ [42]. Then, for every $L \in \overline{L}$ we can find positive integers a_L^* and b_L^*, such that $x_L^* = a_L^*/b_L^*$ and $a_L^* b_L^* \leq 2^{\mathrm{poly}(n2^{n-k})}$. Let c^* be the least common multiple of all b_L^*s, i.e.,

$$c^* = \mathrm{lcm}(\{b_L^*\}_{L \in \overline{L}}). \tag{34}$$

Clearly, $c^* < (2^{\mathrm{poly}(n2^{n-k})})^{2^{n-k}} = 2^{\mathrm{poly}(n2^{n-k})}$. Let $\mathcal{A}^*$ be a collection of representatives of $\overline{L}$ obtained by taking $x_L^* c^*$ copies of each $L \in \overline{L}$. Since $\mathcal{A}^*$ is c^*-disjoint and $|\mathcal{A}^*| = \sum_{L \in \overline{L}} x_L^* c^*$, we obtain that

$$\Delta^* = \sum_{L \in \overline{L}} x_L^* = |\mathcal{A}^*|/c^* \leq \Delta_{c^*}(\overline{L}) \leq \sup_{c \in \mathbb{Z}_+} \Delta_{c^*}(\overline{L}). \tag{35}$$

Thus, the inequalities in Eqs. (33) and (35) have to become equalities, which implies Eqs. (30)-(31). $\square$

Theorem 5 implies the following proposition.

Proposition 6. *The disjointness of any stabilizer code $\mathcal{S}$ satisfies the following equality*

$$\Delta(\mathcal{S}) = \min_{\overline{L} \in \mathcal{L}} \sup_{c \in \mathbb{Z}_+} \Delta_c(\overline{L}). \tag{36}$$

Proof. By definition of the disjointness, we have $\Delta(\mathcal{S}) = \sup_{c \in \mathbb{Z}_+} \min_{\overline{L} \in \mathcal{L}} \Delta_c(\overline{L})$. Theorem 5 guarantees that for any non-trivial logical Pauli operator $\overline{L} \in \mathcal{L}$ the supremum of the c-disjointness for $\overline{L}$ is attained at some positive integer $c_{\overline{L}}$. Let c^* be the least common multiple of all $c_{\overline{L}}$s, i.e.,

$$c^* = \mathrm{lcm}(\{c_L\}_{\overline{L} \in \mathcal{L}}). \tag{37}$$

We then obtain

$$\min_{\overline{L} \in \mathcal{L}} \Delta_{c^*}(\overline{L}) \leq \sup_{c \in \mathbb{Z}_+} \min_{\overline{L} \in \mathcal{L}} \Delta_c(\overline{L}) \leq \min_{\overline{L} \in \mathcal{L}} \sup_{c \in \mathbb{Z}_+} \Delta_c(\overline{L}) \tag{38}$$

$$= \min_{\overline{L} \in \mathcal{L}} \Delta_{c_{\overline{L}}}(\overline{L}) \leq \min_{\overline{L} \in \mathcal{L}} \Delta_{c^*}(\overline{L}), \tag{39}$$

where we use the max-min inequality and the fact that $\Delta_a(\overline{L}) \leq \Delta_{ab}(\overline{L})$ for any $\overline{L} \in \mathcal{L}$ and $a,b \in \mathbb{Z}_+$. Thus, all the above inequalities in Eqs. (38)-(39) have to become equalities, which, in turn, implies Eq. (36). $\square$

We remark that the linear program in Eqs. (27)-(29), with 2^{n-k} variables and n constraints, can be solved in time $O(2^{2.5(n-k)})$. Thus, the disjointness of the stabilizer code $\mathcal{S}$ can be found in time $O(2^{2.5n-0.5k})$, as we can find it by solving this linear program for all $2^{2k} - 1$ logical Pauli operators for $\mathcal{S}$.
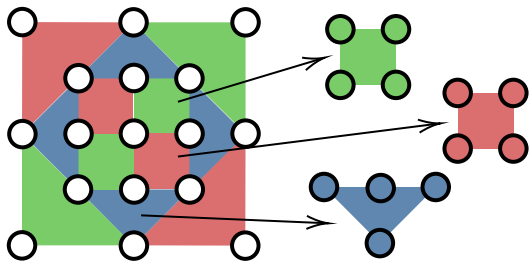
FIG. 2. The $[\![14,3,3]\!]$ stabilizer code is defined by placing qubits (white dots) on the vertices of a rhombic dodecahedron and introducing X-, Y- and Z-type stabilizer generators for every red, green and blue face, respectively. We depict "the azimuthal projection", where the four corner qubits are identified. Dots colored in red, green and blue correspond to Pauli X, Y and Z operators, respectively.

B. An illustrative example

Now, we focus on an illustrative example of the recently introduced $[\![14,3,3]\!]$ stabilizer code [34]; see Fig. 2. We would like to understand whether any non-Clifford logical operators for this code can be implemented via transversal gates. If this was the case, then due to its small size the $[\![14,3,3]\!]$ stabilizer code could prove useful in, for instance, magic state distillation protocols. Since the $[\![14,3,3]\!]$ stabilizer code is a non-CSS code, there are no off-the-shelf techniques to find transversal gates; however, we can use the disjointness to rule out the possibility of certain logical operations.

A brute-force approach to computing the c-disjointness, even for $c = 1$ and small stabilizer codes, is infeasible. In the case of the $[\![14,3,3]\!]$ stabilizer code, there are $4^3 - 1 = 63$ different non-trivial logical Pauli operators, and each of them has $2^{11} = 2048$ representatives. Thus, for each logical Pauli operator there are $2^{2048} \approx 3.2 \times 10^{616}$ possible subsets of its representatives, and we would need to check the qubit overlap for each of them (as the 1-disjointness is achieved with a set rather than a multiset).

We can, however, use the linear program specified in Eqs. (27)-(29), which has 2048 variables and 14 constraints (excluding the positivity constraints). We numerically find that the disjointness of the $[\![14,3,3]\!]$ stabilizer code is 2; see the source code [43]. We also find that the max-distance of the $[\![14,3,3]\!]$ stabilizer code is $d_\uparrow = 6$. Thus, using the bound from Eq. (1) we obtain that any transversal gate can only implement logical operations within the third level of the logical Clifford hierarchy.

We now obtain an improvement of the bound in Eq. (1), which we subsequently use to rule out the possibility of any transversal gate that implements a non-Clifford logical operation for the $[\![14,3,3]\!]$ stabilizer code. Recall that the main proof idea in Ref. [30] is to evaluate the (nested) group commutator of the transversal logical gate and M logical Pauli operators for the stabilizer code $\mathcal{S}$. If for every M-tuple of logical Pauli operators the
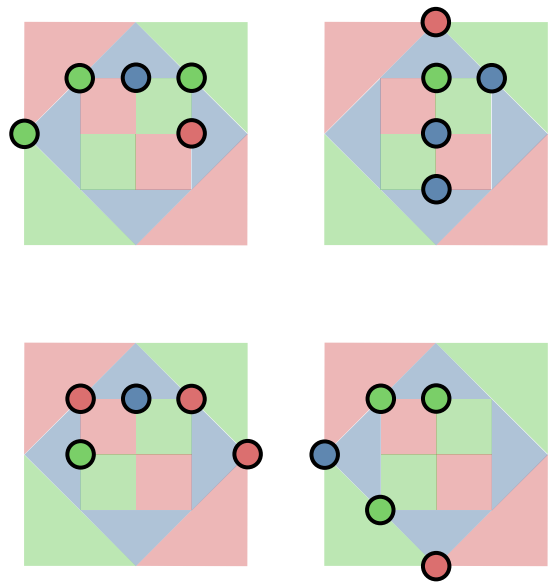

FIG. 3. For the $[\![14,3,3]\!]$ stabilizer code there are four logical Pauli operators with disjointness 2. We depict their smallest-weight representatives, where dots colored in red, green and blue correspond to Pauli X, Y and Z operators, respectively.

resulting operator is a trivial logical operator, then the transversal logical gate is in the Mth level of the logical Clifford hierarchy. To recast this condition, it is useful to introduce the following quantity

$$\Omega_M(\mathcal{S}) = \max_{\{\overline{L}_i\}_{i\in[M]}} \min_{L_i\in\overline{L}_i} \left| \bigcap_{i\in[M]} \text{supp}\, L_i \right|, \quad (40)$$

where for each M-tuple of logical Pauli operators $\{\overline{L}_i\}_{i\in[M]}$ we seek an M-tuple of corresponding representatives $L_i \in \overline{L}_i$, whose intersection $\bigcap_{i\in[M]} \text{supp}\, L_i$ is the smallest. Then, we can formulate the following strengthening of Theorem 5 from Ref. [30].

Theorem 7. *Consider a stabilizer code $\mathcal{S}$ with the min-distance $d_\downarrow$. If M is a positive integer satisfying*

$$\Omega_M(\mathcal{S}) < d_\downarrow, \quad (41)$$

then any transversal logical gate for $\mathcal{S}$ is in the Mth level of the logical Clifford hierarchy.

To establish that for the $[\![14,3,3]\!]$ stabilizer code any transversal gate can only implement a logical Clifford gate, it suffices to show that $\Omega_2(\mathcal{S}) < 3$. First, consider any pair of logical Pauli operators $\overline{L}_1$ and $\overline{L}_2$, for which, without loss of generality, $\Delta(\overline{L}_1) > 2$. Because the max-distance is $d_\uparrow = 6$, we can find a representative $L_2 \in \overline{L}_2$ of weight six. By Lemma 4 from Ref. [30], there exists a representative $L_1 \in \overline{L}_1$ such that L_1 and L_2 overlap on at most two qubits. Thus, we restrict our attention to the logical Pauli operators $\overline{L}_1$ and $\overline{L}_2$, such that $\Delta(\overline{L}_i) = 2$ for $i \in [2]$. There are exactly four of these operators; see Fig. 3. By a brute-force search we verify that for

all ten pairs of these operators there exists a choice of representatives that overlap on at most one qubit; see the source code [43]. We finally conclude that $\Omega_2(\mathcal{S}) < 3$ for the $[\![14, 3, 3]\!]$ stabilizer code.

IV.   BOUNDS ON THE DISJOINTNESS

In this section, we provide bounds on the disjointness for the CSS codes, concatenated codes and hypergraph product codes, which, in turn, lead to the limitations on transversal gates available in each code family.

A.   CSS codes

Proposition 8. *Let $\mathcal{S}$ be a CSS code and $\overline{L} = \overline{L}^X \overline{L}^Z$ be a logical Pauli operator written in a standard logical basis. Then, for any positive integer c and $P \in \{X, Z\}$ the largest c-disjoint collection for $\overline{L}^P$ can be formed using P-type operators and the following inequalities hold*

$$\Delta_c\left(\overline{L}\right) \leq \min_{P \in \{X,Z\}} \Delta_c\left(\overline{L}^P\right), \qquad (42)$$

$$\Delta\left(\overline{L}\right) \geq \frac{\Delta\left(\overline{L}^X\right)\Delta\left(\overline{L}^Z\right)}{\Delta\left(\overline{L}^X\right)+\Delta\left(\overline{L}^Z\right)-1}, \qquad (43)$$

where $\Delta\left(\overline{L}\right)$, $\Delta\left(\overline{L}^X\right)$ and $\Delta\left(\overline{L}^Z\right)$ denote the supremum of the c-disjointness for $\overline{L}$, $\overline{L}^X$ and $\overline{L}^Z$, respectively.

Proof. Let $\mathcal{A}$ and $\mathcal{A}^P$ be the largest c-disjoint collections for $\overline{L}$ and $\overline{L}^P$, respectively. Since $\overline{L}^P$ is a P-type logical Pauli operator, there exists a P-type operator L^P, which is a representative of $\overline{L}^P$. Then, for every representative $L \in \mathcal{A}^P$ we can express it in the following way

$$L = L^P S_L^X S_L^Z, \qquad (44)$$

where S_L^X and S_L^Z are some X- and Z-type stabilizer generators of $\mathcal{S}$. Note that $\{L^P S_L^P \mid L \in \mathcal{A}^P\}$ is an example of the desired c-disjoint collection for $\overline{L}^P$. Similarly, for every representative $L \in \mathcal{A}$ we have

$$L = L^X S_L^X L^Z S_L^Z. \qquad (45)$$

Since $\{L^P S_L^P \mid L \in \mathcal{A}\}$ is a c-disjoint collection for $\overline{L}^P$, we conclude that $\Delta_c\left(\overline{L}\right) \leq \Delta_c\left(\overline{L}^P\right)$ and establish Eq. (42).

Theorem 5 guarantees that we can find an integer c^P and a c^P-disjoint collection $\mathcal{A}'^P$, such that $\left|\mathcal{A}'^P\right| = c^P \Delta\left(\overline{L}^P\right)$. Note that $\mathcal{A}' = \{LL' \mid L \in \mathcal{A}'^X, L' \in \mathcal{A}'^Z\}$ is a collection of representatives of $\overline{L}$ of size

$$|\mathcal{A}'| = c^X \Delta\left(\overline{L}^X\right) c^Z \Delta\left(\overline{L}^Z\right). \qquad (46)$$

Since there are at most

$$c' = c^X \left|\mathcal{A}'^Z\right| + c^Z \left|\mathcal{A}'^X\right| - c^X c^Z \qquad (47)$$

$$= c^X c^Z \left(\Delta\left(\overline{L}^X\right) + \Delta\left(\overline{L}^Z\right) - 1\right) \qquad (48)$$

elements of $\mathcal{A}'$ that are supported on any given qubit, thus $\mathcal{A}'$ is c'-disjoint. This, in turn, allows us to conclude that $\Delta\left(\overline{L}\right) \geq |\mathcal{A}'|/c'$ and establish Eq. (43). $\qquad\square$

We remark that Proposition 6 and Proposition 8 immediately imply that

$$\min_{\overline{L} \in \mathcal{L}^X \cup \mathcal{L}^Z} \Delta\left(\overline{L}\right)/2 < \Delta(\mathcal{S}) \leq \min_{\overline{L} \in \mathcal{L}^X \cup \mathcal{L}^Z} \Delta\left(\overline{L}\right), \qquad (49)$$

where $\mathcal{L}^P$ denotes the set of all non-trivial P-type logical operators in a standard logical basis for $P \in \{X, Z\}$. Thus, in order to obtain an approximation up to a multiplicative factor of two to the disjointness of any CSS code it suffices to consider the c-disjointness for its X- and Z-type logical operators.

B.   Concatenated stabilizer codes

Proposition 9. *Let $\mathcal{S}_i$ be a stabilizer code with the min-distance $d_\downarrow^{(i)} > 1$, max-distance $d_\uparrow^{(i)}$ and disjointness $\Delta(\mathcal{S}_i)$, where $i \in [2]$. Then, the disjointness of the concatenated code $\mathcal{S}_1 \lhd \mathcal{S}_2$ satisfies the following inequality*

$$\Delta(\mathcal{S}_1 \lhd \mathcal{S}_2) \geq \Delta(\mathcal{S}_1)\Delta(\mathcal{S}_2). \qquad (50)$$

Moreover, if M is the level of the logical Clifford hierarchy attainable by transversal logical gates for $\mathcal{S}_1 \lhd \mathcal{S}_2$, then

$$M \leq M_{\max} = \max_{i \in [2]} \left\lfloor \log_{\Delta^{(i)}} \left(d_\uparrow^{(i)}/d_\downarrow^{(i)} \right) \right\rfloor + 2. \qquad (51)$$

Proof. Let $\mathcal{P} = \{I, X, Y, Z\}$ be the set of single-qubit Pauli operators (modulo the phase) and n_i be the number of physical qubits of the stabilizer code $\mathcal{S}_i$ for $i \in [2]$. Let K^P be a representative of the logical operator $\overline{K}^P$ for the stabilizer code $\mathcal{S}_2$, where we always choose K^I to be the identity operator. Note that if $L = \bigotimes_{i=1}^{n_1} P_i$ implements a logical Pauli operator $\overline{L}$ for the stabilizer code $\mathcal{S}_1$, where $P_i \in \mathcal{P}$ for $i \in [n_1]$, then the following operator

$$L \lhd \left\{K^P\right\}_{P \in \mathcal{P}} = \bigotimes_{i=1}^{n_1} K^{P_i} \qquad (52)$$

implements $\overline{L}$ for the concatenated code $\mathcal{S}_1 \lhd \mathcal{S}_2$.

Using Theorem 5 and Proposition 6 we can find a positive integer $c^{(2)}$, such that for any $P \in \mathcal{P} \setminus \{I\}$ we have

$$\Delta_{c^{(2)}}\left(\overline{K}^P\right) \geq \Delta(\mathcal{S}_2). \qquad (53)$$

Subsequently, for $\overline{K}^P$ we can find a $c^{(2)}$-disjoint collection

$$\left\{K_i^P \mid i \in [m]\right\}, \qquad (54)$$

where $m = \lceil c^{(2)}\Delta(\mathcal{S}_2)\rceil$. Similarly, we can find a positive integer $c^{(1)}$, such that for any non-trivial logical Pauli operator $\overline{L}$ for the stabilizer code $\mathcal{S}_1$ we have

$$\Delta_{c^{(1)}}\left(\overline{L}\right) \geq \Delta(\mathcal{S}_1). \qquad (55)$$

This, in turn, implies the existence of a $c^{(1)}$-disjoint collection $\mathcal{A}$ for $\overline{L}$ that satisfies the following inequality

$$|\mathcal{A}| \geq \lceil c^{(1)} \Delta(\mathcal{S}_1) \rceil. \tag{56}$$

Let $\mathcal{A}'$ be a collection of representatives of $\overline{L}$ for the concatenated code $\mathcal{S}_1 \lhd \mathcal{S}_2$ defined as follows

$$\mathcal{A}' = \left\{ L \lhd \left\{ K_i^P \right\}_{P \in \mathcal{P}} \,\middle|\, L \in \mathcal{A}, i \in [m] \right\}. \tag{57}$$

By construction, we have

$$|\mathcal{A}'| = m|\mathcal{A}| \geq c^{(1)} c^{(2)} \Delta(\mathcal{S}_1) \Delta(\mathcal{S}_2). \tag{58}$$

It is straightforward to show that $\mathcal{A}'$ is $c^{(1)}c^{(2)}$-disjoint. Thus, we obtain

$$\Delta(\mathcal{S}_1 \lhd \mathcal{S}_2) \geq \Delta_{c^{(1)}c^{(2)}}(\overline{L}) \tag{59}$$
$$\geq |\mathcal{A}'|/(c^{(1)}c^{(2)}) \geq \Delta(\mathcal{S}_1)\Delta(\mathcal{S}_2), \tag{60}$$

which establishes the inequality in Eq. (50)

Let $M_i = \log_{\Delta(\mathcal{S}_i)} \left(d_\uparrow^{(i)}/d_\downarrow^{(i)} \right)$. One can easily establish the following inequalities

$$d_\uparrow(\mathcal{S}_1 \lhd \mathcal{S}_2) \leq d_\uparrow^{(1)} d_\uparrow^{(2)}, \tag{61}$$
$$d_\downarrow(\mathcal{S}_1 \lhd \mathcal{S}_2) \geq d_\downarrow^{(1)} d_\downarrow^{(2)}, \tag{62}$$

which, together with the inequality in Eq. (50), lead to

$$d_\uparrow(\mathcal{S}_1 \lhd \mathcal{S}_2) \leq d_\uparrow^{(1)} d_\uparrow^{(2)} = d_\downarrow^{(1)} d_\downarrow^{(2)} \Delta(\mathcal{S}_1)^{M_1} \Delta(\mathcal{S}_2)^{M_2} \tag{63}$$
$$\leq d_\downarrow(\mathcal{S}_1 \lhd \mathcal{S}_2)\Delta(\mathcal{S}_1)^{M_1} \Delta(\mathcal{S}_2)^{M_2} \tag{64}$$
$$< d_\downarrow(\mathcal{S}_1 \lhd \mathcal{S}_2) \left(\Delta(\mathcal{S}_1) \Delta(\mathcal{S}_2) \right)^{M_{\max}} \tag{65}$$
$$\leq d_\downarrow(\mathcal{S}_1 \lhd \mathcal{S}_2)\Delta(\mathcal{S}_1 \lhd \mathcal{S}_2)^{M_{\max}}, \tag{66}$$

where we use the fact that $\Delta(\mathcal{S}_i) > 1$ and $M_{\max} > M_i$ for $i \in [2]$. Then, using the inequality in Eq. (1), we obtain a bound on the level of the logical Clifford hierarchy attainable by transversal gates for $\mathcal{S}_1 \lhd \mathcal{S}_2$, which is the inequality in Eq. (51). □

We remark that Proposition 9 asserts that the level of the logical Clifford hierarchy attainable by transversal logical gates for the concatenated code $\mathcal{S}_1 \lhd \mathcal{S}_2$ cannot exceed the bounds in Eq. (1) for the stabilizer codes $\mathcal{S}_1$ and $\mathcal{S}_2$. However, these bounds are not necessarily saturated. Thus, we cannot immediately rule out the possibility that transversal logical gates for $\mathcal{S}_1 \lhd \mathcal{S}_2$ may attain a level of the logical Clifford hierarchy higher than the levels attained by transversal logical gates for $\mathcal{S}_1$ and $\mathcal{S}_2$. Lastly, we acknowledge that the bound in Eq. (50) has been previously independently derived in Ref. [31].

C. Hypergraph product codes

Let $\mathcal{S}$ be a CSS code and $\mathcal{L}^P$ be the set of all non-trivial P-type logical operators in a standard logical basis, where $P \in \{X, Z\}$. We now introduce the P-type disjointness $\Delta^P(\mathcal{S})$ of the CSS code $\mathcal{S}$ as follows

$$\Delta^P(\mathcal{S}) = \sup_{c \geq 1} \min_{\overline{L} \in \mathcal{L}^P} \Delta_c(\overline{L}). \tag{67}$$

Similarly as for the disjointness, we can show that the P-type disjointness satisfies the following equality

$$\Delta^P(\mathcal{S}) = \min_{\overline{L} \in \mathcal{L}^P s} \sup_{c \geq 1} \Delta_c(\overline{L}). \tag{68}$$

By definition, we have

$$\Delta^P(\mathcal{S}) \geq \Delta(\mathcal{S}). \tag{69}$$

The P-type disjointness $\Delta^P(\mathcal{S})$ is useful when the stabilizer group $\mathcal{S}$ is generated by only P-type operators. In such a case, the min-distance of the stabilizer code $\mathcal{S}$ is one, i.e., $d_\downarrow = 1$. Subsequently, Lemma 2(ii) in Ref. [30] implies that $\Delta(\mathcal{S}) = 1$; however, $\Delta^P(\mathcal{S})$ might still be greater than one. This proves useful in the following proposition, where we establish an upper bound on the disjointness of hypergraph product codes.

Proposition 10. *Let H_i be a full-rank binary matrix of size $m_i \times n_i$ and $\mathcal{S}_i$ be a stabilizer code specified by the binary matrix $(H_i|0_{m_i,n_i})$, where $i \in [2]$. Let $\mathcal{S}$ be a hypergraph product code specified by the binary matrix in Eq. (2). Then, the following inequality holds*

$$\Delta(\mathcal{S}) \leq \min_{i \in [2]} \Delta^X(\mathcal{S}_i). \tag{70}$$

Proof. In what follows, we perform arithmetic operations modulo 2. Let span M denote the row span of a binary matrix M. Note that a Pauli operator L_1 is a representative of some non-trivial X-type logical operator $\overline{L}_1$ in a standard logical basis for the stabilizer code $\mathcal{S}_1$ iff L_1 is specified by the row vector $(l_1|0_{1,n_1}) \in \{0,1\}^{2n_1}$, such that $l_1 \notin \text{span } H_1$. Let $l_2 \in \{0,1\}^{m_2}$ be a non-zero row vector, such that $l_2 H_2 = 0_{1,n_2}$, and define

$$l = (l_1 \otimes l_2 | 0_{1,n_2 m_1}). \tag{71}$$

Then, we have

$$\left(I_{n_1} \otimes H_2^{\mathrm{T}} \middle| H_1^{\mathrm{T}} \otimes I_{n_2} \right) l^T = 0_{n_1 n_2, 1}. \tag{72}$$

Moreover, $l \notin \text{span}(H_1 \otimes I_{m_2}|I_{m_1} \otimes H_2)$; otherwise, we would obtain that $l_1 \in \text{span } H_1$, leading to a contradiction. We thus conclude that a Pauli operator L specified by the row vector $(l|0_{1,n_1 m_2 + n_2 m_1})$ is a representative of some X-type logical operator for the stabilizer code $\mathcal{S}$.

Let c be a positive integer and $\mathcal{A}$ be a c-disjoint collection for $\overline{L}$ of the largest possible size, i.e., $|\mathcal{A}| = c\Delta_c(\overline{L})$. Since the stabilizer code $\mathcal{S}$ is a CSS code and $\overline{L}$ is an X-type logical operator, thus Proposition 8 implies that

we can select $\mathcal{A}$ in a way that it contains only X-type operators. Let $\lambda \in [m_2]$ be a position of any non-zero entry of l_2 and define the index subset

$$\Lambda = \{\lambda + (i-1)m_2 \mid i \in [n_1]\}. \tag{73}$$

We also treat Λ as the subset of qubits whose indices belong to Λ. By definition of the stabilizer group of the hypergraph product code, if S is an X-type stabilizer operator from $\mathcal{S}$, then its restriction $S|_\Lambda$ to the subset of qubits Λ is an X-type stabilizer operator from $\mathcal{S}_1$. Moreover, for the representative L of the logical operator $\overline{L}$, its restriction $L|_\Lambda$ is a representative of the logical operator $\overline{L}_1$. Thus, a collection

$$\mathcal{A}_\Lambda = \{K|_\Lambda \mid K \in \mathcal{A}\} \tag{74}$$

comprises representatives of $\overline{L}_1$. Since $\mathcal{A}$ is c-disjoint, it has to be c-disjoint on the subset of qubits Λ, which immediately implies that $\mathcal{A}_\Lambda$ is c-disjoint. Thus, we obtain

$$\Delta_c(\overline{L}_1) \geq |\mathcal{A}_\Lambda|/c = |\mathcal{A}|/c = \Delta_c(\overline{L}), \tag{75}$$

and, subsequently, $\Delta^X(\mathcal{S}_1) \geq \Delta(\mathcal{S})$.

To establish the inequality in Eq. (70), we show that $\Delta^X(\mathcal{S}_2) \geq \Delta(\mathcal{S})$ in a similar way. First, we note that for any X-type logical operator $\overline{L}_2$ for the stabilizer code $\mathcal{S}_2$ a Pauli operator L_2 is a representative of $\overline{L}_2$ iff L_2 is specified by a row vector $(l_2'|0_{1,n_2}) \in \{0,1\}^{2n_2}$, such that $l_2' \notin \text{span } H_2$. Then, we choose $l_1' \in \{0,1\}^{m_1}$ to be any non-zero row vector, such that $l_1' H_1 = 0_{1,n_1}$, and define $l' = (0_{1,n_1 m_2}|l_1' \otimes l_2')$. Finally, let c be a positive integer. We can then show that any c-disjoint collection for the logical operator $\overline{L}'$, whose representative is specified by the row vector $(l'|0_{1,n_1 m_2 + n_2 m_1})$, gives rise to a c-disjoint collection for $\overline{L}_2$. This, in turn, implies that $\Delta_c(\overline{L}_2) \geq \Delta_c(\overline{L}')$, and, subsequently, $\Delta^X(\mathcal{S}_2) \geq \Delta(\mathcal{S})$. $\qquad\square$

We remark that unlike other bounds in this section, Eq. (70) provides an upper bound on the disjointness. As such, it does not lead to a bound on the level of the Clifford hierarchy attainable by transversal logical gates for the hypergraph product code.

V. DISCUSSION

The main result of our work, which is Theorem 4, established that for any positive integer constant c the problem of calculating the c-disjointness (or even approximating it up to within a multiplicative factor) is NP-complete. Although we have not shown that calculating the disjointness of stabilizer codes is hard, Theorem 4 suggests so. In general, our results indicate that finding fault-tolerant logical gates for generic quantum error-correcting codes is a computationally challenging task. Other results presented in our work included: (i) formulating a linear program to calculate the disjointness, (ii) strengthening the main result of Ref. [30], and (iii) providing bounds on the disjointness for various stabilizer code families. Lastly, we remark that the source code [43] can be used to numerically estimate the disjointness of small stabilizer codes and, subsequently, rule out the existence of certain transversal logical gates.

We hope that our work initiates and motivates a thorough search of methods of finding the disjointness, as well as fault-tolerant logical gates at large. We expect that there exist efficient algorithms to calculate or approximate the disjointness for certain code families, such as topological quantum codes. We emphasize that clever usage of the underlying code symmetries might further simplify this problem. For example, for codes that are invariant under certain permutations of qubits, the number of variables in the linear program in Eqs. (27)-(29) to find the disjointness can be reduced from 2^n to $2^n/n$. Moreover, it would be interesting to make the connection between the disjointness and other code quantities, such as the price [44].

[1] P. Shor, Fault-tolerant quantum computation, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE Comput. Soc. Press, 1996) pp. 56–65.

[2] A. M. Steane, Active stabilization, quantum computation, and quantum state synthesis, Physical Review Letters **78**, 2252 (1997).

[3] E. Knill, Quantum computing with realistically noisy devices, Nature **434**, 39 (2005).

[4] E. Knill, Scalable quantum computing in the presence of large detected-error rates, Physical Review A **71**, 042322 (2005).

[5] H. Bombín, Gauge color codes: Optimal transversal gates

and gauge fixing in topological stabilizer codes, New Journal of Physics **17**, 083002 (2015).

[6] A. Kubica and M. E. Beverland, Universal transversal gates with color codes: A simplified approach, Physical Review A **91**, 032330 (2015).

[7] F. H. Watson, E. T. Campbell, H. Anwar, and D. E. Browne, Qudit color codes and gauge color codes in all spatial dimensions, Physical Review A **92**, 022312 (2015).

[8] A. Kubica, B. Yoshida, and F. Pastawski, Unfolding the color code, New Journal of Physics **17**, 083026 (2015).

[9] H. Bombín, Transversal gates and error propagation in 3D topological codes, arXiv:1810.09575 (2018).

[10] T. Jochym-O'Connor and T. J. Yoder, Four-dimensional toric code with non-Clifford transversal gates, Physical Review Research **3**, 13118 (2021).

[11] M. Vasmer and A. Kubica, in preparation (2021).

[12] H. Bombín, Dimensional jump in quantum error correction, New Journal of Physics **18**, 043038 (2016).

[13] H. Bombín, 2D quantum computation with 3D topological codes, arXiv:1810.09571 (2018).

[14] A. Kubica, *The ABCs of the color code: A study of topological quantum codes as toy models for fault-tolerant quantum computation and quantum phases of matter*, Ph.D. thesis (2018).

[15] M. Vasmer and D. E. Browne, Three-dimensional surface codes: Transversal gates and fault-tolerant architectures, Physical Review A **100**, 012312 (2019).

[16] B. J. Brown, A fault-tolerant non-clifford gate for the surface code in two dimensions, Science Advances **6**, 10.1126/sciadv.aay4929 (2020).

[17] J. Iverson and A. Kubica, in preparation (2021).

[18] E. Knill, Fault-Tolerant Postselected Quantum Computation: Schemes, arXiv:quant-ph/0402171 (2004).

[19] E. Knill, Fault-Tolerant Postselected Quantum Computation: Threshold Analysis, arXiv:quant-ph/0404104 (2004).

[20] S. Bravyi and A. Kitaev, Universal quantum computation with ideal Clifford gates and noisy ancillas, Physical Review A **71**, 022316 (2005).

[21] M. E. Beverland, A. Kubica, and K. M. Svore, Cost of Universality: A Comparative Study of the Overhead of State Distillation and Code Switching with Color Codes, PRX Quantum **2**, 020341 (2021).

[22] B. Eastin and E. Knill, Restrictions on Transversal Encoded Quantum Gate Sets, Physical Review Letters **102**, 110502 (2009).

[23] B. Zeng, A. Cross, and I. L. Chuang, Transversality versus universality for additive quantum codes, IEEE Transactions on Information Theory **57**, 6272 (2011).

[24] P. Faist, S. Nezami, V. V. Albert, G. Salton, F. Pastawski, P. Hayden, and J. Preskill, Continuous Symmetries and Approximate Quantum Error Correction, Physical Review X **10**, 041018 (2020).

[25] M. P. Woods and Á. M. Alhambra, Continuous groups of transversal gates for quantum error correcting codes from finite clock reference frames, Quantum **4**, 245 (2020).

[26] A. Kubica and R. Demkowicz-Dobrzański, Using Quantum Metrological Bounds in Quantum Error Correction: A Simple Proof of the Approximate Eastin-Knill Theorem, Physical Review Letters **126**, 150503 (2021).

[27] S. Bravyi and R. König, Classification of Topologically Protected Gates for Local Stabilizer Codes, Physical Review Letters **110**, 170503 (2013).

[28] F. Pastawski and B. Yoshida, Fault-tolerant logical gates in quantum error-correcting codes, Physical Review A **91**, 13 (2015).

[29] M. E. Beverland, O. Buerschaper, R. Koenig, F. Pastawski, J. Preskill, and S. Sijher, Protected gates for topological quantum field theories, Journal of Mathematical Physics **57**, 44 (2016).

[30] T. Jochym-O'Connor, A. Kubica, and T. J. Yoder, Disjointness of Stabilizer Codes and Limitations on Fault-Tolerant Logical Gates, Physical Review X **8**, 021047 (2018).

[31] P. Webster, M. Vasmer, T. R. Scruby, and S. D. Bartlett, Universal Fault-Tolerant Quantum Computing with Stabiliser Codes, arXiv:2012.05260 (2020).

[32] D. Gottesman, Class of quantum error-correcting codes saturating the quantum Hamming bound, Physical Review A **54**, 1862 (1996).

[33] D. Gottesman and I. L. Chuang, Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations, **402**, 390 (1999).

[34] A. J. Landahl, The surface code on the rhombic dodecahedron, arXiv:2010.06628 (2020).

[35] A. Calderbank and P. Shor, Good quantum error-correcting codes exist, Physical Review A **54**, 1098 (1996).

[36] A. Steane, Multiple-Particle Interference and Quantum Error Correction, Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences **452**, 2551 (1996).

[37] E. Knill and R. Laflamme, Concatenated Quantum Codes, arXiv:quant-ph/9608012 (1996).

[38] J. P. Tillich and G. Zemor, Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength, IEEE Transactions on Information Theory **60**, 1193 (2014).

[39] M. R. Garey and D. S. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness* (W. H. Freeman and Company, 1979).

[40] J. Hastad, Clique is hard to approximate within n/sup 1-/spl epsiv, in *Proceedings of 37th Conference on Foundations of Computer Science* (IEEE, 1996) pp. 627–636.

[41] D. Zuckerman, Linear degree extractors and the inapproximability of max clique and chromatic number, Theory of Computing **3**, 103 (2007).

[42] G. B. Dantzig, Maximization of a linear function of variables subject to linear inequalities, Activity analysis of production and allocation **13**, 339 (1951).

[43] J. Bostanci, Disjointness, https://github.com/quantumCodewords/disjointness (2020).

[44] F. Pastawski and J. Preskill, Code Properties from Holographic Geometries, Physical Review X **7**, 021022 (2017).