No-Go Theorem on Fault Tolerant Gadgets for Multiple Logical Qubits

Aranya Chakraborty[1,*] and Daniel Gottesman[2,†]
[1]Department of Physics, University of Maryland, College Park, MA, USA
[2]QUICS, University of Maryland, College Park, MA, USA

(Received 27 February 2026; revised 29 April 2026; accepted 23 June 2026; published 13 August 2026)

Identifying stabilizer codes that admit fault tolerant implementations of the full logical Clifford group would significantly advance fault tolerant quantum computation, as such implementations prevent uncontrolled error propagation. Motivated by this goal, we study several classes of fault tolerant gadget constructions consisting of Clifford gates acting on the physical qubits, including transversal gadgets, code automorphisms, and fold-transversal gadgets. While stabilizer codes encoding a single logical qubit—most notably the $[[7,1,3]]$ Steane code—are known to admit transversal implementations of the full logical Clifford group, no analogous examples are known for codes encoding multiple logical qubits. In this work, we prove a no-go theorem establishing that no stabilizer code admits a fully transversal implementation of the Clifford group on more than one logical qubit. We further strengthen this result by showing that fold-transversal implementations of the full logical Clifford group are impossible for stabilizer codes encoding more than two logical qubits. More generally, we introduce the notion of k-fold transversal gadgets and prove that implementing the full Clifford group on k logical qubits requires at least k-fold transversal gadgets at the physical level. In addition, we analyze code-automorphism-based constructions and demonstrate that they also fail to realize the full Clifford group on multiple logical qubits for any stabilizer code. Together, these results place fundamental constraints on fault tolerant Clifford gadget design and show that stabilizer codes supporting the full logical Clifford group on multiple logical qubits via these architectures do not exist. Since the Clifford group is a core component of universal gate sets, our findings imply that quantum computing with codes encoding multiple logical qubits within a single code block necessarily entails more complex constructions for fault tolerance.

DOI: 10.1103/y14y-7kp3

I. INTRODUCTION

Quantum computers leverage the laws of quantum mechanics in order to solve a variety of problems in physics and chemistry far beyond the capabilities of a classical computer [1]. However, working on quantum computers also leaves us vulnerable to a much wider range of noise than that seen on a classical computer. Thus, one of the cornerstones in realizing the *quantum advantage* is the ability to encode the logical information and protect it from errors. The most widely used and reliable way to protect quantum information is by using the class of stabilizer codes [2], where we encode the logical information onto a larger number of highly entangled physical qubits. This allows us to protect the encoded information from the inherent noise in a quantum system as long as the error is below a certain threshold [3,4]. Recently, significant strides have been made toward demonstrating practical error correction in a variety of qubit architectures [5–8]. However, in order to do meaningful computation, in addition to encoding the information, we need a way of performing operations on the encoded logical information. In particular, a class of logical operations that are of interest is the Clifford group (Cl_n) due to its significance in quantum error correction and simulation [9]. More importantly, the Clifford group also plays an important role in the construction of universal gate sets [10] as any non-Clifford gate in addition to the full Clifford group gives us the universal gate set [11,12].

At the same time we need a way to perform the desired logical operations in a *fault tolerant* manner, that is without spreading errors onto multiple qubits, thereby preventing uncontrolled error propagation. A natural approach is to employ transversal gates, which are composed of single-qubit Clifford operations acting independently on each physical qubit. Because transversal gates do not involve multi-qubit interactions, they are inherently fault tolerant and suppress the propagation of errors during

*Contact author: aranyac@umd.edu
†Contact author: dgottesm@umd.edu

implementation. Moreover, the product of two transversal gates is itself transversal; consequently, any logical operation realizable through transversal gates can be implemented with constant circuit depth.

Even though no stabilizer code admits a transversal implementation of the universal gate set [13,14], there exist various methods to introduce non-Clifford gates, including magic state injection [15,16] and error correction [17]. These methods, combined with stabilizer codes having a transversal implementation of the full logical Clifford group, provide the most realistic method to obtain fault tolerant universal quantum computation in the near term. Several quantum codes encoding a single logical qubit and permitting transversal implementation of the entire logical Clifford group have been identified, with the $[[7, 1, 3]]$ Steane code [18] being the smallest such example. More recently, attention has shifted toward quantum codes that encode multiple logical qubits and support addressable transversal gates [19,20], which allow us to interact with each logical qubit independently. Such codes not only provide a better rate of encoding and reduce qubit overhead but also allow more flexible code switching and gauge fixing frameworks [17,21] and enable more efficient implementations of non-Clifford operations [22,23]. This could allow us to circumvent Eastin-Knill restrictions and expand the accessible fault tolerant gate sets.

However, the number of logical actions that we can perform while limiting ourselves to transversal gadgets at the physical level is usually small for general stabilizer codes, and indeed, no stabilizer code has been found that supports the transversal implementation of the full logical Clifford group for multiple qubits. Hence, there exists a need to expand the notion of transversality, and there exist two leading ways to achieve this. The first is by using code automorphisms where we allow for qubit permutations in addition to transversal Clifford gates at the physical level [24–27]. The second is by using fold-transversal gates [28–32] which use multi-qubit Clifford gates within a single physical code block, which spreads errors but can implement a greater number of logical actions.

In this paper, we establish a no-go theorem that limits the performance of both transversal and fold transversal Clifford gadgets, proving that no stabilizer code has a Clifford transversal implementation of the full Clifford group on multiple logical qubits or a Clifford fold transversal implementation of the full Clifford group on more than two logical qubits.

The theorem can be generalized for $[[n, k, d]]$ codes by introducing k-fold transversal gadgets and proving that in order to implement the full logical Clifford gate on k logical qubits we need to implement at least k-fold transversal Clifford gadgets at the physical level. In the worst case, a k-fold transversal gadget can propagate a single qubit error onto k physical qubits and hence becomes decreasingly fault tolerant as k increases. An important example of k-fold transversal gates is transversal gates between k blocks of a code. Our theorem shows that even by using multiple blocks, Clifford transversal gates cannot achieve the full logical Clifford group unless each block has only a single logical qubit.

We further prove a constraint on using code automorphisms as physical gadgets and show that they can never implement the full logical Clifford group for any stabilizer code with multiple logical qubits. Thus the theorems presented in this paper establish a clear constraint on high rate codes, limiting the number of logical actions which can be implemented using the fault tolerant Clifford gadgets considered in this paper. As a result it becomes important to explore other pathways to universal fault tolerant computation including code switching [17,33–35], flag fault tolerance [36–38], lattice surgery [39,40] etc.

The rest of the paper is organized as follows. In Sec. II we go over some preliminaries which will be required for the rest of the paper. We then define the central theorems of our paper in Sec. III and provide a proof for the theorems in Sec. IV. We then end our work with a discussion on the impact of the theorems presented and future directions which can be pursued in Sec. V.

II. PRELIMINARIES

A. Stabilizer codes

The stabilizer formalism provides a compact description of a large class of quantum codes and operations [2]. A $[[n, k, d]]$ stabilizer code encodes k logical qubits into n physical qubits with distance d. It is defined by an abelian subgroup $\mathcal{S} \subset \mathcal{P}_n$ of the n-qubit Pauli group

$$\mathcal{P}_n = \langle iI, X_j, Z_j \colon j = 1, \ldots, n \rangle, \tag{1}$$

where each element of $\mathcal{S}$ fixes the codespace. The codespace is therefore the joint $+1$ eigenspace of all stabilizer generators.

a. Binary symplectic representation: Every Pauli operator on n qubits can be written uniquely (up to phase) as

$$P(\mathbf{a}, \mathbf{b}) = \bigotimes_{j=1}^{n} X_j^{a_j} Z_j^{b_j}, \tag{2}$$

with binary vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}_2^n$. This identifies each Pauli operator with a binary vector

$$(\mathbf{a}|\mathbf{b}) \in \mathbb{F}_2^{2n}. \tag{3}$$

The commutation relation between two Pauli operators $P(\mathbf{a}, \mathbf{b})$ and $P(\mathbf{a}', \mathbf{b}')$ is determined by the *symplectic inner product*

$$\langle(\mathbf{a},\mathbf{b}),(\mathbf{a}',\mathbf{b}')\rangle = (\mathbf{a},\mathbf{b})^T J(\mathbf{a}',\mathbf{b}') = \mathbf{a}\cdot\mathbf{b}' + \mathbf{b}\cdot\mathbf{a}' \pmod 2. \quad (4)$$

where $J = \begin{bmatrix} 0 & \mathrm{Id}_n \\ \mathrm{Id}_n & 0 \end{bmatrix}$.

Two Paulis commute if their symplectic inner product vanishes. The stabilizer $\mathcal{S}$ can be represented as the row space of a binary $(n-k)\times 2n$ matrix

$$S = \begin{bmatrix} \mathbf{a}_1 & \mathbf{b}_1 \\ \vdots & \vdots \\ \mathbf{a}_{n-k} & \mathbf{b}_{n-k} \end{bmatrix}, \quad (5)$$

with rows encoding the stabilizer generators.

B. Clifford group

The Clifford group Cl_n is defined as the normalizer of $\mathcal{P}_n$, i.e.

$$U\in\mathrm{Cl}_n \quad\Leftrightarrow\quad UPU^\dagger\in\mathcal{P}_n \quad \forall\, P\in\mathcal{P}_n. \quad (6)$$

We define the projective Clifford group as the Clifford group mod phases and the Paulis, i.e. $\mathrm{Proj}\,\mathrm{Cl}_n := \mathrm{Cl}_n/(U(1),\mathcal{P}_n)$. The projective Clifford group is isomorphic to the classical symplectic group such that for each $U\in\mathrm{Proj}\,\mathrm{Cl}_n$ there exists a $M_U\in\mathrm{Sp}(2n,2)$ such that,

$$U^\dagger P(\mathbf{a},\mathbf{b})U = P(\mathbf{a}',\mathbf{b}') \quad\Leftrightarrow\quad \mathrm{M_U}\cdot(\mathbf{a},\mathbf{b}) = (\mathbf{a}',\mathbf{b}'). \quad (7)$$

Thus, each operator in the projective Clifford group can be represented as a $2n\times 2n$ binary matrix acting linearly on the binary representation of the Pauli operators. Since Clifford operators are unitaries, they must preserve the commutation relation between Pauli operators, and this constraint is equivalent to their binary representation satisfying the symplectic condition.

$$M_U^T J M_U = J, \quad (8)$$

For instance, on a single qubit:

$$H:\ \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \mapsto M_H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad (9)$$

$$S:\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix} \mapsto M_S = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}. \quad (10)$$

The **order** of a Clifford operator $U\in\mathrm{Cl}_n$ is defined as the smallest integer r such that

$$U^r = \mathrm{Id}_n. \quad (11)$$

Since the Clifford group is a finite group, all Clifford operators have a finite order. For example, the single-qubit projective Clifford group only has elements having order: 1 (e.g. Id), 2 (e.g. H and S) and 3 (e.g. HS).

C. Field theory

A field $\mathbb{F}$ is an algebraic structure where mathematical operations such as addition, subtraction, multiplication etc. are well defined. Since we are working with qubits, a field of particular importance to us is the binary field $\mathbb{F}_2 = \{0,1\}$. Finite fields having sizes which are powers of primes can be constructed using field extensions: for any finite field $\mathbb{F}_p$ of size p and for each prime power $q = p^m$, there exists a unique field $\mathbb{F}_q$ which can be constructed as

$$\mathbb{F}_q \cong \mathbb{F}_p[x]/(f(x)), \quad (12)$$

where $\mathbb{F}_p[x]$ is the ring of polynomials with coefficients in the field $\mathbb{F}_p$ and $f(x)$ is an irreducible polynomial of degree m.

Let $\alpha = x \bmod f(x)$.α is then called the residue class of x and every element of $\mathbb{F}_q$ can be written as

$$a_0 + a_1\alpha + \cdots + a_{m-1}\alpha^{m-1}, \qquad a_i\in\mathbb{F}_p. \quad (13)$$

An irreducible polynomial is called *primitive* if α has multiplicative order $p^m - 1$; in this case, α generates the multiplicative group $\mathbb{F}_q^\times = \mathbb{F}_q\backslash\{0\}$ that is,

$$\mathbb{F}_q^\times = \langle\alpha\rangle = \{\alpha^k :\ 1\le k\le p^m - 1\}.$$

Example. Let us define $\mathbb{F}_9 = \mathbb{F}_{3^2}$ using the primitive polynomial

$$f(x) = x^2 + x + 2\in\mathbb{F}_3[x], \quad (14)$$

and let $\alpha = x \bmod f(x)$. Then α satisfies $\alpha^2 = 2\alpha + 1$. Every element of $\mathbb{F}_9$ can therefore be uniquely written as

$$a_0 + a_1\alpha, \qquad a_i\in\mathbb{F}_3. \quad (15)$$

Then, the multiplicative group $\mathbb{F}_9^\times$ can be generated by successive powers of α such that

$$\begin{aligned} \alpha^1 &= \alpha \\ \alpha^2 &= 2\alpha + 1 \\ \alpha^3 &= 2\alpha^2 + \alpha = 2\alpha + 2 \\ \alpha^4 &= 2\alpha^2 + 2\alpha = 2 \\ \alpha^5 &= 2\alpha \\ \alpha^6 &= 2\alpha^2 = \alpha + 2 \\ \alpha^7 &= \alpha^2 + 2\alpha = \alpha + 1 \\ \alpha^8 &= \alpha^2 + \alpha = 1. \end{aligned} \quad (16)$$

D. Fault tolerance

In order to do computation using stabilizer codes, we use physical *gadgets* $\in \mathrm{SU}(n)$ which act on the physical qubits and perform some *logical action* $\in \mathrm{SU}(k)$ on the logical qubits. In this paper, we limit our attention to the physical Clifford group and only consider gadgets $\in \mathrm{Cl}_n$. In order for a physical operation (U) to be a gadget it must preserve the code space i.e. map codewords to other codewords. Equivalently, for stabilizer codes, the above condition is equivalent to

$$U^\dagger S U \in \mathcal{S} \quad \forall\ S \in \mathcal{S} \tag{17}$$

However, in order to prevent fault propagation we need the physical gadgets to be *fault tolerant*. One of the best ways to ensure fault tolerance is to make the physical gadgets local and minimize multi-qubit operators. Inherently fault tolerant gadgets, known as transversal gadgets, act independently on each individual physical qubit,

$$U_{\mathrm{trans}} = \bigotimes_{i=1}^{n} U_i \tag{18}$$

and prevent fault propagation within a code block. We further limit each U_i to be in the single-qubit Clifford group and call these gadgets Clifford transversal gadgets.

The term transversal is also used for gadgets between k code blocks, but now each U_i acts on the qubit number i within each of the k blocks. Thus, each U_i acts only one qubit within a block. In this paper, we focus on transversal gadgets on single blocks, but multiple-block transversal gadgets can be seen as an example of k-fold transversal gadgets, which we introduce in Sec. III.

The idea of transversal gates has been expanded recently to include some two qubit operations within a single code block. These are known as fold transversal gadgets. They are constructed with respect to a chosen ZX duality τ. A unitary U is considered to be fold transversal if it is supported on $\{i, \tau(i)\}$ for $i = 1, 2, \cdots, n$ [41]. We further restrict $U \in \mathrm{Cl}_n$ and call them Clifford fold transversal gadgets. These gadgets exploit the geometrical properties of the code and only propagate single-qubit errors to two qubit errors which have independent syndromes. This allows us to identify and correct these two qubit errors which show up most commonly during implementation, even if these codes cannot correct any arbitrary two qubit errors.

The idea of transversal gates can also be relaxed on certain qubit architectures such as Rydberg atoms [42], ions [43] etc. In these architectures, SWAP gates can also be made fault tolerant since qubit relabeling can be done in software, enabling us to construct fault tolerant gadgets made up of a Clifford transversal gadget followed by a permutation of physical qubits. Such gadgets are known as code automorphisms.

In order to prevent the spread of errors it is important to enforce as strict a notion of fault tolerance as possible. In terms of their fault tolerant properties, that is how well they are able to limit the spread of errors, the gadgets can be ranked as follows:

$$\begin{aligned}\text{Transversal Gadgets} &\geq \text{Code Automorphisms} \\ &> \text{Fold Transversal Gadgets.}\end{aligned}$$

However, compromising on fault tolerance could allow us more flexibility in terms of the logical actions available to us.

III. CENTRAL THEOREM

Transversal gadgets, though a very desirable fault tolerant method to construct physical gadgets, are very limited in their ability to perform logical actions for most stabilizer codes. Our first theorem provides a fundamental constraint on the logical actions that can be implemented by using transversal Clifford gadgets at the physical level. We limit ourselves only to the logical Clifford group since it is an important ingredient to achieve universal quantum computation. We then show that for stabilizer codes having more than one logical qubit, it is impossible to perform the whole logical Clifford group using just transversal Clifford gadgets. In order to generalize it to stabilizer codes having multiple logical qubits we introduce the notion of k-fold transversal gadgets (similar to the notion of transversality introduced in Ref. [44]). This notion generalizes the concept of fold transversality and under this convention, transversal gates are 1-fold transversal gadgets, and fold transversal gates under a fixed pairing of physical qubits are an example of 2-fold transversal gadgets.

A set of gadgets (G) is k-fold transversal if there exists a disjoint partition of physical qubits:

$$\{s_i\}_{i=1}^{m} : \bigcup_{i=1}^{m} s_i = \{1, 2, \ldots, n\}, \tag{19}$$

such that

$$|s_i| \leq k \quad \forall\ i, \tag{20}$$

and every $U \in G$ can be expressed as a tensor product

$$U = \bigotimes_{i=1}^{m} U_{s_i}, \tag{21}$$

where U_{s_i} acts only on the qubits in s_i. In this work, we restrict $U_{s_i} \in \mathrm{Cl}_{|s_i|}$ and call these gadgets Clifford k-fold transversal gadgets.

Under this definition, k-fold transversal gadgets share a fundamental property with both transversal gadgets and fold-transversal gadgets. If U and V are k-fold transversal

with respect to a given partition, then their product UV is also k-fold transversal for the same partition. Consequently, the collection of k-fold transversal gadgets associated with a fixed partition forms a group. It follows that any logical operation realizable through k-fold transversal gadgets can be implemented with constant circuit depth.

This is a significant property, since for every stabilizer code, fold-transversal gates over arbitrary partitions always permit implementation of the full logical Clifford group. This is because {H, S, CNOT}, which is a two qubit gate set, allows us to implement an arbitrary physical Clifford circuit and thus all logical Clifford gates. However, these gates no longer form a group structure themselves, and, as a result, implementing an arbitrary logical Clifford operation would generally require non-constant circuit depth.

We now present the first theorem.

Theorem 1. In order to implement the full Clifford group on stabilizer codes with k logical qubits, we need at least Clifford k-fold transversal gadgets acting on the physical qubits.

Corollary. There exists no stabilizer code which permits a Clifford transversal implementation of the full Clifford group on multiple logical qubits or a Clifford fold transversal implementation of the full Clifford group on more than two logical qubits.

The corollary follows from the definition of the k-fold transversal gates. A simple example to illustrate Theorem 1 would be to consider k blocks of the $[[7, 1, 3]]$ code. The full logical Clifford group can be performed for this code by using transversal gadgets in each code block and transversal CNOT gadgets between code blocks. However, we can also think of the aggregate code as a single $[[7k, k, 3]]$ code. Since the CNOT gates now act within a single code block, they are no longer transversal and are in fact k-fold transversal (as they act on one qubit from each block). Thus, this is a straightforward example of a stabilizer code having the full Clifford group on k logical qubits implemented using Clifford k-fold transversal gadgets, which also shows our theorem is tight. An important observation is that, if one instead considers k blocks of a CSS code encoding two logical qubits, yielding an $[[nk, 2k, d]]$ code, then even in the presence of transversal inter-block CNOT gates, one can realize at most the full logical Clifford group on only k logical qubits.

We present another theorem that constrains the logical actions of code automorphisms at the physical level. Code automorphisms, similar to k-fold transversal gadgets, form a group; therefore, any logical operation realizable through code automorphisms can be implemented with constant circuit depth.

Theorem 2. There exists no stabilizer code that admits the full Clifford group on multiple logical qubits implemented using code automorphisms.

The above two theorems provide a fundamental constraint on the logical actions available for a stabilizer code using fault tolerant Clifford gadgets such as fold transversal gates and code automorphisms. This shows that in order to do the full Clifford group on multiple logical qubits in a fault tolerant manner, more complex constructions would be required.

IV. PROOF OF THE CENTRAL THEOREMS

A. Proof of theorem 1

In this section, we provide a proof for Theorem 1 stated in Sec. III. We first show a correspondence between the order of the physical gadget acting on the physical qubits and the logical operation implemented by it.

Lemma 1. [Order Lemma]: The order of the physical gadget must be a multiple of the order of the logical action induced by it.

Proof. Let $U \in \mathrm{Cl}_n$ be the physical gadget on the physical qubits and $\bar{U} \in \mathrm{Cl}_k$ be its logical action and let $\mathrm{ord}(U) = r$. Then, U^r acts trivially on the physical qubits and would by definition induce a trivial logical action i.e.

$$U^r = \mathrm{Id_n} \quad \Rightarrow \quad \bar{U}^{\,r} = \overline{\mathrm{Id_k}}. \tag{22}$$

Hence, the logical action induced by the physical gadget U must have an order that is a factor of $\mathrm{ord}(U)$. ∎

Corollary. If the order of the physical gadget is prime, then the induced logical action either has the same order or acts trivially on the logical qubits.

The above lemma is valid for any physical gadget $U \in \mathrm{SU(n)}$, but we will be restricting our attention to physical gadgets belonging to a subset of the full unitary group, the Clifford group (Cl_n). Physical Clifford gadgets always induce a Clifford logical action on a stabilizer code

$$U \in \mathrm{Cl}_n \quad \Rightarrow \quad \bar{U} \in \mathrm{Cl}_k.$$

An important thing to note is that elements in the Clifford group can, in principle, have any order by adding a phase term. However, since a phase as well as a Pauli gadget induces a trivial logical action we would be ignoring their effect and restricting ourselves to physical gadgets in the projective Clifford group.

The key idea of the proof is to show that there exists a unitary operation in Cl_k of prime order that cannot be realized within Cl_{k-1}. Consequently, to implement this Clifford operation on the logical qubits, one must realize a physical gadget belonging to Cl_k, corresponding to a k-fold transversal gate. We show this with the help of the following lemmas.

Lemma 2. [Zsigmondy's Theorem]: There exists a prime p which divides $2^{2k} - 1$ and does not divide $2^{2i} - 1$ for all $i < k$.

Proof. The above lemma is an application of Zsigmondy's theorem, which was first presented in Ref. [45] (for a more abridged proof see Ref. [46]). The theorem states that for any a, b which are positive

co-prime integers and any integer $n \geq 1$, there exists a prime (also known as primitive prime divisor) number p that divides $a^n - b^n$ and does not divide $a^i - b^i$ for any positive integer $i < n$.

Lemma 2 is a specific case of the above theorem with $a = 2$ and $b = 1$ (this simplification is also known as Bang's theorem and was proved earlier in Ref. [47]). We further simplify by only considering even exponents i.e. $n = 2k$ for any integer $k \geq 1$. An important thing to note is that Bang's theorem has an exception for $n = 6$, where we get $2^6 - 1 = 63 = 7 \times 3^2$, and the two primes 3 and 7 appear previously for $n = 2$ and $n = 3$, respectively, and hence we do not get a primitive prime divisor. However, as we are only considering even exponents, the above exception is no longer valid, and 7 becomes a "primitive" prime divisor, and our lemma can be applied without exceptions. ∎

From Lemma 2, we know there exists a prime p that divides $2^{2k} - 1$ but no smaller term of that form. Such a prime must divide either $2^k + 1$ or $2^k - 1$. We now show that there exists a Clifford operator in Cl_k that has order $2^k + 1$ and $2^k - 1$.

Lemma 3. [Existence Lemma]: There exists a $V, W \in \mathrm{Cl}_k$ such that $\mathrm{ord(V)} = 2^{\mathrm{k}} - 1$ and $\mathrm{ord(W)} = 2^{\mathrm{k}} + 1$.

Proof. As shown in Sec. II, each gate in the projective Clifford group Cl_k can be written as a matrix in the classical symplectic group $\mathrm{Sp}(2k, 2)$. Hence, it is enough to show that there exists a classical matrix, M of dimension $2k \times 2k$ which satisfies that symplectic condition, $M^T J M = J$, and has the required order.

(a) We first show how to construct an operator $V \in \mathrm{Sp}(2k, 2)$ such that $\mathrm{ord(V)} = 2^{\mathrm{k}} - 1$.

(1) Select a primitive polynomial

$$p(x) = x^k + a_{k-1}x^{k-1} + \cdots + a_1 x + a_0 \in \mathbb{F}_2[x],$$

of degree k and construct the field extension $\mathbb{F}_{2^k}$

$$\mathbb{F}_{2^k} \cong \mathbb{F}_2[x]/(p(x)). \tag{23}$$

Let α denote the residue class of x. Since $p(x)$ is primitive, α has multiplicative order $2^k - 1$.

(2) Construct the $k \times k$ dimensional companion matrix of $p(x)$ given by

$$C_p = \begin{bmatrix} 0 & 0 & \cdots & 0 & a_0 \\ 1 & 0 & \cdots & 0 & a_1 \\ 0 & 1 & \cdots & 0 & a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & a_{k-1} \end{bmatrix}. \tag{24}$$

With respect to the basis $\{1, \alpha, \ldots, \alpha^{k-1}\}$, this matrix represents multiplication by α on $\mathbb{F}_{2^k}$. Because α has multiplicative order $2^k - 1$, the matrix C_p also has multiplicative order $2^k - 1$.

(3) Using C_p we construct a symplectic matrix $\in \mathrm{Sp}(2k, 2)$

$$V = \begin{bmatrix} C_p & \mathbf{0}_{k\times k} \\ \mathbf{0}_{k\times k} & (C_p^{-1})^T \end{bmatrix}. \tag{25}$$

The matrix V always satisfies the symplectic condition, $V^T J V = J$ and moreover

$$V^r = \begin{bmatrix} C_p^r & \mathbf{0}_{k\times k} \\ \mathbf{0}_{k\times k} & (C_p^{-r})^T \end{bmatrix}. \tag{26}$$

Hence, V has the same order as C_p, that is $\mathrm{ord(V)} = 2^{\mathrm{k}} - 1$.

(b) We then show how to construct an operator $W \in \mathrm{Sp}(2k, 2)$ such that $\mathrm{ord(W)} = 2^{\mathrm{k}} + 1$.

(1) Select a monic primitive polynomial $p(x)$ of degree $2k$ and construct

$$\mathbb{F}_{2^{2k}} \cong \mathbb{F}_2[x]/(p(x)). \tag{27}$$

On $\mathbb{F}_{2^{2k}}$ we use the standard bilinear form (used in Equation 8.41 in Ref. [48])

$$B(a, b) = \mathrm{Tr}_{\mathbb{F}_{2^{2k}}/\mathbb{F}_2}(\theta \cdot (ab^{2^k} - a^{2^k} b)), \tag{28}$$

which is known to be alternating and can be made non-degenerate by choosing a $\theta \in \mathbb{F}_{2^{2k}}$. Therefore there exists a basis of $\mathbb{F}_{2^{2k}}$ over $\mathbb{F}_2$ in which

$$B(a, b) = a^T J b, \tag{29}$$

where $J = \begin{bmatrix} 0 & \mathrm{Id}_k \\ \mathrm{Id}_k & 0 \end{bmatrix}$.

(2) Let α be the residue class of x in $\mathbb{F}_{2^{2k}}$. Since $p(x)$ is primitive, α has multiplicative order $2^{2k} - 1$. Define

$$t = \alpha^{2^k - 1}. \tag{30}$$

The element t has multiplicative order $2^k + 1$. Let W be the linear transformation on $\mathbb{F}_{2^{2k}}$ defined by

$$Wa = t \cdot a. \tag{31}$$

Thus

$$\mathrm{ord}(W) = 2^k + 1. \tag{32}$$

(3) The matrix W preserves the symplectic form B that is

$$\begin{aligned} B(Wa,Wb) &= \mathrm{Tr}_{\mathbb{F}_{2^{2k}}/\mathbb{F}_2}(\theta\cdot(ta\cdot(tb)^{2^k}-(ta)^{2^k}\cdot tb)) \\ &= \mathrm{Tr}_{\mathbb{F}_{2^{2k}}/\mathbb{F}_2}(\theta\cdot t^{2^k+1}(ab^{2^k}-a^{2^k}b)) \\ &= \mathrm{Tr}_{\mathbb{F}_{2^{2k}}/\mathbb{F}_2}(\theta\cdot(ab^{2^k}-a^{2^k}b)) \\ &= B(a,b) \quad \forall\ a,b. \end{aligned}$$

Hence,

$$a^TJb = (Wa)^TJ(Wb) = a^T(W^TJW)b \quad \forall\ a,b. \tag{33}$$

Hence, $W^TJW = J$. We construct an explicit example to illustrate Lemma 3 in Appendix A. ∎

Since we know that p divides either 2^k+1 or 2^k-1 then by Lemma 3, either $U = V^{\frac{2^k-1}{p}}$ or $U = W^{\frac{2^k+1}{p}}$ will be a valid Clifford operator $\in \mathrm{Cl}_k$ such that $\mathrm{ord}(U) = p$. We now show that such an operator cannot exist in Cl_{k-1}.

Lemma 4. [Uniqueness Lemma]: The order of any $U \in \mathrm{Proj}\,\mathrm{Cl}_k$ must divide

$$2^{k^2}\prod_{i=0}^{k}(2^{2i}-1). \tag{34}$$

Proof. Weyl showed in Ref. [49] that the order of the symplectic group over the binary field is

$$|\mathrm{Sp}(2k,2)| = 2^{k^2}\prod_{i=0}^{k}(2^{2i}-1). \tag{35}$$

As shown in Sec. II, the projective Clifford group is isomorphic to the classical symplectic group and thus has the same size. Let $U \in \mathrm{Cl}_k$ have order r. The cyclic subgroup it generates,

$$\langle U\rangle = \{I, U, U^2, \cdots, U^{r-1}\}, \tag{36}$$

has size r. By Lagrange's theorem, the order of any subgroup of a finite group divides the order of the group itself. Thus the order of any element of the Clifford group must divide $2^{k^2}\prod_{i=0}^{k}(2^{2i}-1)$. ∎

Since Lemma 2 guarantees that the prime p does not divide $(2^{2i}-1)$ for any $i<k$ using Lemma 4, we show that an operator having an order p can never exist in $\mathrm{Proj}\,\mathrm{Cl}_{k-1}$. Since appending a Pauli can only multiply the order of an operator by a power of 2, there cannot exist an operator of order p even in Cl_{k-1} where p is the primitive prime divisor.

The above lemmas combine to show that there exists a operator in $U \in \mathrm{Cl}_k$ which has an order p where p is the primitive prime divisor of $2^{2k}-1$ and such an element cannot exist in Cl_{k-1}. Hence, if we want to implement $\bar{U}$ on the logical qubits of a $[[n,k,d]]$ stabilizer code, by Lemma 1 we need to implement a physical gadget having the same order p which must belong in $\mathrm{Cl}_k\backslash\mathrm{Cl}_{k-1}$ and must be implemented using a k-fold transversal gadget. Hence, Theorem 1 is proved. ∎

B. Proof of theorem 2

We now prove Theorem 2 stated in Sec. III, which constrains gates implemented by code automorphisms, including any combination of single-qubit Clifford gates and permutations of the qubits. We show that in particular there is a logical gate, the Bell gate in Cl_2, which cannot be implemented using code automorphisms, and hence, there does not exist any stabilizer code such that the full Clifford group on multiple logical qubits can be implemented using code automorphisms.

The Bell gate is a Clifford gate that yields a matrix transformation that conjugates any real matrix in $\mathrm{SU}(4)$ to a tensor product of two single-qubit gates [50–52]. In other words, Bell gate conjugates the subgroup $\mathrm{SO}(4)$ to the subgroup $\mathrm{SU}(2)\otimes\mathrm{SU}(2)$. The matrix transformation is given by

$$\mathrm{BELL} = \frac{e^{\frac{3\pi i}{4}}}{\sqrt{2}}\begin{bmatrix} 1 & i & 0 & 0 \\ 0 & 0 & i & 1 \\ 0 & 0 & i & -1 \\ 1 & -i & 0 & 0 \end{bmatrix}. \tag{37}$$

The projective Bell gate or the equivalent classical symplectic matrix of the Bell gate is given by

$$M_{\mathrm{BELL}} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}. \tag{38}$$

The Bell gate belongs to $\mathrm{Cl}_k\ \forall\ k \geq 2$ and thus any code which has the full logical Clifford group on multiple qubits must have the logical Bell gate as well. Since the Bell gate is a Clifford gate, its conjugation action on the Pauli basis is given by

$$\begin{aligned} X\otimes I &\Rightarrow Y\otimes Z \qquad Z\otimes I \Rightarrow Z\otimes Z \\ I\otimes X &\Rightarrow X\otimes Y \qquad I\otimes Z \Rightarrow X\otimes X. \end{aligned} \tag{39}$$

The Bell gate (and any power of the Bell gate not divisible by 5) has two important properties which would be used during the proof.

(1) The projective Bell gate is an order 5 gate, i.e., $\mathrm{BELL}^5 = \mathrm{Id}_2$.
(2) The Bell gate cannot preserve a maximal abelian subgroup of $\mathcal{P}_2$ i.e. for every maximal abelian subgroup $G \subseteq \mathcal{P}_2$ there exists $g \in G$ such that $\{\mathrm{BELL}^\dagger\, g\, \mathrm{BELL}, g\} = 0$.

Let us now assume that there exists a stabilizer code for which the logical Bell gate can be implemented using a code automorphism, and let us call this automorphism U'. According to Lemma 1, U' must have an order which is a multiple of 5. Then without loss of generality let us assume

$$\mathrm{ord}(U') = 5^m \times q \tag{40}$$

where $m, q \in \mathbb{Z}$ and q is not divisible by 5. Since the code automorphisms form a group, $U = (U')^q$ will also be a code automorphism which will have exactly order 5^m and will implement the logical gate $\overline{\mathrm{BELL}}^q$. Note that the logical gate still satisfies both the properties listed above since q is not divisible by 5.

We now show that any automorphism that has an order 5^m must be $\mathbf{5^m}$ local automorphism. A 5^m local automorphism is defined as a permutation of physical qubits that contain only cycles of order 5^m followed by a transversal gadget that acts trivially on the non-permuted qubits (but could be non-trivial on the permuted qubits).

Lemma 5. [Locality Lemma]. Any code automorphism that has an order which a power of a prime, p^m with $p > 3$, must be a p^m local automorphism.

Proof. Code automorphism consists of a permutation of the physical qubits followed by a transversal gadget. Since transversal gadgets consist of single-qubit Cliffords acting individually on each physical qubit, they can only have order $= 1, 2, 3$ or 6. In order for the code automorphism to have an order p^m with $p > 3$, the permutation must have cycles of order p^m.

In addition, if the transversal gadget part of the automorphism acts non-trivially on a non-permuted qubit, then applying the automorphism p^m times would still result in a non-trivial action on that qubit, because the order of the single-qubit Clifford gate on that qubit does not divide p^m. The order of the automorphism would then be a multiple of p^m. Hence, to have an automorphism of order exactly p^m, it must act trivially on the non-permuted qubits. ∎

We have now shown that any stabilizer code that has the logical BELL gate implemented using code automorphisms must have a 5^m local automorphism that implements the logical BELL^q gate for some integers m, q where q is not divisible by 5.

Lemma 6. [Equivalence Lemma]. For any p^m local automorphism U there exists a transversal gadget V such that $V^\dagger U V$ is just a qubit permutation.

Proof. Without loss of generality let

$$U = \left(\bigotimes_{1 \le i \le p^m} U_i\right) * (1, 2, \ldots, p^m), \tag{41}$$

where $(1, 2, \cdots, p^m)$ is a permutation that cyclically permutes the first p^m physical qubits. Since U has order p^m,

$$\prod_{i=1}^{p^m} U_i = \mathrm{Id}. \tag{42}$$

Now,

$$V^\dagger U V = \left(\bigotimes_{1 \le i \le p^m} V_i^\dagger\right)\left(\bigotimes_{1 \le i \le p^m} U_i * (1, 2, \ldots, p^m)\right)$$
$$\left(\bigotimes_{1 \le i \le p^m} V_i\right) = \left(\bigotimes_{1 \le i \le p^m} V_i^\dagger U_i V_{i+1 \bmod p^m}\right) * (1, 2, \ldots, p^m).$$

So we can choose $V_1 = \mathrm{Id}$ and $V_i = (\prod_{j<i} U_j)^\dagger$ such that

$$V^\dagger U V = \bigotimes_{1 \le i \le p^m} \mathrm{Id} * (1, 2, \ldots, p^m). \tag{43}$$

Hence, the lemma is proved. If the automorphism has multiple permutation cycles of size p^m then the above procedure can be carried out independently for each cycle of permuted qubits since they must be disjoint. ∎

Given Lemma 6, we can construct an equivalent stabilizer code to the one with the logical Bell gate by choosing its stabilizer group to be $V^\dagger \mathcal{S} V$ where $\mathcal{S}$ is the stabilizer group of the original code. Then for the equivalent code,

$$(V^\dagger U V)^\dagger (V^\dagger S V)(V^\dagger U V) \in V^\dagger \mathcal{S} V \quad \forall\ S \in \mathcal{S}. \tag{44}$$

Thus, $V^\dagger U V$ is an automorphism for the new code. Moreover, the logical gate implemented by $V^\dagger U V$ for the new code must satisfy both the properties of the Bell gate. In addition, $V^\dagger U V$ is just a permutation of physical qubits. In the next Lemma we show that a physical gadget, which is just a permutation, can never satisfy the second property.

Lemma 7. [Permutation Lemma]. Any permutation of the physical qubits must implement a logical gate that preserves the abelian subgroup $\{\overline{Z_i}\}$.

Proof. It was shown in Ref. [53] [Chapter 4, Equation 4.3] that all stabilizer codes in their binary symplectic form can be reduced to a standard form by performing Gaussian elimination.

$$\begin{array}{r} \\ r\{ \\ n-k-r\{ \end{array} \left(\begin{array}{ccc|ccc} \overbrace{I}^{r} & \overbrace{A_1}^{n-k-r} & \overbrace{A_2}^{k} & \overbrace{B}^{r} & \overbrace{C_1}^{n-k-r} & \overbrace{C_2}^{k} \\ 0 & 0 & 0 & D & I & E \end{array} \right).$$

For a code in the standard form we can choose a set of logical Pauli representatives,

$$L = \begin{bmatrix} L_x \\ L_z \end{bmatrix} = \left[\begin{array}{ccc|ccc} 0 & E^T & I & E^T C_1^T + C_2^T & 0 & 0 \\ 0 & 0 & 0 & A_2^T & 0 & I \end{array}\right]. \quad (45)$$

The important thing to note is that in the standard form, the logical Z operators are completely made up of physical Z terms.

Now let us consider a permutation of physical qubits U. Since $\bar{Z}_i \ \forall \ i$ is completely made up of physical Z terms, $U\overline{Z}_i U^\dagger$ must also be completely made up of physical Z terms. As a result $[\overline{Z}_i, U\overline{Z}_i U^\dagger] = 0 \ \forall \ i$. ∎

Corollary. There exists no stabilizer code that admits the full Clifford group on the logical qubits using qubit permutations. This is true even for codes with a single logical qubit, whereas for code automorphisms the result only holds for codes with multiple logical qubits.

Hence, according to Lemma 7 all permutation gadgets preserve a maximal abelian subgroup ($\{\overline{Z}_i\}$ given one particular choice of logical operators). Since the Bell gate can never preserve a maximal abelian subgroup, permutation gadgets can never implement the logical Bell gate. Following the train of thought backward, we prove that no stabilizer code can have a code automorphism which implements the logical Bell gate, therefore the full logical Clifford group on multiple qubits. Hence, the lemma is proved. ∎

V. DISCUSSION

In this paper we present and prove two theorems that impose constraints on existing fault tolerant Clifford architectures such as fold-transversal gadgets and code automorphisms. First, we show that no stabilizer code has a transversal Clifford implementation of the full Clifford group on multiple logical qubits or a fold-transversal Clifford implementation of the full Clifford group on more than two logical qubits. We also study code automorphisms, which are physical gadgets made up of transversal Clifford gates followed by a permutation of physical qubits, and show that there exists no stabilizer code that admits the full Clifford group on multiple logical qubits implemented using code automorphisms. We then introduce the notion of k-fold transversal gadgets and show that, in order to implement the full Clifford group on k logical qubits, we need at least k-fold transversal gadgets at the physical level.

A k-fold transversal gadget can, in the worst case, propagate a single-qubit error onto k physical qubits. Stabilizer codes encoding multiple logical qubits within a single code block are desirable not only to reduce qubit overheads but also because they provide advantages such as more flexible code-switching and gauge-fixing frameworks. It is also desirable to perform the full Clifford group with a simple implementation, such as a k-fold transversal implementation for small k. However, here we show that finding stabilizer codes with both of these desired properties is not possible within the fault tolerant frameworks considered in this paper. Thus, to implement the full Clifford group on multiple logical qubits, we need more complex fault tolerant constructions.

An important feature of our result is that each class of gadget constructions we consider forms a group, in the sense that compositions of such gadgets do not lead to increased fault propagation. Consequently, any subgroup of logical operations realized via code automorphisms or k-fold transversal gadgets can be implemented in a depth that depends on k but is constant in the number of physical qubits n. In contrast to our result, adding power beyond k-transversal gates can result in more powerful logical gate sets. However, doing so requires non-constant circuit depth and additional fault tolerant circuitry (such as extra error-correction steps, which typically require significant space-time volume). Two recent works provide examples:

(1) In Ref. [54], the author constructs families of codes with an arbitrary number of logical qubits for which the generators of the logical Clifford group can be implemented using fold-transversal gadgets. However, these constructions rely on fold-transversal operations defined with respect to different partitions. As a result, the corresponding physical gadgets do not form a group, and intermediate error-correction steps are required when composing them. This leads to implementations of logical Clifford operations that are not constant depth. Thus, this approach circumvents the no-go theorem by allowing increased circuit depth.

(2) In Ref. [55], the authors present an efficient scheme for implementing the full logical Clifford group in quantum LDPC codes. Their approach uses subsystem codes and syndrome-extraction measurements to realize certain Clifford operations. These ingredients fall outside the framework considered here, but they suggest alternative avenues for circumventing the no-go theorem.

Moreover, the theorems established in this work also have important implications for the addressability of stabilizer codes encoding multiple logical qubits. In particular, since addressable logical H, S, and CNOT gates acting on all logical qubits generate the full logical Clifford group, our results imply that no stabilizer code can admit a fully addressable implementation of these gates on all logical qubits using only Clifford transversal constructions or code automorphisms.

A. Future directions

In this work, we have limited our attention only to physical gadgets which belong to the Clifford group in order to perform logical actions belonging to the Clifford group. It remains an interesting open question whether non-Clifford transversal or fold transversal gates at the physical level could be used as part of a construction of the full logical Clifford group. In addition, we could also investigate the applicability of these theorems to qudit stabilizer codes. As qudit fault tolerance has been explored less than qubit fault tolerance, establishing the existence (or absence) of analogous no-go constraints would be an important contribution to the field. Another possible direction could be to search for stabilizer codes which have certain geometrical advantages that allow for a favorable implementation of k-fold transversal gadgets. In such codes, even though k-fold transversal gadgets spread errors onto k physical qubits, they only result in high weight errors that correspond to unique syndromes. These codes are then able to detect and correct such errors resulting from the implementation of the k-fold transversal gadget, despite having a smaller distance. A possible example could be to consider k blocks of the $[[7,1,3]]$ code as a single block encoding k logical qubits.

ACKNOWLEDGMENTS

A. C would like to thank Xiaozhen Fu, Victor V. Albert, Jin Ming Koh, Shayan Majidy, Alexander Frei and Zachary Mann for their helpful discussions. D. G is partially supported by the National Science Foundation (RQS QLCI Grant No. OMA-2120757).

DATA AVAILABILITY

There are no publicly available research data or software supporting this manuscript. Requests for further information or data should be sent to the authors.

APPENDIX

Here we illustrate Lemma 3 by explicitly constructing $V, W \in \mathrm{Sp}(4,2)$ which correspond to Clifford operators $\in \mathrm{Cl}_2$ acting on two qubits. According to the lemma, $\mathrm{ord}(V) = 2^k - 1 = 3$ and $\mathrm{ord}(W) = 2^k + 1 = 5$.

(a) We first choose a primitive polynomial of degree 2,

$$p(x) = x^2 + x + 1. \tag{A1}$$

We then construct the field extension $\mathbb{F}_4 \cong \mathbb{F}_2[x]/(p(x))$. Then the companion matrix of $p(x)$ is given by

$$C_p = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}. \tag{A2}$$

It can be easily verified that $C_p^3 = \mathrm{Id}$ hence $\mathrm{ord}(C_p) = 3$. We then construct $V \in \mathrm{Sp}(4,2)$ such that

$$V = \begin{bmatrix} C_p & \mathbf{0}_{k\times k} \\ \mathbf{0}_{k\times k} & (C_p^{-1})^T \end{bmatrix} = \left[\begin{array}{cc|cc} 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 \end{array}\right] \tag{A3}$$

It can be verified that $\mathrm{ord}(V) = \mathrm{ord}(C_p) = 3$, and it satisfies the symplectic condition i.e. $V^T J V = J$. An interesting thing to note is that in this construction V acts on the X and Z Pauli subspaces independently, which means it can be realized by a CNOT circuit.

(b) We select the primitive polynomial

$$p(x) = x^4 + x + 1, \tag{A4}$$

and construct $\mathbb{F}_{16} = \mathbb{F}_2[x]/(p(x))$. Let α be the residue class of x and since $p(x)$ is a primitive polynomial each element of $\mathbb{F}_{16}$ can be written $\{0, \alpha^k\}$ for $1 \le k \le 15$.

In order to make the bilinear form non-degenerate we define the kernel of the bilinear form as

$$\mathrm{Ker}(B) = \{a \in \mathbb{F}_{16} | B(a,b) = 0 \quad \forall\ b \in \mathbb{F}_{16}\}. \tag{A5}$$

In order for the standard form to be non-degenerate its kernel must contain only the zero element i.e. $\mathrm{Ker}(B) = \{0\}$. Simplifying, we can show that in order for the form to be non-degenerate $(\theta + \theta^4) \neq 0$ which implies $\theta \notin \{0, 1, \alpha^5, \alpha^{10}\}$. Hence, we choose $\theta = \alpha$ and the non-degenerate standard bilinear form is given by

$$B(a,b) = \mathrm{Tr}_{\mathbb{F}_{16}/\mathbb{F}_2}(\alpha \cdot (ab^4 - a^4 b)), \tag{A6}$$

We now choose a symplectic basis for $\mathbb{F}_{16}$ such that $B(a,b) = a^T J b$. Let the symplectic basis be $\{u_1, u_2, v_1, v_2\}$ which must satisfy the following conditions:

$$\begin{aligned} &B(u_i, u_j) = 0 \qquad B(v_i, v_j) = 0 \\ &B(u_i, v_j) = \delta_{ij} \quad \forall\ i,j \end{aligned} \tag{A7}$$

One such symplectic basis can be chosen as $\{1, \alpha^5, \alpha^3, \alpha^4\}$. Then, W is the linear transformation generated by multiplication by α^3.

$$\begin{aligned} W(1) &= \alpha^3 \\ W(\alpha^5) &= \alpha^8 = \alpha^5 + \alpha^4 \\ W(\alpha^3) &= \alpha^6 = 1 + \alpha^5 + \alpha^3 + \alpha^4 \\ W(\alpha^4) &= \alpha^7 = \alpha^3 + \alpha^4 \end{aligned} \tag{A8}$$

Hence,

$$W = \left[\begin{array}{cc|cc} 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \end{array}\right] \tag{A9}$$

It can be verified that $\mathrm{ord}(W) = 5$, and it satisfies the symplectic condition $W^T J W = J$. ∎

[1] S. Majidy, C. Wilson, and R. Laflamme, *Building Quantum Computers: A Practical Introduction* (Cambridge University Press, Cambridge, England, 2024).

[2] D. Gottesman, *Theory of fault-tolerant quantum computation*, Phys. Rev. A **57**, 127 (1998).

[3] D. Aharonov and M. Ben-Or, *Fault-tolerant quantum computation with constant error*, in *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997). pp. 176–188.

[4] E. Knill, R. Laflamme, and W. H. Zurek, *Resilient quantum computation*, Science **279**, 342 (1998).

[5] B. W. Reichardt, D. Aasen, R. Chao, A. Chernoguzov, W. van Dam, J. P. Gaebler, D. Gresh, D. Lucchetti, M. Mills, S. A. Moses *et al.*, *Demonstration of quantum computation and error correction with a tesseract code*, arXiv:2409.04628.

[6] M. D. Reed, L. DiCarlo, S. E. Nigg, L. Sun, L. Frunzio, S. M. Girvin, and R. J. Schoelkopf, *Realization of three-qubit quantum error correction with superconducting circuits*, Nature (London) **482**, 382 (2012).

[7] A. Paetznick, M. da Silva, C. Ryan-Anderson, J. Bello-Rivas, J. Campora III, A. Chernoguzov, J. Dreiling, C. Foltz, F. Frachon, J. Gaebler *et al.*, *Demonstration of logical qubits and repeated error correction with better-than-physical error rates*, arXiv:2404.02280.

[8] R. Acharya *et al.*, *Quantum error correction below the surface code threshold*, Nature (London) **638**, 920 (2025).

[9] D. Gottesman, *The Heisenberg representation of quantum computers*, arXiv:quant-ph/9807006.

[10] V. Kliuchnikov, D. Maslov, and M. Mosca, *Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and T gates*, arXiv:1206.5236.

[11] C. M. Dawson and M. A. Nielsen, *The Solovay-Kitaev algorithm*, arXiv:quant-ph/0505030.

[12] D. Aharonov, *A simple proof that Toffoli and Hadamard are quantum universal*, arXiv:quant-ph/0301040.

[13] B. Eastin and E. Knill, *Restrictions on transversal encoded quantum gate sets*, Phys. Rev. Lett. **102**, 110502 (2009).

[14] B. Zeng, A. Cross, and I. L. Chuang, *Transversality versus universality for additive quantum codes*, IEEE Trans. Inf. Theory **57**, 6272 (2011).

[15] E. Knill, *Fault-tolerant postselected quantum computation: Schemes*, arXiv:quant-ph/0402171.

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

[17] A. Paetznick and B. W. Reichardt, *Universal fault-tolerant quantum computation with only transversal gates and error correction*, arXiv:1304.3709.

[18] A. M. Steane, *Error correcting codes in quantum theory*, Phys. Rev. Lett. **77**, 793 (1996).

[19] J. Guyot and S. Jaques, *On the addressability problem on css codes*, arXiv:2502.13889.

[20] Z. He, V. Vaikuntanathan, A. Wills, and R. Y. Zhang, *Quantum codes with addressable and transversal non-Clifford gates*, arXiv:2502.01864.

[21] H. Bombín, *Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes*, New J. Phys. **17**, 083002 (2015).

[22] S. Bravyi and J. Haah, *Magic-state distillation with low overhead*, Phys. Rev. A **86**, 052329 (2012).

[23] S. Bravyi and R. König, *Classification of stabilizer operations over qubits*, Phys. Rev. Lett. **110**, 170503 (2013).

[24] A. R. Calderbank, E. M. Rains, P. M. Shor, and N. J. Sloane, *Quantum error correction via codes over gf (4)*, IEEE Trans. Inf. Theory **44**, 1369 (1998).

[25] H. Sayginel, S. Koutsioumpas, M. Webster, A. Rajput, and D. E. Browne, *Fault-tolerant logical Clifford gates from code automorphisms*, arXiv:2409.18175.

[26] N. Berthusen, M. J. Gullans, Y. Hong, M. Mudassar, and S. J. S. Tan, *Automorphism gadgets in homological product codes*, arXiv:2508.04794.

[27] J. M. Koh, A. Gong, A. C. Diaconu, D. B. Tan, A. A. Geim, M. J. Gullans, N. Y. Yao, M. D. Lukin, and S. Majidy, *Entangling logical qubits without physical operations*, arXiv:2601.20927.

[28] K. Sahay, P.-K. Tsai, K. Chang, Q. Su, T. B. Smith, S. Singh, and S. Puri, *Fold-transversal surface code cultivation*, arXiv:2509.05212.

[29] N. P. Breuckmann and S. Burton, *Fold-transversal Clifford gates for quantum codes*, Quantum **8**, 1372 (2024).

[30] J. N. Eberhardt and V. Steffan, *Logical operators and fold-transversal gates of bivariate bicycle codes*, IEEE Trans. Inf. Theory **71**, 1140 (2024).

[31] J. E. Moussa, *Transversal Clifford gates on folded surface codes*, Phys. Rev. A **94**, 042316 (2016).

[32] A. O. Quintavalle, P. Webster, and M. Vasmer, *Partitioning qubits in hypergraph product codes to implement logical gates*, Quantum **7**, 1153 (2023).

[33] F. Butt, S. Heußen, M. Rispler, and M. Müller, *Fault-tolerant code-switching protocols for near-term quantum processors*, PRX Quantum **5**, 020345 (2024).

[34] L. Daguerre, R. Blume-Kohout, N. C. Brown, D. Hayes, and I. H. Kim, *Experimental demonstration of high-fidelity logical magic states from code switching*, Phys. Rev. X **15**, 041008 (2025).
[35] S. J. S. Tan, Y. Hong, T.-C. Lin, M. J. Gullans, and M.-H. Hsieh, *Single-shot universality in quantum IDPC codes via code-switching*, arXiv:2510.08552.
[36] R. Chao and B. W. Reichardt, *Quantum error correction with only two extra qubits*, Phys. Rev. Lett. **121**, 050502 (2018).
[37] B. Anker and M. Marvian, *Universal fault tolerance with non-transversal clifford gates*, arXiv:2510.08402.
[38] N. Berthusen, S. J. S. Tan, E. Huang, and D. Gottesman, *Adaptive syndrome extraction*, PRX Quantum **6**, 030307 (2025).
[39] D. Horsman, A. G. Fowler, S. Devitt, and R. Van Meter, *Surface code quantum computing by lattice surgery*, New J. Phys. **14**, 123011 (2012).
[40] A. G. Fowler and C. Gidney, *Low overhead quantum computation using lattice surgery*, arXiv:1808.06709.
[41] N. Berthusen and E. Durso-Sabina, *Simple logical quantum computation with concatenated symplectic double codes*, arXiv:2510.18753.
[42] Y. Xiao, Y.-H. Kang, R.-H. Zheng, J. Song, Y.-H. Chen, and Y. Xia, *Effective nonadiabatic holonomic swap gate with Rydberg atoms using invariant-based reverse engineering*, Phys. Rev. A **109**, 062610 (2024).
[43] D. Kielpinski, C. Monroe, and D. J. Wineland, *Architecture for a large-scale ion-trap quantum computer*, Nature (London) **417**, 709 (2002).
[44] T. Jochym-O'Connor, A. Kubica, and T. J. Yoder, *Disjointness of stabilizer codes and limitations on fault-tolerant logical gates*, Phys. Rev. X **8**, 021047 (2018).
[45] K. Zsigmondy, *Zur theorie der potenzreste*, Monatsh. Math. Phys. **3**, 265 (1892).
[46] B. Michels, *Zsigmondy's theorem* (2014).
[47] A. Bang, *Taltheoretiske undersøgelser*, Tidsskr. Math. **4**, 70 (1886).
[48] D. Gottesman, *Surviving as a quantum computer in a classical world—draft* (2024).
[49] H. Weyl, *The Classical Groups: Their Invariants and Representations* (Princeton university press, Princeton, NJ, 1946), Vol. 1.
[50] S. Hill and W. K. Wootters, *Entanglement of a pair of quantum bits*, arXiv:quant-ph/9703041.
[51] F. Vatan and C. Williams, *Optimal quantum circuits for general two-qubit gates*, Phys. Rev. A **69**, 032315 (2004).
[52] E. Kubischta and I. Teixeira, *Classification of the subgroups of the two-qubit clifford group*, J. Math. Phys. **66**, 122203 (2025).
[53] D. Gottesman, *Stabilizer Codes and Quantum Error Correction* (California Institute of Technology, Berkeley, 1997).
[54] T. Tansuwannont, T. Chan, and R. Takagi, *Construction of the full logical clifford group for high-rate quantum reed-muller codes using only transversal and fold-transversal gates*, arXiv:2602.09788.
[55] A. J. Malcolm, A. N. Glaudell, P. Fuentes, D. Chandra, A. Schotte, C. DeLisle, R. Haenel, A. Ebrahimi, J. Roffe, A. O. Quintavalle *et al.*, *Computing efficiently in QLDPC codes*, arXiv:2502.07150.