This is the accepted manuscript made available via CHORUS. The article has been published as:

Fault-tolerant logical gates in quantum error-correcting codes

Fernando Pastawski and Beni Yoshida

Phys. Rev. A **91**, 012305 — Published 8 January 2015

DOI: 10.1103/PhysRevA.91.012305

Fault-tolerant logical gates in quantum error-correcting codes

Fernando Pastawski[1] and Beni Yoshida[1]

[1]Institute for Quantum Information and Matter,
California Institute of Technology, Pasadena, California 91125, USA

Recently, Bravyi and König have shown that there is a trade-off between fault-tolerantly implementable logical gates and geometric locality of stabilizer codes. They consider locality-preserving operations which are implemented by a constant-depth geometrically-local circuit and are thus fault-tolerant by construction. In particular, they show that, for local stabilizer codes in D spatial dimensions, locality-preserving gates are restricted to a set of unitary gates known as the D-th level of the Clifford hierarchy. In this paper, we explore this idea further by providing several extensions and applications of their characterization to qubit stabilizer and subsystem codes.

First, we present a new no-go theorem for self-correcting quantum memory. Namely, we prove that a three-dimensional stabilizer Hamiltonian with a locality-preserving implementation of a non-Clifford gate cannot have a macroscopic energy barrier. This result implies that non-Clifford gates do not admit such implementations in Haah's cubic code and Michnicki's welded code.

Second, we prove that the code distance of a D-dimensional local stabilizer code with non-trivial locality-preserving m-th level Clifford logical gate is upper bounded by $O(L^{D+1-m})$. For codes with non-Clifford gates ($m > 2$), this improves the previous best bound by Bravyi and Terhal. Topological color codes, introduced by Bombin and Martin-delgado, saturate the bound for $m = D$.

Third, we prove that the qubit erasure threshold for codes with non-trivial transversal m-th level Clifford logical gate is upper bounded by $1/m$. This implies that no family of fault-tolerant codes with transversal gates in increasing level of the Clifford hierarchy may exist. This result applies to arbitrary stabilizer and subsystem codes, and is not restricted to geometrically-local codes.

Fourth, we extend the result of Bravyi and König to subsystem codes. Unlike stabilizer codes, the so-called union lemma does not apply to subsystem codes. This problem is avoided by assuming the presence of error threshold in a subsystem code, and a conclusion analogous to that of Bravyi-König is recovered.

I. INTRODUCTION

Quantum error-correcting codes constitute an indispensable ingredient in the roadmap to fault-tolerant quantum computation. They provide a framework enabling imperfect quantum gates and resources to implement arbitrarily reliable quantum computation [1, 2]. An essential feature for such codes is to admit a fault-tolerant implementation of a universal gate set with the property that physical errors propagate in a benign and controlled manner. A paragon for fault-tolerant implementation of logical gates is provided by transversal unitary operations, *i.e.* single qubit rotations acting independently on each physical qubit.

However, Eastin and Knill have proved that the set of transversal gates constitutes a finite group, and hence is not universal for quantum computation [3]. This suggests a trade-off between computational power and fault-tolerance. Recently, Bravyi and König have further sharpened this tension for topological stabilizer codes supported on a lattice with geometrically-local generators [4]. In their article, logical gates implementable by constant-depth local circuits (i.e. circuits composed a constant number of layers of neighboring qubit gates) in D spatial dimensions are found to be restricted to a set of unitary gates, known as the D-th level of the Clifford hierarchy [5]. This result establishes a connection between two seemingly unrelated notions: fault-tolerance and geometric locality. In this article, we address some of the open questions arising from the work of Bravyi and König and obtain extensions of their result.

A. Clifford hierarchy

As in BK [4], the group consisting of tensor product Pauli operators on n qubits (denoted by $\mathsf{Pauli} = \langle X_j, Y_j, Z_j\rangle_{j\in[1,n]}$) and the corresponding Clifford hierarchy [5] will play a central role. We provide a formal definition of the m-th level of the Clifford hierarchy $\mathcal{P}_m$.

Definition 1. *We define the **Clifford hierarchy** as $\mathcal{P}_0 \equiv \mathbb{C}$ (i.e. global complex phases), and then recursively as*

$$\mathcal{P}_{m+1} = \{\text{unitary } U : \forall P \in \mathsf{Pauli},\ UPU^\dagger P^\dagger \in \mathcal{P}_m\}. \quad (1)$$

The above definition coincides with the standard one for $m \geq 2$ [4, 5]. (See appendix A for comparison). $\mathcal{P}_1$ is the group of Pauli operators with global complex phases. $\mathcal{P}_2$ coincides with the *Clifford group* and includes the Hadamard gate, the $\pi/2$-phase gate and the CNOT gate. The $\mathcal{P}_3$ includes some non-Clifford gates such as the $\pi/4$-phase gate and the Toffoli gate. Similarly, the $\pi/2^m$-phase gate is in $\mathcal{P}_{m+1}$ but not to $\mathcal{P}_{m+1}$. Note that for $m \geq 3$, $\mathcal{P}_m$ is a set and is not a group since it is not closed under multiplication.

In principle, gates in the Clifford group can be implemented with arbitrary precision by using concatenated stabilizer codes [7] or topological codes. Realistic systems also offer decoherence-free implementation of some

Clifford gates. For instance, braiding of Ising anyons (believed to exist as excitations of the fractional quantum Hall state at filling fraction $\nu = 5/2$) implements certain Clifford gates with an estimated error-rate of 10^{-30} [8]. However, the Gottesman-Knill theorem assures that any quantum circuit composed exclusively from Clifford gates in $\mathcal{P}_2$, together with preparation and measurement of in the computational basis, can be simulated efficiently by a classical computer[6]. In contrast, incorporating any additional non-Clifford gate to $\mathcal{P}_2$ results in a universal gate set for quantum computation. For this reason, it is important to fault-tolerantly perform *non-Clifford* logical gates outside of $\mathcal{P}_2$.

B. Summary of results

Let us now summarize the main contributions of this work. We begin by providing a self-contained and arguably simpler derivation of BK's result. We then derive a key technical lemma to assess fault-tolerant implementability of logical gates for both stabilizer and subsystem error-correcting codes (lemma 4 in section II). In addition, there are four original results. Below, we provide an intuitive description of them, deferring a rigorous treatment to later sections.

1. No-go result for self-correction

First of all, we show that the property of self-correction imposes a further restriction on logical gates implemented by constant-depth local circuits. Namely, we find that the assumption of having no string-like logical operators reduces the accessible level of the Clifford hierarchy by one with respect to BK's result (see Theorem 2).

This leads to a new no-go result for self-correcting quantum memory in three spatial dimensions: a three-dimensional topological stabilizer Hamiltonian with a locality-preserving non-Clifford gate cannot have a macroscopic energy barrier. This result is presented in section V. It establishes a somewhat surprising connection between ground state properties and excitation energy landscape.

2. Upper bound on code distance

Our second, establishes a trade-off between the accessibility of logical gates from the m-th level of the Clifford hierarchy and the code distance $d$of topological stabilizer codes. Namely, assuming an L^D lattice in D spatial dimensions and a locality preserving gate in $\mathcal{P}_m$, we find that $d \leq O(L^{D+1-m})$ (see theorem 3). For a code with a non-Clifford gate ($m > 2$), this result improves the previous best bound $d \leq O(L^{D-1})$ for topological stabilizer codes [9]. The bound is found to be tight for $m = D$ as some topological color codes saturate it [10–13]. This result also applies to topological subsystem codes provided that the stabilizer subgroup admits a complete set of geometrically-local generators as in Bombin's topological *gauge* color code [13]. The proof is presented in section V.

3. Erasure threshold

Our third result relates the erasure threshold in stabilizer and subsystem error-correcting codes with the set of transversally implementable logical gates. Namely, if the erasure threshold p_l is larger than $1/n$, only gates in $\mathcal{P}_{n-1}$ might be transversely implemented (see theorem 1). We would like to emphasize that this result holds for all stabilizer and subsystem codes regardless of generator locality. The proof is presented in section III.

4. Subsystem code and the Clifford hierarchy

Finally, the main technical result is to generalize BK's result to subsystem codes with local generators. A difficulty is that the so-called union lemma [9] does not apply to topological subsystem codes [14, 15]. Minimal extra assumptions, such as a finite erasure threshold for the code and a logarithmically increasing code distance, are required in order to recover a statement analogous to the one obtained by BK for topological stabilizer codes. The strengthened assumptions for our theorem are automatically satisfied by fault-tolerant codes. Namely, a finite erasure threshold is necessary for a finite error threshold against depolarization. Furthermore, a code distance d increasing logarithmically with the number of phsysical qubits n is necessary for the recovery probability to remain polynomially close to unity under constant noise rate. The proof of this result is presented in section IV. In addition, we provide new algebraic definitions for dressed and bare logical operators in subsystem codes which include arbitrary logical operators and are not restricted to those implemented by tensor product Paulis operators. These definitions, are of independent interest, as they might be general enough to analyze codes beyond the Pauli stabilizer/subsystem formalism.

C. Organization of the paper

The paper is organized as follows. In section II, we provide a definition of subsystem codes and derive a key technical tool to study fault-tolerant implementability of logical gates. We then provide a derivation of BK's result. In section III, we derive a trade-off between the erasure threshold and transversal implementability of logical gates. In section IV, we generalize BK's result to topological subsystem codes. In section V, we connect the property of self-correction of a code with a strengthened restriction on the set of locality-preserving logical gates. We then derive an upper bound on the distance of topological stabilizer codes. Section VI is devoted to summary and discussion.

II. FAULT-TOLERANCE VERSUS LOCALITY

In this section, we review the framework of subsystem error-correcting codes and derive a tool relating fault-tolerant implementability of logical gates and locality (or non-locality) of logical gates with respect to a partition of the physical qubits. We also present a qualitative derivation of BK's result for topological stabilizer codes.

A. Fault-tolerant implementation of logical gates

Let us begin with a brief review of the stabilizer formalism [16]. Given the Hilbert space of n qubits $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, a Pauli stabilizer group $\mathcal{S}$ is an abelian subgroup of the Pauli group on n qubits. Moreover, $\mathcal{S}$ does not contain $-\mathbb{1}$. The codeword space of the stabilizer group $\mathcal{S}$ is defined to be the subspace $\mathcal{C}(\mathcal{S}) \subseteq \mathcal{H}$ of common $+1$ eigenvectors for all stabilizers in $\mathcal{S}$ (*i.e.* $\mathcal{C}(\mathcal{S}) = \{|\psi\rangle \in \mathcal{H} : \forall S \in \mathcal{S},\ S|\psi\rangle = |\psi\rangle\}$). In this article, *topological* stabilizer codes refers to codes presenting the following characteristics i) they are defined on a regular lattice of physical qubits with bounded density (number of qubits per lattice site) ii) the stabilizer group $\mathcal{S}$ admits geometrically local generators $\mathcal{S} = \langle S_1, \ldots, S_{n-k}\rangle$ (*i.e.* the support of each generator S_j is contained in a ball of radius ξ). When S_j are independent generators, k is the number of logical qubits encoded in the codespace $\mathcal{C}(\mathcal{S})$.

Ideally, one hopes for a logical gate U to be implemented by a transversal unitary, *i.e.* an operator with a tensor product form $U = \otimes_{j=1}^{n} U_j$, with U_j being single qubit rotations acting on j-th physical qubit. In this way, errors on physical qubits do not propagate to other qubits. Pauli logical gates in $\mathcal{P}_1$ are an example of gates admitting a transversal implementation for all stabilizer codes, furthermore CSS stabilizer codes admit a quasi-transveral [47] implementations of certain CNOT gates in $\mathcal{P}_2$. Logical gates U admitting an implementation by a constant-depth quantum circuit are also desirable for similar reasons. Here, error propagation is kept under control and can be bounded to a *light-cone* like regions associated to the circuit. The gates in such a circuit should be geometrically local[48] to simplify their physical realization and contain growth of such light-cone regions. For this reason, it is important to classify logical gates of quantum error-correcting codes admitting such an implementation. We use the term *locality-preserving* to refer to a logical unitary that can be implemented by a constant-depth geometrically-local circuit. The main feature is that the support of geometrically local observable which are conjugated by such a unitary remains geometrically local and may only grow its support by incorporating a constant radius neighborhood.

Bravyi and König[4] consider the set of logical gates implementable by locality-preserving quantum circuits that may be realized on a topological stabilizer code. The following is a restatement of their main result:

Theorem. [Bravyi and König] *Let U be a morphism between two topological stabilizer codes $\mathcal{C}_1$ and $\mathcal{C}_2$ defined on a sufficiently large D-dimensional lattice. Then if U admits a locality-preserving implementation, the gate associated to U is contained in $\mathcal{P}_D$.*

The theorem by Bravyi and König deals with code deformations [17] *i.e.* transformations mapping code $\mathcal{C}_1$ onto code $\mathcal{C}_2$. For simplicity, here we assume $\mathcal{C}_1 = \mathcal{C}_2$. Our arguments may then be made applicable to code deformations $\mathcal{C}_1 \neq \mathcal{C}_2$ by fixing the interpretation of a locality-preserving reversible morphism U between codes to be the logical identity. The circuit implementing $U^\dagger : \mathcal{C}_2 \to \mathcal{C}_1$ may then be composed with any other code deformation circuit V to obtain a code-preserving locality-preserving gate $U^\dagger V : \mathcal{C}_1 \to \mathcal{C}_1$ which is covered by the special case considered.

B. Gauge and logical qubits

One of our aims is to generalize BK's result to topological subsystem codes as specified by the Pauli stabilizer formalism [18]. Some of our definitions, are aimed to tackle the more general (less-structured) setting of operator quantum error correction formalism [19, 20]. Intuitively, a subsystem code is a stabilizer code defined by $\mathcal{S}$ for which we encode quantum information into only a subset of the qubits associated to the stabilized subspace. Encoded qubits in this subset are called logical qubit whereas the remaining qubits are called gauge qubits (*i.e.* the stabilized subspace may be decomposed into $\mathcal{H}_{\text{logical}} \otimes \mathcal{H}_{\text{gauge}} = \mathcal{C}(\mathcal{S})$ as in Fig. 1).

A subsystem code is concisely defined by its gauge group $\mathcal{G} \subseteq \mathsf{Pauli}$ which may be non-abelian and contain $-\mathbb{1}$, unlike the stabilizer group $\mathcal{S}$. Up to global phases, the stabilizer subgroup consists of the center $\mathcal{S} \equiv Z(\mathcal{G})/\mathbb{C}$ of the gauge group $\mathcal{G}$ (i.e. the elements of $\mathcal{G}$ that commute with all the elements in $\mathcal{G}$). In fact, there are multiple consistent choices for the global phases for the operators in $\mathcal{S}$ such that they are all in the Pauli group and $-\mathbb{1}$ is not included in the group. The freedom for the choice of $\mathcal{S}$ is associated to the signs of its generators. The codespace of the subsystem code, denoted by $\mathcal{C}(\mathcal{S})$, is the joint $+1$ eigenspace of $\mathcal{S}$. Under this definition, gauge operators act trivially on the subsystem composed of logical qubits while still allowing for a non-trivial action on the stabilized subspace. In particular, the case of $\mathcal{S} = \mathcal{G}$ corresponds to a stabilizer code.

A potential advantage of subsystem codes is that, by not requiring to keep track of errors affecting gauge qubits [21], recovery procedures may admit simpler realizations. One might hope that requiring locality-preserving implementation for gates on subsystem codes to be less restrictive than it is for stabilizer codes. However, our results show that the set of locality-preserving gates for subsystem codes is similarly restricted.

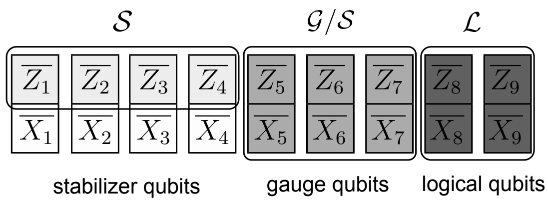

FIG. 1: The algebraic structure of the gauge group $\mathcal{G}$ and the stabilizer subgroup $\mathcal{S}$ in a subsystem code, is depicted by associating an independent generator to each box such that all generators commute except pairs in the same column. The figure illustrates an example with $n = 9$ physical qubits, three gauge qubits and two logical qubits. The full Pauli algebra for the gauge qubits $\mathcal{G}/\mathcal{S}$ (middle grey). The stabilizer group $\mathcal{S}$ is generated by the Z operators on the stabilized qubits (light grey). The remaining qubits (dark grey) represent the algebra of logical qubits $\mathcal{L}$. An appropriate Clifford transformation U can reduce the the generators to a canonical form such that $\tilde{X}_j = UX_jU^\dagger$ and $\tilde{Z}_j = UZ_jU^\dagger$ for $j = 1, \ldots, n$.

C. Bare and dressed logical operators

Logical operators preserve the codespace $\mathcal{C}(\mathcal{S})$ and act non-trivially on logical qubits. In a subsystem code, there are two types of logical operators, called *bare* and *dressed* logical operators, depending on how they act on gauge qubits [15]. Given a decomposition of the codespace as $\mathcal{C}(\mathcal{S}) = \mathcal{H}_{\text{logical}} \otimes \mathcal{H}_{\text{gauge}}$, bare logical operators act exclusively on logical qubits and act trivially on gauge qubits: $[U_{\text{bare}}] = [U]_L \otimes [I]_G$ where $[U]_L$ represents a logical action of U_{bare} on logical qubits, and $[I]_G$ represents a trivial action on gauge qubits. Formally, bare Pauli logical operators are the elements of the centralizers of the gauge group $\mathcal{G}$ (i.e. Pauli operators that commute with all the elements of $\mathcal{G}$): $\mathcal{L}_{\text{bare}} = C(\mathcal{G}) = \{z \in \mathsf{Pauli} : \forall g \in \mathcal{G}, zg = gz\}$ denotes the centralizer of $\mathcal{G}$. Bare logical operators are identified up to stabilizer operators $\mathcal{S}$ since stabilizers act trivially on both gauge and logical qubits. For this reason, non-trivial bare logical operators are elements of $C(\mathcal{G})$ which are not in $\mathcal{S}$.

Existing definitions of bare and dressed logical operators [15] rely on the Pauli centralizer group $C(\mathcal{G})$, thus restricting logical operators to $\mathcal{P}_1$. In order to allow for other logical operations, such as higher order Clifford gates, we must provide a more general definition.

In particular, a bare logical unitary will be a unitary generated by the algebra of bare logical Paulis. It has a logical action on the code space described by $[A] = [A]_L \otimes \mathbb{1}_G$ which factorizes with respect to the $\mathcal{C}(\mathcal{S}) = \mathcal{H}_{\text{logical}} \otimes \mathcal{H}_{\text{gauge}}$ decomposition of the codes and is trivial on the gauge qubits. This means that a Hermitian bare logical operator can be thought of as an observable associated to the encoded information independent of the state of the gauge quits. Dressed logical operators, must also respect a tensor product form $[A] = [A]_L \otimes [A]_G$, but may admit a non-trivial action $[A]_G$ on the gauge qubits. We will say that $[A]_L$ is the logical action of such a dressed logical operator.

We are particularly interested in dressed logical unitaries as they preserve bare logical observables under conjugation.

Lemma 1. *Let U be a dressed logical unitary and B be a bare logical operator for a subsystem code. Then $UBU^\dagger$ is also a bare logical operator.*

It is interesting to provide alternate operator algebraic definitions for bare and dressed logical operators which coincide in the case of qubit subsystem codes. In appendix C, we provide such definitions, in the hope that these will be useful to a broader context of quantum error correcting codes beyond the qubit subsystem codes.

D. Cleaning lemma

The notion of *cleaning*, initially introduced for stabilizer codes [9], can be generalized to subsystem codes [15]. Let us begin by reviewing the cleaning procedure for stabilizer codes. Consider a logical Pauli operator $P \in \mathcal{L}$ that has non-trivial support on a subset R of qubits. For stabilizer codes and logical operators P of tensor product Pauli form, cleaning P within the subset R refers to a procedure of multiplying P by an operator $S \in \mathcal{S}$ to obtain a logically equivalent operator PS that has a trivial action on R. The cleaning is not always possible, and can be performed if and only if there exists a stabilizer S whose action on R is identical to the action of P on R. Indeed, it is necessary that for some S, $P|_R = S|_R$ up to a complex phase where $P|_R$ and $S|_R$ represent restriction of P and S to the subset R, *i.e.* the respective operators obtained by considering only tensor factors supported on the subset R.

The cleaning lemma by Bravyi and Terhal states that if a subset R supports no logical operators (except the one with trivial action), then any logical operator P can be cleaned within R [9]. Namely, there exists a stabilizer S such that PS is supported exclusively on $\overline{R}$, the complement of R. For any subset R of qubits, one may define $l(R)$ to be the number of independent Pauli logical operators supported exclusively on R. A result in [22] concisely relates the set of independent logical operators supported on two complementary subsets of qubits.

Lemma 2. *Suppose a stabilizer code has k logical qubits. Then $l(R) + l(\overline{R}) = 2k$.*

The cleaning lemma is recovered from lemma 2 by imposing that there be no logical operator supported on R,

$l(R) = 0$, which leads to $l(\overline{R}) = 2k$. Since there are $2k$ independent Pauli logical operators for a stabilizer code with k logical qubits, all the logical operators have representation with support only on $\overline{R}$. Thus cleaning of subset R is always possible.

In the case of subsystem codes, multiplication by an element of $\mathcal{S}$ preserves bare logical operators, whereas multiplication by an element of $\mathcal{G}$ preserves dressed logical operators. A result due to Bravyi [15] generalizes lemma 2 to relate the respective number of independent bare and dressed logical operators supported on two complementary regions. In particular, we may define $l_{\text{dressed}}(R)$ and $l_{\text{bare}}(R)$ to be the number of independent dressed and bare Pauli logical operators supported on R.

Lemma 3. *Suppose a subsystem code has k logical qubits. Then $l_{\text{dressed}}(R) + l_{\text{bare}}(\overline{R}) = 2k$.*

This lemma implies that if there are no non-trivial dressed (bare) logical operators fully supported on R, all bare (dressed) logical operators can be cleaned within the region R so that they are supported exclusively on $\bar{R}$. This leads to the following definition of bare (dressed)-cleanable regions.

Definition 2. *A region R is* ***bare (dressed)-cleanable****, if it supports no non-trivial dressed (bare) logical operators.*

Cleanability is closely related to coding properties of the code. The distance d of a subsystem code is defined as the size of the smallest possible support of an operator in $\mathcal{L}_{\text{dressed}} \setminus \mathcal{G}$ (a dressed Pauli logical operator with non-trivial action on $\mathcal{H}_{\text{logical}}$). Furthermore, a subset R of qubits is *correctable* if and only if it supports no dressed logical operator. In other words, the subset R is correctable if and only if R is bare-cleanable.

E. Fault-tolerant logical gate and cleanability

Let us now present a key technical lemma which plays a central role in deriving our main results.

Lemma 4. *Let $\{R_j\}_{j\in[0,m]}$ be a set of regions where R_0 is bare-cleanable and each of the regions $\{R_j\}_{j\in[1,m]}$ is dressed-cleanable in a subsystem code. If a dressed logical unitary U is supported on the union $\bigcup_{j\in[0,m]} R_j$ and is transversal with respect to regions R_j, then the logical action of U factorizes with respect to the logical and gauge qubits $[U] = [U]_L \otimes [U]_G$ and $[U]_L$ is an element of $\mathcal{P}_m$ (the m-th level of the Clifford hierarchy).*

The above theorem does not require any form of locality of the gauge or stabilizer generators, and thus applies to arbitrary subsystem codes. Furthermore, the regions $\bigcup_{j\in[0,m]} R_j$ need not cover the full set of qubits of the code, giving rise to some interesting observations.

Proof. The proof proceeds by induction on m, the number of regions considered. Assuming $m = 0$, the operator U is fully supported on a bare-cleanable region R_0. The full algebra of bare logical Pauli operators may be supported on $\overline{R}_0$ hence they must commute with U. Thus, $[U]_L = \mathbb{1}_L$ must be a trivial logical operator in $\mathcal{P}_0$ (proportional to identity) and $[U] = \mathbb{1}_L \otimes [U]_G$.

Let us now prove the inductive step. We assume that all the dressed transversal operators supported on the union $R_0 \cup \bigcup_{j=1}^{m} R_j$ are in $\mathcal{P}_m$. Consider a transversal dressed logical operator U such that

$$\text{supp}(U) \subseteq R_0 \cup \bigcup_{j=1}^{m+1} R_j. \tag{2}$$

By definition, the U is a dressed logical operator. Since R_{m+1} is dressed-cleanable, all the dressed Pauli operators may be supported on $\bar{R}_{m+1}$. Hence, the group commutator $UPU^\dagger P^\dagger$ is also a dressed logical operator with a tensor product form with respect to the gauge and logical qubits. Furthermore, the transversality of U and P with respect to the subsets R_j implies

$$\text{supp}(UPU^\dagger P^\dagger) \subseteq R_0 \cup \bigcup_{j=1}^{m} R_j, \tag{3}$$

which in turn requires $[UPU^\dagger P^\dagger]_L \in \mathcal{P}_m$. By definition of the Clifford hierarchy, $[U]_L \in \mathcal{P}_{m+1}$. $\square$

III. ERASURE THRESHOLD AND TRANSVERSAL LOGICAL GATES

One conclusion that may be reached at this point is a trade-off between erasure threshold and the set of achievable transversal gates. Quantum error-correcting codes should ideally tolerate errors (such as depolarization) on a subset of physical qubits randomly drawn with a small but constant probability p. Loosely, a family of codes parametrized by the number of physical qubits n has an error threshold p_e if the probability of correcting independent errors, occurring with probability $p < p_e$ approaches unity as n grows. Erasure errors, which correspond to the loss of physical qubits from the system, are an important special case for such errors, as they are unavoidable in many realistic physical systems. Furthermore, any form of depolarizing noise is more severe than qubit erasure since, in the latter, full information on the location of errors is available. [49] For this reason, the erasure threshold is necessarily larger than the depolarization error threshold for any quantum error-correcting code. The following corollary elucidates the existing trade-off between erasure threshold and the set of transversally implementable gates.

Theorem 1. [Erasure threshold] *Suppose we have a family of subsystem codes with a erasure threshold $p_l > 1/n$ for some natural number n. Then, any transversally implementable logical gate must belong to $\mathcal{P}_{n-1}$.*

Proof. Suppose $p_l > 1/n$, and assign each qubit to one of n regions $\{R_j\}_{j\in[0,n-1]}$ uniformly at random. Each of the regions chosen this way will be correctable with a probability which is arbitrarily close to unity as we take larger codes from the family. Finally, we may conclude by applying lemma 4 to the n correctable regions obtained in this way, which are both bare and dressed cleanable. $\square$

Theorem 1 applies to arbitrary stabilizer and subsystem codes, and is *not* restricted to codes with geometrically local generators.

Example 1. The toric code saturates the bound of theorem 1. It has an erasure threshold of $p_l = 1/2 > 1/3$ and can still transversely implement some logical operators in $\mathcal{P}_2$ (such as CNOT) [23].

Example 2. The Reed-Muller code $[[2^m-1, 1, 3]]$ admits the transversal implementation of the $\pi/2^{m-1}$ phase gate which belongs to $\mathcal{P}_m$ and not to $\mathcal{P}_{m-1}$ [6]. As a family of codes with increasing m, it must have a zero erasure threshold.

Example 3. D-dimensional topological color codes admit the transversal implementation of gates in $\mathcal{P}_D$ but not of gates in $\mathcal{P}_{D+1}$. Their erasure threshold is hence upper bounded by $1/D$. This conclusion may likely be recovered by other arguments related to percolation in D-dimensional lattices.

IV. CONSTANT DEPTH CIRCUITS AND GEOMETRIC LOCALITY

The discussion so far does not rely on geometric locality of the generators of the code. The underlying assumption of geometric locality is that physical qubits are placed on a regular lattice, the density of qubits is finite and the stabilizer/gauge generators involve only particles within a neighborhood of constant size. More precisely, the gauge group $\mathcal{G}$ may be generated by a set of Pauli operators, with support restricted to a ball of diameter $\xi = O(1)$. In this section, we generalize BK's result to topological subsystem codes that are supported on a D-dimensional lattice with geometrically local gauge generators.

A. Union lemma

The first challenge in generalizing BK's result is that the so-called *union lemma* does apply to topological subsystem codes. The union lemma for a topological stabilizer code states that *the union of two spatially disjoint cleanable regions is also cleanable.* We say that two regions are spatially disjoint if local stabilizer generators overlap with at most one of the regions.

Lemma 5. **[Union lemma for stabilizer codes]** *For a topological stabilizer code, let R_1 and R_2 be two spatially disjoint regions such that there exists a complete set of stabilizer group generators $\{S_j\}$ each intersecting at most one of $\{R_1, R_2\}$. If R_1 and R_2 are cleanable, then the union $R_1 \cup R_2$ is also cleanable.*

At this point, let us review the derivation of BK's result in order to illustrate the use of the union lemma. For a topological stabilizer code with a growing code distance, one is able to split the D-dimensional space into $D+1$ regions R_m for $m = 0, \dots, D$, where R_m consists of small regions with connected components of constant size which are spatially disjoint. Let us demonstrate it for $D = 2$ (see Fig. 2). We first split the entire lattice into square tiles so that the diameter of local stabilizer generators is much shorter than the spacing of the tiles. This square tiling has three geometric object: points, lines and faces. First, we "fatten" points to create regions R_0. We then fatten lines and create regions R_1. The remaining regions are identified to be R_2. Therefore R_m is the union of fattened m-dimensional objects. For a D-dimensional lattice, we start with a D-dimensional hyper-cubic tiling and fatten m-dimensional objects to obtain R_m for $m = 0, \dots, D$. Region R_D is actually composed by the original D dimensional tiles which have been "eroded" by the fattening of lower dimensional objects.

Every connected component in R_m is cleanable as the code distance is growing with the system size n. Also, connected components in R_m are spatially disjoint. Due to the union lemma, the union of spatially disjoint small regions is correctable, and thus R_m is correctable. Then lemma 4 implies that transversally implementable logical gates are restricted to $\mathcal{P}_D$, recovering BK's result (Theorem II A).

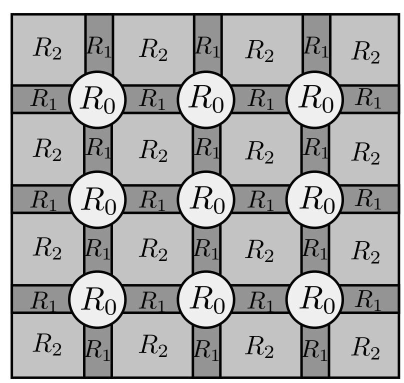

FIG. 2: The partition of a two-dimensional lattice into three regions R_0, R_1, R_2 which consist of smaller regions that are correctable and spatially disjoint.

For a topological subsystem code, two regions are said to be spatially disjoint if local gauge generators may overlap with at most one of the regions. Unlike a topological stabilizer code, however, geometric locality of stabilizer generators is not always guaranteed since the stabilizer subgroup $\mathcal{S}$ is defined to be the center of the gauge group $\mathcal{G}$, and generators of $\mathcal{S}$ are, in general, products of multiple local gauge generators. As such, the union lemma holds only for dressed-cleanable regions as summarized below.

Lemma 6. [Union lemma for subsystem codes] *For a topological subsystem code, let R_1 and R_2 be two spatially disjoint regions such that there exists a complete set of gauge group generators $\{G_j\}$ each intersecting at most one of $\{R_1, R_2\}$. If R_1 and R_2 are dressed-cleanable, then the union $R_1 \cup R_2$ is also dressed-cleanable.*

By taking a complete set of geometrically-local gauge generators the union lemma for dressed-cleanable regions can be given a geometric interpretation. A geometric interpretation for the union lemma for bare-cleanable regions can be obtained as long as the stabilizer subgroup admits a complete set of geometrically local stabilizer generators. This is the case for Bombin's gauge color code, which is a three-dimensional subsystem code [13]. However, a complete set of local stabilizer generators is not guaranteed for arbitrary topological subsystem codes, as is exemplified by the quantum compass model [21]. The absence of a geometric union lemma for bare-cleanable regions is the main difficulty in generalizing the result by BK to topological subsystem codes.

B. Generalization of Bravyi-König theorem to topological subsystem codes

BK's derivation relies only on a macroscopic code distance, which is a requirement for a finite error threshold. A macroscopic code distance is sufficient to guarantee a finite error threshold only in the case of constant weight stabilizer generators as proven by Kovalev and Pryadko [25]. This does not apply to all topological subsystem codes. For example, two- and three-dimensional quantum compass models have a macroscopic code distance and local gauge generators yet still lack a positive error threshold due to their high weight stabilizer generators[26]. This justifies the approach taken in the present work, where we use the fault-tolerance itself as the guiding principle. Namely, in order to generalize BK's result to topological subsystem codes, we assume that (i) the code distance grows at least logarithmically, and (ii) the code has a finite (erasure) error threshold.

The distance $d(,)$ between physical qubits on the lattice will be used to define the r-neighborhood $\mathfrak{B}(R, r)$ of a region R which includes R and all physical qubits within distance r from it. Furthermore, we define the spread s_U of a unitary as the smallest possible distance such that $\forall A : \mathrm{supp}(UAU^\dagger) \subseteq \mathfrak{B}(\mathrm{supp}(A), s_U)$. In particular, if U is implemented by a constant depth circuit composed of geometrically local gates, the spread s_U will also be bounded by a constant.

A version of lemma 4 involving the lattice geometry can now be stated.

Lemma 7. *Let U be a dressed logical unitary operator supported on the union of mutually non-intersecting regions R_0 and $\{R_j\}_{j\in[1,m]}$. If R_0 is bare-cleanable and each $R_j^+ := \mathfrak{B}(R_j, 2^{j-1}s_U)$ is dressed-cleanable for $j > 0$, then the logical unitary implemented by U belongs to $\mathcal{P}_m$.*

This means that when dealing with locality-preserving circuits which implement logical unitary gates, it is sufficient to use extended correctable regions such that they overlap within a boundary of width $2^{m-1}s_U$, where m is the number of regions to be used. As such, much of discussion dealing with transversal gates applies to finite depth circuits. The proof is presented in appendix B.

With an assumption of macroscopic code distance alone, one is able to obtain the following statement for topological subsystem codes.

Corollary 1. *Consider a family of subsystem codes with increasing code distance defined by geometrically local gauge generators of diameter bounded by ξ in D spatial dimensions. Then the set of dressed logical unitary gates implementable by constant depth circuits is included in $\mathcal{P}_{D+1}$.*

Proof. Since gauge generators are geometrically local with diameter bounded by ξ, the union lemma (lemma 6) applies to dressed cleanable regions that are separated by a distance ξ or larger. Furthermore, by the definition, any region with volume smaller than the code distance d is dressed-cleanable. Let s_U be the spread of the circuit U. One has $d > (2^D s_U + \xi)^D$ for sufficiently large n since the code has a macroscopic distance. Then the lattice may be partitioned into $D+1$ disjoint regions $\{R_j\}_{j\in[1,D+1]}$ such that $R_j^+ := \mathfrak{B}(R_j, 2^{j-1}s_U)$ is dressed-cleanable for all $j > 0$. For instance, we construct a D-dimensional hyper-cubic tiling and fatten m-dimensional objects to obtain R_{m+1} for $m = 0, \ldots, D$. By taking R_0 to be an empty set $\emptyset$, we conclude that the logical action of U is included in P_{D+1}. □

Note that R_j is dressed-cleanable, but not necessarily bare-cleanable since the union lemma does not hold for bare-cleanable regions. Taking R_0 to be the empty set results in loosening the bound on the implementable level of the Clifford hierarchy by one with respect to BK's result for topological stabilizer codes. An interesting open problem is to find subsystem codes with growing distance which achieve the bound stated in corollary 1. If such subsystem codes exist, we believe that they would be highly artificial and would possess highly non-local stabilizer generators.

From now on, we assume that the family of codes has a non-zero erasure threshold $p_l > 0$ and that the code distance d grows at least logarithmically with the number of particles n. Under these reasonable and perhaps indispensable assumptions for fault-tolerance of the code, we obtain the same conclusion as BK's result for topological subsystem codes.

Theorem 2. **[Subsystem code]** *Consider a family of subsystem codes with geometrically-local gauge generators in D spatial dimensions with i) an erasure threshold $p_l > 0$ and ii) a code distance $d = \Omega(\log^{1-1/D}(n))$. Then any dressed logical unitary that can be implemented by a constant depth geometrically local circuit U belongs to $\mathcal{P}_D$.*

Our proof technique borrows an idea used by Hastings in a different context [27].

Proof. For simplicity, let us assume that U is transversal. The argument leading to lemma 7 suffices to make the current proof applicable to constant depth geometrically local circuits by taking care of some cumbersome yet inessential technical details.

Imagine that some subset of qubits, denoted as R_{loss}, is lost. This subset R_{loss} is chosen so that each site has an independent probability $p_0 < p_l$ of being included in R_{loss}. By definition of erasure threshold, R_{loss} must be correctable (in other words, bare-cleanable) with probability approaching to unity as the system size n grows. The key idea is to make use of this randomly generated bare-cleanable region R_{loss} to construct a bare-cleanable region R_0 which consists of spatially disjoint balls of constant radius.

For any fixed region R, the probability that R is included in R_{loss} is given by $\Pr(R \subseteq R_{\mathrm{loss}}) = p_0^{|R|}$. Thus, a ball of constant radius $r \gg \xi$ is included in R_{loss} with some constant probability independent of n. Let us now split the full lattice into unit cells of volume $v_c = c\log(n)$ as in Fig. 3. Inside a given unit cell, the probability of having no ball of radius r included in R_{loss} is $O(1/\operatorname{poly}(n))$, where the power of n can be made arbitrary large by increasing a finite constant c. Hence, with probability approaching to unity, R_{loss} includes at least one ball of radius r in each unit cell. We choose one ball from each unit cell so that they are spatially disjoint, and denote their union as R_0. Region R_0 is bare-cleanable and contains one ball of diameter r per tile. We may construct a skewed D-dimensional hyper-cubic tiling by connecting balls in R_0 corresponding to neighboring tiles, which are separated by at most $O(\log(n)^{1/D})$.(see Fig. 3). We then fatten m-dimensional objects to construct a covering of the full lattice with R_m for $m = 0, \dots, D-1$. The region R_D is composed of the skewed cells which have been eroded by thickened lower dimensional objects.

It remains to prove that R_m for $m > 0$ are dressed-cleanable. Any region with volume smaller than $d = \Omega(\log^{1-1/D}(n))$ is cleanable. For $m < D$, R_m consists of connected components with volume at most $O(\log^{1-1/D}(n))$, and hence are dressed-cleanable. For R_D, suppose that there exists a non-cleanable D-dimensional connected component, denoted as R, with volume $O(\log(n))$. Then R must support at least one bare logical Pauli operator U_{bare}. Yet, the disentangling lemma [14] tells that U_{bare} can be supported on qubits within the boundary of R. The volume of the boundary is at most $O(\log^{1-1/D}(n))$ which leads to a contradiction. Therefore, R_D is dressed-cleanable. Given a bare-cleanable region R_0 and dressed cleanable regions R_m for $m = 1, \dots, D$, lemma 4 implies that transversally implementable U must be included in $\mathcal{P}_D$. □

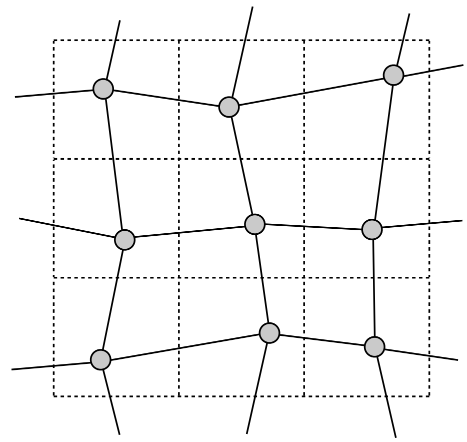

FIG. 3: A construction of a bare-cleanable region R_0. Circles represent balls that are included in randomly generated subset R_{loss} of qubits. Dotted lines mark unit cells with volume $O(\log(n))$.

A further observation is that constant-depth circuits supported on a string-like region must be Pauli operators and, in general, constant-depth logical operators supported on a m-dimensional region must be in $\mathcal{P}_m$ regardless of the spatial dimension of the lattice $D \geq m$.

V. NON-CLIFFORD GATE PROHIBITS SELF-CORRECTION

A self-correcting quantum memory is a system that allows reliable storage of quantum information for macroscopic times when put in contact with a thermal environment [29]. At low enough temperatures, the energy landscape provided by the system Hamiltonian should make it unlikely for the accumulation of physical errors to result in a logical error [9, 28]. An important question is whether such a system may exist in three spatial dimensions. No-go results have ruled out most two-dimensional systems and a certain class of three-dimensional systems [9, 29–31]. Furthermore, at the moment, there are no known three-dimensional model with macroscopic quantum memory time.

In this section, we derive a new no-go result on three dimensional self-correcting quantum memory that arises from fault-tolerant implementability of a non-Clifford gate. In particular, we show that a stabilizer Hamiltonian with a locality-preserving non-Clifford gate cannot have a macroscopic energy barrier, and thus it is not expected to provide a practical increase in memory time in terms of the system size n. We then derive an upper bound on the code distance of topological stabilizer codes with locality-preserving logical gates from the higher-level Clifford hierarchy.

A. Upper bound on code distance

The presence of locality-preserving logical gates from the higher-level Clifford hierarchy imposes a restrictions on the geometric locality of other logical operators. Here, we find a trade-off between the code distance and fault-tolerant implementability of logical gates.

Theorem 3. **[Distance trade-off]** *If a topological stabilizer code in D spatial dimensions admits a locality-preserving implementation for a logical gate from $\mathcal{P}_m$, but outside of $\mathcal{P}_{m-1}$, its code distance is upper bounded by $d \leq O(L^{D+1-m})$.*

Proof. Let $R_0, R_1, \ldots, R_{m-1}$ be regions which jointly cover the whole lattice. Each region is a collection of disjoint aligned $(D+1-m)$-dimensional objects (Fig. 4 corresponds to the case for $D = 3$ and $m = 3$). Suppose that there is no logical operator supported on any of the regions R_j. Applying lemma 4, implementable logical operators are restricted to $\mathcal{P}_{m-1}$, leading to a contradiction. Thus, at least one region R_j supports a logical operator. Due to the union lemma, such a logical operator can be supported on a single $(D+1-m)$-dimensional object of volume $O(L^{D+1-m})$, which implies $d \leq O(L^{D+1-m})$. □

Bravyi and Terhal have derived an upper bound on the code distance for topological stabilizer and subsystem codes: $d \leq O(L^{D-1})$ [9]. Whether the Bravyi-Terhal bound is tight for $D \geq 3$ remains open. For $m = 2$, our bound is reduced to the Bravyi-Terhal bound [9] whereas for $m > 2$ we obtain a stronger bound on the code distance of topological stabilizer codes.

Topological color codes, proposed in a seminal work by Bombin and Martin-Delgado [10–12], are families of D-dimensional topological stabilizer codes. Some of these codes admit transversal implementations of logical gates in $\mathcal{P}_D/\mathcal{P}_{D-1}$-th level of the Clifford hierarchy. For these codes, there is a string-like logical operator, and thus $d = O(L)$, implying that our bound is tight for $m = D$.

Example 4. Topological color codes in D spatial dimensions saturate the bound in theorem 3.

B. Self-correction and fault-tolerance

For a topological stabilizer code, the stabilizer Hamiltonian is composed of geometrically local operators in the stabilizer group: $H = -\sum_j S_j$, where $S_j \in \mathcal{S}$. A non-rigorous yet commonly used proxy to verify whether self-correction can be achieved is the presence of a macroscopic energy barrier that scales with the system size. A macroscopic energy barrier seems to be a necessary but not sufficient condition for the system to exhibit macroscopic memory time [50]. For stabilizer Hamiltonians, the presence of string-like logical operators implies the absence of a macroscopic energy barrier [51].

The previous theorem also imposes a trade-off on locality-preserving logical gates arising from a macroscopic energy barrier in a stabilizer Hamiltonian. It can be obtained as a converse for the case $m = D$.

Corollary 2. **[Self-correction]** *If a stabilizer Hamiltonian in D spatial dimensions has a macroscopic energy barrier, the set of fault-tolerant logical gates is restricted to $\mathcal{P}_{D-1}$.*

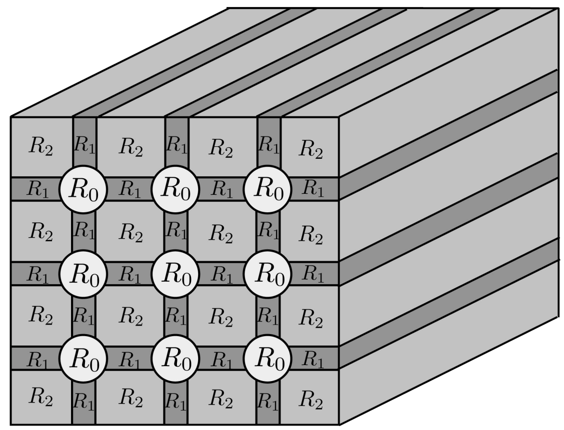

FIG. 4: The partition of the lattice into $R_0, R_1, \ldots, R_{D-1}$ for $D = 3$.

Haah [32, 33] provided the first example of a three-dimensional topological stabilizer code without string-like logical operators. The code is defined on a three dimensional $L \times L \times L$ cubic lattice with an energy barrier scaling as $O(\log L)$. There also exists a number of three-dimensional translation symmetric stabilizer codes without string-like logical operators [34, 35]. By theorem 2, for $D = 3$ the presence of a macroscopic energy barrier implies that the set of locality-preserving logical gates is restricted to $\mathcal{P}_2$.

Corollary 3. *Haah's 3D stabilizer code [32] has no constant depth logical gates outside of $\mathcal{P}_2$.*

A different approach to construct stabilizer codes with a macroscopic energy barrier has been proposed by Michnicki [36], who introduced the notion of code welding to construct new codes by combining existing ones. The welding technique leads to a construction of a topological stabilizer code with a polynomially growing energy barrier in three spatial dimensions. Our theorem 2 also applies to this code.

Corollary 4. *Michnicki's 3D welded stabilizer code has no constant depth logical gates outside of $\mathcal{P}_2$.*

A model of a six-dimensional self-correcting quantum memory with fault-tolerantly implementable non-Clifford gates has been proposed [37]. An intriguing question is whether or not such a code may exist in four (or five) spatial dimensions.

We then move to the discussion of topological subsystem codes. A generic recipe to construct Hamiltonians for topological subsystem codes is not known. A candidate Hamiltonian, often discussed in the literature, is composed of geometrically local terms in the gauge group: $H = -\sum_j G_j$ [52]. As long as Hamiltonian terms consist only of local generators of the gauge group $\mathcal{G}$, the presence of bare-logical operators with string-like support implies the absence of an energy barrier .

For topological subsystem codes, we obtain a less restrictive trade-off between fault-tolerant implementability and geometric non-locality of logical gates.

Corollary 5. *If a topological subsystem code in D spatial dimensions has macroscopic energy barrier, the set of transversal operators is restricted to $\mathcal{P}_D$.*

The three-dimensional gauge color code has transversal gates in $\mathcal{P}_2$ and do not have string-like bare logical operators, and hence are not ruled out from having a macroscopic energy barrier.

VI. CONCLUSIONS

We have provided several extensions of BK's characterization of locality-preserving logical gates which constitute a natural approach to achieve fault tolerance in topological stabilizer and subsystem codes.

Our results are summarized as follows: (i) A three-dimensional stabilizer Hamiltonian admitting a locality-preserving non-Clifford gate is not self-correcting. (ii) The code distance of a D-dimensional topological stabilizer code with a non-trivial m-th level locality-preserving logical gate is upper bounded by $O(L^{D+1-m})$. (iii) An erasure threshold of a subsystem code with non-trivial m-th level transversal logical gate is upper bounded by $1/m$. (iv) Locality-preserving logical gates in a D-dimensional topological subsystem code belong to the D-th level $\mathcal{P}_D$ in the presence of a finite error threshold.

An interesting open problem, is the further generalization of the result of Bravyi and König to other families of codes such as frustration-free commuting projector codes. In this direction, a characterization of locality-preserving logical operations in the context of topological quantum field theories has been presented [46]. Another interesting generalization concerns topological codes with geometrically non-local gates, and quantum low density parity check (LDPC) codes. It has been recently proven by the authors that, for families of the toric code and color codes, local constant-depth gates (not necessarily geometrically-local) do not increase the level of the implementable Clifford hierarchy.

The definition of quantum phases, widely accepted in the literature, is that two ground state wavefunctions belong to different phases if there is no local unitary transformation connecting them [39]. Yet, even within the ground space of a Hamiltonian, it is possible that different ground states are in different phases. Perhaps, BK-type characterization will give a coherent insight into the classification of ground state wavefunctions with long-range entanglement.

Fault-tolerant implementability of non-Clifford logical gates is an important ingredient for magic-state distillation protocols [40]. An interesting future problem includes the asymptotic rate of the number of magic states that can be distilled with a desired precision. In general, it may be interesting to study whether similar restrictions apply to the gauge-fixing technique [13, 41] , code concatenation [42] and other approaches to achieve universal fault-tolerant quantum computation.

Acknowledgements

We would like to thank Michael Beverland for pointing out non-geometric interpretation of BK's results, and Robert König, Kristan Temme, David Poulin and John Preskill for fruitful discussions. We would also like to thank an anonymous referee for valuable feedback. We acknowledge funding provided by the Institute for Quantum Information and Matter, a NSF Physics Frontiers Center with support of the Gordon and Betty Moore Foundation (Grants No. PHY-0803371 and PHY-1125565). BY is supported by the David and Ellen Lee Postdoctoral fellowship.

Appendix A: Observations on the Clifford hierarchy

In the present work, we have adopted a slightly different definition of the Clifford hierarchy $\mathcal{P}_n$ from the one introduced by Gottesman and Chuang [5], and used by Bravyi and König [4]. In this appendix, we would like to justify that these two definitions are mostly equivalent. However, our definition lead to more compact proofs and result statements.

Definition 3. *The Clifford hierarchy is usually defined as follows. The first level of the hierarchy is taken to be the Pauli group* $\mathsf{Clifford}_1 \equiv \mathsf{Pauli}$. *Successive levels of the hierarchy are defined recursively as*

$$\mathsf{Clifford}_{m+1} = \left\{ U : \forall P \in \mathsf{Pauli},\ UPU^\dagger \subseteq \mathsf{Clifford}_m \right\}. \tag{A1}$$

The following statement shows how definition 1 is equivalent to definition 3.

Lemma 8. $\mathcal{P}_1 = \mathbb{C} \cdot \mathsf{Pauli}$ *and* $\mathcal{P}_n = \mathsf{Clifford}_n$ *for* $n \geq 2$.

Proof. Let us first show that $\mathcal{P}_1 = \mathbb{C} \cdot \mathsf{Pauli}$. Suppose that $U \in \mathcal{P}_1$. By hypothesis, the group commutator of U with any Pauli operator $P \in \mathcal{P}$ is trivial up to a phase $UPU^\dagger P^\dagger = e^{i\theta}$. This phase must be ± 1, since it is an eigenvalue for the rank one superoperator resulting from conjugation by a Pauli operator $PU^\dagger P^\dagger = e^{i\theta}U^\dagger$. Conversely, we may consider the rank one superoperator $U \cdot U^\dagger$ for which the Pauli operators constitute a full set of eigenoperators with eigenvalues ± 1. This uniquely determines U to be equivalent to a Pauli operator itself up to a global phase. Here, we have used the fact that the Pauli operators linearly span the full operator algebra.

We will now prove by induction in $n \geq 2$ that $U \in \mathcal{P}_n \Leftrightarrow U \in \mathsf{Clifford}_n$. The proof relies on the observation that all the levels of the standard Clifford hierarchy are closed under right multiplication by Pauli operators $\mathsf{Clifford}_n = \mathsf{Clifford}_n \cdot \mathsf{Pauli}$ which can be proven inductively.

Suppose $U \in \mathcal{P}_{n+1}$. Hence, for any $P \in \mathsf{Pauli}$ we have that $UPU^\dagger P^\dagger \in \mathcal{P}_n$ and consequently $UPU^\dagger \in \mathsf{Clifford}_n$. This implies that $U \in \mathsf{Clifford}_{n+1}$. The converse can be similarly proven. $\square$

The hierarchy is composed of increasingly larger sets of gates, where $\mathsf{Clifford}_n \subset \mathsf{Clifford}_{n+1}$. These sets are closed under group multiplication only for $n \leq 2$. Furthermore, $\mathsf{Clifford}_n/\mathbb{C}$ is a finite set. For $n > 2$, $\mathsf{Clifford}_n$ generates a dense subset of the full unitary group. A full characterization of subgroups included in $\mathcal{P}_n$ remains an interesting open problem.

Appendix B: Constant depth local circuits (proof of lemma 7)

Proof. Let us assume that the unitary U preserves the codespace and is implementable by a constant-depth local quantum circuit with the spread s_U. The proof proceeds by induction in the number of regions. Assuming $m = 0$, the dressed logical operator U is supported on a bare-cleanable region and by definition 2 must be a trivial logical operator in $\mathcal{P}_0$. This is true regardless of the spread s_U.

In order to prove the inductive step for $m + 1$ and spread s_U, assume that our statement is true up to m for any spread (and in particular for $2s_U$). Consider a unitary U with the spread s_U, such that

$$\mathrm{supp}(U) \subseteq \bigcup_{j=0}^{m+1} R_j. \tag{B1}$$

Any logical Pauli operator $[P]_L$ has a dressed realization P fully supported on $\overline{R_1^+}$. Observe that

$$\mathrm{supp}(UPU^\dagger P^\dagger) \subseteq \left[\bigcup_{j=0}^{m+1} R_j\right] \cap \mathfrak{B}(\overline{R_1^+}, s_U) \tag{B2}$$

$$\subseteq R_0 \cup \bigcup_{j=2}^{m+1} R_j. \tag{B3}$$

The last expression has the form required by the assumption of lemma 7. However, regions $R_2, \ldots, R_{m+1}$ play the role of regions $R_1, \ldots, R_m$ and $2s_U \geq s_{UPU^\dagger P^\dagger}$ plays the role of s_U. Hence, by inductive hypothesis, $[UPU^\dagger P^\dagger]_L$, which is also a dressed logical operator, must belong to $\mathcal{P}_m$ when restricted to the codespace. Thus, by definition of the Clifford hierarchy, $[U]_L \in \mathcal{P}_{m+1}$. $\square$

Appendix C: Operator algebraic definition of bare and dressed logical operators

One of the motivations for introducing subsystem codes is to have a larger flexibility when seeking a realization of the code as the ground space of some local Hamiltonian. In this setting, operators belonging to the algebra generated by the gauge group $\mathcal{G}$ can be included in the Hamiltonian while maintaining the stabilized subspace as an invariant subspace. Which subspaces stabilized by the stabilizer group will depend on microscopic details of the strength of the terms included in the Hamiltonian. In quantum error correcting codes, the convention is to choose a sign consistent stabilizer code such that the code space $\mathcal{C}(\mathcal{S})$ corresponds to the common $+1$ eigenspace of the stabilizer group $\mathcal{S}$.

Logical operators preserve the codespace $\mathcal{C}(\mathcal{S})$ with associated projector P and may act non-trivially on logical qubits. Existing definitions of bare and dressed logical operators [15] rely on the centralizer group $C(\mathcal{G})$ with respect to the Pauli group, thus restricting logical operators to $\mathcal{P}_1$. Here, we provide a natural extension of these definitions, which admits logical operators beyond Paulis and may potentially be applicable beyond the framework of Pauli subsystem codes.

Definition 4. *An operator B is a bare logical operator of the subsystem code defined by the gauge group $\mathcal{G}$ and code space associated to the projector P if and only if*

$$[B, P] = 0 \quad \text{and} \quad [U, G]P = 0 \quad \forall G \in \mathcal{G}. \tag{C1}$$

Dressed logical operators may act non-trivially on both logical and gauge qubits. There exist unitary operators that preserve the stabilized subspace $\mathcal{C}(\mathcal{S})$, but do not have tensor product structure with respect to $\mathcal{H}_{\mathrm{logical}} \otimes \mathcal{H}_{\mathrm{gauge}}$. The action of such unitary operators on the logical qubits is characterized by its dependence on the state of the gauge qubits, which we wish to exclude. Such a dependence, would violate the premise of subsystem codes by which the state of gauge qubits is irrelevant. This may be formalized by demanding that the application of a dressed logical unitary commute with twirling of the unitary gauge group (which includes depolarization of gauge qubits). This can be specified as follows:

Definition 5. *An operator U is a dressed logical unitary on a subsystem code defined by the gauge group $\mathcal{G}$ and the associated stabilizer subgroup $\mathcal{S}$ if and only if for all $\rho = P_{\mathcal{C}(\mathcal{S})}\rho P_{\mathcal{C}(\mathcal{S})}$,*

$$\Delta_{\mathcal{G}}(U\rho U^\dagger) = U\Delta_{\mathcal{G}}(\rho)U^\dagger, \tag{C2}$$

where $\Delta_{\mathcal{G}}(\rho) = \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} G\rho G^\dagger$ *is the twirling with respect to the gauge group.*

In the case of a full Pauli algebra, twirling is known to have the effect of fully depolarizing the qubits associated to the algebra. In the case of an incomplete Pauli Algebra, the resulting twirl gives rise to dephasing of the corresponding qubits. Intuitively, the definition imposes that tracing out gauge qubits does not affect the action on logical qubits.

The bare and dressed logical operators beyond the Pauli group are indeed algebraically well defined. For instance, the set of all dressed logical unitaries form a closed group under multiplication. Furthermore, the set of bare logical operators is preserved under conjugation by dressed logical operators:

Lemma 9. *Let U_d be a dressed logical unitary and U_b be a bare logical operator for a subsystem code. Then $U_d U_b U_d^\dagger$ is also a bare logical operator.*

Proof. Both U_d and U_b preserve the codespace $P_{\mathcal{C}(\mathcal{S})}$ and so does their product. We will now prove that $U_d U_b U_d^\dagger$ commutes with any gauge operator $G_0 \in \mathcal{G}$ restricted to $P_{\mathcal{C}(\mathcal{S})}$.

$$\begin{aligned}
U_d U_b U_d^\dagger G_0 P_{\mathcal{C}(\mathcal{S})} &= U_d \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} G G^\dagger U_b P_{\mathcal{C}(\mathcal{S})} U_d^\dagger G_0 \\
&= \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} U_d G U_b P_{\mathcal{C}(\mathcal{S})} G^\dagger U_d^\dagger G_0 \\
&= \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} G U_d U_b P_{\mathcal{C}(\mathcal{S})} U_d^\dagger G^\dagger G_0 \\
&= \frac{1}{|\mathcal{G}|}\sum_{G\in\mathcal{G}} G_0 G U_d U_b P_{\mathcal{C}(\mathcal{S})} U_d^\dagger G^\dagger
\end{aligned}$$

The first step consists of simply multiplying by an identity. Then, the commutation of $G^\dagger$ with U_b on the subspace $P_{\mathcal{C}(\mathcal{S})}$ is used. The main equation of definition for a dressed logical operator 5 is then applied. In order to relabel the sum we recall that $G_0 G \in \mathcal{G}$. Finally, tracking back the previous steps one may recover the expression $G_0 U_d U_b U_d^\dagger P_{\mathcal{C}(\mathcal{S})}$, which concludes the proof. □

Lemma 1 allows us to formally prove that dressed logical operators may transform logical qubits in a way independent of the state of gauge qubits. This is the content of lemma 10 which represents the tensor factorization of dressed logical operators with respect to gauge and logical qubits.

Lemma 10. *Let $|\psi\rangle$ and $|\psi'\rangle$ be two arbitrary states in the codespace $\mathcal{P}_{C(\mathcal{S})}$, such that $\langle\psi|U_b|\psi\rangle = \langle\psi'|U_b|\psi'\rangle$ for all the bare logical operators U_b. Then, for any dressed logical operator U_d, one has $\langle\psi|U_d U_b U_d^\dagger|\psi\rangle = \langle\psi'|U_d U_b U_d^\dagger|\psi'\rangle$ for all U_b.*

Proof. Lemma 1 implies that $U_b' = U_d U_b U_d^\dagger$ is a bare logical operator. Then

$$\begin{aligned}
\langle\psi|U_d U_b U_d^\dagger|\psi\rangle &= \langle\psi|U_b'|\psi\rangle \\
&= \langle\psi'|U_b'|\psi'\rangle \\
&= \langle\psi'|U_d U_b U_d^\dagger|\psi'\rangle.
\end{aligned}$$

□

[1] P. W. Shor, in *Proceedings of the 37th Annual Symposium on Foundations of Computer Science (FOCS)* (IEEE Computer Society, Los Alamitos, CA, 1996), p. 56.
[2] J. Preskill, Proc. Roy. Soc. Lond. **454**, 385 (1998).
[3] B. Eastin and E. Knill, Phys. Rev. Lett. **102**, 110502 (2009).
[4] S. Bravyi and R. König, Phys. Rev. Lett. **110**, 170503 (2013).
[5] D. Gottesman and I. L. Chuang, Nature **402**, 390 (1999), 0906.1579v1.
[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, 2000).
[7] D. Gottesman, Phys. Rev. A **57**, 127 (1998).
[8] S. Bravyi, Phys. Rev. A **73**, 042313 (2006).
[9] S. Bravyi and B. Terhal, New. J. Phys. **11**, 043029 (2009).
[10] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).
[11] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **98**, 160502 (2007).
[12] H. Bombin and M. A. Martin-Delgado, Phys. Rev. B **75**, 075103 (2007).
[13] H. Bombin, arXiv:1311.0879.
[14] S. Bravyi, D. Poulin, and B. Terhal, Phys. Rev. Lett. **104**, 050503 (2010).
[15] S. Bravyi, Phys. Rev. A **83**, 012320 (2011).
[16] D. Gottesman, Phys. Rev. A **54**, 1862 (1996).
[17] H. Bombin and M. A. Martin-Delgado, Journal of Physics A: Mathematical and General **42**, 095302 (2009).
[18] D. Poulin, Phys. Rev. Lett. **95**, 230504 (2005).
[19] D. W. Kribs, R. Laflamme, D. Poulin, and M. Lesosky, Quant. Inf. Comp. **6**, 383 (2006).
[20] M. A. Nielsen and D. Poulin, Phys. Rev. A **75**, 064304 (2007).
[21] D. Bacon, Phys. Rev. A **73**, 012340 (2006).
[22] B. Yoshida and I. L. Chuang, Phys. Rev. A **81**, 052302 (2010).
[23] T. M. Stace, S. D. Barrett, and A. C. Doherty, Phys. Rev. Lett. **102**, 200501 (2009).
[24] J. Dorier, F. Becca, and F. Mila, Physical Review B **72**, 024448 (2005).
[25] A. A. Kovalev and L. P. Pryadko, Phys. Rev. A **87**,

020304 (2013).
[26] F. Pastawski, A. Kay, N. Schuch, and I. Cirac, Quantum Inf. Comput. **10**, 580 (2010).
[27] M. B. Hastings, Phys. Rev. Lett. **107**, 210501 (2011).
[28] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, J. Math. Phys. **43**, 4452 (2002).
[29] B. Yoshida, Ann. Phys. **326**, 2566 (2011).
[30] J. Haah and J. Preskill, arXiv:1011.3529.
[31] O. Landon-Cardinal and D. Poulin, Phys. Rev. Lett. **110**, 090502 (2013).
[32] J. Haah, Phys. Rev. A **83**, 042330 (2011).
[33] S. Bravyi and J. Haah, Phys. Rev. Lett. **111**, 200501 (2013).
[34] I. H. Kim, arXiv:1202.0052.
[35] B. Yoshida, Phys. Rev. B **88**, 125122 (2013).
[36] K. Michnicki, arXiv:1208.3496.
[37] H. Bombin, R. W. Chhajlany, M. Horodecki, and M. A. Martin-Delgado, New. J. Phys. **15**, 055023 (2013).
[38] F. Pastawski, L. Clemente, and J. I. Cirac, Phys. Rev. A **83**, 012304 (2011).
[39] X. Chen, Z.-C. Gu, and X.-G. Wen, Phys. Rev. B **82**, 155138 (2010).
[40] S. Bravyi and A. Kitaev, Phys. Rev. A **71**, 022316 (2005).
[41] A. Paetznick and B. W. Reichardt, Phys. Rev. Lett. **111**, 090505 (2013).
[42] T. Jochym-O'Connor and R. Laflamme, Phys. Rev. Lett. **112**, 010505 (2014).
[43] B. Yoshida, arXiv:1404.0457.
[44] S. Bravyi, B. Leemhuis, and B. M. Terhal, Ann. Phys. **326**, 839 (2011).
[45] S. A. Ocko, X. Chen, B. Zeng, B. Yoshida, Z. Ji, M. B. Ruskai, and I. L. Chuang, Phys. Rev. Lett. **106**, 110501 (2011).
[46] M. E. Beverland, R. König, F. Pastawski, J. Preskill, S. Sijher, arXiv:1409.3898 (2014).
[47] If two copies of a CSS code are stacked such that corresponding qubits are geometrically close, performing pairwise CNOT on all physical qubits implements a CNOT gates on all pairs of encoded qubits in the two copies.
[48] We say that a circuit is geometrically local if it is composed of elementary gates acting only on neighboring qubits up to some radius ξ.
[49] Formally, erasure errors are modeled by extending the space associated to each qubit with one additional state $|l\rangle$ which indicates erasure of the corresponding qubit. An error-correcting recovery map for erasure errors may mimic the one for depolarizing noise by mapping all particles marked as $|l\rangle$ to the fixed-point of corresponding depolarizing channel. Hence, the erasure threshold p_l must necessarily be no smaller than any depolarization threshold $p_l \geq p_e$.
[50] Models proposed in [32, 36] have a macroscopic energy barrier, yet quantum memory time is upper bounded by constant, which is perhaps due to topological transition temperature being zero. Finite transition temperature is not sufficient to guarantee exponentially growing quantum memory time [43]
[51] The absence of string-like logical operators does not necessarily imply the presence of macroscopic energy barrier [44].
[52] Due to the non-commutativity of G_j, a ground state $|\psi\rangle$ of the Hamiltonian $H = \sum_j \alpha_j G_j$ is not necessarily inside the stabilized subspace. Indeed it is analytically and computationally difficult to find values of S_j in the ground space. For CSS subsystem codes, the ground space of $H = -\sum_j G_j$ is guaranteed to be in the stabilized subspace defined by $\mathcal{S}$ due to the Perron-Frobenius theorem [45].