Recoverable information and emergent conservation laws in fracton stabilizer codes

A. T. Schmitz, Han Ma, Rahul M. Nandkishore, and S. A. Parameswaran

Phys. Rev. B **97**, 134426 — Published 26 April 2018

DOI: 10.1103/PhysRevB.97.134426

Recoverable Information and Emergent Conservation Laws in Fracton Stabilizer Codes

A.T. Schmitz,[1, *] Han Ma,[1, †] Rahul M. Nandkishore,[1, 2, ‡] and S. A. Parameswaran[3, §]

[1] *Department of Physics, University of Colorado at Boulder, Boulder CO 80309, USA*
[2] *Center for Theory of Quantum Matter, University of Colorado at Boulder, Boulder CO 80309, USA*
[3] *The Rudolf Peierls Centre for Theoretical Physics, University of Oxford, Oxford OX1 3NP, UK*

We introduce a new quantity, that we term *recoverable information*, defined for stabilizer Hamiltonians. For such models, the recoverable information provides a measure of the topological information, as well as a physical interpretation, which is complementary to topological entanglement entropy. We discuss three different ways to calculate the recoverable information, and prove their equivalence. To demonstrate its utility, we compute recoverable information for *fracton models* using all three methods where appropriate. From the recoverable information, we deduce the existence of emergent Z_2 Gauss-law type constraints, which in turn imply emergent Z_2 conservation laws for point-like quasiparticle excitations of an underlying topologically ordered phase.

CONTENTS

I. INTRODUCTION

Ideas from quantum information have provided powerful tools for the study of complex quantum many-body systems and the myriad phases of matter they host. Quantum entanglement has been a uniquely influential concept in this regard — specifically, its characterization in terms of entanglement entropy. While discussions of entanglement entropy pervade many areas of modern theoretical physics, it has proven particularly potent in the characterization of topological order (see, e.g., Ref. [1] for a review). For two-dimensional gapped topologically ordered phases, the scaling of the entanglement entropy of a subsystem with its size contains a universal subleading 'constant' term [2, 3]. This is intimately related to the topological quantum field theory (TQFT) description of the topological phase and provides a partial characterization of the nature of topological order. However, while two dimensional topological orders are relatively well understood, the book of topological order in higher dimensions is still being written.

Indeed, the discovery of three-dimensional *fracton* models [4–11] has begun a new chapter in this book. These models exhibit many of the characteristic properties of topological order, such as locally indistinguishable degenerate ground states on manifolds of non-trivial topology, and local excitations that cannot be created by purely local operators. However, unlike more familiar topologically ordered states, the point-like excitations of these models are restricted to move only within lower-dimensional subspaces such as along a line or in a plane. This leads to a sub-extensive ground state degeneracy on manifolds of nontrivial topology, rather than the *finite* ground state degeneracy expected of topological orders described by emergent discrete gauge theories. As such, fracton models have begun to draw intensive interest [9, 12–24]. Given the insights previously gleaned

[*] albert.schmitz@colorado.edu
[†] han.ma@colorado.edu
[‡] rahul.nandkishore@colorado.edu
[§] sid.parameswaran@physics.ox.ac.uk

into topological order by the consideration of topological entanglement, an investigation of the topological entanglement of fracton models is both timely and interesting.

Two recent works have made important progress in this direction. Ref. 25 places a linear-scaling bound on the dependence of the non-local entanglement on subsystem size in several fracton models, thereby establishing that the fracton models considered do possess non-local entanglement. Meanwhile in a recent paper [26] we have explicitly computed the topological entanglement entropy of the same archetypal fracton models presented here. Our results are consistent with the bounds of Ref. 25, and match those from a distinct approach that uses a tensor-network representation of the ground-state fracton wavefunction [27]. Given the absence of a TQFT description of fracton models, however, it is natural to ask what *physical* interpretation may be ascribed to these results. Here, we attempt to address this question in the restricted setting of a special class of *stabilizer* Hamiltonians built from commuting sets of Pauli operators. For such models, we introduce a concept that we dub the 'recoverable information', that is closely related yet distinct from the topological entanglement entropy. For a bipartition (A, B) of a system described by an eigenstate of a stabilizer Hamiltonian, the recoverable information is given by

$$\mu = \min_{\mathcal{B}} \left[d_{\partial A}(\mathcal{B}) - S_A - S_B \right], \tag{1}$$

where $S_{A(B)}$ is the entanglement entropy of $A(B)$ and $d_{\partial A}(\mathcal{B})$ is a characterization of the number of stabilizers 'cut' by the entanglement surface ∂A for a choice of basis $\mathcal{B}$ for the group of all stabilizers. The final computation of μ entails minimization over all such basis choices. The precise definition of a stabilizer basis and how this minimization is accomplished are elaborated below.

To understand the relationship between the recoverable information and the topological entanglement entropy, it is useful to recall two distinct interpretations that may be given to the topological entanglement entropy in $d = 2$. Recall that the topological entanglement entropy is a negative quantity, indicating that there is *less* information lost than one would naïvely expect based solely on the area law for gapped ground states, with no further assumptions. When a TQFT description exists and the topological phase is non-chiral, it is possible to relate the extra information to the geometric structure of the ground state wavefunction — e.g. for the toric code/Z_2 gauge theory, this reflects the closed-loop constraint implied by the Gauss law for the Z_2 gauge field. Where such a description is absent, clearly we cannot immediately ascribe this additional 'topological' information to a physical conservation law. However, by leveraging the special structure of stabilizer Hamiltonians, we show here that we can link the recoverable information to the existence of a set of independent Z_2 Gauss law-type constraints that must be satisfied with respect to the entanglement cut, permitting us in some cases to deduce that the phase is described by an underlying generalized 'gauge' theory. It is this interpretation that we formalize in this paper.

Stabilizer codes were first introduced into quantum information theory in the context of quantum error correction and quantum memory [28]. A stabilizer code is defined as a subspace of states stabilized, i.e. left invariant, by a set of mutually-commuting Pauli operators —also known as the *stabilizer set* — acting in some tensor product Hilbert space. Clearly, this is also the ground state manifold ('ground space') of the associated stabilizer Hamiltonian given by the negative sum over all members of the stabilizer set. If knowledge of the eigenvalues of the stabilizer set does not suffice to specify a complete basis for the Hilbert space, then there exists some complementary set of *logical operators*; these form a Pauli algebra in the stabilizer code, where quantum information can be stored and processed. A logical operator will generally involve a large number of single-qubit Pauli operators, typically as many as characterized by the *code distance*, that in the ideal scenario for error correction, grows with system size. The resulting non-locality of the information stored via logical operators means that it is robust to errors, making quantum error correction feasible. The special structure of stabilizer codes and the associated mathematical machinery, reviewed in part below, are central in our ability to define and understand recoverable information.

At this point, it is worth remarking that, within the mathematical machinery used to study stabilizer codes, the choice of the stabilizer basis is a key *physical* input. A spatially local basis choice is precisely what makes stabilizer codes useful in a physical setting, since most reasonable perturbations of an actual system will be spatially local. The information stored in logical operators is hence robust against such physical perturbations. Our choice of basis thus implicitly gives locality a concrete meaning. It will be apparent in what follows that this restriction to a spatially local basis is also what makes the recoverable information a non-zero quantity, since it restricts the possible allowed measurements on a system. We will henceforth assume a spatially local basis (suitable defined on a lattice) unless otherwise specified.

The remainder of this paper is organized as follows. First, we review some basic facts about stabilizer codes and the models we consider here, and develop some additional machinery that we require later. We then motivate and define recoverable information and relevant concepts. This leads to the central result of this paper, namely that recoverable information counts the number of independent *non-local surface stabilizers* (NLSS). These are the generalizations of the non-contractible loop operators generated in the toric code by the removal of a bounded subsystem. We also show that the NLSSs allow us to identify a set of generalized Z_2 Gauss law relations satisfied by the entanglement cut, and obtain a lower bound (tight for some models) for recoverable information. We conclude by calculating the recoverable information of different stabilizer models using three methods: (i) by using the definition; (ii) when tight, the lower bound; and (iii) by counting NLSSs.

II. STABILIZER HAMILTONIANS AND FRACTON MODELS

We begin this section by reviewing some basic results on stabilizer codes and the particular models we study to set the stage for the discussion, before deriving several new results that will be relevant in the remainder. In certain cases, we will draw upon results from Ref. 29, and in those cases we will often only provide details telegraphically or omit them entirely.

A. Models and Hilbert Space Structure

Each of the models we study here consists of a set of N qubits (i.e., spin-1/2 variables) arranged on the edges or vertices of a simple graph, with q qubits per edge or vertex. We denote the graph set (vertices or edges) as V, so as to have $N = q|V|$. We use $|\dots|$ to denote the size of a set. The Hilbert space $\mathcal{H} = (\mathbb{C}^2)^{\otimes N}$ is the product space of all the qubits. We will consider graphs with either periodic or open boundary conditions.

We define the *Pauli group* P to be the set of all Pauli operators (that we denote $\{X, Y, Z\}$) of the qubits modulo any phase of $\pm1, \pm i$. P can be mapped to a vector space $V[P]$ over the field $\mathbb{F}_2$. Addition in $V[P]$ corresponds to multiplication of distinct Pauli operators, and scalar multiplication in $V[P]$ (modulo 2, as appropriate in $\mathbb{F}_2$) is the power of a Pauli operator; the choice of field should now be obvious since every Pauli operator squares to the identity. For example, in a system of three qubits, $X_1Y_3 \leftrightarrow \begin{pmatrix} 1 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$, $Z_2Z_3 \leftrightarrow \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix}$, and their product $X_1Z_2X_3 \leftrightarrow \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ is given by adding the corresponding $V[P]$-vectors. Additionally, commutation relations can be encoded by use of a symplectic form, but as we will not use this for what follows, we do not discuss it further. Since the set of single-qubit X- and Z-type Pauli operators generate all of P, we conclude that the dimension of $V[P]$ is $2N$, and that of the Pauli group P itself is 2^{2N}. It is useful to define the *support* of a Pauli operator $p \in P$ as the set of vertices (edges) upon which it acts nontrivially:

$$\mathrm{Supp}(p) = \{v \in V : p_v \neq I_v\}, \tag{2}$$

where $I_v = (I_2)^{\otimes q}$ is the identity operator on the q qubits on vertex (edge) v. Note that we have implicitly defined a map p_v from the set of vertices (edges) to the Pauli group. It is natural to likewise define the support of a set $F \subseteq P$ of Pauli operators, via $\mathrm{Supp}(F) = \bigcup_{f\in F} \mathrm{Supp}(f)$.

We identify a subset $\mathcal{S} \subset P$ of the Pauli group as the *stabilizer set*. $\mathcal{S}$ satisfies the following properties:

- $[\mathcal{S}, \mathcal{S}] = \{0\}$, i.e. it is made of commuting operators
- $|\mathcal{S}| \geq N$, i.e. there are at least as many stabilizers as there are qubits in the system
- $-I_{\mathcal{H}} \notin \mathcal{S}$, i.e. the negative identity is not in $\mathcal{S}$, so that all of $\mathcal{S}$ can have a positive eigenvalue
- $\mathrm{Supp}(\mathcal{S}) = V$, i.e. every vertex (edge) is acted upon nontrivially by at least one stabilizer in $\mathcal{S}$.

The Hamiltonian is then taken to be the negative sum of all stabilizers:

$$H = -\sum_{s\in\mathcal{S}} s. \tag{3}$$

For specificity, we now list the particular examples of stabilizer Hamiltonians that we focus on in this work. Each is defined on a Euclidean lattice of linear size L. While our central focus is on fracton models, for pedagogical reasons, we will also discuss a few more conventional models, both with and without topological order, that serve to illustrate various properties of recoverable information as well as various considerations necessary for its proper definition. The first of these is the cluster model. On a d-dimensional hypercubic lattice with a single qubit per vertex, this has the Hamiltonian

$$H_{\mathrm{clu}_d} = -\sum_v K_v, \tag{4}$$

where K_v is the product of the single Z-type operator at v and all the X-type operators surrounding v. (there are $2d$ such X operators on a d-dimensional hypercubic lattice). Each stabilizer thus consists of $2d$ X-type operators 'clustered' around a single Z-type operator, hence the name. On a d-dimensional hypertorus, the (unique) ground state of this model is not topologically ordered, but is instead either trivial or else in the symmetry-protected topological (SPT) phase [30–35].

We also discuss two simple instances of topological order, namely the $d = 2$ and 3 toric codes. These consist of one qubit per edge of the square or cubic lattice, and are described by the Hamiltonian

$$H_{\mathrm{TC}_d} = -\sum_v A_v - \sum_p B_p, \tag{5}$$

where B_p is the product of the four Z-type operators surrounding the plaquette p, and A_v is the product of the $2d$ X-type operators connected to vertex v. The entanglement entropy of the $d = 3$ toric code was studied in Ref. 36.

The two fracton models we consider are the 'X-cube' model [8, 9] and 'Haah's code.' [6] The X-cube model is defined on a cubic lattice with a qubit on each edge, with Hamiltonian

$$H_{\mathrm{XC}} = -\sum_v \left(A_v^{(xy)} + A_v^{(yz)} + A_v^{(zx)}\right) - \sum_c B_c, \tag{6}$$

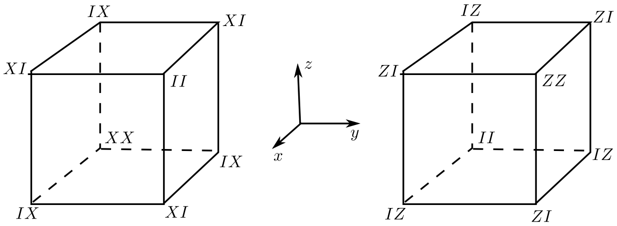


FIG. 1. G_c^X and G_c^Z terms in Haah's code. Each site has two spins. X and Z denote the corresponding Pauli operators. I represents the identity operator.

where $A_v^{(\mu)}$ is the product of the four Z-type operators that surround vertex v in one of three mutually orthogonal planes (labeled by their normal direction μ), and B_c is the product of the twelve X-type operators surrounding the cube c. Haah's code is also defined on a cubic lattice, but now with two qubits on every vertex. The Z- and X-type stabilizer operators now consist of products of Z and X Pauli matrices around a cube with a Hamiltonian of the form

$$H_{\text{Haah}} = -\sum_c (G_c^Z + G_c^X), \tag{7}$$

where $G_c^{X(Z)}$ denotes the product of the Pauli operators specified in Fig. 1 for cube c. While much of the following discussion will deal with the abstract structure of stabilizer codes, these examples are worth keeping in mind as the particular models of interest, to which we will eventually turn. Note that we do not discuss many of the striking properties of these fracton models in this work; we refer the interested reader to Refs. 4–11 for more details.

B. Stabilizer Group and Constraint Subspaces

In this subsection, we review some properties of stabilizer groups and their constraints. While much of this will be familiar to readers acquainted with stabilizer codes, the next subsections develop technical details relevant to the discussion in Section III and so may be useful in the balance of the paper.

The stabilizer set, $\mathcal{S}$, naturally generates the stabilizer group $G = \{\prod_{s\in F} s : F \in \mathbb{P}[\mathcal{S}]\}$, where the power set of $\mathcal{S}$, denoted $\mathbb{P}[\mathcal{S}]$, is the set of all subsets of $\mathcal{S}$. In other words, the stabilizer group is the group generated by taking products of Pauli operators in $\mathcal{S}$. (Note the distinction between the stabilizer set $\mathcal{S}$ and the stabilizer group G — the latter is much larger than the former.) We also impose an additional requirement of the stabilizer set, that, while not necessary from the perspective of error correcting codes and quantum information, is reasonable from a physical standpoint. Namely, we require that if a bijective mapping of the qubits into themselves induces an automorphism on G, then the same mapping induces an automorphism on $\mathcal{S}$ consistent with the automorphism on G. In simpler terms, if the group has some symmetry, then we require that the stabilizer set (and hence, the Hamiltonian) has that same symmetry. Besides being physically reasonable, this has the consequence of imposing certain contraints in the stabilizer set, as we will discuss shortly.

Now, G is isomorphic to a subspace $V[G] \subseteq V[P]$. This follows, again, from the prescription used to define G, and is related naturally to the commutative ring formalism [29] used in Section IV D. We will denote the dimension of the vector space $V[G]$ as d_G, so that the size of the stabilizer group is $|G| = 2^{d_G}$ (Note that we sometimes use the shorthand $\dim G$; this should be understood as the dimension of the vector space generated by group G, rather than the size of the group itself. As noted previously, we reserve the symbol $|\ldots|$ for the size of a group or a set).

The definition of G implicitly defines a map $\phi : \mathbb{P}[\mathcal{S}] \to V[G]$, that takes an element from the power set into G. Let us define a basis, $\mathcal{B} \in \mathbb{P}[\mathcal{S}]$, as a minimal generating set for G. We emphasize that this basis is a subset of $\mathcal{S}$ and does not contain any non-local stabilizers. Now, since $\mathcal{B} \in \mathbb{P}[\mathcal{S}]$ generates all of G and is minimal, it follows from the definitions above that $\phi(\mathbb{P}[\mathcal{B}])$, the image of the power set of $\mathcal{B}$ under ϕ, is all of the vector space $V[G]$, and thus $\mathcal{B}$ maps onto a basis for $V[G]$. It then follows that $|\mathcal{B}| = d_G$. We define a *constraint* C on the stabilizer group as a subset of $\mathcal{S}$, $C \in \mathbb{P}[\mathcal{S}]$, such that $\prod_{s\in C} s = I_{\mathcal{H}}$. If $d_G < |\mathcal{S}|$, then there must exist $|\mathcal{S}| - d_G$ *independent* constraints in $\mathcal{S}$. To define independence, we require a vector space structure. Let $W[\mathcal{S}]$ be an *arbitrary* vector space over $\mathbb{F}_2$ with dimension $|\mathcal{S}|$, and hence has $|W| = 2^{|\mathcal{S}|}$ distinct elements. For concreteness, we may take $W[\mathcal{S}] = \{0,1\}^{|\mathcal{S}|}$ where addition is given by bitwise addition modulo 2 ($\oplus$). As this is the same size as $\mathbb{P}[\mathcal{S}]$, we can construct a bijection $f : W[\mathcal{S}] \to \mathbb{P}[\mathcal{S}]$, with inverse $f^{-1} : \mathbb{P}(\mathcal{S}) \to W[\mathcal{S}]$. In other words,

$$W[S] \underset{f^{-1}}{\overset{f}{\rightleftharpoons}} \mathbb{P}[\mathcal{S}] \xrightarrow{\phi} V[G]. \tag{8}$$

Therefore, $\phi \circ f : W[\mathcal{S}] \to V[G]$ is a surjective linear map between vector spaces. Since f is bijective, it follows that the pre-image under f of the kernel of ϕ is itself the kernel of $\phi \circ f$, i.e. $\ker(\phi \circ f) = f^{-1}(\ker(\phi))$. Now, $\ker(\phi)$ contains all sets of stabilizers that map to the identity in $V[G]$, i.e. every element of $\ker(\phi)$ is a constraint. We may now identify the set of constraints as $\mathcal{C} = \ker(\phi)$. Since $f^{-1}(\mathcal{C}) = \ker(\phi \circ f)$ is the kernel of a linear map between vector spaces, it must be a subspace of $W[\mathcal{S}]$ and therefore is itself a vector space with its own basis and dimension. So we define a set of constraints to be independent if the preimage under f of this set is independent in $W[\mathcal{S}]$. Similarly, we can identify a basis for the constraint set $\mathcal{C}$ as the image of a basis for $\ker(\phi \circ f)$ under f. Since the orthogonal compliment of $\ker(\phi \circ f)$ consists of every member of $W[\mathcal{S}]$ which maps injectively onto $V[G]$, we then have $\dim \mathcal{C} =$

$\dim[\ker(\phi\circ f)] = \dim W[\mathcal{S}] - \dim V[G] = |\mathcal{S}| - d_G$ as we asserted earlier. We will henceforth refer to $\mathcal{C}$ as the 'constraint subspace', with the understanding that we refer to this structure.

C. Logical Pauli Operators

We next identify a special class of *logical* Pauli operators. If a Pauli operator p anti-commutes with any member of $\mathcal{S}$, it creates an excitation above the ground state of the stabilizer Hamiltonian (3). This implies that $\mathcal{P}^{(\mathbf{0})} p \mathcal{P}^{(\mathbf{0})} = 0$, where $\mathcal{P}^{(\mathbf{0})}$ is a projector onto the ground space. If instead p commutes with all members of $\mathcal{S}$, then p can be separated as $p = p_\ell p_G$ where $p_G \in V[G]$ and $p_\ell \in V^\perp[G]$, where $V^\perp[G]$ is the complement of $V[G]$ in $V[P]$. If in this decomposition we have $p_\ell = I_\mathcal{H}$, then $p \in V[G]$ and we actually have $\mathcal{P}^{(\mathbf{0})} p \mathcal{P}^{(\mathbf{0})} = \mathcal{P}^{(\mathbf{0})}$. This follows since the ground space (the stabilizer code) is stabilized by all of G. So the set of logical Pauli operators, that we denote P_ℓ, is the maximal set such that

- $[\mathcal{S}, P_\ell] = \{0\}$, and
- $P_\ell \subseteq V^\perp[G]$.

Given that for any p *not* in P_ℓ, $\mathcal{P}^{(\mathbf{0})} p \mathcal{P}^{(\mathbf{0})} \propto \mathcal{P}^{(\mathbf{0})}$, it follows that only logical operators can distinguish between different sectors of the ground state manifold, as any other operator acts trivially when restricted to that manifold. This is equivalent to the statement that the quantum information contained in the stabilizer code is difficult to modify without performing non-local operations, and is the origin of their topological robustness. Furthermore, a maximal set of mutually commuting, independent operators in P_ℓ generates the *logical Pauli group*, G_ℓ. It then follows from the above definitions that G_ℓ is isomorphic to a vector space $V[G_\ell]$ over $\mathbb{F}_2$. Since the total number of mutually commuting, independent operators cannot exceed N, we may show (see Ref. 29 for a proof) that $d_\ell \equiv \dim(G_\ell) = N - d_G$.

The logical Pauli operators define the *code distance* $D = \min\{d \in \mathbb{Z} : \mathrm{Supp}(p) \subseteq B_v(d) \text{ for some } v \in V \text{ and } p \in P_\ell\}$. Here $B_v(d)$ is a ball about vertex v of radius d using some appropriate distance function on the graph. For example, we could use a Euclidean distance for some embedding of the graph, or the actual graph distance. The choice of distance function must be tuned to the objective. For our purposes, we wish to relate logical operators to topological information, i.e. non-local correlation, and so we use a Euclidean distance function as it has the most natural link to spatial locality. For quantum error correction on the other hand, we might instead restrict the distance function to simply count the smallest number of bit flips needed to change the value of a logical operator (i.e. the size of the support). For all systems considered here and either distance function, we have $D \sim L$, the linear size of the system. The code distance is the size of the smallest logical Pauli operator, and plays an important role in providing a topological perspective on stabilizer codes, as it gives a sense of which operators are local. To wit, a subset $A \subseteq V$ is *bounded* iff there exists a vertex v such that $A \subseteq B_v(D)$, and an operator is bounded iff its support is bounded. A set of operators is *local* iff every member is bounded. Since no logical operator is bounded, and these are the only ones that can distinguish between distinct ground states, this shows that the ground states are indistinguishable to local operators. This indistinguishability is what makes information encoded in the ground space robust against decoherence, since the sources of decoherence are generically local perturbations that cannot 'measure' an eigenstate and are hence unable to decohere it.

The Pauli group has another important property provided by the *cleaning lemma* [29, 37–40]. This states that for any bounded subset of the space, A, any member of $p_\ell \in P_\ell$ can be 'cleaned' out of A. That is, there exists an element $g \in G$, such that gp is not supported in A. For the example of the toric code on a torus, this corresponds to string-like logical operators which encircle any one direction and can be deformed around any A, so long as A itself does not wrap the torus in any direction. The cleaning lemma is actually far more general [29] than the statement above, but this is sufficient for our purposes.

D. Topological Constraints and Boundary Conditions

With the notion of locality defined as above, we may now apply these ideas to the constraint subspace $\mathcal{C}$, and link it with the logical Pauli operators. If $d_G < N$, then from our definition $d_\ell > 0$. Furthermore, since $|\mathcal{S}| \geq N$ we have $\dim \mathcal{C} \geq d_\ell > 0$, i.e. the set of constraints is non-trivial.

A set of constraints $\mathcal{T} \subseteq \mathcal{C}$ is *topological* if it satisfies the following conditions:

- $\mathcal{T}$ is a subspace of $\mathcal{C}$ (as understood in $W[S]$); and
- No nonempty member of $\mathcal{T}$ remains a constraint if we go from periodic (pbc) to open (obc) boundary conditions.

This definition invokes the process of changing boundary conditions from periodic to open that we must make more concrete. We do so by first discussing the illuminating example of the toric code. With pbc, we have two constraints: the product of all stars and the product of all plaquettes always contain their respective type of single-qubit Pauli operator twice for every edge, and therefore must equal the identity. When changing boundary conditions, we retain the original Hilbert space, but modify the Hamiltonian. For stabilizer codes, this corresponds implicitly to changing the stabilizer set while preserving all the original qubits. Thus, for the toric code case, obc requires that we consider a planar graph with two 'ragged' edges and two 'smooth' edges (see Fig.

2); we must then consider carefully all stabilizers affected by this change in boundary conditions. The two ragged edges have complete star operators at each vertex, but incomplete, U-shaped plaquettes; nevertheless, these all commute with each other and with the complete stabilizers in the interior of the graph. Similarly, the smooth edges have complete plaquettes, but T-shaped 'stars' that are all mutually commuting. While the *total* number of stabilizers is unchanged, both constraints are lost, since some of the Pauli operators on the edges only appear once when taking the product of all the star or plaquette stabilizers. Simultaneously, we observe that the number of stabilizers is equal to the number of qubits so that there can be no logical operators. Instead, the product of all stars (that equals the identity with pbc) is now an L-shaped string-like operator on the two perpendicular ragged edges; this is the descendant of what would be a logical operator with pbc but is included in the stabilizer group with obc. A similar statement holds for plaquettes. Thus, the change of boundary conditions provides a natural link between constraints and logical operators: every time a change in boundary conditions removes a constraint, a logical operator is transformed into a string operator included in the *modified* stabilizer group.

We may formalize this intuition by defining a *change of boundary conditions* to be a map from the original set of stabilizers $\mathcal{S}$ to an alternative stabilizer set $\mathcal{S}'$. $\mathcal{S}'$ satisfies all the conditions of a stabilizer set, but generates a group G' that is not equal to G. We also require that $|\mathcal{S}\cap\mathcal{S}'|\sim\mathcal{O}(L^d)$, i.e. their intersection is extensive (well-defined on a lattice). Finally, for every member $s'\in\mathcal{S}'$ there exists a unique member $s\in\mathcal{S}$ such that $\mathrm{Supp}_\mu(s')\subseteq\mathrm{Supp}_\mu(s)$, where we use the subscript μ to emphasize that we consider the support of the same Pauli operator in each case. Thus, every member of $\mathcal{S}$ has a corresponding member of $\mathcal{S}'$ which is either the same or has some "deleted" support. As a corollary to the last condition $|\mathcal{S}|=|\mathcal{S}'|$. If there exists a change of boundary conditions $\mathcal{S}\to\mathcal{S}'$ such that a subspace of $\mathcal{C}$ (the constraints in $\mathcal{S}$) is lost, then the change has removed topological constraints. The maximal set of constraints lost by a set of equivalent changes of boundary conditions (where equivalency is related by symmetry where it exists) is then the set $\mathcal{T}$ of topological constraints. Clearly $\dim(\mathcal{T})\leq d_\ell$ since every time the constraint subspace dimension is lowered by one, d_G increases by one until it is equal to N, at which point one has exactly enough independent stabilizers to specify a complete eigenbasis for $\mathcal{H}$. Since a change in boundary conditions maintains the same number of stabilizers, all other constraints beyond a number d_ℓ must necessarily remain unchanged, leading to the bound. Therefore, we conclude that the subspace of topological constraints is isomorphic to a subset of the logical Pauli group $\mathcal{T}\cong G_\mathcal{T}\subseteq G_\ell$.

It is worth noting that *typically* any topological constraint will 'wrap' all the way around the system (defined with periodic boundary conditions) and thus will have non-local support in the thermodynamic limit. This justifies an implicit assumption that we make below: namely that non-topological constraints are typically (though possibly not always) local constraints. Note that in all the examples studied in Section IV, this can be verified explicitly. In the remainder, we will assume that topological constraints are in one-to-one correspondence with logical operators.

E. Entanglement Entropy of Stabilizer Eigenstates

We now calculate the entanglement entropy of any eigenstate of the stabilizer Hamiltonian (i.e., *any* simultaneous eigenstate of all members of G and G_ℓ) for a bipartition (A,B) of the space $\mathcal{H}$. This derivation reviews and extends the results from Refs. 41–43; more details and explicit computations using this approach may be found in our previous work [26]. Let A be bounded as defined in Section II C. Also let our basis for G be $\mathcal{B}$ as before and $\mathcal{B}_\ell$ be a basis for G_ℓ. With the cleaning lemma, we can assume without loss of generality that the support of $\mathcal{B}_\ell$ is contained entirely in B.

As before, let $W[\mathcal{B}]=\{0,1\}^{d_G}$ so that elements $g\in G$ may then be labeled by a binary vector $\mathbf{n}\in W[\mathcal{B}]\simeq V[G]$, via

$$g(\mathbf{n})=s_1^{n_1}s_2^{n_2}\dots s_{d_G}^{n_{d_G}} \tag{9}$$

where $s_i\in\mathcal{B}$ and n_i is the i^{th} bit of the string $\mathbf{n}$. Note the ordering of Eq. (9) defines the isomorphism for $W[\mathcal{B}]\rightleftharpoons V[G]$. As this is isomorphic, we see that $(W[\mathcal{B}],\oplus)$ is a faithful representation of G. Furthermore, one can see that the irreps of G can be labeled by another string $\mathbf{k}\in W_{\mathrm{irrep}}[\mathcal{B}]=W[\mathcal{B}]$ and given by $\{(-1)^{\mathbf{k}\cdot\mathbf{n}}\}_\mathbf{n}$, where $\mathbf{k}\cdot\mathbf{n}$ is the binary dot product. This should be obvious as any Abelian group has 1D irreps and one can show that

$$(-1)^{\mathbf{k}\cdot(\mathbf{m}\oplus\mathbf{n})}=(-1)^{\mathbf{k}\cdot\mathbf{m}}(-1)^{\mathbf{k}\cdot\mathbf{n}}. \tag{10}$$

Thus they preserve the multiplication rules for $W[\mathcal{B}]$. It then follows from standard orthogonality relations that projection operators can be formed as

$$\mathcal{P}^{(\mathbf{k})}=\frac{1}{|G|}\sum_\mathbf{n}(-1)^{\mathbf{k}\cdot\mathbf{n}}g(\mathbf{n}). \tag{11}$$

As the stabilizer Hamiltonian is just a sum over group elements, it is easy to show that these operators project onto the various eigenvalue subspaces as labeled by the quantum number, $\mathbf{k}$. Clearly, this labels the stabilizer excitations and so $\mathbf{k}=0$ is the ground space i.e. the one with no excitations. To find the dimension of these subspaces, we can take the trace and use the fact that $\mathrm{tr}\,(g(\mathbf{n}))=N\delta_{\mathbf{n},0}$, to find that for any $\mathbf{k}$, the *topological* degeneracy is $N-d_G=d_\ell$, as expected. (Note that for a translationally-invariant code there are additional degeneracies among the various values of $\mathbf{k}$, hence our use of the term *topological* degeneracy to distinguish that linked to the logical Pauli operators). To obtain pure state projectors and hence the density matrix, we can construct analogous projection operators using $\mathcal{B}_\ell$,

$$\mathcal{P}_\ell^{(\mathbf{a})} = \frac{1}{|G_\ell|}\sum_{\mathbf{z}}(-1)^{\mathbf{a}\cdot\mathbf{z}}\mathfrak{g}(\mathbf{z}), \tag{12}$$

where $\mathbf{a},\mathbf{z} \in W[\mathcal{B}_\ell]$ and $\mathfrak{g}(\mathbf{z}) \in G_\ell$ as given by a basis expansion analogous to Eq. (9). This gives the pure state density matrix

$$|\mathbf{k},\mathbf{a}\rangle\langle\mathbf{k},\mathbf{a}| = \mathcal{P}^{(\mathbf{k})}\mathcal{P}_\ell^{(\mathbf{a})} \tag{13}$$

With an explicit form of the density matrix, we take the partial trace to get the reduced density matrix of A as

$$\begin{aligned}\rho_A =& \mathrm{Tr}_B\,|\mathbf{k},\mathbf{a}\rangle\langle\mathbf{k},\mathbf{a}| = \\ =& \frac{1}{|G||G_\ell|}\sum_{\mathbf{n},\mathbf{z}}(-1)^{\mathbf{k}\cdot\mathbf{n}}(-1)^{\mathbf{a}\cdot\mathbf{z}}\mathrm{Tr}_B\left(g(\mathbf{n})\mathfrak{g}(\mathbf{z})\right).\end{aligned} \tag{14}$$

Any element $g(\mathbf{n})\mathfrak{g}(\mathbf{z})$ that is *not* equal to the identity on B contains at least one X or Z operator acting in B and thus has a zero partial trace. The only non-zero contributions to the sum in Eq. (14) is from operators supported only on A, i.e. equal to the identity on B. For the logical group, only $\mathbf{z} = 0$ survives as any other value of $\mathbf{z}$ is not the identity in B. However, there is an entire subgroup, $G_A \subseteq G$, which survives the trace. Such a subgroup is defined as all elements of G which only have support in A. As a subgroup, it has an analogous basis labeled by $\mathbf{n}_A$ and excitations labeled by $\mathbf{k}_A$, where we assume that $\mathbf{k_A}\cdot\mathbf{n}_A = \mathbf{k}\cdot\mathbf{n} \mod 2$ for the same element as labeled by $\mathbf{n}_A$ in G_A and $\mathbf{n}$ in G. Since the identity on B has trace $\mathrm{Tr}\, I_B = 2^{|B|}$ we have that

$$\begin{aligned}\rho_A &= \frac{2^{|B|}}{|G||G_\ell|}\sum_{n_A}(-1)^{\mathbf{k}_A\cdot\mathbf{n}_A}g_A(\mathbf{n}_A) \\ &= \frac{2^{|B|}|G_A|}{|G||G_\ell|}\mathcal{P}_A^{(\mathbf{k}_A)}.\end{aligned} \tag{15}$$

Since Eq. (15) expresses ρ_A as a projector, its entanglement entropy follows straightforwardly:

$$S_A = (|B| - d_G - d_\ell) - d_{G_A} = |A| - d_{G_A}, \tag{16}$$

where $d_{G_A} = \dim G_A$ and we have used the fact that $N = |A| + |B| = d_G + d_\ell$.

This same calculation can be carried out from the perspective of B, and we would similarly find that there exists a subgroup, $G_B \subseteq G$, consisting of all elements of G which only have support in B. The only difference in the argument is that all of G_ℓ survives the partial trace over A, so that the reduced density matrix for B is now a projection defined by the group $G_B \times G_\ell$. Here, the $\times$ product of two groups generates all products between elements of the two groups. Note that taking such a product of trivially intersecting Abelian groups generates the $\mathbb{F}_2$ vector space given by the direct sum of each individually, i.e. $V[G_B \times G_\ell] = V[G_B] \oplus V[G_\ell]$. Defining $d_{G_B} = \dim G_B$, one has that the entanglement entropy for B is

$$S_B = |B| - d_{G_B} - d_\ell. \tag{17}$$

Even though $S_A = S_B$ as the original state was pure, we will use both forms in the discussion of recoverable information.

III. RECOVERABLE INFORMATION IN STABILIZER CODES

We now introduce the concept of 'recoverable information.' In a subsequent section we will show that for stabilizer codes, this quantity is intimately tied to topological properties of the stabilizer set, and hence to those of the Hamiltonian naturally generated by it. Furthermore, it can be calculated with information obtained entirely from system A, and in this sense, is locally measurable.

To motivate the idea of recoverable information, consider the following situation. A binary password is encoded into the eigenstate of a stabilizer code of linear size L (the $\sim L^d$ bits of the password being simply the eigenvalues of the stabilizer and logical operators). Then an R^d sub-region A of the eigenstate is removed and given to Alice, with the remaining piece $B = \bar{A}$ given to Bob. Alice and Bob are only allowed local operations and measurements, while afterwards, they are allowed to combine all measurement results via classical communication. How many bits of the binary password can they extract?

Since the stabilizers form a commuting set and the wavefunction is an eigenstate of the stabilizers, Alice can faithfully and without disturbing the state measure all the bits of the password encoded in stabilizers contained entirely in her subsystem. Bob can do the same with its complement. However, because Alice's density matrix is proportional to a projection, she is in a completely mixed state for all the degrees of freedom which lie on the boundary of A. That is, if she tries to measure any of the partial stabilizers that were cut when separating A and B, she could measure ±1 with equal probability. Naïvely, Alice and Bob might assume they have lost all the information about cut stabilizers, but the *actual* amount of information lost is given by the entropy for each of their states. There is thus a difference between the naïve and actual information loss, viz.

$$\mu(\mathcal{B}) = d_{\partial A}(\mathcal{B}) - S_A - S_B, \tag{18}$$

where $d_{\partial A}$ is the number of basis stabilizers cut by the entanglement surface. Eq. (18) emphasizes that value of $d_{\partial A}$ can depend on the choice of stabilizer basis, and a 'bad basis' in which the stabilizers are less spatially local will lead to more cut stabilizers and hence a higher value of μ. So, in order to unambiguously define the recoverable information we minimize (18) over all possible choices of stabilizer basis (effectively assuming obc as discussed below). This removes the basis-dependence, leading to the definition (1) quoted in the introduction. This definition implicitly involves the 'most spatially local' set of stabilizers (analogous to the 'clipped stabilizers' of Ref. 44). A second caveat also implicit in this definition is that the set of cut stabilizers should not include any *logical* operators. This is trivially true for a system with open boundary conditions. For a system with periodic boundary conditions, this may be ensured by working in a basis where the logical operators are contained entirely in $B = \bar{A}$ as was assumed in Section II E.

This discussion motivates a complementary perspective on recoverable information. Imagine we cut the system as described, but also *forbid* Alice and Bob from communicating by any means. They then perform all their measurements of any combination of stabilizers accessible to them, and use this to estimate how much information they have lost about the system. Then, classical two-way communication between Alice and Bob is reestablished, allowing them to compare their measurements; it is possible that in the process, they can get some additional information that was missing when they were forbidden from communicating. It is this additional information that is the 'recoverable' information.

How may we compute the recoverable information of a bipartition? The most naïve procedure is to count cut stabilizers (suitably defined) to determine $d_{\partial A}$, and then compute the entanglement entropy using the methods of Section II E. An alternative approach is to count topological constraints. This provides a lower bound on μ that is tight for some of the models studied here. Finally and most powerfully, we will show in this section that μ may be determined directly by counting the number of independent NLSSs. Each NLSS is typically a product of stabilizers in A and on its boundary, which has support only in B. We will demonstrate the use of each of these methods in the examples of Section IV.

A. Warm-up: the $d = 1$ cluster model and the $d = 2$ toric code

Before embarking on an abstract analysis of recoverable information, we discuss two simple illustrative examples, the $d = 1$ cluster model and the $d = 2$ toric code. The $d = 1$ cluster model with periodic boundary conditions is not topologically ordered (i.e., it is short range entangled and has no ground space degeneracy), and so we should expect that its recoverable information is zero. Note that it also has no constraints, topological or otherwise. We take subsystem A to be a contiguous set of R qubits. Every stabilizer in A except for the two vertices at either end are completely in A and so applying Eq. (16) we find the entropy is $S_A = 2$. As for cut stabilizers, both edge vertices in A and edge vertices in B represent cut stabilizers, in which case the recoverable information is

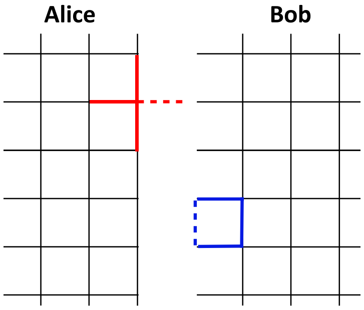


FIG. 2. Depiction of the boundary of Alice and Bob's system. The red (blue) edges represent a good cut stabilizer for Alice (Bob). The dotted edges are not actually present but represent the completion of the cut stabilizer.

$$\mu_{\mathrm{clu}_1} = 4 - 2(2) = 0. \tag{19}$$

So in this case, Alice and Bob gain no additional information when allowed to communicate classically. We will discuss cluster models in $d > 1$ in Sec. IV A.

We now turn to the more interesting case of the $d = 2$ toric code. We take subsystem A to be the square $R \times R$ cut (in edges) so that Alice does not have 'ragged' edges, but Bob does. This is depicted in Fig. 2. Then, Alice has parts of cut plaquette and vertex operators. Consider only the cut vertex operators for Alice. Since they still commute with all complete stabilizers in A, they still represent good qubits, the measurement of which will not disturb the other stabilizer qubits. Likewise, the cut plaquette operators still commute with all of Bob's stabilizers and thus are good qubits for him. There are $4R$ such good boundary qubits for Alice and $4R$ good boundary qubits for Bob, all of which are apparently in a mixed state (±1 with equal probability). However, the actual amount of lost information for either Alice or Bob is given by the entanglement entropy, $S_A = 4R - 1$. We thus conclude that the recoverable information is

$$\mu_{\mathrm{TC}_2} = 2 \times 4R - 2(4R - 1)) = 2, \tag{20}$$

which is twice the topological entanglement of the toric code. The existence of this recoverable information comes from the fact that there is a constraint between Alice's stabilizers and boundary qubits, namely that

$$\prod_{v \in A} (A_v)_A = I_A, \tag{21}$$

— in other words, the product of all the vertex stabilizers, both cut and whole, is the identity in A. This implies that Alice has one additional bit of information in her system. To understand the nature of the information recoverable by Alice and Bob, and how they can recover it in practice, note that there are global constraints on the stabilizers, namely

$$\prod_p B_p = I_{AB} \text{ and } \prod_v A_v = I_{AB}. \qquad (22)$$

Alice can measure her vertex operators (all via local operations) and communicate the results (purely classical information) to Bob. Bob now measures all his vertex operators and together with the constraint,

$$\prod_{v\in\partial A} A_v = \prod_{v\in\text{bulk}(A)} A_v \prod_{v\in\text{bulk}(B)} A_v \qquad (23)$$

In this way, Alice and Bob have gained one additional piece of information about the system than they would naïvely have expected, since they now know the *parity* of the boundary stabilizers. Likewise, they can do the same for the plaquettes, so that the actual information gained is two bits. We may also attribute this result to the presence of two NLSSs. Each consists of a string loop (one of each type, X and Z) in B which are non-contractible due to the hole created by removing A. These strings are created by the product of all stabilizers of a given type in A and on the boundary. We also note that we have cut two independent topological constraints.

To relate this back to topological considerations, we mentioned in the introduction that the topological entanglement entropy of the toric code can be related to a geometric constraint on the wavefunction. In the loop condensation picture, the net parity of edges perpendicular to the boundary as measured in the X basis must be 1, i.e. every loop in the condensate is cut twice by the boundary. This is shown to be the source of the one bit of topological entanglement entropy. However as we have shown, this bit is not special to a ground state: all simultaneous eigenstates of the stabilizers (and members of the logical group) have the same entanglement entropy and thus the same topological entanglement entropy. This is because the constraint for the ground state is generalized to a Z_2 Gauss law for excited states. Namely, the parity of the boundary is equal to the number of topological charges within the bounded region modulo 2. An equivalent statement can be made for the magnetic sector. To see that these two Gauss's laws correspond to the two bits of recoverable information, we rewrite Eq. (21) as

$$\prod_{v\in\text{bulk}(A)} A_v = \prod_{v\in\partial A} (A_v)_A. \qquad (24)$$

This is exactly the electric Z_2 Gauss's law. An analogous statement can be made for the magnetic sector using plaquette stabilizers. We show that this is not coincidental; recoverable information for other stabilizer models also matches the number of independent Z_2 Gauss law-type relations satisfied by the entanglement cut. We return to the $d = 3$ toric code in Sec. IV B.

B. Recoverable Information and Topological Properties of Stabilizer Codes

With these preliminaries, we may now clarify various facts about recoverable information. In Section III, (see (18) and the subsequent discussion) we defined the recoverable information via a process of minimization over choices of stabilizer basis, leading to (1), reproduced here for convenience:

$$\mu = \min_{\mathcal{B}}[d_{\partial A}(\mathcal{B}) - S_A - S_B]. \qquad (25)$$

We now discuss how to compute the recoverable information via three different methods.

1. *Computing recoverable information by counting cut stabilizers*

Using (16) and (17) in (25), and $|A| + |B| = N = d_G + d_\ell$ again, we have

$$\mu = \min_{\mathcal{B}}[(d_{G_B} + d_{G_A} + d_{\partial A}(\mathcal{B})) - d_G]. \qquad (26)$$

To go further and compute μ directly by counting stabilizers we need to clarify the meaning of $d_{\partial A}(\mathcal{B})$, which will also motivate our minimization procedure for (26). Recall that we chose the basis $\mathcal{B}$ to be a subset of $\mathcal{S}$; this is always possible as we generated G from $\mathcal{S}$. Given any basis $\mathcal{B}$, we can clearly partition it as $\mathcal{B} = \mathcal{B}_A \cup \mathcal{B}_B \cup \mathcal{B}_\partial$ where $\mathcal{B}_A = \{g \in \mathcal{B} : \text{Supp}(g) \subseteq A\}$, $\mathcal{B}_B = \{g \in \mathcal{B} : \text{Supp}(g) \subseteq B\}$, and $\mathcal{B}_\partial = \{g \in \mathcal{B} : \text{Supp}(g) \cap A \neq \emptyset \text{ and } \text{Supp}(g) \cap B \neq \emptyset\}$, respectively corresponding to basis elements entirely in A, entirely in B, and supported in both. Note that it is not necessarily true that $G_A \cong \text{Span}(\mathcal{B}_A)$ and $G_B \cong \text{Span}(\mathcal{B}_B)$: the subgroup of the stabilizer group obtained by restricting its support to some subregion need not coincide with the subgroup generated by the restriction of the basis to that subregion. Below, we will frequently and without explicit comment use subscripts A, B and ∂ to denote the partitioning of a set into elements only supported in A, only supported in B, and supported in both A and B. It follows that

$$d_G = |\mathcal{B}_A| + |\mathcal{B}_B| + |\mathcal{B}_\partial|. \qquad (27)$$

$\mathcal{B}_\partial$ in turn generates the *cut stabilizer group* $G_\partial = \text{Span}(\mathcal{B}_\partial)$ with dimensions $|\mathcal{B}_\partial|$, motivating us to define $d_{\partial A} = |\mathcal{B}_\partial|$. Now, we may easily show that $G_A \times G_B \times G_\partial = G$; note that dim G is not necessarily equal to dim G_A + dim G_B + dim G_∂. Then, from (26), we see that the only way for the recoverable information to be nonzero is if there is some nontrivial pairwise intersection between G_A, G_B and G_∂ (for, if they were disjoint then $d_G = d_{G_B} + d_{G_A} + d_{\partial A}$). Note that G_A and G_B correspond to disjoint subspaces, since any product of two elements of G_A is also in G_A, and likewise for G_B, and the support of G_A is disjoint from that of G_B.

We note that the definition of G_∂ is basis-dependent, but this is expected given the motivation for recoverable information: the encoding of a password in stabilizer quantum numbers is inherently basis-dependent. However, we can restrict the basis choice by exploiting constraints and the definition of recoverable information from Section III. By doing so, we automatically perform the minimization procedure that makes the recoverable information unambiguous. Our approach, elaborated below, will also resolve subtleties in quantifying the quantum information encoded in sets of stabilizers that are required to satisfy local constraints.

From the definition of $d_{\partial A} = |\mathcal{B}_\partial|$ and (27) we see that (26) reduces to

$$\begin{aligned}\mu &= \min_{\mathcal{B}}[(d_{G_A} - |\mathcal{B}_A|) + (d_{G_B} - |\mathcal{B}_B|)] \\ &= \min_{\mathcal{B}}[\dim\left(G_A/\mathrm{Span}(\mathcal{B}_A) \times G_B/\mathrm{Span}(\mathcal{B}_B)\right)]. \end{aligned} \quad (28)$$

The second equality follows from the fact that $G_A/\mathrm{Span}(\mathcal{B}_A)$ and $G_B/\mathrm{Span}(\mathcal{B}_B)$ are disjoint subgroups of G with dimensions $d_{G_A} - |\mathcal{B}_A|$ and $d_{G_B} - |\mathcal{B}_B|$, respectively. Note (28) proves that μ is the (minimum) dimension of a group and is hence non-negative, $\mu \geq 0$.

We have yet to fully fix the basis that minimizes μ. To do so, and fully remove ambiguity in the basis choice, we recall an important fact: namely, that the difference between the full stabilizer set $\mathcal{S}$ and the basis $\mathcal{B} \subseteq \mathcal{S}$ is linked to the existence of constraints. So we can think of creating a basis as a process of removing stabilizers: we need to remove exactly one stabilizer for each independent constraint.

To proceed, we define the following: a Pauli operator, p, is said to be *cut* by a bipartition (A, B) iff $\mathrm{Supp}(p) \cap A \neq \emptyset$ and $\mathrm{Supp}(p) \cap B \neq \emptyset$; a set of Pauli operators is cut iff at least one member is cut. Constraints that are not cut by definition contain only elements entirely in A or B, and so which stabilizer is removed to form the basis is inconsequential. However, when a constraint is cut, we have the choice to remove either a non-cut stabilizer or a cut stabilizer, leading to some remaining ambiguity in the basis choice. We propose the following rules to remove this ambiguity and implement the minimization which defines recoverable information in Eq. (25). The rules are simple: when a constraint is cut, we remove a non-cut (cut) stabilizer if the constraint is topological (non-topological). (Note that typically, a non-cut stabilizer removed due to a cut topological constraint will be contained entirely in B.) In practice, our prescription means that we simply compute $d_{\partial A}$ by counting cut stabilizers, with the caveat that we reduce the number of cut stabilizers by one for every cut non-topological constraint. Note that per this algorithm, when subtracting cut constraints, we ignore any topological constraints (effectively assuming obc), since they involve removing only non-cut stabilizers from B. Since decreasing $d_{\partial A} = |\mathcal{B}_\partial|$ decreases μ, this procedure yields the minimal value of μ, modulo changes to the boundary conditions.

We may justify this method of generating the basis as follows. When the constraint is topological, the relation between topological constraints and logical Pauli operators tells us that there must exist a logical operator which contains a piece of the code. The logical bit implied by the topological constraint is always recoverable via the cleaning lemma, so always contributes to μ. (Note that this is consonant with the fact that a topological constraint will never reduce the count of cut stabilizers in our algorithm above). Put differently, we are effectively minimizing the recoverable information over *all possible* local *bases used to encode the information*, thereby retaining only the non-local information that is recoverable for any local basis choice. The basis connection lemma, proved in Appendix A, ensures that this procedure will not get "stuck" in local minima and instead will converge on a globally minimizing basis that yields μ.

2. *A bound on the recoverable information*

As a corollary, we also derive a useful **bound on** μ, providing another route to characterizing recoverable information. Define the set of *cut topological constraints*, $\mathrm{Cut}(\mathcal{T}) = \{C \in \mathcal{T} : C \text{ is cut by } (A, B)\}$. Then

$$\mu \geq \dim(\mathrm{Cut}(\mathcal{T})). \quad (29)$$

This is obvious as we specifically did not remove cut stabilizers when the constraints involved were topological.

3. *Computing recoverable information from non-local surface stabilizers*

Finally, we can relate recoverable information to a special subgroup of G defined for a bipartition (A, B) and a basis $\mathcal{B}$, that we term the *non-local surface stabilizer group*, denoted G_{NLSS}. This group captures all of the non-trivial overlap between the groups G_A, G_B and G_∂ and is defined as

$$G_{\mathrm{NLSS}} = G_\partial \cap (G_A \times G_B). \quad (30)$$

It follows from this definition that any element $g_\partial \in G_{\mathrm{NLSS}}$ may be generated in one of two ways: (i) as a product over some set of cut basis members, $F_\partial \subseteq \mathcal{B}_\partial$ and (ii) as a product of some elements $g_A \in G_A$ and $g_B \in G_B$, i.e. $g_\partial = \prod_{s \in F_\partial} s = g_A g_B$. We show (via relation (A3)) that given G_{NLSS}, we can always choose a basis $\mathcal{B}$ such that either g_A or g_B, but not both, can be written as a product over members of $\mathcal{B}_A(\mathcal{B}_B)$. Thus we have a product of stabilizers in A (B) and on the boundary which only has support in $B(A)$.

As in the $d = 2$ toric code, the elements of G_{NLSS} are topological in nature, and are directly linked to the Gauss's law-type constraints satisfied by the entanglement cut. To see this, let us suppose all of G_A is generated by stabilizers which are confined to A (as is usually the case when A is bounded and simply-connected), and assume that we choose a basis $\mathcal{B}$ that contains all such stabilizers. Now consider any element $g_\partial \in G_{\text{NLSS}}$; from our discussion above, there exist elements $g_A \in G_A$ and $g_B \in G_B$ so that $g_\partial = g_A g_B$. By restricting both sides of this expression to its support in A (denoted $(\cdot)_A$) we find $(g_\partial)_A = (g_A g_B)_A = (g_A)_A (g_B)_A = g_A$, as g_B is supported only in B by definition. By our assumptions about G_A, g_A is generated by a basis set $F_A \subseteq \mathcal{B}_A$, i.e. $g_A = \prod_{s\in F_A} s$, leading to the operator relation

$$\prod_{s\in F_A} s = (g_\partial)_A. \tag{31}$$

Since each member of F_A is a local stabilizer that squares to the identity, it can be thought of as a Z_2 charge. In this sense, (31) expresses the fact that the number of charges modulo 2 within some subset of the bulk of a compact subregion A is equal to a measurement on the boundary of A — a relation akin to Gauss's law. Thus, any NLSS can be mapped to a Gauss's law-type relation (or combination of such relations) satisfied by the entanglement cut. Though we assumed g_A was formed by stabilizers in A, we can see that if g_B is formed by a set $F_B \subseteq \mathcal{B}_B$ instead, an analogous statement holds when the equation is restricted to its support in B.

Note that G_{NLSS} is dependent on the basis (which we say *generates* G_{NLSS}); we prove in Appendix A that

$$G_{\text{NLSS}} \simeq G_A/\text{Span}(\mathcal{B}_A) \times G_B/\text{Span}(\mathcal{B}_B). \tag{32}$$

Comparing this to Eq. (28), we see non-local surface stabilizer groups can be related to recoverable information by the following equivalent statements:

(S1) $\dim(G_{\text{NLSS}}) = \mu$ (see the minimization definition in (1) along with Eqs. (28) and (32)),

(S2) For any element $g_\partial \in G_{\text{NLSS}}$ with its associated $g_A \in G_A$ and $g_B \in G_B$ for which $g_\partial = g_A g_B$, if there exist sets of stabilizers, $F_A \subseteq \mathcal{S}_A$ and $F_B \subseteq \mathcal{S}_B$ whose product is g_A, g_B respectively, then $F_A \cup F_B \cup F_\partial \in \mathcal{T}$, where $F_\partial \subseteq \mathcal{B}_\partial$ is the basis set which generates g_∂ and

(S3) For all $g_\partial \neq I \in G_{\text{NLSS}}$ such that $F_\partial \subseteq \mathcal{B}_\partial$ is the basis set which generates g_∂, if there exists a non-topological constraint $C \in \mathcal{C} - \mathcal{T}$ such that $F_\partial \subseteq C$, then $C_\partial - F_\partial \neq \emptyset$.

Recall $\mathcal{C}$ is the set of all constraints and $\mathcal{T}$ is the set of topological constraints as discussed in Sections II B and II D. See Appendix A for proof of the equivalence of these statements. (S2) and (S3) can be useful for computing μ as they make no reference to a generating basis for G_{NLSS} (outside of $\mathcal{B}_\partial$, but generally, this can be replaced by $\mathcal{S}_\partial$, the set of stabilizers at the boundary). For example, suppose we find some independent set of stabilizer group elements which look like NLSSs (i.e. satisfy a relation of the form $g_\partial = g_A g_B$). However, there is no immediate way by which we can check that these form some G_{NLSS} outside of finding a basis which generates it, let alone assess whether such a basis is minimal. However, if our set of NLSSs satisfy statements (S2) or (S3), we know that its dimension is a lower bound for μ. Then we can continue to add independent NLSSs to our candidate set until (S2) and (S3) are violated, at which point we must have had a G_{NLSS} which is generated by a minimal basis. This is useful in cases where calculating μ via the definition (25) is considerably more difficult than finding possible NLSSs.

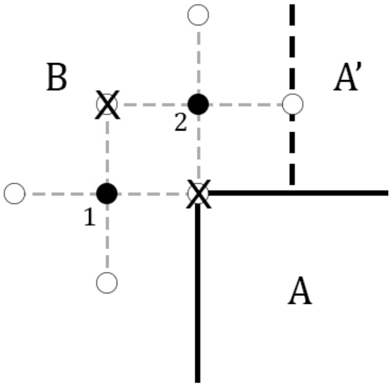


FIG. 3. Visualization of the 'quasi-local NLSS' at the corner of a square entanglement cut. For simplicity, the black circles represent Z-type operators and the white circles represent the X-type operators.

Given this result, it is useful to consider what the minimization procedure implies for the emergent Gauss's law relations. Usually, Z_2 charges in a stabilizer code can be defined consistently only if we choose a stabilizer basis. However, some choices of basis may result in conservation laws of the type in (31) that are not universal, i.e. do not appear for other basis choices. When the basis $\mathcal{B}$ is minimal, any alternative basis $\mathcal{B}'$ generates a possibly larger non-local surface stabilizer group, i.e. $G_{\text{NLSS}} \subseteq G'_{\text{NLSS}}$, up to equivalent choices of minimal basis (a more precise statement is discussed in Appendix A). Thus a minimal basis generates a universal set of Gauss laws.

Note that both the stabilizer-counting and NLSS counting approaches to determining recoverable information involve counting stabilizers to determine the dimension of an appropriate stabilizer subspace, so that there is an occasional formal similarity between the two in practice. The reader is urged to keep in mind that the two are still quite different — the former requires additionally the knowledge of entanglement entropy (e.g. by the methods of Ref. 26) whereas the latter gives the recoverable information directly and is therefore arguably more powerful.

FIG. 4. Depiction of the constraint in the $d = 3$ toric code which is formed by the product (sum in $W[\mathcal{S}]$) of all cut local cube constraints, but is itself not a cut constraint.

IV. COMPUTATIONS OF RECOVERABLE INFORMATION

We now apply the techniques of the preceding section to four different stabilizer codes introduced in Sec. II : the cluster model in any dimension, the $d = 3$ toric code, the X-Cube model and Haah's code. Though first two are not fracton models, they demonstrate some of the necessary considerations involved in the minimization process used to define and compute the recoverable information. The cluster models also serve to illustrate the expected behavior for the recoverable information in SPT phases. Note that we often identify elements of G_{NLSS} by the element $g_B \in G_B$ for which there exist $g_A \in G_A$ and $g_\partial \in G_\partial$ such that $g_A g_B = g_\partial \in G_{\text{NLSS}}$. We do so as elements in B are more intuitively thought of as NLSSs.

A. Cluster model in arbitrary dimension d

We return to the cluster model, starting with $d = 2$ and then arguing the same results hold in all other dimensions. We first take subsystem A to be an $R \times R$ square of vertices. The entropy is simply $S_A = 4R - 4$, as there are R^2 qubits and $(R-1)^2$ complete stabilizers in A (one for each internal vertex not along the edge). Turning to the cut stabilizers, note that every vertex on the boundary of A represents one cut stabilizer which contributes $4(R-1)$ cut stabilizers. Similarly, almost every vertex in B and on the boundary with A is cut, *except* for the four vertex stabilizers at the corners; so vertices in B contribute $4R$ to the count of cut stabilizers. Therefore, the recoverable information is

$$\mu_{\text{clu}_2}(\text{square}) = 4R - 4 + 4R - 2(4R - 4) = 4. \tag{33}$$

This may seem counterintuitive, as one would expect that the recoverable information vanish, in accord with the fact that the cluster model does not exhibit topological order. The extra bits of recoverable information can be attributed to the sharp corners of the entanglement cut. At a corner, we technically have a NLSS (though the name is not particularly apt in this case) coming from the product of two stabilizers oriented diagonally across a lattice plaquette and adjacent to the corner. Though each of these are cut by the corner, their product exactly removes support from that corner and as such is an NLSS (a product of cut stabilizers which is only supported in B; $g_A = I$ in this case). Such a situation is depicted in Fig. 3 for regions A and B. However, if we choose the subregion to be $A \cup A'$ from Fig. 3, stabilizer 2 is now additionally cut along the boundary of A'. Then the product of stabilizers 1 and 2 is no longer entirely supported in B and thus does not contribute to the recoverable information. So instead of a square entanglement cut, if we choose a sufficiently 'smooth' cut (i.e. one such that no corners are formed such as that shown in Fig. 3 for cut A), then we have zero recoverable information. This is analogous to topological entanglement entropy, where one only considers the universal constant piece of entanglement entropy for smooth boundaries. It should also be clear that this behavior is general in any dimension, where these 'quasi-local NLSSs' appear in the recoverable information for sharp $d-2$ edges. For example in $d = 3$, Fig. 3 can be viewed as a 2D slice of a 3D boundary where the cluster stabilizers extend into and out of the page. So we conclude that in any dimensions d and for a sufficiently smooth boundary, the recoverable information is

$$\mu_{\text{clu}_d}(\text{smooth}) = 0. \tag{34}$$

This result is the first demonstration of the power of independent NLSS counting; we can make very general statements without explicitly computing the entropy or the number of cut stabilizers. Furthermore, the fact that recoverable information is sensitive to sharp corners can be useful in certain cases — e.g., the existence of corner entanglement can be indicate important universal features for certain models[45] (also see discussion in Sec. V).

B. $d = 3$ toric code

The $d = 3$ toric code provides an example where counting cut stabilizers requires some caution to properly account for local constraints. We will take sub-region A to be an $R \times R \times R$ cube (as measured by edge length on the lattice) with smooth surfaces. Qubits on the entanglement surface are taken to lie 'inside' the entanglement cut. As in the $d = 2$ toric code, we have the constraint that the product of all vertex stabilizers in the system is the identity. However, there is an extensive set of independent *local* constraints: namely, the set of all plaquette stabilizers around any cube (as their product is the identity). The latter reflects a redundancy in the information contained in the stabilizers: naïvely, it seems as though the plaquette stabilizers on a single cube encode 6 bits of information, but thanks to this constraint, only 5 of these are independent. If such a cube is cut, Alice and Bob each receive one of two opposite faces, and between

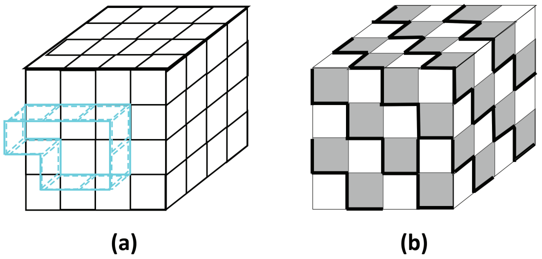


FIG. 5. Construction of an appropriate basis for the cut stabilizer group in the $d = 3$ toric code. (a) Depiction of the straddling surface ribbons a minimal basis must avoid generating in G_∂. (b) Basis construction using the checkerboard method as described in the text. Here, the thick lines indicate the intersection of a cut plaquette stabilizer in the basis with A.

them they have two bits of information. The remaining 3 bits are shared between the four cut stabilizers, and so if we simply counted cut stabilizers, we would *overestimate* the information loss when Alice and Bob are in the 'no communication' state. This motivates the caveat on the definition of cut stabilizers and the cut stabilizer group as discussed in Section III B.

We first compute μ by the cut-stabilizer-counting approach. According to the prescription given in Section III B, we can minimize the number of cut stabilizers by counting all cut stabilizers and subtracting the number of cut non-topological constraints. There are $6R^2+2$ cut vertex stabilizers (with no local constraints among these). There are $12R(R+1)$ cut plaquette stabilizers, but there are $6R^2+12R$ cut local cubic constraints involving plaquettes, one for each cube with an edge intersecting A. Naively, subtracting these gives $6R^2$ for the contribution of plaquette stabilizers to μ. However, under the map f (defined in Sec. II B) the product of all the cubic constraints (alternatively understood as the sum of constraints in $W[\mathcal{S}]$) is mapped to the product of all plaquettes in the boundary and all those in the layer of plaquettes just outside the boundary and parallel to it (excepting the three plaquettes that close each corner of the outer cube; see Fig. 4), and is therefore *not* cut. That is, there is no cut stabilizer to remove from this enveloping constraint in accordance with the minimization and cannot be subtracted from the total number of cut stabilizers. Note that this example neatly illustrates the importance of counting constraints with the group structure imposed (i.e., inside $W[\mathcal{S}]$) rather than simply taking unions of sets. Therefore, we find in effect that, for any choice of $\mathcal{B}_\partial$, there are $6R^2+1$ cut plaquette stabilizers, leading to $\min_{\mathcal{B}}[d_{\partial A}(\mathcal{B})] = 6R^2+2+6R^2+1 = 12R^2+3$.

An alternative method to obtain this result is to recognize that no matter what basis we choose, it satisfies our requirements only if no set of cut plaquettes can be combined to form a closed ribbon operator whose parallel edges straddle the entanglement cut (i.e forms parallel strings in A and B; see Fig. 5(a)).These ribbons represent a cut local constraint when one includes the 'caps' of the ribbon: the plaquettes on the boundary of A encircled by the ribbon, and their parallel counterpart in B. An algorithm for constructing a basis free of such straddling ribbon operators is as follows: first, we label the squares on the surface (not to be confused with plaquette stabilizers) in a "checkerboard" fashion. Then, we only include the cut plaquettes that intersect the "black" squares on two of the four edges, say the right and bottom. On the surface, this looks like several lines that stair-step diagonally across a face of the cube, as shown in Fig. 5(b). At the intersection of two planes of the entanglement surface, we only include the plaquette perpendicular to the surface containing the black square. Careful consideration at those intersections shows that this does not allow closed ribbon loops. This implies that exactly half the edges on the boundary support a cut stabilizer in the basis, but we can add exactly one additional plaquette anywhere without creating a closed ribbon loop. Counting this way, we once again have $6R^2+1$ cut plaquettes and $6R^2+2$ cut vertex stabilizers (as the vertices satisfy no local constraints). A simple counting argument using Eq. (16) yields $S_A = S_B = 6R^2+1$, whence we find, using the definition (25), that

$$\mu_{\mathrm{TC}_3} = 1. \tag{35}$$

We can also compute the result in terms of the lower bound on μ as given by cut topological constraints. The set of all vertex stabilizers in the system satisfies a single topological constraint with pbcs, that is cut by the presence of A, leading to the bound $\mu \geq 1$, which is clearly saturated in this example.

Finally, this also agrees with the counting in terms of non-local surface stabilizers: there is precisely one NLSS, namely the non-contractible membrane operator wrapping the cut (product of all vertex stabilizers in $A \cup \partial A$). There are no other NLSSs left after local constraints are implemented. In particular, to convince ourselves that no string operator can be in a minimal G_{NLSS}, the provided construction explicitly forbids creating cut ribbons which when combined with some element of G_A would create a string. One can readily check that such strings violate statements (S2) and (S3) and thus any basis which admits them cannot be minimal.

Note that the definition of μ also clearly satisfies our criterion of identifying the number of Gauss law-type conditions satisfied by the emergent degrees of freedom: it is well-known that the $d = 3$ toric code can be identified with a Z_2 gauge theory, with a single such emergent conservation law for point-like particles (i.e. excitations with zero-dimensional support). The Z_2 magnetic fluxes in the 3D toric code form closed loops, rather than point-like particles as in $d = 2$. The corresponding conservation law is linked to a closed-membrane rather than a closed-loop constraint of the Gauss-law type, and hence does not contribute to the recoverable information for simply connected entanglement cuts.

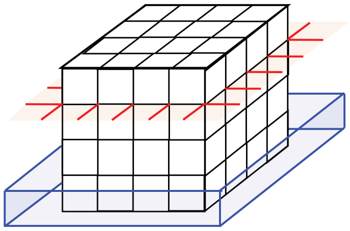


FIG. 6. Depiction of the NLSS ribbon operators for the X-cube model. The red operator is the Z-type ribbon operator generated from A-type stabilizers and the blue operator is the X-type ribbon operator generated from B-type operators.

C. X-cube model

We consider next the X-cube model, with the same choice of A as for the $d = 3$ toric code. As in that example, we have an extensive number of local (non-topological) constraints, namely the set of all three A-type stabilizers for every vertex (since their product is the identity). However, these constraints are more easily handled as compared to the $d = 3$ toric code, as follows. Clearly, when the six edges that coordinate a vertex are split between A and B, at least two of the three associated vertex stabilizers are cut. When only two are cut, the minimization prescription dictates that one is removed from the basis, leaving one cut stabilizer; when all three are cut, we have no choice but remove one from the basis, leaving two cut stabilizers. By this counting, there are $6(R-1)^2$ cut A type stabilizers on the faces (one for each vertex), $24(R-1)$ on the edges and 16 at the corners (two for each vertex). As for the B-type stabilizers, all the constraints are topological and every cube which intersects A represents a cut B-type stabilizer. There are $6R^2$ intersecting cubes on the faces and $12R$ along the edges. Note that the cubes that touch A at its corners do not share an edge with it and are hence are not cut. So we find that $d_{\partial A} = 12R^2 + 24R - 2$. Again, a counting argument using Eq. (16) gives $S_A = S_B = 6R^2 + 9R - 4$, whence we find that

$$\mu_{\rm XC} = 6(R+1). \tag{36}$$

The result is consistent with the number of cut topological constraints characterized by

$$\prod_{v\in P_k} A_v^{(k)} = I \text{ and} \tag{37}$$

$$\prod_{c @ P_k} B_c = I, \tag{38}$$

where P_k is the set of all vertices in a given plane perpendicular to the direction k, and we use the adjacency symbol ('@') to denote that the cube c contains vertices in P_k. The constraints on the cubes are written in an apparently redundant manner since cubes are adjacent both above and below a given plane of vertices, and each plane of cubes (corresponding to a dual lattice plane) contains an independent constraint. However, the constraints given above are still independent and are natural in terms of fracton excitation, in that this represents the subsystem in which bound pairs are mobile[17]. As each of these constraints is characterized by a plane and the constraint is cut if the plane is cut, we can see that the number of cut topological constraints is $6(R+1)$.

Finally, we can also interpret these results in terms of a natural basis for $G_{\rm NLSS}$. From the A-type stabilizers, we can create non-contractible Z-type ribbon loops confined to a single layer of vertices (all vertex stabilizers entirely supported in a given layer of $A \cup \partial A$; see Fig. 6) and there are $3(R+1)$ such operators. Meanwhile from the B-type stabilizers we can construct non-contractible X-type ribbon operators which are confined to a single plane of cubes (the product of all cube stabilizers in a layer of $A \cup \partial A$; see Fig. 6). Note that these ribbons are supported on the joints where the ribbon changes direction. There are $3R$ such ribbons near the surface, but also 6 additional ribbons which straddle the outer edge of any plane of the cut. However, not all the operators we have identified above are truly independent. For the Z-type ribbons, consider multiplying all of them together. This is equivalent to multiplying all vertex stabilizers in A and on the boundary which, put a different way, is the product of the three star stabilizers at every vertex in or on the surface of A. By the local vertex constraints, this is the identity, and we lose one NLSS from the basis. Furthermore, if we multiply all X-type ribbons wrapping around the cube in any one direction, we remove support where the ribbons overlap, but not on the outer edges or joints of the ribbons. Thus this operator is identical to having taken the same product in any other direction. This gives two additional constraint and we lose two more NLSSs from the basis. Altogether, there are $6(R+1)$ NLSSs in the basis after we implement the constraints, consistent with our earlier calculations for recoverable information.

The Gauss's law interpretation follows immediately from this basis for $G_{\rm NLSS}$. Each Z-type ribbon yields a Gauss's law akin to that found in Eq. (24) for the $d = 2$ toric code, one for each of the $3R + 3$ layers. Likewise, each X-type ribbon has an associated Gauss's law that again looks similar to the magnetic sector of the $d = 2$ toric code, but within one of the $3R + 9$ layers of cubes. However, while all of these Gauss law constraints are true, only $6(R+1)$ are independent in the following sense: given $6(R+1)$ of these conservation law statements, the other three can be deduced via direct correspondence with the discussion of constraints for $G_{\rm NLSS}$.

D. Haah's Code

We now turn to Haah's code, where we consider as sub-region A a cube consisting of R *vertices* on a side, always including both qubits at a vertex. Since the number of constraints is a complicated function of the boundary conditions, we conjecture that all of them are topological. Therefore, we need not be concerned about eliminating any of the cut stabilizers from $\mathcal{B}_\partial$. The number of cut stabilizers is therefore simply given by $d_{\partial A} = 2N_{\text{cubes}}(\partial A) - 2$, where $N_{\text{cubes}}(\partial A)$ is the number of cubes wrapping the entanglement cut (each of which has two types of stabilizers), and the subtraction accounts for the fact that exactly one of each type of stabilizer intersects A only at its double identity corner. So, although the qubits on that corner are in A, the stabilizer formed in that cube by definition has no support in A and therefore should not be counted as cut. To find $N_{\text{cubes}}(\partial A)$, we simply count the difference between the number of elementary cubes contained in an $(R+2)\times(R+2)\times(R+2)$ volume, less those in the $R \times R \times R$ interior volume. As these are $(R \pm 1)^3$, respectively, we find that $d_{\partial A} = 2[(R+1)^3 - (R-1)^3] - 2 = 12R^2 + 2$. Using Eq. (16), we obtain $S_A = S_B = 6R^2 - 6R + 2$ whence we find that

$$\mu_{\text{Haah}} = 12R - 2. \tag{39}$$

Before proceeding to compare this to the other methods for calculating recoverable information, we caution that there are some subtleties that need to be taken into account. As mentioned above, the ground state degeneracy of Haah's code and hence the number of topological constraints is a complicated function of L: while a certain sub-sequence of lattice sizes L has a degeneracy exponential in L, there is a different sub-sequence of sizes where the degeneracy is independent of L. Since the size of the entanglement cut R can be any suitable fraction of the total system size L, the recoverable information can exceed the number of topological constraints. So not only is it difficult to count cut constraints, the bound they provide for recoverable information is far from tight. This emphasizes the fact that recoverable information is locally measurable and thus insensitive to the boundary conditions which dictate the number of global constraints and the ground state degeneracy. This distinction is hidden in more conventional topological orders where the local information in the entanglement is more directly linked to the global constraint and ground-state degeneracy properties.

We now wish to associate the recoverable information in Haah's code with a basis for G_{NLSS}; this is challenging owing to the rather complicated structure of the stabilizers in Haah's code. The counting is most easily achieved by means of the polynomial ring formalism described, e.g., in Ref. 8. The full power of the polynomial ring formalism requires results in commutative algebra that are standard but possibly unfamiliar to our readers; however, we only require some basic ideas of this formalism, that we reproduce here for convenience.

The polynomial ring formalism provides a representation of translationally-invariant Hamiltonians built from products of Pauli operators in terms of 'stabilizer maps', matrices whose entries are drawn from the Laurent polynomial ring $\mathbb{F}_2(x, y, z)$ over the finite field on two elements (In simple terms, this is the set of Laurent polynomials in three variables with coefficients in the set $\{0, 1\}$ where addition and multiplication are both taken modulo 2). As we describe below, this is merely an alternative representation of $V[P]$ and $V[G]$ which exploits polynomial algebra to represent translation symmetry. Let us consider a model with a q-site basis, i.e. q qubits per unit cell: for example, $q = 3$ for the X-cube model, while $q = 2$ for Haah's code. The left dimension (number of rows) of the stabilizer map labels each of the distinct Pauli operators for each qubit in a unit cell. As in our discussion of the vector space $V[P]$, we can represent Y-type operators by the product of the X- and Z-type operators, $Y = iXZ$, so that the stabilizer map has $2q$ rows. For example, we may take the first q rows to correspond to X on each basis site, and the next q to Z. The right dimension (number of columns), m of the stabilizer map corresponds to the independent stabilizers associated to each unit cell. For instance, $m = 3$ for the X-cube model as each unit cell includes a single cubic stabilizer and two independent vertex stabilizers (since $A_v^{(xz)} = A_v^{(xy)} \cdot A_v^{(yz)}$). For Haah's code, $m = 2$, corresponding to the two stabilizer types depicted in Fig. 1. A stabilizer map is therefore a $2q \times m$ matrix.

The entries of this matrix elegantly capture the structure of the stabilizers, as follows. Each stabilizer can be associated to a given 'origin' unit cell and involves some set of Pauli operators, X or Z acting on the q qubits in some set of (usually adjacent) unit cells, linked by translation from the origin cell. Monomials in d variables provide a convenient representation of the d-dimensional translation group: for instance, a lattice point $\mathbf{r} = a\hat{\mathbf{x}} + b\hat{\mathbf{y}} + c\hat{\mathbf{z}}$ is maps to the monomial $x^a y^b z^c$. We may thus associate a monomial in the matrix with the position (as measured from the origin cell) of a single qubit Pauli operator of a given type (X or Z) and index q (that together specify the row within the matrix) which makes up a given stabilizer type (that fixes the column). As in the case of $V[P]$, multiplication of Pauli operators which make up a stabilizer is represented by summation over the appropriate monomials. Translations of a stabilizer from the origin to an arbitrary point (*e.g.*, (a, b, c)) on the lattice is then given by multiplication of the corresponding column by an appropriate monomial ($x^a y^b z^c$) which represents that position. Thus, by using arbitrary sums (which represent products) over individual monomials (which represent translations), we can generate all elements of the stabilizer group by multiplying the resulting polynomial by a column of the stabilizer map and using distributivity.

In order to illustrate this and set the stage for our next calculation, let us apply this approach to Haah's code, where as we have noted $q = m = 2$, so that the stabilizer map is a 4×2 matrix. Let us focus on the X-type stabilizers G_c^X, corresponding to the first column of the stabilizer map. As such a stabilizer only involves the X-type Paulis, only the first two rows will be non-zero. Now, for a given site $\mathbf{r}$, G_c^X involves the 'right' qubit at the four sites $\{\mathbf{r}, \mathbf{r}+\hat{\mathbf{x}}, \mathbf{r}+\hat{\mathbf{y}}, \mathbf{r}+\hat{\mathbf{z}}\}$ and so the first row of the first column is the polynomial $1+x+y+z$, since we may, without loss of generality, choose the origin at $\mathbf{r}$. Similarly, G_c^X involves the 'left' cubit at $\{\mathbf{r}, \mathbf{r}+\hat{\mathbf{x}}+\hat{\mathbf{y}}, \mathbf{r}+\hat{\mathbf{y}}+\hat{\mathbf{z}}, \mathbf{r}+\hat{\mathbf{z}}+\hat{\mathbf{x}}\}$, so that the second row of the first column corresponds to $1+xy+yz+zx$. Proceeding analogously for G_c^Z, we find the stabilizer map for Haah's code,

$$S_{\text{Haah}} = \begin{pmatrix} 1+x+y+z & 0 \\ 1+xy+yz+zx & 0 \\ 0 & 1+\overline{xy}+\overline{yz}+\overline{zx} \\ 0 & 1+\overline{x}+\overline{y}+\overline{z} \end{pmatrix}, \tag{40}$$

where $\overline{x} \equiv x^{-1}$, and we take the 'origin' at the XX (ZZ) site (Fig. 1) for G_c^X (G_c^Z).

This formalism allows us to count stabilizers with relative ease, by reducing the problem to one of counting independent polynomials. We focus on just the X-type stabilizers; the arguments carry over, *mutatis mutandis* to the Z-type ones. We define the pair of polynomials

$$\alpha = 1+x+y+z, \;\; \beta = 1+xy+yz+zx. \tag{41}$$

All other X-type stabilizers are obtained by simultaneous translations of these two polynomials, i.e. by multiplication by a monomial in the ring:

$$\begin{pmatrix} \alpha \\ \beta \end{pmatrix} \to x^a y^b z^c \begin{pmatrix} \alpha \\ \beta \end{pmatrix}. \tag{42}$$

An arbitrary product of X-type stabilizers is then associated to an element of the polynomial ring via $g = P \cdot \begin{pmatrix} \alpha & \beta \end{pmatrix}^\top$, where $P \in \mathbb{F}_2(x,y,z)$. For instance, the product of X-type stabilizers located at $3\hat{\mathbf{x}}+2\hat{\mathbf{y}}$ and $\hat{\mathbf{y}}+4\hat{\mathbf{z}}$ is obtained by choosing $P = x^3y^2+yz^4$. Similar relations hold for Z-type stabilizers, so that the entire stabilizer group is represented by elements of the polynomial ring.

For the cubic entanglement cut, we may represent every stabilizer group element contained in $A \cup \partial A$ as follows. First, take the origin (corresponding to 1 in the ring) to be the lowest corner of the $(R+2)\times(R+2)\times(R+2)$ cube surrounding A; then, the elements correspond to those polynomials where any single variable has degree $\leq R+1$. We use this construction to find a dimension for G_{NLSS} by noting that all elements of $\text{Span}(\mathcal{B}_A) \times G_\partial$ (of X-type) can be generated by local stabilizers and thus are represented by the polynomial description above (see the Appendix of Ref. 26 for a rigorous proof of this). We then impose the condition that such elements are also in G_B (i.e., only supported in B) and count the number of independent solutions, thereby computing the dimension of $G_B \cap (\text{Span}(\mathcal{B}_A) \times G_\partial)$. We follow this by arguing that $G_A \cap (\text{Span}(\mathcal{B}_B) \times G_\partial)$ is trivial (i.e., our equations have no solutions), which completes our calculation of the dimension of the X-type stabilizers within G_{NLSS} via the relation (A3) . Then, we simply draw on the inversion symmetry $\mathbf{r} \leftrightarrow -\mathbf{r}$ of Haah's code under $Z \leftrightarrow X$, while exchange the two qubits on the same site, to conclude that the answer for Z-type stabilizers is identical, so that the dimension of G_{NLSS} is simply twice that of the X-type stabilizers alone.

First, if we write

$$P(x,y,z) = \sum_{\mathbf{a}} p(a_1,a_2,a_3) x^{a_1} y^{a_2} z^{a_3}, \tag{43}$$

where $p(a_1,a_2,a_3) \in \{0,1\}$ and the sum is over all integer vectors $\mathbf{a}$, then we see that an arbitrary product of X-type stabilizers has the form

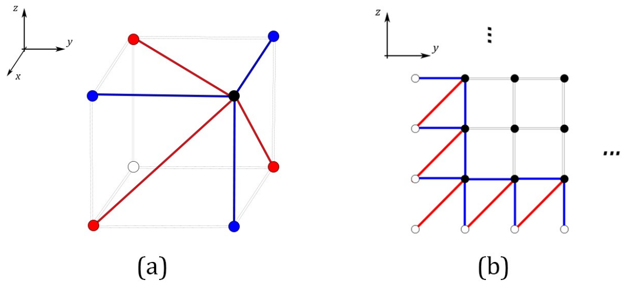
$$P \cdot \begin{pmatrix} \alpha & \beta \end{pmatrix}^\top = \sum_{\mathbf{a}} \begin{pmatrix} p_\alpha(\mathbf{a}) & p_\beta(\mathbf{a}) \end{pmatrix}^\top x^{a_1} y^{a_2} z^{a_3}, \tag{44}$$

where $p_\alpha(\mathbf{a}), p_\beta(\mathbf{a})$ follow from (41) by straightforward algebra. Since the stabilizer is assumed to belong to $A \cup \partial A$, we assume that it has degree $\leq R+1$. If we want it to be supported entirely within B, we want no terms of degree $\leq R$ in *any* variable: in other words, for any term $\mathbf{a}$ in (44) with $0 < a_i \leq R$, we must have $p_\alpha(\mathbf{a}) = p_\beta(\mathbf{a}) = 0$. Using the form of $p_{\alpha,\beta}$ and the fact that $-p = p$ for $p \in \mathbb{F}_2$, we find a pair of conditions on the coefficients of P (one for each set of qubits), when $0 < a_i \leq R$,

$$\begin{aligned} p(\mathbf{a}) = p(a_1-1,a_2,a_3) + p(a_1,a_2-1,a_3) \\ +p(a_1,a_2,a_3-1); \end{aligned} \tag{45}$$

$$\begin{aligned} p(\mathbf{a}) = p(a_1-1,a_2-1,a_3) + p(a_1,a_2-1,a_3-1) \\ +p(a_1-1,a_2,a_3-1). \end{aligned} \tag{46}$$

Each of these relations is for one qubit at a given vertex, $\mathbf{a}$. As there are two equations for each vertex, the system appears to be over-determined. However, if we apply (46) to all terms on the RHS of (45) and vice-versa, we get the same equation, so that at a given vertex, the equations are redundant so long as A includes the neighboring vertices on the three faces of the elementary cube bounding the negative octant. See Fig. 7 (a) for a visual representation of this. These relations severely limit the degrees of freedom which may be used to build non-local surface stabilizers. All vertices in the 'positive octant' relative to the origin are fully fixed by the relations, and thus not available. This potentially leaves vertices in the three boundary planes containing the origin. However even here, every vertex in the outer surfaces of A adjacent to these planes no longer has the required neighbors to make (45) and (46) equivalent. To visualize this, picture the structure in Fig. 7(a) "puncturing" the surface of A so as to dangle into B. Whenever this happens, we gain one additional relation and so to satisfy all equations, we must fix a coefficient in the boundary planes for every puncture. However, because the boundary planes 'wrapping' A have larger surface area than A, this leaves



(a)

(b)

FIG. 7. Visual representation of Eqs. (45) in blue and (46) in red. (a) Representation for any vertex in A (black vertex). (b) Representation of equations in a 2D slice near the intersection of two boundary planes (white vertices).

a few degrees of freedom left over, out of which we can build NLSSs. This can be seen in Fig, 7(b) which depicts a 2D slice and the relevant punctures. The number of vertices in the boundary planes containing the origin is $3R^2 + 3R + 1$, where the $3R^2$ comes from the faces, the $3R$ from the edges, and the $+1$ from the point at the origin. Meanwhile, the number of vertices fixed by the adjacent surface layer of A is $3(R-1)^2 + 3(R-1) + 1$. This leaves $6R$ vertices that are not fixed by Eqs. (45) and (46). However, the origin corner does not count as a true boundary degree of freedom (because the stabilizer here contains a double identity, as discussed before). Generalizing this argument, we realize that at the intersection of boundary planes (not at a corner), there are 3 degrees of freedom to every one additional relation, thus giving 2 NLSSs per vertex in that intersection. Had we done the same analysis using stabilizers in B and on the boundary, we would find that this is reversed: there are 3 relations for each degree of freedom. This implies that such a system is over determined, yielding no solutions, so we conclude that $G_A \cap (\mathrm{Span}(\mathcal{B}_B) \times G_\partial)$ is trivial. Therefore, including the Z-type stabilizers (by doubling the answer above) we find that the basis for G_{NLSS} has $12R - 2$ elements in agreement with our earlier results for recoverable information.

We note that the process of finding NLSSs is similar to that of finding constraints: we are looking for solutions which remove support from some subset of the space. While the topological degeneracy of Haah's code is a complicated function of system size, and admits of a simple answer only for certain system sizes, the largest topological degeneracy that is currently known to occur is $\sim 4L$, for certain L. Our results suggest that $4L$ is not maximal, and that a degeneracy of $12L-2$ may be possible for certain boundary conditions, potentially including unequal lengths in each dimension. Using methods similar to those described above, it would be worth searching for the possible sequence of boundary conditions which saturate this upper bound. Such a sequence could be useful from the perspective of quantum error correcting codes as this would be a much denser coding than other models such as X-Cube or stacked toric codes.

We close this section by noting that while the Gauss's law picture for Haah's code is not presently clear, it would be an interesting topic for future work.

V. DISCUSSION: CONNECTIONS BETWEEN RECOVERABLE INFORMATION, NON-LOCAL ENTANGLEMENT, AND TOPOLOGICAL PROPERTIES

Now that we have calculated recoverable information, we briefly analyze the advantages and disadvantages of recoverable information over the non-local entanglement (NLE), given by multi-cut constructions such as those presented in 2, 3, 26, 36, and 46. The clear disadvantage is that we are restricted to stabilizer codes. However, when a stabilizer framework exists, there is real power in using recoverable information. The first and most important is the ability to extract the structure of Gauss-like laws associated to the boundaries of closed subsystems: every bit of recoverable information can be linked to an important piece of information about the excitations of the system. In contrast, as pointed out in Ref. 26 (in particular, the section on Haah's code), NLE calculations indicate long-range entanglement and can possibly distinguish different phases, but there is no universal interpretation of the exact value in $d > 3$ and the interpretation is quite complicated in $d = 3$[47]. Additionally, the Gauss laws accessed via the recoverable information can further be interpreted as braiding or braiding-like rules for the excitations (e.g., Eq. (24) or its generalization Eq. (31)). The boundary operator can be broken into pieces, namely the cut stabilizers, each of which must generate excitations (or else they would have been in either the stabilizer or logical group). Thus applying these operators sequentially is akin to braiding. For example, in the $d = 2$ toric code, $(A_v)_A$ along the boundary is an operator which hops an electric particle. So applying the full boundary operator, $(g_\partial)_A$, is equivalent to creating a pair of electric excitations and threading one around the boundary to annihilate with the other. Then by the Gauss law, Eq. 24, we know that if this process is sensitive to the parity of the number N_m of magnetic excitations it encircles: the resulting phase is -1^{N_m}. A similar picture can be given for all other Gauss law-type relations given above. Finally, as recoverable information is defined for any bipartition (as long as it is less than the code distance), one can ask about other patterns in the entanglement. The most obvious example is sharp-edge contributions in any d dimensional cluster model. Another example is the $d = 3$ toric code, where we conjecture that a more general formula for recoverable information is

$$\mu_{TC_3} = 1 + 2g, \tag{47}$$

where g is the genus of the boundary of A (we are assume that the bipartition is connected). For such general bipartitions, loop NLSS operators can be included in a minimal $G_{\rm NLSS}$ as they now satisfy statements (S2) and (S3), so long as they wrap non-contractible loops. For an example using the X-cube model, we consider more general but still zero-genus entanglement surfaces. Once again with the benefit of independent NLSS counting, one can see the only property of the surface which recoverable information depends on is the bounding rectangular parallelepiped. If the edge lengths of this parallelepiped are R_1, R_2 and R_3, then we conjecture the general formula for recoverable information is

$$\mu_{\rm XC} = 2(R_1 + R_2 + R_3 + 3), \tag{48}$$

where we once again have all the same dependency among the ribbon operators as before. Though we have assumed a set of principal axes (a dependency we naturally expect) and a certain kind of scale dependence, recoverable information demonstrates that there is a deformation invariance in the entanglement structure, that is in a sense topological. Presumably, Haah's code has a similar deformation invariance though we leave a more rigorous treatment of this to future work.

In summary, recoverable information is sensitive to finer but often relevant features in the entanglement structure that can be lost to multi-cut constructions. So even though its use is limited, when possible, it can give us a more complete picture of the structure of entangled and topological states of matter.

VI. CONCLUDING REMARKS

We have introduced a new concept - recoverable information - which is at present defined only for stabilizer codes, but which provides for a natural (albeit Hamiltonian dependent) characterization and interpretation of topological information, complementary to topological entanglement entropy. We have introduced three distinct ways of calculating recoverable information - via counting cut stabilizers, counting topological constraints, and counting non-local surface stabilizers. We have also illustrated these methods of calculating recoverable information for the cluster model in arbitrary dimension, the $d = 2$ and 3 toric codes and two archetypal 'fracton' models - the X-cube model and Haah's code. We argue that from recoverable information we may deduce the existence of Z_2 Gauss' law type constraints, which in turn suggests an interpretation in terms of emergent Z_2 conservation laws for point-like quasiparticles.

While our present discussion of recoverable information is restricted to stabilizer codes, it is worth noting that small perturbations about stabilizer codes can be treated using the method of Schrieffer-Wolff transformations (as discussed in [26]). This allows us to find 'dressed stabilizer' operators, which are eigenoperators of the ground state, and may provide a (Hamiltonian dependent) route to defining recoverable information outside of the stabilizer limit. This perspective appears similar in spirit to the Hamiltonian-dependent definition of "dressed" Wilson line operators by Hastings and Wen [48]. Finally, note that for any given state, one can always find a (not necessarily local) Hamiltonian for which this is the ground state. In this respect, it may be possible to extract recoverable information directly from a wavefunction (perhaps by minimizing over inferred Hamiltonians), thereby relaxing the Hamiltonian dependence of the definition. We leave such extensions of the concept of recoverable information, as well as further applications of the formalism introduced herein, to future work.

ACKNOWLEDGEMENTS

We acknowledge extremely useful discussions and a previous collaboration (Ref. 26) with M. Hermele, and thank B. Ware for valuable discussions. We are also very grateful to Jeongwan Haah for valuable comments on the manuscript. H.M. is supported by the U.S. Department of Energy, Office of Science, Basic Energy Sciences (BES) under Award number DE-SC0014415. SAP and RMN acknowledge support from the the Foundational Questions Institute (fqxi.org; grant no. FQXi-RFP-1617) through their fund at the Silicon Valley Community Foundation. This work was also supported in part by the NSF under Grant No. DMR-1455366 (SAP) at the University of California, Irvine.

Appendix A: Relating non-local surface stabilizers to recoverable information

In this appendix, we prove the claims made in the main text that connect non-local surface stabilizer groups to recoverable information. For the following, it is useful to introduce some notational conventions in order to work conveniently with subsets of the stabilizer set. We denote the *symmetric difference* of any two subsets $F, H \subseteq \mathcal{S}$, by $F \uplus H = F \cup H - F \cap H$. This is isomorphic to summation in $W[\mathcal{S}]$: we have

$$\begin{aligned}\prod_{s \in F \uplus H} s &= \prod_{s \in F\cup H - F\cap H} s \\ &= \prod_{s\in F - F\cap H} s \prod_{s' \in H - F\cap H} s' \prod_{s''\in F\cap H} (s'')^2 \\ &= \prod_{s\in F} s \prod_{s'\in H} s',\end{aligned} \tag{A1}$$

since any elements in both $F \cap H$ will square to the identity in the RHS. (On a more technical level, any power set equipped with a symmetric product forms an Abelian group; all we have done is to show that this group is isomorphic to $W[\mathcal{S}]$.) This allows us to use both set-theoretic and algebraic properties without constantly toggling between $W[\mathcal{S}]$ and $\mathbb{P}[\mathcal{S}]$.

We will also require a few standard properties of the symmetric difference. First, we note that it is both commutative and associative — these were already implicit in our statement that it lends an Abelian group structure to the power set. Another frequently-used relationship is that $F \uplus H = F \cup H$ when the two F, H are disjoint. Intersection on both right and left distributes over the symmetric difference, i.e. $J \cap (F \uplus H) = (F \uplus H) \cap J = (J \cap F) \uplus (J \cap H)$; however, set difference is only left distributive, i.e $(F \uplus H) - J = (F - J) \uplus (H - J)$. Also, we will frequently switch back and forth between a stabilizer group element and a set of stabilizers that generates it. Then, for any $F \subseteq \mathcal{S}$, we say $F \sim g \in G$ iff $\prod_{s\in F} s = g$. We use the same notation if two sets generate the same group member, i.e $F \sim H$ iff $\prod_{s\in F} s = \prod_{s\in H} s$. Clearly, this implies that if $F \sim H$ then $F \uplus H \in \mathcal{C}$. We also continue to employ the convention that if a subset $F \subseteq \mathcal{S}$ includes a subscript of A, B or ∂, then this represents the unique partitioning of that set into members only supported in A ($F_A = \{s \in F : \mathrm{Supp}(s) \subseteq A\}$), members only supported in B ($F_B = \{s \in F : \mathrm{Supp}(s) \subseteq B\}$), and members with support in both A and B ($F_\partial = \{s \in F : \mathrm{Supp}(s) \cap A \neq \emptyset$ and $\mathrm{Supp}(s) \cap B \neq \emptyset\}$).

First we prove Eq. (32), restated here for convenience:

$$G_{\mathrm{NLSS}} \simeq G_A/\mathrm{Span}(\mathcal{B}_A) \times G_B/\mathrm{Span}(\mathcal{B}_B). \tag{A2}$$

Proof. We prove (A2) by constructing a linear (in the sense of $V[G]$) bijective function, $\psi : G_{\mathrm{NLSS}} \to G_A/\mathrm{Span}(\mathcal{B}_A) \times G_B/\mathrm{Span}(\mathcal{B}_B)$.

Consider any $g_\partial \in G_{\mathrm{NLSS}}$. We know that there exists unique elements $g_\partial \sim F_\partial \subseteq B_\partial$, $g_A \in G_A$, and $g_B \in G_B$ such that $g_\partial = g_A g_B$. Furthermore, there exists a pair of sets $H, J \subseteq \mathcal{B}$ such that $g_A \sim H = H_A \cup H_B \cup H_\partial$ and $g_B \sim J = J_A \cup J_B \cup J_\partial$. Define, $g_A^{n\mathcal{B}} \sim H_B \cup H_\partial$, and $g_B^{n\mathcal{B}} \sim J_A \cup J_\partial$. Clearly, $g_A^{n\mathcal{B}} \in G_A/\mathrm{Span}(\mathcal{B}_A)$ and $g_B^{n\mathcal{B}} \in G_B/\mathrm{Span}(\mathcal{B}_B)$. As these sets uniquely represent these operators in the given basis, we define the map $g_\partial \mapsto \psi(g_\partial) = g_A^{n\mathcal{B}} g_B^{n\mathcal{B}}$, which makes ψ injective. Clearly, this is also linear.

Now consider any element of $g_A \in G_A/\mathrm{Span}(\mathcal{B}_A)$ and $g_B \in G_B/\mathrm{Span}(\mathcal{B}_B)$. Each of these has a unique representation in our given basis, $g_A \sim H = H_A \cup H_B \cup H_\partial \subseteq \mathcal{B}$ and $g_B \sim J = J_A \cup J_B \cup J_\partial \subseteq \mathcal{B}$, which by definition must be such that $H_A = J_B = \emptyset$. Using the stabilizer-generating equivalence we may write $H_\partial \sim g_A \prod_{s\in H_B} s$ and $J_\partial \sim g_B \prod_{s\in J_A} s$. Rearranging, we have $H_\partial \uplus J_\partial \sim (g_A \prod_{s\in J_A} s)(g_B \prod_{s\in H_B} s) \in G_A \times G_B$, which must also be in G_∂ and therefore in G_{NLSS}. So we have that for all $g_A g_B \in G_A/\mathrm{Span}(\mathcal{B}_A) \times G_B/\mathrm{Span}(\mathcal{B}_B)$, there exists a $g_\partial \in G_{\mathrm{NLSS}}$ such that $\psi(g_\partial) = g_A g_B$. Therefore, ψ is surjective and is thus a linear bijection. □

It is a corollary of the above argument that

$$\begin{aligned} G_{\mathrm{NLSS}} \simeq\, & (G_A \cap (\mathrm{Span}(\mathcal{B}_B) \times G_\partial)) \\ & \times G_B \cap (\mathrm{Span}(\mathcal{B}_A) \times G_\partial), \end{aligned} \tag{A3}$$

since $(G_A \cap (\mathrm{Span}(\mathcal{B}_B) \times G_\partial)) = G_A/\mathrm{Span}(\mathcal{B}_A)$ and $(G_B \cap (\mathrm{Span}(\mathcal{B}_A) \times G_\partial)) = G_B/\mathrm{Span}(\mathcal{B}_B)$.

Our ultimate goal is to prove the equivalence of (S1) -(S3). First, however, we prove a pair of lemmas that are both necessary to this goal, as well as being illuminating in their own right. Before proceeding, we recall the standard result that if $G_1, G_2 \subseteq G$ are both subgroups of G, then $G_1 \cap G_2$ is also a subgroup of G; we will use this in what follows without explicit comment.

Lemma 1. *Given two subgroups G_1 and G_2 of a stabilizer group G which we studied in this context, there exists a factorization for either group in the form $G_1 = (G_1 \cap G_2) \times [G_1/(G_1 \cap G_2)]$ or vice-versa, where $G_1/(G_1 \cap G_2)$ is the quotient group of G_1 with respect to $G_1 \cap G_2$.*

Proof. This follows from the fact that $G_1 \cap G_2$ is a subgroup of both G_1 and G_2, and is clearly normal as both $G_{1,2}$ are abelian. Therefore, we can construct the quotient group $G_1/(G_1 \cap G_2)$ and the factorization is an immediate consequence. Of course, there is some ambiguity in this factorization: any $g \in G_1/(G_1 \cap G_2)$ is equivalent to any other member g' of its coset, $g' \in g(G_1 \cap G_2) \subseteq G_1$. □

To remove ambiguity, we assume the factor $G_1/(G_1 \cap G_2)$ contains one and only one member of each coset, and in particular, I from the identity coset $I(G_1 \cap G_2)$, so that $G_1/(G_1 \cap G_2)$ is a proper subgroup.

Lemma 2. *(Basis Connection Lemma) For any basis $\mathcal{B}$, there exists a unique constraint basis $\mathfrak{C}_\mathcal{B}$ (i.e., a basis for the constraint subspace $\mathcal{C}$), with the following properties:*

(L2.1) Every stabilizer not in the basis, $s \in \mathcal{S} - \mathcal{B}$, belongs to a unique element C_s of the constraint basis, $s \in C_s \in \mathfrak{C}_\mathcal{B}$, and is in turn the unique non-basis stabilizer contained in C_s, i.e. $(C_s - \{s\}) \cap (\mathcal{S} - \mathcal{B}) = \emptyset$,

(L2.2) Any constraint $C \in \mathcal{C}$ is uniquely represented in the basis $\mathfrak{C}_\mathcal{B}$ by combining via symmetric difference the constraints $C_s \in \mathfrak{C}_\mathcal{B}$ that correspond under (L2.1) to non-basis elements of C, i.e. $C = \biguplus_{s\in C-\mathcal{B}} C_s$, and

(L2.3) Given any other basis, $\mathcal{B}'$, there exists a bijection $\psi : \mathcal{B} - \mathcal{B}' \to \mathcal{B}' - \mathcal{B}$, such that for all $s \in \mathcal{B} - \mathcal{B}'$, we have $s, \psi(s) \in C'_s \in \mathfrak{C}_{\mathcal{B}'}$.

(L2.3) gives this lemma its name. Stated simply, for every element in one basis but not the other, we have a constraint which can be used to exchange that element with one from the other basis, thereby connecting the two bases. See Fig. 8 for a pictorial depiction of this result.

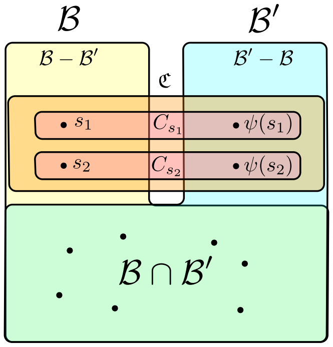


FIG. 8. Pictorial depiction of the basis connection lemma, specifically (L2.3).

Proof. Consider any element $s \in \mathcal{S}-\mathcal{B}$. As $s \notin \mathcal{B}$, and $\mathcal{B}$ is a basis, there must be a unique subset $F(s) \subseteq \mathcal{B}$ such that $s \sim F(s)$. It then follows that $F(s)\cup\{s\} = C_s \in \mathcal{C}$, i.e. is a constraint. C_s is uniquely specified by s and clearly $(C_s - \{s\}) \cap (\mathcal{S}-\mathcal{B}) = \emptyset$. Therefore, if we define $\mathfrak{C}_{\mathcal{B}} = \{C_s : s \in \mathcal{S}-\mathcal{B}\}$, this demonstrated (L2.1).

We now prove the set $\mathfrak{C}_{\mathcal{B}}$ is a basis for $\mathcal{C}$, first by showing it is independent. We do so by contradiction: assume that there exists some nonempty set $H \subseteq \mathcal{S}-\mathcal{B}$ such that $\emptyset = \biguplus_{s\in H} C_s = \biguplus_{s\in H}(F(s)\cup\{s\}) = \biguplus_{s\in H}(F(s)\uplus\{s\})$, where in the second step we used the fact that $F(s)\uplus\{s\} = F(s)\cup\{s\}$ since $F(s), \{s\}$ are disjoint. We then have, using the commutativity and associativity properties of the symmetric difference, that

$$\begin{aligned}\emptyset &= \left[\biguplus_{s\in H} F(s)\right] \uplus \left[\biguplus_{s\in H}\{s\}\right]\\ \implies \biguplus_{s\in H} F(s) &= \biguplus_{s\in H}\{s\} = H. \qquad \text{(A4)}\end{aligned}$$

This is a contradiction, since H is nonempty by assumption and therefore must contain at least one element of $\mathcal{S}-\mathcal{B}$, whereas $F(s)$ contains no elements of $\mathcal{S}-\mathcal{B}$ by (L2.1). As well as being independent, the set $\mathcal{C}_{\mathcal{B}}$ also contains the exact number of constraints needed to span the constraint set ($|\mathfrak{C}_{\mathcal{B}}| = |\mathcal{S}-\mathcal{B}| = |\mathcal{S}| - \dim(G) = \dim(\mathcal{C})$ as discussed in Section II B). Thus $\mathfrak{C}_{\mathcal{B}}$ is a constraint basis for $\mathcal{C}$.

To prove (L2.2), Let $C \in \mathcal{C}$ be any constraint. As the constraint subspace is closed under symmetric difference, the following must be a constraint:

$$\begin{aligned}\mathcal{C} \ni &\left\{\biguplus_{s\in C-\mathcal{B}} C_s\right\} \uplus C\\ &= \left\{\biguplus_{s\in C-\mathcal{B}} (F(s)\uplus\{s\})\right\} + C\\ &= \left\{\biguplus_{s\in C-\mathcal{B}} F(s)\right\} \uplus (C-\mathcal{B}) \uplus C\\ &= \left\{\biguplus_{s\in C-\mathcal{B}} F(s)\right\} \uplus (C\cap\mathcal{B}), \qquad \text{(A5)}\end{aligned}$$

where we have again used the equivalence of the union and symmetric difference of disjoint sets as well as the disjoint partitioning of $C = (C\cap\mathcal{B}) \cup (C-\mathcal{B}) = (C\cap\mathcal{B}) \uplus (C-\mathcal{B})$, which implies $(C-\mathcal{B})\uplus C = C\cap\mathcal{B}$. Both terms in (A5) are subsets of $\mathcal{B}$, again as $F(s)\subseteq\mathcal{B}$ by (L2.1), which means (A5) is a constraint among members of $\mathcal{B}$. As $\mathcal{B}$ is a basis, there can be no nonempty constraints among its members, so it follows that (A5) is an empty constraint. Therefore $\biguplus_{s\in C-\mathcal{B}} C_s = C$, which is the unique representation of C in the basis $\mathfrak{C}_{\mathcal{B}}$.

Finally, we prove (L2.3) by demonstrating the bijection ψ exists. We first show that $\bigcup_{s\in\mathcal{B}-\mathcal{B}'}(F'(s)-\mathcal{B}) = \mathcal{B}'-\mathcal{B}$, and then demonstrate that all the sets in the union are distinct and nonempty (the prime is included on $F'(s)$ to indicate that $F'(s)\subseteq\mathcal{B}'$, but is similarly defined by $s \sim F'(s)$). This immediately implies that we can construct a bijection that maps $s\in\mathcal{B}-\mathcal{B}'$ to some $\psi(s)\in F'(s)-\mathcal{B}\subseteq\mathcal{B}'-\mathcal{B}$. Evidently, $s, \psi(s) \in C'_s \in \mathfrak{C}_{\mathcal{B}'}$.

As $F'(s)\subseteq\mathcal{B}'$, we have trivially that $\bigcup_{s\in\mathcal{B}-\mathcal{B}'}(F'(s)-\mathcal{B}) \subseteq \mathcal{B}'-\mathcal{B}$. To show inclusion in the other direction, consider any $s'\in\mathcal{B}'-\mathcal{B}$. By (L2.1), we conclude that s' uniquely specifies a constraint $C_{s'} = F(s')\cup\{s'\}$ where $s' \sim F(s') \subseteq \mathcal{B}$. We also observe that $F(s')-\mathcal{B}' \neq \emptyset$. (For, if this were not the case, then $F(s')\subseteq\mathcal{B}'$, but then $C_{s'} = F(s')\cup\{s'\}\subseteq\mathcal{B}'$ would be a nonempty constraint among members of $\mathcal{B}'$, contradicting the fact that $\mathcal{B}'$ is a basis.) Then by (*L*2.2), the constraint $C_{s'}$ is represented in the $\mathfrak{C}_{\mathcal{B}'}$ basis by $C_{s'} = \sum_{s\in C_{s'}-\mathcal{B}'} C'_s = \sum_{s\in F(s')-\mathcal{B}'} C'_s$ (note we are using the fact that $s'\in\mathcal{B}'$ which implies $C_{s'}-\mathcal{B}' = F(s')-\mathcal{B}'$). As $s'\in C_{s'}$, this implies that $s' \in C'_s - \mathcal{B} = F'(s)-\mathcal{B}$ for some $s\in F(s')-\mathcal{B}'\subseteq\mathcal{B}-\mathcal{B}'$. Thus $\mathcal{B}'-\mathcal{B}\subseteq\bigcup_{s\in\mathcal{B}-\mathcal{B}'}(F'(s)-\mathcal{B})$, and we have proved inclusion in both directions, so that $\bigcup_{s\in\mathcal{B}-\mathcal{B}'}(F'(s)-\mathcal{B}) = \mathcal{B}'-\mathcal{B}$.

To show that no two sets of the form $F'(s)-\mathcal{B}$ are equal, suppose instead that there exist $s, s^*\in\mathcal{B}$ such that $F'(s)-\mathcal{B} = F'(s^*)-\mathcal{B}$. Using the disjoint partitioning of $F'(s) = (F'(s)\cap\mathcal{B})\uplus(F'(s)-B)$, and similarly for $F'(s^*)$, we find that $\{s\}\uplus\{s^*\} \sim (F'(s)\cap\mathcal{B})\uplus(F'(s^*)\cap\mathcal{B})$. From this, we have $\{s\}\uplus\{s^*\}\uplus(F'(s)\cap\mathcal{B})\uplus(F'(s^*)\cap\mathcal{B})\in\mathcal{C}$. As this constraint contains only members of a basis $\mathcal{B}$, it must be empty, and therefore from the uniqueness of $F'(s)$ and $F'(s^*)$, we must have $s = s^*$. Now $F'(s)-\mathcal{B}$

is nonempty for any s by an argument exactly like that used above to show $F(s') - \mathcal{B}' \neq \emptyset$.

From these results, we see that if we write $\bigcup_{s\in\mathcal{B}-\mathcal{B}'}(F'(s)-\mathcal{B}) = \mathcal{B}'-\mathcal{B}$, then there are exactly as many distinct nonempty sets in the union on the LHS as there are elements in $\mathcal{B}'-\mathcal{B}$, and every member of $\mathcal{B}'-\mathcal{B}$ is contained in at least one of the sets in the union. Thus, we can bijectively map s onto $\psi(s) \in F'(s)-\mathcal{B} \subseteq \mathcal{B}'-\mathcal{B}$, in which case $s, \psi(s) \in F'(s)\cup\{s\} = C'_s \in \mathfrak{C}_{\mathcal{B}'}$. □

One consequence of the basis connection lemma is that in minimizing over bases to find $\dim(G_{\text{NLSS}}) = \mu$, there are no 'local minima'. That is, every basis is directly connected to a minimal basis via some independent constraint set, and so the minimizing procedure described in the main text will eventually reach this minimal basis, which is necessary in order for it to reproduce the recoverable information μ.

With these lemmas, we are ready to prove the equivalence of (S1)-(S3), reproduced here for convenience:

(S1) $\dim(G_{\text{NLSS}}) = \mu$,

(S2) For any element $g_\partial \in G_{\text{NLSS}}$ with its associated $g_A \in G_A$ and $g_B \in G_B$ for which $g_\partial = g_A g_B$, if there exist sets of stabilizers, $F_A \subseteq \mathcal{S}_A$ and $F_B \subseteq \mathcal{S}_B$ whose product is g_A, g_B respectively, then $F_A \cup F_B \cup F_\partial \in \mathcal{T}$, where $F_\partial \subseteq \mathcal{B}_\partial$ is the basis set which generates g_∂ and

(S3) For all $g_\partial \neq I \in G_{\text{NLSS}}$ such that $F_\partial \subseteq \mathcal{B}_\partial$ is the basis set which generates g_∂, if there exists a non-topological constraint $C \in \mathcal{C}-\mathcal{T}$ such that $F_\partial \subseteq C$, then $C_\partial - F_\partial \neq \emptyset$.

Proof. Without loss of generality, let $\mathcal{S}$ be such that $\mathcal{T} = \{\emptyset\}$, as we can always change boundary conditions so that this is the case without changing μ. Let G_{NLSS} represent the non-local surface stabilizer group generated by the basis $\mathcal{B}$.

(S1) $\Longrightarrow$ (S2) (by contraposition). For the sake of argument, suppose there exists a $g_A g_B = g_\partial \in G_{\text{NLSS}}$ with $g_\partial \sim F_\partial \subseteq \mathcal{B}_\partial$, $g_A \in G_A$ and $g_B \in G_B$ such that there exist sets $F_A \subseteq \mathcal{S}_A$ and $F_B \subseteq \mathcal{S}_B$ with $g_A \sim F_A$ and $g_B \sim F_B$. Then, since $g_\partial = g_A g_B$ implies that $F_\partial \sim F_A \uplus F_B$, we have that $F_A \uplus F_B \uplus F_\partial = C \in \mathcal{C}$. Now, suppose this constraint is not topological, i.e. $C \notin \mathcal{T}$; since $\emptyset \in \mathcal{T}$, it follows that $C \neq \emptyset$. From (L2.2), we have that the expansion of C in the $\mathfrak{C}_\mathcal{B}$ basis is given by $C = \biguplus_{s\in C-\mathcal{B}} C_s$, where $C_s \in \mathfrak{C}_\mathcal{B}$ is the unique constraint associated with $s \in C - \mathcal{B}$. Let s_∂ be any member of $F_\partial \subseteq C$. According to the expansion of C, $s_\partial \in C_s$ for some $s \in C-\mathcal{B}$, which is clearly not a member of the basis. Also by (L2.1), $C_s - \{s\} \subseteq \mathcal{B}$. With this, consider the set $\mathcal{B}' = (\mathcal{B} - \{s_\partial\}) \cup \{s\}$. As any member of $\mathcal{B}$ is a product of members from $\mathcal{B}'$ (namely $s_\partial \sim C_s - \{s_\partial\} \subseteq \mathcal{B}'$), this is a basis for which $F_\partial \nsubseteq \mathcal{B}'$ and we have that $\mathcal{B}'$ generates $G'_{\text{NLSS}} \subsetneq G_{\text{NLSS}}$. Therefore, $\mathcal{B}$ is not minimal and by contraposition, (S1) implies (S2).

(S2) $\Longrightarrow$ (S3). Suppose G_{NLSS} satisfies (S2). Now suppose there exists $g_\partial \neq I \in G_{\text{NLSS}}$ such that $g_\partial \sim F_\partial \subseteq C \in \mathcal{C}$, where $F_\partial \subseteq \mathcal{B}_\partial$. It follows then that we can also write $g_\partial \sim C - F_\partial$. Using the standard decomposition of set C, we may rewrite this as $g_\partial \sim C_A \cup C_B \cup (C_\partial - F_\partial)$. Now, suppose, contrary to (S3), that $C_\partial - F_\partial = \emptyset$. Then, we have $g_\partial \sim C_A \cup C_B$, from which it follows that $g_\partial = g_A g_B$ with $G_A \ni g_A \sim C_A$ and $G_B \ni g_B \sim C_B$. Now, using (S2) we have that $C \in \mathcal{T}$, but since we also have $C \neq \emptyset$, this contradicts $\mathcal{T} = \{\emptyset\}$ (as we have assumed at the outset). So, it must be that $C_\partial - F_\partial \neq \emptyset$, which proves (S3).

(S3) $\Longrightarrow$ (S1). Finally, suppose G_{NLSS} satisfies (S3), and let $\mathcal{B}'$ be any other basis which generates the non-local surface stabilizer group G'_{NLSS}. We now construct an injective map $\chi : G_{\text{NLSS}} \to G'_{\text{NLSS}}$; this immediately requires that $\dim(G_{\text{NLSS}}) \leq \dim(G'_{\text{NLSS}})$ and hence we can conclude that the basis $\mathcal{B}$ is minimal.

First, we observe that G_{NLSS} and G'_{NLSS} are both subgroups of G. Using Lemma 1, and defining $G_\cap = G_{\text{NLSS}} \cap G'_{\text{NLSS}}$ we may factor $G_{\text{NLSS}} = G_\cap \times [G_{\text{NLSS}}/G_\cap]$. We now show that, given any $g_\partial \neq I \in G_{\text{NLSS}}/G_\cap$ with $g_\partial \sim F_\partial \subseteq \mathcal{B}_\partial$, $F_\partial - \mathcal{B}'$ is uniquely specified by g_∂. To see this, let $g_{\partial 1} \sim F_{\partial 1} \subseteq \mathcal{B}_\partial$ be any member of $G_{\text{NLSS}}/G_\cap$ such that $F_{\partial 1} - \mathcal{B}' = F_\partial - \mathcal{B}'$, and consider the disjoint partition $F_\partial = (F_\partial \cap \mathcal{B}') \uplus (F_\partial - \mathcal{B}')$, and similarly for $F_{\partial 1}$. Then $g_\partial g_{\partial 1} \sim F_\partial \cap \mathcal{B}' \uplus F_{\partial 1} \cap \mathcal{B}'$. Thus, $g_\partial g_{\partial 1}$ is generated by some subset of $\mathcal{B}'_\partial$, and is therefore contained in G'_{NLSS}; but, since $g_\partial, g_{\partial 1} \in \mathcal{B}_\partial$, they must both also be contained in G_{NLSS}. From this, we have that $g_\partial g_{\partial 1} \in G_\cap$, but as the factor $G_{\text{NLSS}}/G_\cap$ is closed and only trivially intersects $G_\cap$, it follows that $g_\partial = g_{\partial 1}$, i.e. $F_\partial - \mathcal{B}'$ is uniquely specified by g_∂ (for the given factorization of G_{NLSS} or, equivalently, unique to the coset of g_∂).

By (L2.1), for every $s \in F_\partial - \mathcal{B}'$ there exists a unique $C'_s \in \mathfrak{C}_{\mathcal{B}'}$ such that $s \in C'_s$ and $(C'_s - \{s\}) \cap (F_\partial - \mathcal{B}') = \emptyset$. As each member of $F_\partial - \mathcal{B}'$ is in one and only one of these constraints, we have $F_\partial - \mathcal{B}' \subseteq \biguplus_{s\in F_\partial-\mathcal{B}'} C'_s = C \in \mathcal{C}$. Furthermore, we have

$$\begin{aligned} C - (F_\partial - \mathcal{B}') &= \biguplus_{s\in F_\partial-\mathcal{B}'} C'_s - (F_\partial - \mathcal{B}') \\ &= \biguplus_{s\in F_\partial-\mathcal{B}'} (C'_s - \{s\}) \subseteq \mathcal{B}', \end{aligned} \tag{A6}$$

where we have used the left-distributivity of set difference over symmetric difference and the fact that $F_\partial - \mathcal{B}' \subseteq \mathcal{B}'$ with $(F_\partial - \mathcal{B}') \cap C'_s = \{s\}$. Note that the constraint C is unique to $F_\partial - \mathcal{B}'$ and thus unique to $g_\partial \in G_{\text{NLSS}}/G_\cap$. So, we have that $F_\partial - \mathcal{B}' \sim C - (F_\partial - \mathcal{B}') = C_A \uplus C_B \uplus (C_\partial - (F_\partial - \mathcal{B}'))$ (as $C_{A(B)} \cap (F_\partial - \mathcal{B}') = \emptyset$ by definition). Using this result and the disjoint partitioning of F_∂, we have

$$\begin{aligned} g_\partial \sim F_\partial =& (F_\partial \cap \mathcal{B}') \uplus (F_\partial - \mathcal{B}') \\ \sim& C_A \uplus C_B \uplus [\{F_\partial \cap \mathcal{B}'\} \uplus \{C_\partial - (F_\partial - \mathcal{B}')\}] \, . \end{aligned} \tag{A7}$$

But we also have that $g_\partial = g_A g_B$ for some $g_A \in G_A$ and $g_B \in G_B$. This implies that

$$\begin{aligned}&\{F_\partial \cap \mathcal{B}'\} \uplus \{C_\partial - (F_\partial - \mathcal{B}')\}\\ &\qquad \sim \left(g_A \prod_{s\in C_A} s\right)\left(g_B \prod_{s\in C_B} s\right) \in G'_{\mathrm{NLSS}},\end{aligned} \tag{A8}$$

where in the second line, we use $(F_\partial \cap \mathcal{B}') \uplus (C_\partial - (F_\partial - \mathcal{B}')) \subseteq \mathcal{B}'$ to argue that we have an element in G'_{NLSS}. From this, we can define a map $\chi : G_{\mathrm{NLSS}} \to G'_{\mathrm{NLSS}}$ such that

$$g_\partial \mapsto \chi(g_\partial) \sim (F_\partial \cap \mathcal{B}') \uplus (C_\partial - (F_\partial - \mathcal{B}'))\,. \tag{A9}$$

Note that for any $g_\partial \in G_\cap$ with $g_\partial \sim F_\partial$, $F_\partial - \mathcal{B}'$ is empty by definition; in this case it generates an empty constraint via the preceding construction. For such an element, $\chi(g_\partial) \sim F_\partial \cap \mathcal{B}' = F_\partial \sim g_\partial$.

We now show that χ is linear. To do this, consider any $g_{\partial 1}, g_{\partial 2} \in G_{\mathrm{NLSS}}$, with $g_{\partial 1} \sim F_{\partial 1} \subseteq \mathcal{B}$ and $g_{\partial 2} \sim F_{\partial 2} \subseteq \mathcal{B}$. This implies $g_{\partial 1} g_{\partial 2} \sim F_{\partial 1} \uplus F_{\partial 2} \subseteq \mathcal{B}$, which is its unique basis representation. By our previous arguments, we have the constraint $C_{12} \supseteq (F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}'$ uniquely associated with $(F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}'$. Moreover

$$\begin{aligned}C_{12} &= \biguplus_{s\in(F_{\partial 1}+F_{\partial 2})-\mathcal{B}'} C_s\\ &= \left\{\biguplus_{s\in F_{\partial 1}-\mathcal{B}'} C_s\right\} \uplus \left\{\biguplus_{s\in F_{\partial 2}-\mathcal{B}'} C_s\right\}\\ &= C_1 \uplus C_2,\end{aligned} \tag{A10}$$

where $C_{1(2)}$ is the unique constraint for $F_{\partial 1(2)} - \mathcal{B}'$ as discussed above. Here we have used the fact that for $s \in (F_{\partial 1} \cap F_{\partial 2}) - \mathcal{B}'$, C_s cancels out in the expansion of $C_1 \uplus C_2$, allowing us to combine the two sums into the one, yielding C_{12}. Using this relation, we have $F_{\partial 1} - \mathcal{B}' \sim C_1 - (F_{\partial 1} - \mathcal{B}')$, $F_{\partial 2} - \mathcal{B}' \sim C_2 - (F_{\partial 2} - \mathcal{B}')$ and $(F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}' \sim (C_1 \uplus C_2) - ((F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}')$. This in turn implies

$$\begin{aligned}(F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}' =&(F_{\partial 1} - \mathcal{B}') \uplus (F_{\partial 2} - \mathcal{B}')\\ &\sim (C_1 - (F_{\partial 1} - \mathcal{B}')) \uplus (C_2 - (F_{\partial 2} - \mathcal{B}'))\\ &\sim (C_1 \uplus C_2) - ((F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}')\,.\end{aligned} \tag{A11}$$

Since both sets on either side of the last $\sim$ relation are subsets of $\mathcal{B}'$, and any constraints among members of $\mathcal{B}'$ must be empty, we have that the last $\sim$ of Eq. (A11) can be replaced by set equality. To further simplify this equality, we recognize $C_{1(2)} = (C_{1(2)})_A \uplus (C_{1(2)})_B \uplus (C_{1(2)})_\partial$ and observe all sets being subtracted from these constraints in Eq. (A11) contain only cut stabilizers. In that case, $(C_{1(2)})_A \uplus (C_{1(2)})_B$ is unaffected and can be canceled from both sides of Eq. (A11) so that

$$\begin{aligned}&((C_1)_\partial - (F_{\partial 1} - \mathcal{B}')) \uplus ((C_2)_\partial - (F_{\partial 2} - \mathcal{B}'))\\ &= ((C_1)_\partial \uplus (C_2)_\partial) - ((F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}')\,.\end{aligned} \tag{A12}$$

We now compute the action of χ on the product $g_{\partial 1} g_{\partial 2}$ using the definition (A9):

$$\begin{aligned}\chi(g_{\partial 1} g_{\partial 2}) \sim&[(F_{\partial 1} \uplus F_{\partial 2}) \cap \mathcal{B}'] \uplus [\{(C_1)_\partial + (C_2)_\partial\}\\ &- \{(F_{\partial 1} \uplus F_{\partial 2}) - \mathcal{B}'\}]\\ &=(F_{\partial 1} \cap \mathcal{B}') \uplus [(C_1)_\partial - (F_{\partial 1} - \mathcal{B}')]\\ &\quad \uplus (F_{\partial 2} \cap \mathcal{B}') \uplus [(C_2)_\partial - (F_{\partial 2} - \mathcal{B}')]\\ &\sim\chi(g_{\partial 1})\chi(g_{\partial 2}),\end{aligned} \tag{A13}$$

where we have used Eq. (A12) and distributivity of intersection over symmetric difference. This completes the argument that χ is linear.

Up to this point, we have yet to use the fact that $\mathcal{B}$ and G_{NLSS} satisfy (S3). However, consider some $g_\partial \in G_{\mathrm{NLSS}}$ with $g_\partial \sim F_\partial \subseteq \mathcal{B}$ and suppose $\chi(g_\partial) = I$. Using the unique constraint $C \supseteq F_\partial - \mathcal{B}'$ associated with $F_\partial - \mathcal{B}'$, we can apply the definition of χ, which implies $F_\partial \cap \mathcal{B}' \uplus (C_\partial - (F_\partial - \mathcal{B}')) \in \mathcal{C}$. But all members of this constraint are members of $\mathcal{B}'$, which implies this constraint is empty. Thus $F_\partial \cap \mathcal{B}' = (C_\partial - (F_\partial - \mathcal{B}'))$. Using the disjoint partitioning of F_∂, this implies $F_\partial = (F_\partial \cap \mathcal{B}') \cup (F_\partial - \mathcal{B}') = (C_\partial - (F_\partial - \mathcal{B}')) \cup (F_\partial - \mathcal{B}') \subseteq C$. With this, we consider

$$\begin{aligned}C_\partial - F_\partial =&C_\partial - ((F_\partial \cap \mathcal{B}') \cup (F_\partial - \mathcal{B}'))\\ &= (C_\partial - (F_\partial \cap \mathcal{B}')) \cap (C_\partial - (F_\partial - \mathcal{B}'))\\ &= (C_\partial - (F_\partial \cap \mathcal{B}')) \cap (F_\partial \cap \mathcal{B}') = \emptyset,\end{aligned} \tag{A14}$$

using $F_\partial \cap \mathcal{B}' = (C_\partial - (F_\partial - \mathcal{B}'))$ and DeMorgan's rule for distributing set difference over union in the second line. But by our hypothesis, such a set can only be empty for $g_\partial = I$. Thus only $\chi(I) = I$, which as χ is linear, implies χ is injective (i.e has a trivial kernel).

Injectivity implies that the imagine of G_{NLSS} under χ has the same dimension as G_{NLSS} itself. As $\chi[G_{\mathrm{NLSS}}] \subseteq G'_{\mathrm{NLSS}}$, we therefore have that $\dim(G_{\mathrm{NLSS}}) = \dim(\chi[G_{\mathrm{NLSS}}]) \leq \dim(G'_{\mathrm{NLSS}})$, which implies $\mathcal{B}$ is minimal. □

As this proof is quite complicated, it is worth making a general observation. Essentially the proof constructs a 'canonical' mapping between any two non-local surface stabilizer groups G_{NLSS} and G'_{NLSS}. The map χ is well-defined regardless of whether or not (S3) is satisfied. (S3) only guarantees the resulting map is injective. The importance of the basis connection lemma in allowing us to construct a well-defined, linear χ, should be evident from the proof.

[1] T. Grover, Y. Zhang, and A. Vishwanath, New Journal of Physics **15**, 025002 (2013).

[2] A. Kitaev and J. Preskill, Phys. Rev. Lett. **96**, 110404 (2006).

[3] M. Levin and X.-G. Wen, Phys. Rev. Lett. **96**, 110405 (2006).

[4] C. Chamon, Phys. Rev. Lett. **94**, 040402 (2005).

[5] S. Bravyi, B. Leemhuis, and B. M. Terhal, Ann. Phys. **326**, 839 (2011).

[6] J. Haah, Phys. Rev. A **83**, 042330 (2011).

[7] B. Yoshida, Phys. Rev. B **88**, 125122 (2013).

[8] S. Vijay, J. Haah, and L. Fu, Phys. Rev. B **92**, 235136 (2015).

[9] S. Vijay, J. Haah, and L. Fu, Phys. Rev. B **94**, 235157 (2016).

[10] Z. Nussinov and G. Ortiz, Proceedings of the National Academy of Science **106**, 16944 (2009), cond-mat/0605316.

[11] Z. Nussinov and G. Ortiz, Annals of Physics **324**, 977 (2009).

[12] D. J. Williamson, Phys. Rev. B **94**, 155128 (2016).

[13] M. Pretko, Phys. Rev. B **95**, 115139 (2017).

[14] M. Pretko, Phys. Rev. B **96**, 035119 (2017).

[15] S. Vijay, arXiv e-prints (2017), 1701.00762.

[16] H. Ma, E. Lake, X. Chen, and M. Hermele, ArXiv e-prints (2017), arXiv:1701.00747 [cond-mat.str-el].

[17] A. Prem, J. Haah, and R. Nandkishore, Phys. Rev. B **95**, 155133 (2017).

[18] K. Slagle and Y. B. Kim, ArXiv e-prints (2017), arXiv:1704.03870 [cond-mat.str-el].

[19] T. H. Hsieh and G. B. Halász, ArXiv e-prints (2017), arXiv:1703.02973 [cond-mat.str-el].

[20] M. Pretko, Phys. Rev. B **96**, 125151 (2017).

[21] K. Slagle and Y. B. Kim, ArXiv e-prints (2017), arXiv:1708.04619 [cond-mat.str-el].

[22] T. Devakul, S. A. Parameswaran, and S. L. Sondhi, ArXiv e-prints (2017), arXiv:1709.10071 [cond-mat.str-el].

[23] O. Petrova and N. Regnault, ArXiv e-prints (2017), arXiv:1709.10094 [cond-mat.str-el].

[24] A. Prem, M. Pretko, and R. Nandkishore, ArXiv e-prints (2017), arXiv:1709.09673 [cond-mat.str-el].

[25] B. Shi and Y.-M. Lu, ArXiv e-prints (2017), arXiv:1705.09300 [cond-mat.str-el].

[26] H. Ma, A. T. Schmitz, S. A. Parameswaran, M. Hermele, and R. M. Nandkishore, Phys. Rev. B **97**, 125101 (2018).

[27] H. He, Y. Zheng, B. A. Bernevig, and N. Regnault, arXiv:1710.04220 (2017), arXiv:1710.04220 [cond-mat.str-el].

[28] D. Gottesman, Stabilizer codes and quantum error correction, Ph.D. thesis, California Institute of Technology (1997).

[29] J. Haah, Commun. Math. Phys. **324(2)**, 351 (2013).

[30] R. Raussendorf, S. Bravyi, and J. Harrington, Phys. Rev. A **71**, 062313 (2005).

[31] A. C. Doherty and S. D. Bartlett, Phys. Rev. Lett. **103**, 020506 (2009).

[32] S. Montes and A. Hamma, Phys. Rev. E **86**, 021101 (2012).

[33] V. Lahtinen and E. Ardonne, Phys. Rev. Lett. **115**, 237203 (2015).

[34] W. Son, L. Amico, R. Fazio, A. Hamma, S. Pascazio, and V. Vedral, EPL (Europhysics Letters) **95**, 50001 (2011).

[35] J. Cui, L. Amico, H. Fan, M. Gu, A. Hamma, and V. Vedral, Phys. Rev. B **88**, 125117 (2013).

[36] C. Castelnovo and C. Chamon, Phys. Rev. B **78**, 155120 (2008).

[37] S. Bravyi and B. Terhal, New Journal of Physics **11**, 043029 (2009).

[38] B. Yoshida and I. L. Chuang, Physical Review A **81**, 052302 (2010).

[39] J. Haah and J. Preskill, Physical Review A **86**, 032308 (2012).

[40] S. T. Flammia, J. Haah, M. J. Kastoryano, and I. H. Kim, arXiv preprint arXiv:1610.06169 (2016).

[41] L. Zou and J. Haah, Phys. Rev. B **94**, 075151 (2016).

[42] A. Hamma, R. Ionicioiu, and P. Zanardi, Physical Review A **71**, 022315 (2005).

[43] D. Fattal, T. S. Cubitt, Y. Yamamoto, S. Bravyi, and I. L. Chuang, arXiv preprint quant-ph/0406168 (2004).

[44] A. Nahum, J. Ruhman, S. Vijay, and J. Haah, ArXiv e-prints (2016), arXiv:1608.06950 [cond-mat.stat-mech].

[45] P. Bueno, R. C. Myers, and W. Witczak-Krempa, Phys. Rev. Lett. **115**, 021602 (2015).

[46] T. Grover, A. M. Turner, and A. Vishwanath, Phys. Rev. B **84**, 195120 (2011).

[47] X. Wen, H. He, A. Tiwari, Y. Zheng, and P. Ye, arXiv preprint arXiv:1710.11168 (2017).

[48] M. B. Hastings and X.-G. Wen, Phys. Rev. B **72**, 045141 (2005).