

# Limitations on Transversal Gates for Hypergraph Product Codes

Simon Burton<sup>ID</sup> and Dan Browne

**Abstract**—In a fault tolerant quantum computer, quantum codes are expected to serve the conflicting purposes of protecting quantum information while also allowing that information to be manipulated by fault-tolerant gates. We introduce a new technique for placing limitations on such gates, and apply this technique to a class of quantum codes known as hypergraph product codes contained within the vertical sector. These codes are constructed from input which is a pair of classical linear codes, and generalize the Kitaev surface code which is the hypergraph product of classical repetition codes. We provide a necessary condition on these input codes, under which the resulting hypergraph product code has transversal gates limited to the Clifford group. We conjecture that this condition is satisfied by all  $[n, k, d]$  Gallager codes with  $d \geq 3$  and  $k \leq n/2$ . This work is a generalization of an argument due to Bravyi and König, and we also conjecture this is a refinement of the recent notion of disjointness due to Jochym-O'Connor *et al.*

**Index Terms**—Computers and information processing, quantum computing.

## I. INTRODUCTION

SINCE the dawn of time humankind has striven to calculate and compute. More recently we strive to build a quantum computer. Like any computer, this has two ingredients: a reliable storage of information, and the ability to manipulate this information. (Input/output comes later.) These two ingredients are at odds with each other: one tries to prohibit any change, and the other is all about change. We seek a *quantum code* that can do both.

Because of this antagonistic relationship between state preservation and state manipulation, knowledge of one side often implies something about the other side. Indeed, the quantum code literature has many results that explore this boundary, both “no-go” and “go” results. The first go result

was the realization that CSS codes allow for transversal CNOT gates [1]–[3]. One of the first no-go results was by Eastin and Knill [4] which found that for stabilizer codes the transversal gateset is finite and therefore not universal. A common proof technique has emerged, for both go and no-go results. This technique relates a condition on the intersection of generic logical operators to the existence of transversal gates. Finding codes where the logical operators have large overlap with each other leads to transversal gates, for example the triorthogonal codes lead to transversal non-Clifford gates [5]–[7] as well as the color codes [8]. Conversely, showing that generic logical operators have small overlap with each other restricts the available transversal gates [9], [10]. These no-go results have also been weakened to apply to locality preserving logical gates, both in the stabilizer code setting [11], [12] and for topological quantum field theories [13].

The Kitaev code, which has the geometry of a torus, and its flat cousin, called the surface code, are the leading contenders for quantum codes [14]. Industry has this goal within sight. To reach universality with these codes we can do so via braiding [15], lattice surgery [16]–[18], or twists [19]. Other approaches use a 3D architecture [20], possibly with time [21], [22] as one of the dimensions. The problem with all these topological codes is that they have a vanishing rate: as we build larger codes to decrease the error rate, the number of logical qubits per physical qubit tends to zero.

Looking over the implementation horizon we find codes with vastly superior performance: the family of hypergraph product codes [23]. These still have a bounded number of interactions between physical qubits; with the additional difficulty that these are non-local interactions. The above story performed with the surface code has a parallel story for hypergraph product codes, which are a direct generalization of the surface code. Good decoders exist [24], and we can braid punctures to perform Clifford gates [25]. More generally, the homological product codes [26] can be made universal with code switching techniques [27].

In this work we show that a family of hypergraph product codes are unable to achieve non-Clifford gates transversally. This is a generalization of an argument of Bravyi and König [9] which applies to surface/toric codes. Note, we are specifically *not* referring to the main *theorem* of Bravyi and König [9], which applies to topological stabilizer codes of arbitrary dimension. Instead, we focus on a motivational heuristic from this paper, which we call the Bravyi-König *argument*. This relies on computing the intersections of two sets of generic logical operators, and showing that this

Manuscript received January 29, 2021; revised October 29, 2021; accepted November 10, 2021. Date of publication November 30, 2021; date of current version February 17, 2022. This work was supported by the QCDA Project through the Engineering and Physical Sciences Research Council (EPSRC) funded by the QuantERA ERA-NET Cofund in Quantum Technologies implemented within the European Union's Horizon 2020 Programme under Grant EP/R043647/1. An earlier version of this paper was presented at the TQC 2021. (Corresponding author: Simon Burton.)

Simon Burton is with the Department of Physics and Astronomy, University College London, London WC1E 6BT, U.K. He is now with the Institute of Physics, Jagiellonian University, 30-348 Kraków, Poland (e-mail: simon.burton@uj.edu.pl).

Dan Browne is with the Department of Physics and Astronomy, University College London, London WC1E 6BT, U.K. (e-mail: d.browne@ucl.ac.uk).

Communicated by M. M. Wilde, Associate Editor for Quantum Information Theory.

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TIT.2021.3131043>.

Digital Object Identifier 10.1109/TIT.2021.3131043

0018-9448 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.

See <https://www.ieee.org/publications/rights/index.html> for more information.

intersection is correctable. See Fig. 1. For the surface code, or toric code, this intersection has size  $O(1)$  and so cannot support any logical operator as the weight of these scales with the lattice size.

For hypergraph product codes, the situation is more delicate. In this case the intersection of generic logical operators has support on an extensive set of qubits, and so we cannot rely on constant distance bounds. This intersection region will have size  $O(k^2)$ , where  $k$  is the dimension of the underlying classical codespace. At first glance this region would seem unlikely to be correctable, as  $k$  is comparable to the distance of the quantum hypergraph product code. However, we show that under some mild assumptions on the underlying classical codes (we call this assumption *robustness*) this region is indeed correctable and so the Bravyi-König argument can be extended to these codes. This is the main result of the paper, Theorem 15 below.

The Bravyi and König argument has also been generalized to arbitrary stabilizer codes by the notion of *disjointness* due to Jochym-O'Connor *et al.* [10]. This however is still only sensitive to the distance of the code. Our techniques can therefore lead the way to a refinement of these disjointness bounds.

In summary, these hypergraph product codes may be excellent for protecting quantum information, but there are restrictions on how this quantum information can then be manipulated. This work is the first result demonstrating such restrictions.

## II. QUANTUM STABILIZER CODES

Fix a number of qubits  $n$ . The Pauli group  $\mathcal{P}_n$  is generated by  $n$ -fold tensor products of the Pauli matrices

$$I_2 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

and phases  $\{\pm 1, \pm i\}$ . Writing  $V = \mathbb{C}^2$  for the single qubit Hilbert space, the group  $\mathcal{P}_n$  acts tautologically on the  $n$ -qubit Hilbert space  $V^{\otimes n}$  of dimension  $2^n$ . Any subgroup  $S \leq \mathcal{P}_n$  will fix a subspace of  $V^{\otimes n}$  pointwise:

$$\text{Fix}_{V^{\otimes n}}(S) := \{v \in V^{\otimes n} \mid gv = v, \forall g \in S\},$$

and any subspace  $W \leq V^{\otimes n}$  will be *stabilized* by a subgroup of  $\mathcal{P}_n$ :

$$\text{Stab}_{\mathcal{P}_n}(W) := \{g \in \mathcal{P}_n \mid gw = w, \forall w \in W\}.$$

An abelian subgroup  $S \leq \mathcal{P}_n$  that does not contain  $-I_2$  has particularly nice structure. We call any such subgroup  $S$  a *stabilizer group*. Together with the tautological action this is known as a (quantum) *stabilizer code*. The subspace  $\text{Fix}_{V^{\otimes n}}(S)$  is the protected *codespace*.

Operators  $g \in \mathcal{P}_n$  built as tensor product of  $I_2$  and  $X$  are called  $X$ -type operators and denoted  $\mathcal{P}_n^X$ . Similarly, tensor products of  $I_2$  and  $Z$  are called  $Z$ -type operators and denoted  $\mathcal{P}_n^Z$ . If  $g \in \mathcal{P}_n$  commutes with every  $h \in S$ , then  $G$  will

preserve the codespace, without needing to fix it pointwise. Up to stabilizers, these are the *logical operators* of the code which form a group isomorphic to  $\mathcal{P}_k$ , for some  $k$ , and so there are  $k$  logical encoded qubits [28].

When  $S$  is generated by  $X$ -type or  $Z$ -type operators (inclusively) we call this a *CSS stabilizer code* [1], [29]. We state a simple lemma describing the logical operators of these codes. Enumerating the  $n$  qubits as  $\{1, \dots, n\}$ , we say that an operator  $g \in \mathcal{P}_n$  has *support* on a subset  $\gamma \subseteq \{1, \dots, n\}$  when the tensor factors of  $g$  are equal to  $I_2$  at indexes not in  $\gamma$ .

**Lemma 1:** In a CSS stabilizer code, if a set of qubits  $\gamma$  supports a non-trivial logical operator, then  $\gamma$  supports either a non-trivial  $X$ -type logical operator, or a non-trivial  $Z$ -type logical operator (or both).

*Proof:* An operator  $g \in \mathcal{P}_n$  commutes with every element of  $S$  exactly when it commutes with every generator of  $S$ . Therefore, given a logical operator that is built from both  $X$  and  $Z$  Pauli operators, replacing  $X$  by  $I$  will also give an operator that commutes with the stabilizers, and similarly replacing  $Z$  by  $I$  will give an operator that commutes with the stabilizers. One or both of these must then be a logical operator. ■

We recall the following definition and Lemma about the Clifford hierarchy [30], which is found in [11] Appendix A. The  $n$ -qubit Clifford hierarchy is a sequence  $\{\mathcal{C}_n^1, \mathcal{C}_n^2, \dots\}$  of sets of unitary operators,  $\mathcal{C}_n^l \subset U(V^{\otimes n})$ . These are defined inductively as  $\mathcal{C}_n^1 := \mathcal{P}_n$ , and

$$\mathcal{C}_n^{l+1} := \{u \in U(V^{\otimes n}) \mid upu^{-1} \in \mathcal{C}_n^l, \forall p \in \mathcal{P}_n\}$$

for  $l \geq 1$ . We also define  $\mathcal{C}_n^0$  to be the phase group  $\mathcal{C}_n^0 = \{\pm I_2^{\otimes n}, \pm i I_2^{\otimes n}\}$ . The sets  $\mathcal{C}_n^0, \mathcal{C}_n^1$  and  $\mathcal{C}_n^2$  are closed under inverse and multiplication and so these are groups. Specifically,  $\mathcal{C}_n^2$  is called the  $n$ -qubit *Clifford group*. For  $l > 2$ ,  $\mathcal{C}_n^l$  is no longer a group [5].

**Lemma 2:** A unitary  $u \in U(V^{\otimes n})$  is in the  $n$ -qubit Clifford group exactly when

$$[upu^{-1}, q] = upu^{-1}qup^{-1}u^{-1}q^{-1} \in \mathcal{C}_n^0$$

for all  $p, q \in \mathcal{P}_n$ .

*Proof:* The set of all unitaries  $u \in U(V^{\otimes n})$  such that  $upu^{-1}p^{-1} = \pm I$  for all  $p \in \mathcal{P}_n$ , equals the Pauli group. Therefore the result follows from the definition of  $\mathcal{C}_n^2$ . ■

A *transversal gate* for a stabilizer code  $S \leq \mathcal{P}_n$  is an operator of the form  $u = u_1 \otimes \dots \otimes u_n$  with each  $u_i$  an arbitrary unitary on  $V$ , such that  $u$  commutes with the projector onto the codespace. Just like the logical operators of the code, these transversal gates also preserve the codespace, without needing to fix it pointwise. The set of all such transversal gates is a subgroup of  $U(V)^{\otimes n}$ .

For any operator  $u \in U(V)^{\otimes n}$  the *support* of  $u$  is the set of qubits, or indices, on which  $u$  acts non-trivially. Given a stabilizer code  $S$ , a subset of the qubits is called *correctable* when the only logical operators supported on the subset are in  $\mathcal{C}_n^0$ . Any logical operator in  $\mathcal{C}_n^0$  will be called *trivial logical operator*.

The central idea behind the proof of Theorem 15 is summarized in the following result.

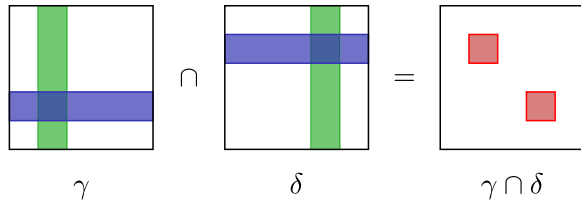


Fig. 1. The intersection of the support of two complete sets of logical operators is correctable under the hypothesis of Theorem 15.

**Theorem 3 (Bravyi-König Argument [9]):** Let  $S$  be a stabilizer code such that the intersection of the support of some representatives for any two logical operators is a correctable set of qubits. Then the action of any transversal gate for  $S$  on the  $k$  logical qubits is an element of the Clifford group  $\mathcal{C}_k^2$ .

*Proof:* This is a consequence of the previous lemma. Let  $u \in U(V)^{\otimes n}$  be a transversal gate for  $S$ . This operator restricts to a unitary on the logical qubits  $\tilde{u} \in U(V^{\otimes k})$ . Similarly for logical operators  $p, q \in \mathcal{P}_n$  and restriction  $\tilde{p}, \tilde{q} \in \mathcal{P}_k$ . Then  $[upu^{-1}, q] \in U(V)^{\otimes n}$  is non-trivial only on the intersection of the support of  $p$  and  $q$ . Because this region is correctable, we have  $[upu^{-1}, q] \in \mathcal{C}_n^0$ , which implies  $[\tilde{u}\tilde{p}\tilde{u}^{-1}, \tilde{q}] \in \mathcal{C}_k^0$ . ■

Note that in the application of this theorem below we choose representatives for the logical operators when computing these intersections. This means, for example, that a logical operator may have trivial intersection with itself, by using two different representatives.

The groups  $\mathcal{P}_n^X$  and  $\mathcal{P}_n^Z$  are abelian, and moreover, these groups are isomorphic to the additive group of the  $\mathbb{F}_2$ -linear vector space  $\mathbb{F}_2^n$ . In the remainder of this work, we switch to this additive group notation. Products of operators are computed as sums of  $\mathbb{F}_2$  vectors. Commutators of  $X$ -type and  $Z$ -type operators are calculated via the evident  $\mathbb{F}_2$  inner product. For  $u \in \mathbb{F}_2^n$  an  $X$ -type operator, and  $v \in \mathbb{F}_2^n$  a  $Z$ -type operator, this is just  $u^\top v \in \mathbb{F}_2$ .

### III. LINEAR CODES

We work using linear vector spaces over the field with two elements  $\mathbb{F}_2$ . Such vector spaces are constructed as  $\mathbb{F}_2^n$  for some number  $n$ . In order to simplify notation, we denote such vector spaces of dimension  $n$  using boldface:  $\mathbf{n} := \mathbb{F}_2^n$ . The idea is to confuse the distinction between the vector spaces and their dimensions. This is because the one determines the other, and calculations done with dimensions often have formally identical calculations done with vector spaces. Dimensions of direct sums add, dimensions of tensor products multiply, etc.

In this paper we make use of an arrow notation for linear maps. In terms of matrices, a linear map  $\mathbf{n} \xrightarrow{f} \mathbf{m}$  has  $n$  columns and  $m$  rows. This is a kind of “dimensional analysis” that keeps track of what we are allowed to multiply  $f$  with. Any arrow terminating on  $\mathbf{n}$  can multiply on the right of  $f$ , and any arrow starting on  $\mathbf{m}$  can multiply on the left of  $f$ . This has the effect, as with all dimensional analysis, of suggesting calculations to do, by combining arrows together. We use this in the proof of Lemma 7 below.

A linear map  $\mathbf{n} \xrightarrow{\partial} \mathbf{m}$  has a *kernel* which records dependencies among the columns of  $\partial$ . This is a linear map  $\mathbf{k} \xrightarrow{\ker(\partial)} \mathbf{n}$  such that  $\partial \cdot \ker(\partial) = 0$ . Furthermore, this map is characterized by a universal property: any other linear map  $\bullet \xrightarrow{f} \mathbf{n}$  with  $\partial f = 0$  factors uniquely through  $\ker(\partial)$ . In other words, there is a unique map  $f' : \bullet \rightarrow \mathbf{k}$  such that  $f = \ker(\partial) \cdot f'$ .

Similarly, the *cokernel* records dependencies among the rows of  $\partial$ . This is a linear map  $\mathbf{m} \xrightarrow{\text{coker}(\partial)} \mathbf{k}^\top$  such that  $\text{coker}(\partial) \cdot \partial = 0$ . (Here, the symbol  $^\top$  in the notation  $\mathbf{k}^\top$  is a formal decoration on the symbol  $\mathbf{k}$ .) Any other linear map  $\mathbf{m} \xrightarrow{g} \bullet$  with  $g\partial = 0$  factors uniquely through  $\text{coker}(\partial)$ . Evidently we have the relation  $\ker(\partial)^\top = \text{coker}(\partial^\top)$ .

Using both the kernel and the cokernel we can construct the following sequence of linear maps:

$$0 \rightarrow \mathbf{k} \xrightarrow{\ker(\partial)} \mathbf{n} \xrightarrow{\partial} \mathbf{m} \xrightarrow{\text{coker}(\partial)} \mathbf{k}^\top \rightarrow 0.$$

This sequence is exact, so we have the fundamental identity

$$k - n + m - k^\top = 0. \quad (1)$$

This is the Euler characteristic of this sequence. (See the book [31], in particular section 5.9, for a user-friendly introduction to sequences, homology and the Euler characteristic.)

The *span* of  $\mathbf{n} \xrightarrow{\partial} \mathbf{m}$  is a linear map  $\mathbf{m} - \mathbf{k}^\top \xrightarrow{\text{span}(\partial)} \mathbf{m}$  that factors through  $\partial$ . This map is also characterized by a universal property: any other linear map with codomain  $\mathbf{m}$  that factors through  $\partial$  factors uniquely through  $\text{span}(\partial)$ . The span is also called the column-span, or image. One way to construct  $\text{span}(\partial)$  is to take  $m - k^\top$  linearly independent columns of the matrix for  $\partial$ . Dually, the *cospan* of  $\mathbf{n} \xrightarrow{\partial} \mathbf{m}$  is a linear map  $\mathbf{n} \xrightarrow{\text{cospan}(\partial)} \mathbf{n} - \mathbf{k}$  that factors through  $\partial$  (on the other side). The universal property characterizing the cospan is that any other linear map with domain  $\mathbf{n}$  factoring through  $\partial$  will factor uniquely through  $\text{cospan}(\partial)$ . The cospan is also called the row-span. One way to construct  $\text{cospan}(\partial)$  is to take  $n - k$  linearly independent rows of the matrix for  $\partial$ . From Eq. (1) we find that  $m - k^\top = n - k$  which is the familiar fact that the column-span and row-span of a matrix have the same dimension.

A classical linear code is usually defined to be a subspace of some finite dimensional vector space  $\mathbb{F}_2^n$ . We are taking a relentlessly “active” approach to linear algebra: the maps are primary, not the spaces. For our purposes, a *classical linear code* is a linear map  $\mathbf{n} \xrightarrow{\partial} \mathbf{m}$ . The *codespace* is the kernel of this map. We will avoid the temptation to define the cocodespace as the cokernel. We also call  $\mathbf{n} \xrightarrow{\partial} \mathbf{m}$  a *parity check matrix*, and  $\ker(\partial)^\top$  the *generator matrix* for this code.

A *quantum code* is defined to be a pair of linear maps

$$\mathbf{m}_Z \xrightarrow{H_Z^\top} \mathbf{n} \xrightarrow{H_X} \mathbf{m}_X$$

such that  $H_X H_Z^\top = 0$ . From this identity it follows we can take the quotients

$$L_Z^\top = \ker(H_X) / \text{span}(H_Z^\top), \quad L_X = \text{coker}(H_Z^\top) / \text{cospan}(H_X)$$

which we call the *logical operators* for the code. We will use the following block matrix notation for these quotients:

$$L_Z^\top = \left( \ker(H_X) \parallel H_Z^\top \right), \quad L_X = \left( \begin{array}{c|c} \text{coker}(H_Z^\top) \\ \hline H_X \end{array} \right),$$

where the dashed line indicates the modulo (quotient). This is read as “logical operators modulo stabilizers.” The dashed line separates the subspace of  $\mathbf{n}$  spanned by logical operators from the subspace of  $\mathbf{n}$  spanned by stabilizer operators.

*Example 4:* The Steane code is a quantum code that uses 7 qubits, with 3 stabilizers of  $X$ -type and 3 stabilizers of  $Z$ -type:

$$\mathbf{3} \xrightarrow{H_Z^\top} \mathbf{7} \xrightarrow{H_X} \mathbf{3}$$

The matrices for  $H_X$  and  $H_Z$  are the same:

$$H_X = H_Z = \begin{pmatrix} 1 & 1 & 1 & 1 & . & . & . \\ 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 \end{pmatrix},$$

$$\ker(H_X) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}, \quad \text{coker}(H_Z^\top) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{pmatrix},$$

and we package all this information into the block matrix notation for the logical operators:

$$L_Z^\top = \left( \ker(H_X) \parallel H_Z^\top \right) = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & . \\ 1 & 1 & . & 1 \\ 1 & 1 & . & . \\ 1 & . & 1 & 1 \\ 1 & . & 1 & . \\ 1 & . & . & 1 \end{pmatrix},$$

$$L_X = \left( \begin{array}{c|c} \text{coker}(H_Z^\top) \\ \hline H_X \end{array} \right) = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & . & . & . \\ 1 & 1 & . & . & 1 & 1 & . \\ 1 & . & 1 & . & 1 & . & 1 \end{pmatrix}.$$

#### IV. PUNCTURES

We write the generic set of size  $j$  as  $[j]$ , which we also identify with the canonical basis of  $\mathbb{F}_2^j$ . Given a subset  $\gamma \subset [j]$ , we say a vector  $v \in \mathbb{F}_2^j$  has *support on*  $\gamma$  if the non-zero components of  $v$  form a subset of  $\gamma$ , in symbols:  $\{i | v_i \neq 0\} \subset \gamma$ .

A linear map  $G : \mathbf{n} \rightarrow \mathbf{k}$  is called *e-puncturable*, for some natural number  $e$ , if there exists a subset  $\gamma \subset [n]$  of size  $e$  such that no non-zero vector in the cospan of  $G$  has support on  $\gamma$ . In terms of the matrix for  $G$  this corresponds to being

able to delete the columns indexed by  $\gamma$  while maintaining the rank of  $G$ . We call such a set  $\gamma$  an *e-puncture* of  $G$ , or a *puncture* of  $G$  when  $e$  is understood. When  $G$  is the generator matrix of a classical code we recover the notion of puncturing a classical code ( $e$  times) well known in the literature; in this case  $\gamma$  corresponds to a correctable set of erasures for the code [32]. However, we also apply this definition to arbitrary linear maps.

Going further, a linear map  $G : \mathbf{n} \rightarrow \mathbf{k}$  is called *e-bipuncturable*, for some natural number  $e$ , if there exists two disjoint subsets  $\gamma \subset [n], \delta \subset [n]$ , such that  $\gamma$  is an  $e$ -puncture of  $G$  and  $\delta$  is an  $e$ -puncture of  $G$ . We call such a pair of sets  $(\gamma, \delta)$  an *e-bipuncture* of  $G$  or a *bipuncture* of  $G$  when  $e$  is understood.

*Example 5:* The repetition code  $\mathbf{4} \xrightarrow{\partial} \mathbf{3}$  is given by the parity check matrix

$$\partial = \begin{pmatrix} 1 & 1 & . & . \\ . & 1 & 1 & . \\ . & . & 1 & 1 \end{pmatrix}.$$

This linear map is only 1-puncturable because it contains all the weight two vectors in its cospan. (If we added a column of zeros to this matrix then there would be certain weight two vectors missing from the cospan, and the map would become 2-puncturable.) The generator matrix for the code  $\mathbf{4} \xrightarrow{\partial} \mathbf{3}$  is given by  $G = \ker(\partial)^\top = (1111)$ , and we can choose any three element subset of  $[n = 4]$  as a puncture of  $G$ . Moreover, we can choose any two element subset  $\gamma \subset [4]$  as a 2-puncture, and then letting  $\delta = [4] - \gamma$ , we see that  $\delta$  is also a 2-puncture of  $G$  such that  $\gamma \cap \delta$  is empty. Therefore we have demonstrated that  $G$  is 2-bipuncturable. ■

More generally, given a classical code with distance  $d$ , any subset of  $[n]$  with size less than  $d$  serves as a puncture for the generating matrix, and so this matrix is  $(d - 1)$ -puncturable.

*Lemma 6:* Any linear map  $\partial : \mathbf{n} \rightarrow \mathbf{m}$  with  $\ker(\partial) : \mathbf{k} \rightarrow \mathbf{n}$ , is  $k$ -puncturable.

*Proof:* We can row-reduce  $\partial$  without changing any of the punctures. By reordering the columns, the row-reduced  $\partial$  has block matrix form

$$\partial = \begin{pmatrix} I_{n-k} & J \\ 0 & 0 \end{pmatrix}$$

where  $J$  is an arbitrary matrix with  $k$  columns. Puncture these  $k$  columns. ■

#### V. HYPERGRAPH PRODUCT CODES

Given classical codes  $A = \{\mathbf{n}_a \xrightarrow{\partial_a} \mathbf{m}_a\}$  and  $B = \{\mathbf{n}_b \xrightarrow{\partial_b} \mathbf{m}_b\}$  we define a quantum code  $A \otimes B$  as:

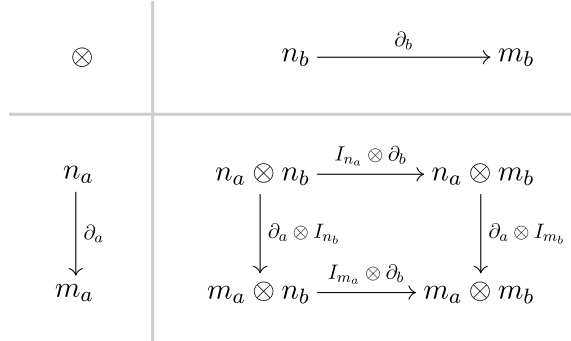
$$\mathbf{n}_a \otimes \mathbf{n}_b \xrightarrow{H_Z^\top} \mathbf{n}_a \otimes \mathbf{m}_b \oplus \mathbf{m}_a \otimes \mathbf{n}_b \xrightarrow{H_X} \mathbf{m}_a \otimes \mathbf{m}_b. \quad (2)$$

Using block matrix notation,

$$H_Z^\top = \begin{pmatrix} I_{n_a} \otimes \partial_b \\ \partial_a \otimes I_{n_b} \end{pmatrix}, \quad H_X = \begin{pmatrix} \partial_a \otimes I_{m_b} & I_{m_a} \otimes \partial_b \end{pmatrix}. \quad (3)$$



A quick calculation shows that  $H_X H_Z^\top = 2\partial_a \otimes \partial_b = 0$  thanks to  $\mathbb{F}_2$  arithmetic, and so this is indeed a quantum code. This *hypergraph product code*  $A \otimes B$  can also be notated as the following diagram



where the inner square is not commutative in general. We are using the tensor symbol for the hypergraph product  $A \otimes B$ , because this is indeed the tensor product of  $A$  and  $B$  as chain complexes. This product is also known as the *homological product* [26].

The following result counts the number of independent logical operators. A similar result is found in the [23], and for completeness we provide another proof.

**Lemma 7:** With the above notation, the quantum code  $A \otimes B$  has  $k$  logical qubits given by  $k = k_a k_b^\top + k_a^\top k_b$ .

*Proof:* By inspecting (3) we find linear maps:

$$\mathbf{k}_a \otimes \mathbf{k}_b \xrightarrow{\ker(H_Z^\top)} \mathbf{n}_a \otimes \mathbf{n}_b$$

and

$$\mathbf{m}_a \otimes \mathbf{m}_b \xrightarrow{\text{coker}(H_X)} \mathbf{k}_a^\top \otimes \mathbf{k}_b^\top.$$

Now use this to extend the sequence (2):

$$\begin{aligned} 0 &\rightarrow \mathbf{k}_a \otimes \mathbf{k}_b \xrightarrow{\ker(H_Z^\top)} \mathbf{n}_a \otimes \mathbf{n}_b \\ &\xrightarrow{H_Z^\top} \mathbf{n}_a \otimes \mathbf{m}_b \oplus \mathbf{m}_a \otimes \mathbf{n}_b \\ &\xrightarrow{H_X} \mathbf{m}_a \otimes \mathbf{m}_b \xrightarrow{\text{coker}(H_X)} \mathbf{k}_a^\top \otimes \mathbf{k}_b^\top \rightarrow 0. \end{aligned}$$

This sequence is exact everywhere except in the middle where the homology is  $k$  (dimensional). Therefore, the Euler character is:

$$\begin{aligned} k &= k_a k_b - n_a n_b + n_a m_b + m_a n_b - m_a m_b + k_a^\top k_b^\top \\ &= k_a k_b + (n_a - m_a)(m_b - n_b) + k_a^\top k_b^\top \\ &= k_a k_b + (k_a - k_a^\top)(k_b^\top - k_b) + k_a^\top k_b^\top \quad \text{using (1)} \\ &= k_a k_b^\top + k_a^\top k_b. \end{aligned}$$

The qubits (vectors) in  $\mathbf{n}_a \otimes \mathbf{m}_b$  are called *vertical* qubits, and the qubits in  $\mathbf{m}_a \otimes \mathbf{n}_b$  are called *horizontal* qubits. This convention is motivated by the following example.

**Example 8 (Surface Code):** Continuing to use the above notation, we take  $A = \{\mathbf{n}_a \xrightarrow{\partial_a} \mathbf{m}_a\}$  to be a repetition code

and  $B = \{\mathbf{n}_b \xrightarrow{\partial_b} \mathbf{m}_b\}$  to be the dual (transpose) code:

$$\partial_a = \begin{pmatrix} 1 & 1 & . & . \\ . & 1 & 1 & . \\ . & . & 1 & 1 \end{pmatrix}, \quad \partial_b = \begin{pmatrix} 1 & . & . \\ 1 & 1 & . \\ . & 1 & 1 \\ . & . & 1 \end{pmatrix}.$$

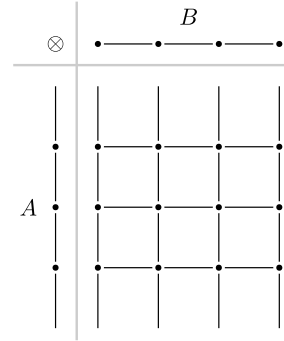
We have  $\partial_a : 4 \rightarrow 3$  and  $\partial_b : 3 \rightarrow 4$ , and also

$$\begin{aligned} k_a &= 1, & k_b &= 0, \\ k_a^\top &= 0, & k_b^\top &= 1. \end{aligned}$$

The quantum code  $A \otimes B$  is then:

$$4 \otimes 3 \xrightarrow{H_Z^\top} 3 \otimes 3 \oplus 4 \otimes 4 \xrightarrow{H_X} 3 \otimes 4.$$

Pictorially this corresponds to  $4 \times 3$  faces,  $3 \times 3$  horizontal edges,  $4 \times 4$  vertical edges, and  $3 \times 4$  vertices:



The surface code is a peculiar example because we are able to combine the  $3 \otimes 3$  horizontal qubits and the  $4 \otimes 4$  vertical qubits together into this single picture. This does not work in general; in Fig. 2 below the sectors are shown separately. ■

The next theorem describes the structure of the logical operators as illustrated in Fig. 2. It is a concrete restatement of the formula  $k = k_a k_b^\top + k_a^\top k_b$  from the previous lemma. The standard basis for  $\mathbf{n}_a$  we write as  $[n_a]$ , similarly for  $\mathbf{m}_a, \mathbf{m}_b$  and  $\mathbf{n}_b$ . Notation such as  $[n_a] \times [m_b]$  denotes the basis for  $\mathbf{n}_a \otimes \mathbf{m}_b$ , which is the space for the vertical qubits.

**Theorem 9:** Let  $A \otimes B$  be a hypergraph product code as above. A complete set of  $L_Z$  operators is supported on the qubits:

$$\begin{aligned} (L_Z\text{-vertical}) & \quad [n_a] \times \gamma_{Zv} \text{ where } \gamma_{Zv} \text{ is any } k_b^\top\text{-puncture of } \partial_b^\top, \text{ and} \\ (L_Z\text{-horizontal}) & \quad \gamma_{Zh} \times [n_b] \text{ where } \gamma_{Zh} \text{ is any } k_a^\top\text{-puncture of } \partial_a^\top. \end{aligned}$$

A complete set of  $L_X$  operators is supported on the qubits:

$$\begin{aligned} (L_X\text{-vertical}) & \quad \gamma_{Xv} \times [m_b] \text{ where } \gamma_{Xv} \text{ is any } k_a\text{-puncture of } \partial_a, \text{ and} \\ (L_X\text{-horizontal}) & \quad [m_a] \times \gamma_{Xh} \text{ where } \gamma_{Xh} \text{ is any } k_b\text{-puncture of } \partial_b. \end{aligned}$$

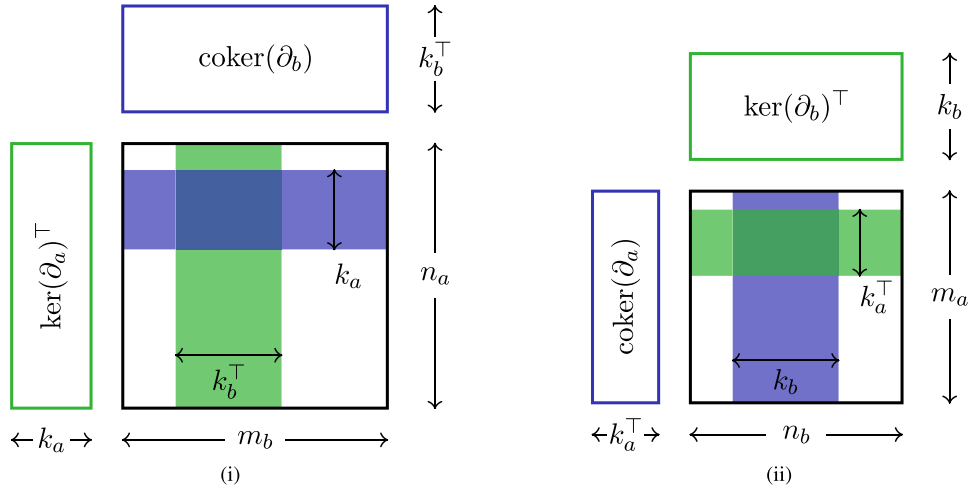


Fig. 2. Illustrating the construction of the logical operators in the hypergraph product code, from the underlying classical codes. (i) There are  $n_a m_b$  vertical qubits. A complete set of  $k_a k_b^T$  many purple  $L_X$  operators is supported on  $k_a$  many horizontal strips, each of which supports  $k_b^T$  many operators from the rows of  $\text{coker}(\partial_b)$ . Similarly, we have  $k_b^T k_a$  many green  $L_Z$  operators supported on vertical strips. (ii) The  $m_a n_b$  horizontal qubits support  $k_b k_a^T$  many purple  $L_X$ , and  $k_a^T k_b$  many green  $L_Z$  operators.

Specifically, we have the following block matrix form for the logical operators

$$\begin{aligned} L_Z^\top &= \begin{pmatrix} \ker(H_X) & \vdots & H_Z^\top \\ \ker(\partial_a) \otimes \gamma_{Zv}^\top & 0 & I_{n_a} \otimes \partial_b \\ 0 & \gamma_{Zh}^\top \otimes \ker(\partial_b) & \partial_a \otimes I_{n_b} \end{pmatrix}, \end{aligned}$$

where the basis for the rows is  $v = n_a \otimes m_b$  and  $h = m_a \otimes n_b$ , and

$$\begin{aligned} L_X &= \begin{pmatrix} \text{coker}(H_Z^\top) \\ H_X \end{pmatrix} \\ &= \begin{pmatrix} \gamma_{Xv} \otimes \text{coker}(\partial_b) & 0 \\ 0 & \text{coker}(\partial_a) \otimes \gamma_{Xh} \\ \partial_a \otimes I_{m_b} & I_{m_a} \otimes \partial_b \end{pmatrix}, \end{aligned}$$

where the basis for the columns is  $v$  and  $h$ . In this notation, a puncture  $\gamma \subseteq [n]$  is promoted to a  $|\gamma| \times n$  matrix by taking the rows of the identity matrix  $I_n$  indexed by  $\gamma$ . The dashed line separates “generators” from “relations,” as in Section III.

*Proof:* We claim that the kernel of  $H_X$  is spanned by the columns of a matrix as:

$$\begin{aligned} \ker(H_X) &= \\ \text{span} \left( \begin{pmatrix} \ker(\partial_a) \otimes I_{m_b} & 0 & I_{n_a} \otimes \partial_b \\ 0 & I_{m_a} \otimes \ker(\partial_b) & \partial_a \otimes I_{n_b} \end{pmatrix} \right), \end{aligned} \quad (4)$$

and the cokernel of  $H_Z^\top$  is spanned by the rows of a matrix as:

$$\text{coker}(H_Z^\top) = \text{cospan} \left( \begin{pmatrix} I_{n_a} \otimes \text{coker}(\partial_b) & 0 \\ 0 & \text{coker}(\partial_a) \otimes I_{n_b} \\ \partial_a \otimes I_{m_b} & I_{m_a} \otimes \partial_b \end{pmatrix} \right).$$

That these matrices lie within  $\ker(H_X)$  and  $\text{coker}(H_Z^\top)$  respectively can be seen by composing with  $H_X, H_Z^\top$  respectively. We next count the independent logical operators and

find  $k_a k_b^T + k_a^T k_b$  many, then by Lemma 7 we will have found all of them.

Let  $\gamma_{Zv}$  be any  $k_b^T$ -puncture of  $\partial_b^\top$ . This puncture always exists by Lemma 6. We claim that any solution of

$$\begin{pmatrix} \ker(\partial_a) \otimes \gamma_{Zv}^\top & 0 & I_{n_a} \otimes \partial_b \\ 0 & I_{m_a} \otimes \ker(\partial_b) & \partial_a \otimes I_{n_b} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$$

will have  $x = 0$ . To see this, we need only consider solutions of

$$(\ker(\partial_a) \otimes \gamma_{Zv}^\top)x + (I_{n_a} \otimes \partial_b)z = 0.$$

If  $z = 0$  then  $x = 0$  because  $\ker(\partial_a)$  has linearly independent columns. If  $z \neq 0$  then we have  $x = 0$  because  $\gamma_{Zv}^\top$  cannot be in the column space of  $\partial_b$ .

Counting the columns of  $\ker(\partial_a) \otimes \gamma_{Zv}^\top$  we find  $k_a k_b^T$ . Therefore, we have found  $k_a k_b^T$  linearly independent  $L_Z$  logical operators:

$$\begin{pmatrix} \ker(\partial_a) \otimes \gamma_{Zv}^\top & \vdots & I_{n_a} \otimes \partial_b \\ 0 & & \partial_a \otimes I_{n_b} \end{pmatrix}.$$

The other  $k_a^T k_b$  operators for  $L_Z$  are found similarly, and also for the  $L_X$  operators. ■

The form of the logical operators in the above theorem has a particular structure which we will use repeatedly below. An  $X$ -type or  $Z$ -type logical operator is called *taut* when it is “as straight and thin as possible.” Within the vertical sector  $\mathbf{n}_a \otimes \mathbf{m}_b$ , the taut  $L_Z$  logical operators are of the form  $v \otimes j$  with  $v \in \text{cospan}(\ker(\partial_a)^\top) \subset \mathbf{n}_a$  and  $j \in [m_b]$  a basis vector of  $\mathbf{m}_b$ . Taut  $L_X$  logical operators are of the form  $i \otimes u$  with  $u \in \text{cospan}(\text{coker}(\partial_b)) \subset \mathbf{m}_b$  and  $i \in [n_a]$  a basis vector of  $\mathbf{n}_a$ . See Fig. 2i. We view the space  $\mathbf{n}_a \otimes \mathbf{m}_b$  as an array with  $n_a$  rows and  $m_b$  columns, and see that a taut  $L_Z$  operator is supported on a vertical strip, and a taut  $L_X$  operator is supported on a horizontal strip. A similar definition holds for taut logical operators in the horizontal sector  $\mathbf{m}_a \otimes \mathbf{n}_b$ .

$\mathbf{n}_b$ . Note that taut operators are not necessarily of minimal weight, for example, the sum of taut operators supported on the same vertical (or horizontal) strip is another taut operator. In general there will be many more taut logical operators than are contained within the complete set of logical operators of Theorem 9.

*Example 10:* Continuing with the surface code from example 8, for the  $L_Z$ -vertical logical operators we compute  $\ker(\partial_a) \otimes \gamma_{Zv}^\top$ :

$$\ker(\partial_a) = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$

and we take  $\gamma_{Zv}$  to be a  $1 \times 4$  matrix that is non-zero at any one coordinate, to get a 1-puncture of  $\partial_b^\top$ . For the  $L_X$ -vertical logical operators we compute  $\gamma_{Xv} \otimes \text{coker}(\partial_b)$ . In this case we have  $\text{coker}(\partial_b) = (1111)$  and  $\gamma_{Xv}$  is a  $1 \times 4$  matrix non-zero at any one coordinate. The  $L_Z$ -horizontal logical operators are trivial because  $k_a^\top = 0$ , and the  $L_X$ -horizontal logical operators are trivial because  $k_b = 0$ . ■

## VI. MAIN RESULT

We say that a hypergraph product code  $A \otimes B$  is *contained within the vertical sector* when  $\ker(\partial_b)$  is trivial and/or  $\text{coker}(\partial_a)$  is trivial. See Fig. 2. This happens precisely when  $k_a^\top k_b = 0$ . The surface code of Example 8 is one such code. The toric code is a counter-example, in this case  $k_a = k_a^\top = k_b = k_b^\top = 1$ . The next lemma shows how to restrict Theorem 9 to the vertical sector.

*Lemma 11:* Given a hypergraph product code  $A \otimes B$  contained within the vertical sector, we have (i) a complete set of  $L_Z$  operators is supported on  $[n_a] \times \gamma_Z$  where  $\gamma_Z$  is any  $k_b^\top$ -puncture of  $\partial_b^\top$ , and (ii) a complete set of  $L_X$  operators is supported on  $\gamma_X \times [m_b]$  where  $\gamma_X$  is any  $k_a$ -puncture of  $\partial_a$ .

*Proof:* We use Theorem 9 and assume  $k_a^\top k_b = 0$ . If  $k_b = 0$  then  $\ker(\partial_b) = 0$  so  $L_Z$ -horizontal is trivial, and any  $k_b$ -puncture of  $\partial_b$  is empty so  $L_X$ -horizontal is trivial. Similarly, if  $k_a^\top = 0$  then  $\text{coker}(\partial_a)$  is trivial so  $L_X$ -horizontal is trivial, and any  $k_a^\top$ -puncture of  $\partial_a^\top$  is empty so  $L_Z$ -horizontal is trivial. The remaining operators are  $L_Z$ -vertical and  $L_X$ -vertical, as required. ■

In Theorem 9 we showed how representatives of the logical operators can be chosen to be taut. The next result shows how more general logical operators can be decomposed into taut logical operators.

*Lemma 12:* Given a hypergraph product code  $A \otimes B$  contained within the vertical sector, any  $Z$ -type (resp.  $X$ -type) logical operator supported on the vertical qubits is a product of disjoint taut  $Z$ -type (resp.  $X$ -type) logical operators.

*Proof:* From the form of the logical operators in the proof of Theorem 9, Eq. (4), we have the block matrix

$$L_Z^\top = \left( \begin{array}{cc|c} \ker(\partial_a) \otimes I_{m_b} & 0 & I_{n_a} \otimes \partial_b \\ 0 & I_{m_a} \otimes \ker(\partial_b) & \partial_a \otimes I_{n_b} \end{array} \right).$$

This notation is to be read as “column vectors modulo column vectors” with the dashed line acting as the modulo. Assuming

the hypothesis we have that  $k_a^\top k_b = 0$  which implies, also from Theorem 9, that the second column is trivial modulo the third column. This gives

$$L_Z^\top = \left( \begin{array}{c|c} \ker(\partial_a) \otimes I_{m_b} & I_{n_a} \otimes \partial_b \\ \hline 0 & \partial_a \otimes I_{n_b} \end{array} \right).$$

An arbitrary logical  $Z$ -type operator  $l_Z^\top$  is got from the span of this matrix, which by the hypothesis must be zero on the bottom row:

$$l_Z^\top = L_Z^\top \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} z \\ 0 \end{pmatrix},$$

for some matrices  $x, y, z$ . Solving this equation in generality, we require  $(\partial_a \otimes I_{n_b})y = 0$  which gives  $y = \ker(\partial_a) \otimes I_{n_b}$ . Expanding out, we get:

$$\begin{aligned} z &= (\ker(\partial_a) \otimes I_{m_b})x + \ker(\partial_a) \otimes \partial_b \\ &= (\ker(\partial_a) \otimes I_{m_b})x + (\ker(\partial_a) \otimes I_{m_b})(I_{k_a} \otimes \partial_b) \\ &= (\ker(\partial_a) \otimes I_{m_b})(x + I_{k_a} \otimes \partial_b). \end{aligned}$$

This means that  $l_Z^\top$  is got from the span of

$$\begin{pmatrix} \ker(\partial_a) \otimes I_{m_b} \\ 0 \end{pmatrix}$$

which is precisely the form of disjoint  $Z$ -type operators we seek. Note that individual columns of this matrix may not be disjoint, but the sum of any non-disjoint columns will be a new taut logical operator. See Example 13 below. The expression  $\ker(\partial_a) \otimes I_{m_b}$  can be read as taking logical operators from the  $A$  code and “copying” them onto vertical strips of the  $n_a \times m_b$  array of vertical qubits; see Fig. 2.

The same calculation transposed shows the result for  $X$ -type logical operators:

$$\begin{aligned} L_X &= \left( \begin{array}{cc|c} I_{n_a} \otimes \text{coker}(\partial_b) & 0 & \\ \hline 0 & \text{coker}(\partial_a) \otimes I_{n_b} & \\ \hline \partial_a \otimes I_{m_b} & I_{m_a} \otimes \partial_b & \end{array} \right) \\ &= \left( \begin{array}{cc|c} I_{n_a} \otimes \text{coker}(\partial_b) & 0 & \\ \hline \partial_a \otimes I_{m_b} & I_{m_a} \otimes \partial_b & \end{array} \right), \end{aligned}$$

and so on. ■

So far, we have seen that we can construct a complete set of logical operators from taut logical operators (Theorem 9 and Lemma 11) and conversely, how to write a more general logical operator as a product of disjoint taut logical operators (Lemma 12). The following example is intended to clarify the relationship between taut logical operators in general, and those that we use to form a complete set of logical operators.

*Example 13:* We take  $A = \{3 \xrightarrow{\partial_a} 1\}$  and  $B = \{1 \xrightarrow{\partial_b} 2\}$  with

$$\begin{aligned} \partial_a &= (1 \ 1 \ 1), \quad \ker(\partial_a) = \begin{pmatrix} 1 & . \\ 1 & 1 \\ . & 1 \end{pmatrix}, \\ \partial_b &= \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad \text{coker}(\partial_b) = (1 \ 1). \end{aligned}$$

The  $A$  code has parameters  $n_a = 3, m_a = 1, k_a = 2, k_a^\top = 0$ , and the  $B$  code has parameters  $n_b = 1, m_b = 2, k_b = 0, k_b^\top = 1$ , therefore we have  $A \otimes B$  contained within the vertical sector. We focus on the logical  $Z$ -type operators. From Theorem 9 we choose a 1-puncture  $\gamma_{Zv} = (1 \ 0)$  of  $\partial_b^\top = (1 \ 1)$ . Then the logical  $Z$ -type operators correspond to the columns of the matrix:

$$L_Z^\top = \begin{pmatrix} \ker(\partial_a) \otimes \gamma_{Zv}^\top & I_{n_a} \otimes \partial_b \\ 0 & \partial_a \otimes I_{n_b} \end{pmatrix}.$$

Ignoring the bottom row, which corresponds to the horizontal sector, we compute:

$$\ker(\partial_a) \otimes \gamma_{Zv}^\top = \begin{pmatrix} 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \\ 1 & 1 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \end{pmatrix}.$$

Each column of this matrix is a taut logical operator, and these columns are not disjoint. So it would seem that this presents a counterexample to Lemma 12, as the logical operator defined by the sum of these columns is a sum of non-disjoint taut logical operators. However, the sum is itself another taut logical operator and so this is not a counterexample to Lemma 12. Finally, we obtain another set of taut logical operators by choosing the other 1-puncture of  $\partial_a^\top$ , which is  $\gamma_{Zv} = (0 \ 1)$ . ■

Two regions (subsets)  $\alpha, \beta \subset [n_a] \times [m_b]$  are called *horizontally separated* when for any  $j \in [m_b]$  we have that  $[n_a] \times \{j\}$  intersects at most one of  $\alpha$  or  $\beta$ . Similarly,  $\alpha$  and  $\beta$  are called *vertically separated* when for any  $i \in [n_a]$  we have that  $\{i\} \times [m_b]$  intersects at most one of  $\alpha$  or  $\beta$ .

**Lemma 14 (Union Lemma):** Consider a hypergraph product code  $A \otimes B$  contained within the vertical sector, with two regions  $\alpha, \beta \subset [n_a] \times [m_b]$  that are separated horizontally and vertically. If there exists a non-trivial logical operator supported on  $\alpha \cup \beta$  then there exists a non-trivial logical operator supported on  $\alpha$ , or there exists a non-trivial logical operator supported on  $\beta$  (inclusive or).

*Proof:* By Lemma 1 we need only consider  $X$ -type and  $Z$ -type operators. Suppose that  $\alpha \cup \beta$  supports a non-trivial  $X$ -type logical operator. Then by the previous lemma,  $\alpha \cup \beta$  must support a non-trivial taut  $X$ -type operator. This operator is supported on  $\{i\} \times [m_b]$  for some  $i \in [n_a]$ , which intersects at most one of  $\alpha$  or  $\beta$  because these are vertically separated. Therefore we must have that at most one of  $\alpha$  or  $\beta$  supports this taut  $X$ -type logical operator. Using the horizontal separation of  $\alpha$  and  $\beta$  a similar argument applies to the  $Z$ -type logical operators. ■

We call a classical code  $\{n \xrightarrow{\partial} m\}$  *robust* if the generator matrix  $\ker(\partial)^\top$  and the parity check matrix  $\partial$  are simultaneously  $k$ -bipuncturable, where  $k$  is the dimension of the codespace.

**Theorem 15:** Let  $A = \{\mathbf{n}_a \xrightarrow{\partial_a} \mathbf{m}_a\}$  and  $B = \{\mathbf{n}_b \xrightarrow{\partial_b} \mathbf{m}_b\}$  be classical codes such that  $A \otimes B$  is contained within

the vertical sector. If  $A$  is robust and  $B^\top = \{\mathbf{m}_b \xrightarrow{\partial_b^\top} \mathbf{n}_b\}$  is robust, then any transversal gate for  $A \otimes B$  is restricted to the Clifford group  $\mathcal{C}^2$ .

*Proof:* We assume the hypothesis of the theorem. Let  $(\gamma_Z, \delta_Z)$  be a simultaneous  $k_b^\top$ -bipuncture of  $\partial_b^\top$  and  $\ker(\partial_b^\top)^\top = \text{coker}(\partial_b)$ . Let  $(\gamma_X, \delta_X)$  be a simultaneous  $k_a$ -bipuncture of  $\partial_a$  and  $\ker(\partial_a)^\top$ . From Lemma 11 we have a complete set of  $L_Z$  operators supported on  $[n_a] \times \gamma_Z$  and a complete set of  $L_Z$  operators supported on  $[n_a] \times \delta_Z$ . Similarly for  $L_X$  operators supported on  $\gamma_X \times [m_b]$  and  $\delta_X \times [m_b]$ . Combining these, we find that  $\gamma := \gamma_X \times [m_b] \cup [n_a] \times \gamma_Z$  supports a complete set of logical operators, as does  $\delta := \delta_X \times [m_b] \cup [n_a] \times \delta_Z$ . See Fig. 1. We now claim that the intersection of these two sets  $\gamma \cap \delta$  is correctable, that is, does not support any non-trivial logical operator. The result then follows from Theorem 3.

Using the disjointness of  $\gamma_X$  and  $\delta_X$ , and the disjointness of  $\gamma_Z$  and  $\delta_Z$  we find

$$\begin{aligned} \gamma \cap \delta &= (\gamma_X \times [m_b] \cup [n_a] \times \gamma_Z) \cap (\delta_X \times [m_b] \cup [n_a] \times \delta_Z) \\ &= (\gamma_X \times [m_b] \cap [n_a] \times \delta_Z) \cup ([n_a] \times \gamma_Z \cap \delta_X \times [m_b]) \end{aligned}$$

and the RHS then satisfies the hypothesis of Lemma 14. Therefore we need only show the correctability of the two pieces,  $\alpha := \gamma_X \times [m_b] \cap [n_a] \times \delta_Z$  and  $\beta := [n_a] \times \gamma_Z \cap \delta_X \times [m_b]$ .

From the hypothesis,  $\delta_Z$  punctures  $\text{coker}(\partial_b)$  and so  $\alpha$  cannot support any taut  $L_X$  logical operator, and by Lemma 12 cannot support arbitrary  $L_X$  logical operator. Also,  $\gamma_X$  punctures  $\ker(\partial_a)^\top$  and so  $\alpha$  cannot support any taut  $L_Z$  logical operator, and by Lemma 12 cannot support arbitrary  $L_Z$  logical operator. By Lemma 1 we find that  $\alpha$  cannot support any logical operator, and so  $\alpha$  is correctable. A similar argument applies to show that  $\beta$  is correctable. ■

By swapping the factors  $A \otimes B$  to  $B \otimes A$  we see that this theorem also applies with “vertical sector” replaced with “horizontal sector.”

In the next section we show that some mild conditions on a classical code will guarantee it is robust.

## VII. CHARACTERIZING ROBUSTNESS

In this section we take  $H = \partial$  to be a parity check matrix that has full rank, without loss of generality. Adding linear dependant rows to  $H$  does not change the punctures on  $H$ . We also write  $G = \ker(H)^\top$  for the generator matrix.

**Lemma 16 (Canonical Form):** Any  $k$ -dimensional code with length  $n$  has canonical form for generator matrix  $G$  and parity check matrix  $H$ ,

$$\begin{aligned} G &= (I_k \quad J) \\ H &= (J^\top \quad I_m) \end{aligned}$$

up to reordering of columns, with  $m = n - k$ .

*Proof:* Row-reduction and column reordering gives the form

$$\begin{aligned} G &= (I_k \quad J) \\ H &= (K \quad I_m). \end{aligned}$$

Using  $HG^\top = 0$  we get  $K = J^\top$ . ■



Writing such a code in this form, the set of indices  $\{1, \dots, k\}$  are called *pivots* and the set of indices  $\{k+1, \dots, n\}$  are called *copivots*. We also use this terminology for matrices in arbitrary row-reduced form (not necessarily column re-ordered to make the pivots sequential).

Next, we show how to puncture the copivots.

**Lemma 17 (Copivot Lemma):** With  $G$  written in canonical form  $G = (I_k \ J)$  any set of indices  $\gamma \subset \{k+1, \dots, n\}$  punctures  $G$ .

*Proof:* For  $v \in \mathbb{F}_2^k$  any non-zero vector,  $vG$  has non-zero components on  $\{1, \dots, k\}$  and so cannot be supported on  $\gamma$ . ■

Slightly more tricky is puncturing the pivots.

**Lemma 18 (Pivot Lemma):** With  $G$  written in canonical form  $G = (I_k \ J)$  a set of indices  $\gamma \subset \{1, \dots, k\}$  punctures  $G$  if and only if  $\gamma$  punctures  $\text{coker}(J)$ .

*Proof:* The only way  $\gamma \subset \{1, \dots, k\}$  can support a non-zero vector in the cospan of  $G$  is for  $\gamma$  to index a linear dependant set of rows of  $J$ . This is precisely the cokernel of  $J$ . ■

The result we are leading to characterizes robust classical codes as being equivalent to a row-reduction condition on the generator matrix. The proof is by successive applications of the Pivot Lemma and the Copivot Lemma.

**Theorem 19 (Robustness Theorem):** Let  $A$  be a classical code of length  $n$ , with generator matrix  $G$ . Then  $A$  is robust if and only if there is some permutation of the columns of  $G$  such that  $G$  has a canonical form  $G = (I_k \ J)$  with  $J$  full rank.

*Proof:* Firstly, to prove the “if” part, we assume that  $G = (I_k \ J)$  with  $J$  full rank. We will find disjoint sets  $\delta, \gamma \subset [n]$  that simultaneously puncture  $G$  and  $H$ .

By Lemma 16, this code has parity check matrix

$$H = (J^\top \ I_m)$$

where  $m = n - k$ . Choose  $\gamma = [k]$ , so that  $\gamma$  is supported on the pivots of  $G$  and the copivots of  $H$ . Because  $J$  is full rank, the cokernel of  $J$  is trivial and so by the Pivot Lemma we have that  $\gamma$  punctures  $G$ . From the Copivot Lemma we have that  $\gamma$  punctures  $H$ .

Also by the Copivot Lemma we have any  $\delta \subset \{k+1, \dots, n\}$  punctures  $G$ , we just need to find such a  $\delta$ , with  $|\delta| = k$ , that also punctures  $H$ . This  $\delta$  is a subset of the pivots of  $H$  so we use the Pivot Lemma to see that  $\delta$  must also puncture  $\text{coker}(J^\top) = \ker(J)^\top$ . We show how to achieve this. The matrix  $J$  is full rank so from Eq. (1) a matrix for  $\text{coker}(J^\top)$  has  $m-k$  rows and  $m$  columns. This  $(m-k) \times m$  matrix is full rank and so has  $m-k$  pivots and  $k$  copivots. We therefore choose  $\delta$  to be these  $k$  copivots of  $\text{coker}(J^\top)$  and by the Copivot Lemma,  $\delta$  will puncture  $\text{coker}(J^\top)$ .

Secondly, to prove the “only if” part, assume we have disjoint sets  $\gamma, \delta \subset [n]$  that simultaneously puncture  $G$  and  $H$ . The goal is to find a canonical form  $G = (I_k \ J)$  with  $J$  full rank. To do this, permute the columns of  $G$  and  $H$  so that  $\gamma = \{1, \dots, k\}$  and  $\delta = \{k+1, \dots, 2k\}$ . With this column order, row-reduce  $G$  to produce the form  $G = (I_k \ J)$ . Using the Pivot lemma, we see that  $\gamma = \{1, \dots, k\}$  punctures

$\text{coker}(J)$  which implies this cokernel is trivial, and so  $J$  is full-rank. ■

## VIII. DISCUSSION

We have shown that this ensemble of good quantum codes nevertheless fails to support transversal gates beyond the Clifford group. This no-go result rules out a wide class of codes, and so assists the search (construction) of codes that do have non-Clifford transversal gates.

The toric code, which is a hypergraph product code, does not satisfy the hypothesis of Theorem 15. This theorem applies to codes which are a generalization of the surface code. It is a natural question to ask if this result can be strengthened to cover these other hypergraph product codes. However, the argument as it stands cannot be strengthened: we have found examples where the region  $\gamma \cap \delta$  (see Fig. 1) is not correctable, even though it satisfies the hypothesis of the Union Lemma. But it may be that a more refined argument does work: numerics suggest that the support of specific commutators  $pqp^{-1}q^{-1}$  of logical operators  $p, q$  is always correctable.

The reader may wonder exactly how “mild” is the robustness condition. We numerically searched through the Gallager code ensembles used for building hypergraph product codes in [33], [34], and did not find any non-robust instance. Specifically, we found evidence that any  $[n, k, d]$  Gallager code with  $d \geq 3$  and  $k \leq n/2$  is robust.

Another natural question to ask is the existence of transversal gates for higher degree hypergraph (homological) product of classical codes  $A \otimes B \otimes C$ , etc. [35]. We expect these codes also have a theory of taut logical operators. Given the intricacy of the calculations in the present work, it seems that better techniques, and deeper insights, are needed before these results can be pushed further.

## ACKNOWLEDGMENT

The authors would like to thank Armanda Quintavalle, Joschka Roffe, Earl Campbell, Michael Vasmer and Paul Webster for useful discussions. They also thank one of the reviewers for suggesting Example 13 which helped to clarify the definition of the taut logical operators.

## REFERENCES

- [1] D. Gottesman, “Class of quantum error-correcting codes saturating the quantum Hamming bound,” *Phys. Rev. A, Gen. Phys.*, vol. 54, no. 3, p. 1862, 1996.
- [2] A. R. Calderbank and P. W. Shor, “Good quantum error-correcting codes exist,” *Phys. Rev. A, Gen. Phys.*, vol. 54, no. 2, p. 1098, 1996.
- [3] A. M. Steane, “Error correcting codes in quantum theory,” *Phys. Rev. Lett.*, vol. 77, no. 5, p. 793, 1996.
- [4] B. Eastin and E. Knill, “Restrictions on transversal encoded quantum gate sets,” *Phys. Rev. Lett.*, vol. 102, no. 11, Mar. 2009, Art. no. 110502.
- [5] B. Zeng, X. Chen, and I. L. Chuang, “Semi-clifford operations, structure of  $C_k$  hierarchy, and gate complexity for fault-tolerant quantum computation,” *Phys. Rev. A, Gen. Phys.*, vol. 77, no. 4, Apr. 2008, Art. no. 042313.
- [6] S. Bravyi and J. Haah, “Magic-state distillation with low overhead,” *Phys. Rev. A, Gen. Phys.*, vol. 86, no. 5, Nov. 2012, Art. no. 052329.
- [7] C. Vuillot and N. P. Breuckmann, “Quantum pin codes,” 2019, *arXiv:1906.11394*.
- [8] H. Bombín, “Gauge color codes: Optimal transversal gates and gauge fixing in topological stabilizer codes,” *New J. Phys.*, vol. 17, no. 8, Aug. 2015, Art. no. 083002.

- [9] S. Bravyi and R. König, "Classification of topologically protected gates for local stabilizer codes," *Phys. Rev. Lett.*, vol. 110, no. 17, Apr. 2013, Art. no. 170503.
- [10] T. Jochym-O'Connor, A. Kubica, and T. J. Yoder, "Disjointness of stabilizer codes and limitations on fault-tolerant logical gates," *Phys. Rev. X*, vol. 8, no. 2, May 2018, Art. no. 021047.
- [11] F. Pastawski and B. Yoshida, "Fault-tolerant logical gates in quantum error-correcting codes," *Phys. Rev. A, Gen. Phys.*, vol. 91, no. 1, Jan. 2015, Art. no. 012305.
- [12] P. Webster and S. D. Bartlett, "Locality-preserving logical operators in topological stabilizer codes," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 1, Jan. 2018, Art. no. 012330.
- [13] M. E. Beverland, O. Buerschaper, R. Koenig, F. Pastawski, J. Preskill, and S. Sijher, "Protected gates for topological quantum field theories," *J. Math. Phys.*, vol. 57, no. 2, Feb. 2016, Art. no. 022201.
- [14] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, "Topological quantum memory," *J. Math. Phys.*, vol. 43, no. 9, pp. 4452–4505, 2002.
- [15] A. G. Fowler, M. Mariantoni, J. M. Martinis, and A. N. Cleland, "Surface codes: Towards practical large-scale quantum computation," *Phys. Rev. A, Gen. Phys.*, vol. 86, no. 3, Sep. 2012, Art. no. 032324.
- [16] C. Horsman, A. G. Fowler, S. Devitt, and R. V. Meter, "Surface code quantum computing by lattice surgery," *New J. Phys.*, vol. 14, no. 12, Dec. 2012, Art. no. 123011.
- [17] A. G. Fowler and C. Gidney, "Low overhead quantum computation using lattice surgery," 2018, *arXiv:1808.06709*.
- [18] D. Litinski, "A game of surface codes: Large-scale quantum computing with lattice surgery," *Quantum*, vol. 3, p. 128, Mar. 2019.
- [19] B. J. Brown, K. Laubscher, M. S. Kesselring, and J. R. Wootton, "Poking holes and cutting corners to achieve Clifford gates with the surface code," *Phys. Rev. X*, vol. 7, no. 2, May 2017, Art. no. 021029.
- [20] M. Vasmer and D. E. Browne, "Three-dimensional surface codes: Transversal gates and fault-tolerant architectures," *Phys. Rev. A, Gen. Phys.*, vol. 100, no. 1, Jul. 2019, Art. no. 012312.
- [21] H. Bombin, "2D quantum computation with 3D topological codes," 2018, *arXiv:1810.09571*.
- [22] B. J. Brown, "A fault-tolerant non-clifford gate for the surface code in two dimensions," *Sci. Adv.*, vol. 6, no. 21, May 2020, Art. no. eaay4929.
- [23] J.-P. Tillich and G. Zemor, "Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1193–1202, Feb. 2014.
- [24] P. Panteleev and G. Kalachev, "Degenerate quantum LDPC codes with good finite length performance," 2019, *arXiv:1904.02703*.
- [25] A. Krishna and D. Poulin, "Fault-tolerant gates on hypergraph product codes," 2019, *arXiv:1909.07424*.
- [26] S. Bravyi and M. B. Hastings, "Homological product codes," in *Proc. 46th Annu. ACM Symp. Theory Comput.*, May 2014, pp. 273–282.
- [27] T. Jochym-O'Connor, "Fault-tolerant gates via homological product codes," *Quantum*, vol. 3, p. 120, Feb. 2019.
- [28] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.
- [29] A. R. Calderbank, E. M. Rains, P. W. Shor, and N. J. A. Sloane, "Quantum error correction and orthogonal geometry," *Phys. Rev. Lett.*, vol. 78, no. 3, p. 405, 1997.
- [30] D. Gottesman and I. L. Chuang, "Demonstrating the viability of universal quantum computation using teleportation and single-qubit operations," *Nature*, vol. 402, no. 6760, pp. 390–393, 1999.
- [31] R. W. Ghrist, *Elementary Applied Topology*. Scotts Valley, CA, USA: Createspace Seattle, 2014.
- [32] F. J. MacWilliams and N. J. A. Sloane, *The Theory of Error-Correcting Codes*. Amsterdam, The Netherlands: Elsevier, 1977.
- [33] A. A. Kovalev, S. Prabhakar, I. Dumer, and L. P. Pryadko, "Numerical and analytical bounds on threshold error rates for hypergraph-product codes," *Phys. Rev. A, Gen. Phys.*, vol. 97, no. 6, Jun. 2018, Art. no. 062320.
- [34] J. Roffe, D. R. White, S. Burton, and E. Campbell, "Decoding across the quantum low-density parity-check code landscape," *Phys. Rev. Res.*, vol. 2, no. 4, Dec. 2020, Art. no. 043423.
- [35] A. O. Quintavalle, M. Vasmer, J. Roffe, and E. T. Campbell, "Single-shot error correction of three-dimensional homological product codes," *PRX Quantum*, vol. 2, no. 2, Jun. 2021, Art. no. 020340.

**Simon Burton** received the Ph.D. degree in theoretical physics from the University of Sydney in 2018. Since then, he has worked as a Post-Doctoral Researcher with the University College London and Jagiellonian University, focusing on quantum codes and fault-tolerant gates.

**Dan Browne** received the M.Sci. and Ph.D. degrees from the Imperial College London in 2000 and 2004, respectively. He is currently a Professor in Physics with the University College London (UCL). He was a Junior Research Fellow with the Merton College Oxford, from 2004 to 2007, and was appointed as a Lecturer at UCL in 2007. He was promoted to a Reader in 2013 and to a Professor in 2017. He is a member of the Institute of Physics. His research is on the theory of quantum computing, including fault-tolerant quantum computation, architectures, and applications of near-term quantum computers.