Fiber Bundle Codes: Breaking the $N^{1/2}\,\mathrm{polylog}(N)$ Barrier for Quantum LDPC Codes

Matthew B. Hastings* Jeongwan Haah† Ryan O'Donnell‡

September 8, 2020

Abstract

We present a quantum LDPC code family that has distance $\Omega(N^{3/5}/\,\mathrm{polylog}(N))$ and $\tilde{\Theta}(N^{3/5})$ logical qubits. This is the first quantum LDPC code construction which achieves distance greater than $N^{1/2}\,\mathrm{polylog}(N)$. The construction is based on generalizing the homological product of codes to a fiber bundle.

1 Introduction

While there are many constructions of "good" classical LDPC codes with linear rate and distance [RU08], the construction of a quantum LDPC code[1] on N qubits with distance greater N^c for some $c > \frac{1}{2}$ has been a longstanding open problem. Even constructing a code with distance $\Omega(N^{1/2}\,\mathrm{polylog}(N))$ is nontrivial, and many natural constructions such as the toric code[Kit03] give only distance $\Theta(N^{1/2})$. The first code to beat $N^{1/2}$ by a polylogarithm was based on the cellulation of a carefully chosen manifold [FML02]. A later construction based on Bruhat–Tits buildings improved the polylogarithm and gave an efficient decoder [EKZ20], but still had distance only $O(N^{1/2}\,\mathrm{polylog}(N))$. In this paper, we give a construction attaining distance $\Omega(N^{3/5}/\,\mathrm{polylog}(N))$, solving this problem. We present partial results toward efficient decoding, giving a polynomial-time algorithm to decode bit flip errors and a conjectured efficient algorithm to decode phase errors.

One of the difficulties in constructing a good quantum code is that the X- and Z-stabilizer generators must commute with each other. If one set of generators — say the X-generators — is chosen randomly (trying to follow randomized constructions of classical codes), it is unlikely that there will exists a set of low weight Z generators that commute with them. So, while these randomized constructions are useful if one allows high-weight generators, it is necessary to incorporate some structure into the code if one wishes to have low-weight generators. The approach in [EKZ20] uses deep algebraic/number-theoretic structure. In this paper we follow a different approach, a fiber bundle construction, that combines a simple fiber (the cycle graph) with a random base (a classical LDPC code).

A precursor to this construction is the homological product of quantum codes [FH14, BH14]. This product takes two quantum codes and constructs a product code from them. This product was used [BH14] to construct quantum codes with linear distance and rate, and with generators of

*Station Q. and Microsoft Quantum.
†Station Q. and Microsoft Quantum.
‡Microsoft Quantum and Carnegie Mellon University Computer Science Department.

[1]Throughout, by "quantum code", we mean a CSS stabilizer code. An LDPC (low-density parity check) should have stabilizer generators of weight $O(1)$, and each qubit should be in the support of at most $O(1)$ stabilizer generators.

weight $O(N^{1/2})$, by taking the product of two random quantum codes with linear-weight generators. This product has several other applications. The hypergraph product [TZ14, LTZ15] is a particular form of the homological product when both codes are classical, giving quantum LDPC codes with linear rate and distance $\Theta(N^{1/2})$. Another application is to distance balancing, taking a quantum code which has different distances d_X and d_Z against X- and Z-errors, and increasing the number of qubits to increase one of the distances: so long as $\sqrt{d_X d_Z} \gg N^{1/2}$, this gives [Has17b, EKZ20] a new quantum code with both $d_X, d_Z \gg N^{1/2}$.

1.1 Results, motivation, and outline

In this paper, we generalize the homological product to a *twisted homological product* based on the idea of fiber bundles from topology, giving what we call *fiber bundle codes*. The main result is the following theorem for codes with *logarithmic weight* stabilizers and with each qubit participating in logarithmically many stabilizers[2]. Then, by applying weight reduction and distance balancing techniques we obtain LDPC codes given in Corollary 1.2.

Theorem 1.1. *There exists a family of quantum codes on N qubits with $d_X = \Omega(N^{1/2}/\operatorname{polylog}(N))$ and $d_Z = \Omega(N^{3/4}/\operatorname{polylog}(N))$ where all stabilizer generators have weight at most $\operatorname{polylog}(N)$ and all qubits participate in at most $\operatorname{polylog}(N)$ stabilizer generators. The code has $\Theta(N^{1/2})$ logical qubits.*

The distances of this code are not balanced since $d_Z \gg d_X$, but as mentioned we can apply the distance balancing technique [Has17b, EKZ20]. The distance balancing procedure of [EKZ20] generalizes that of [Has17b] and improves the rate of the resulting code. This distance-balanced code is not LDPC, but since the stabilizer generators weights are only $\operatorname{polylog}(N)$ and each qubit is in only $\operatorname{polylog}(N)$ stabilizer generators, we can apply the weight reduction procedure [Has17b] to get the following:

Corollary 1.2. *There exists a family of quantum LDPC codes on N qubits having distance $d = \Omega(N^{3/5}/\operatorname{polylog}(N))$ and with $\Omega(N^{3/5}/\operatorname{polylog}(N))$ logical qubits.*

There is a long history [Kit03, FM01] of applying ideas from topology to quantum codes, since a quantum code can naturally be interpreted as a chain complex and such a complex can be derived from cellulations of a manifold. Then, operations which have a natural definition in terms of manifolds can often be translated into useful operations on quantum codes. For example, the product of two manifolds naturally leads to considering the homological product of quantum codes.

A fiber bundle is a generalization of the idea of taking a product of two manifolds; roughly, it is something that locally looks like a product but has richer global structure. A simple example of a fiber bundle is a Möbius strip: locally it "looks like" the product of a circle with an interval. However, there is a global twist: going once around the circle reverses the interval. This makes the Möbius band not homeomorphic to the product of a circle with an interval. More generally, we consider a *base* (a circle, in the Möbius band) and a *fiber*, where the fiber admits automorphisms (reversing the interval, in the Möbius band). Motion along the base can involve acting on the fiber by some automorphism.

Another simple example is the case where both the base and the fiber are a circle, S^1. Their untwisted, usual product is a torus. One can impose a twist so that the fiber is reflected when going

[2] In this paper, for simplicity of presentation we do not bother to optimize polylogarithms in the distance. There are some simple ways in which the polylogarithm in our main theorem can be improved by changing some of our parameter choices later, and we comment on them where appropriate.

around the base circle (for example, using angular coordinates ϕ for the fiber, one maps $\phi \mapsto -\phi$); this changes the topology to that of a Klein bottle.

Beyond the changes in topology, these twists can also have an interesting effect on the geometry. Consider again the example where both the base and the fiber are circles. Rather than imposing reflection, we can impose a rotation by some fixed angle $\phi_0 \in \mathbb{R}$ when going around the base circle. This does not change the topology, so the result is still a torus; it is parameterized by angles $\theta \in \mathbb{R}$ for the base and $\phi \in \mathbb{R}$ for the fiber, and we identify

$$(\theta, \phi) \equiv (\theta, \phi + 2\pi) \equiv (\theta + 2\pi, \phi + \phi_0).$$

For any value of ϕ_0, this is still a torus, but the geometry is different. In the context of quantum codes, one needs some cellulation of the manifold, so one may cellulate the fiber and base circles in the obvious way (by cycle graphs C_{n_B}, C_{n_F} for some integers $n_B, n_F > 0$), where the twist ϕ_0 of the fiber is an integer multiple of $2\pi/n_F$. In this case, the result is still a toric code but with different geometry.

Interestingly, even in this very simple case, the change in geometry resulting from this twist can improve the distance of the code! Taking no twist ($\phi_0 = 0$), the code has $N = 2n_Bn_F$ qubits and a distance equal to $\min(n_B, n_F)$, so that the distance is equal to $\sqrt{N/2}$ when $n_F = n_B$. We leave it to the reader to work out the details, but by imposing an appropriate twist and changing n_B and n_F, one may construct a code whose distance is $\sqrt{cN}$ for some $c > 1/2$. While this does not improve the scaling of the code with N, it is a quantitative improvement in distance. Generalizing this construction to higher dimensions [Has17a], and assuming an unproven conjecture in geometry, this could allow for the construction of LDPC codes with distance $N^{1-\epsilon}$ for any $\epsilon > 0$.

In this paper, we consider a further such idea, combining these twists with the use of randomness. We will take a very simple choice of fiber (a circle), but we will choose the base to be a random LDPC code, considered as a chain complex. While the algebraic ideas will be familiar for those with a topology background, we would like warn these readers that many of our choices of chain complexes do not have a nice interpretation as cellulations of a manifold. For example, the base of our bundle will be a "1"-complex

$$\mathcal{B}_1 \xrightarrow{\partial} \mathcal{B}_0$$

with polylogarithmically many "0"-cells in the boundary of each "1"-cell, while usual cellulations of a manifold (or any topological cell complex) would have by definition only two 0-cells in the boundary of a 1-cell. Even more strangely, the base will have zeroth Betti number equal to zero, $b_0 = 0$, while of course usually the zeroth Betti number is the number of connected components of the manifold. As explained in [BH14], it is possible to "reverse engineer" a manifold of high dimension from the code constructed here, in which our "1"-complex will no longer represent the 1-dimensional skeleton of a high dimensional manifold. One can also reverse engineer a 3-complex from the quantum LDPC code constructed here, and triangulate it with simplices to get a simplicial 3-complex.

The paper is outlined as follows. In Section 1.2, we review the connection between quantum codes and cohomology. In Section 2 we define the fiber bundle code. In this section, we pick a specific choice of fiber, and pick the base to be a classical code but we leave the construction of the classical code for later. Much of this section defines bundles in general and computes (co)homology of bundles, and then in Section 2.5, we define the fiber bundle code and sketch the main results needed to prove Theorem 1.1. Then in Section 3 we give the randomized construction of the base code and prove lower bounds on the weight of cohomology and homology representatives. The homology representative bound depends on some complicated properties of an associated classical code proven in Section 4; this section has the most detailed combinatorial calculations. In Section 5,

we present partial results toward an efficient decoding algorithm. Finally, Appendix A collects some of the notation that we use.

1.2 Quantum codes, chain complexes, and (co)homology

In this work all quantum codes are CSS quantum codes on qubits. All vector spaces will be over $\mathbb{F}_2$ and all homology and cohomology takes coefficients in $\mathbb{F}_2$.

Let us briefly review notions of homological algebra. A *chain complex*

$$\cdots \xrightarrow{\partial_{j+1}} \mathcal{A}_j \xrightarrow{\partial_j} \mathcal{A}_{j-1} \xrightarrow{\partial_{j-1}} \cdots \xrightarrow{\partial_1} \mathcal{A}_0 \xrightarrow{\partial_0=0} 0$$

is a sequence of vector spaces $\mathcal{A}_0, \mathcal{A}_1, \ldots$, each with some preferred basis, together with linear maps $\partial_j : \mathcal{A}_j \to \mathcal{A}_{j-1}$ called *boundary operators* between these vector spaces. The boundary maps obey the condition $\partial_j \partial_{j+1} = 0$ whenever ∂_j and ∂_{j+1} are defined. A *k-complex* is a chain complex with $\mathcal{A}_j = 0$ for all $j > k$ but $\mathcal{A}_k \neq 0$. We refer to basis elements of $\mathcal{A}_j$ as *j-cells*, and to vectors in $\mathcal{A}_j$ as *j-chains*. So, a cell is a particular chain. The *Hamming weight* of a chain is the number of cells in the chain with a nonzero coefficient; we write the Hamming weight using absolute value symbols $|\ldots|$. We sometimes identify a chain with the set of cells that have nonzero coefficient in the chain; since the coefficient field is $\mathbb{F}_2$ this identification does not forget any data of a chain.

The *homology* of a chain complex $\mathcal{A}$ is a sequence of vector spaces

$$H_j(\mathcal{A}) = \ker \partial_j / \operatorname{im} \partial_{j+1}$$

where $j = 0, 1, \ldots$. The *j-th Betti number* is defined[3] as

$$b_j(\mathcal{A}) = \dim_{\mathbb{F}_2} H_j(\mathcal{A}).$$

It is customary to assume $\partial_{k+1} = 0$ for a k-complex even if ∂_{k+1} is not explicitly mentioned. An element of $\ker \partial_j$ is called a *j-cycle*. The *cohomology* $H^j(\mathcal{A})$ is the homology of its dual chain

$$\begin{aligned} &\left(\cdots \xleftarrow{\partial_{j+1}^*} \mathcal{A}_j^* \xleftarrow{\partial_j^*} \mathcal{A}_{j-1}^* \xleftarrow{\partial_{j-1}^*} \cdots \xleftarrow{\partial_1^*} \mathcal{A}_0^* \xleftarrow{0} 0\right) \\ \cong &\left(\cdots \xleftarrow{\partial_{j+1}^\mathsf{T}} \mathcal{A}_j \xleftarrow{\partial_j^\mathsf{T}} \mathcal{A}_{j-1} \xleftarrow{\partial_{j-1}^\mathsf{T}} \cdots \xleftarrow{\partial_1^\mathsf{T}} \mathcal{A}_0 \xleftarrow{0} 0\right) \end{aligned}$$

where $\mathcal{A}_j^*$ is the vector space of linear functionals on $\mathcal{A}_j$. Many authors write the *coboundary maps* ∂_j^* as "δ_{j-1}," but we will not. The indicated isomorphism is nothing but a collection of isomorphisms $\mathcal{A}_j^* \cong \mathcal{A}_j$, which are established thanks to the preferred basis for each $\mathcal{A}_j$. More concretely, a linear functional $c^* \in \mathcal{A}_j^*$ that assigns $1 \in \mathbb{F}_2$ for a cell $c \in \mathcal{A}_j$ but $0 \in \mathbb{F}_2$ for all other cells, is identified with the cell c itself under the isomorphism. Almost always in the context of cohomology, an element of $\mathcal{A}_j^*$ is called a "cochain"; however, in this paper we just call it a *chain* since we always use the isomorphism $\mathcal{A}_j^* \cong \mathcal{A}_j$. That is, a chain will always be a $\mathbb{F}_2$-linear combination of cells regardless of whether we use the chain for homology or cohomology. The isomorphism $\mathcal{A}_j^* \cong \mathcal{A}_j$ is fundamental to any combinatorics of cohomology: we will have to count the weight of a cohomology representative, called a *cocycle*, via this identification of linear functionals with their cell-support.

A chain complex defines a quantum code by picking some integer $q > 0$ and associating q-cells of the complex with qubits, and associating $(q-1)$- and $(q+1)$-cells with X- and Z-stabilizer

[3] Usually, Betti numbers are defined as the rank of the free part of the homology with integer coefficients. Our definition is different from this usual one when the integral homology has 2-torsions.

generators of the code (respectively). The Z logical operators of the code are associated with qth homology classes and the X logical operators are associated with qth cohomology classes. The code has two distances, denoted d_X and d_Z, where d_X is the weight of a lowest weight nontrivial X logical operator (i.e., the lowest possible Hamming weight of a vector that represents nontrivial qth cohomology) and d_Z is the weight of a lowest weight nontrivial Z logical operator (i.e., the lowest possible Hamming weight of a vector that represents nontrivial qth homology). Note that the $\mathbb{F}_2$-dimensions of $H_q(\mathcal{A})$ and $H^q(\mathcal{A})$ are always the same as seen by counting vector space dimensions and matrix ranks.

2 Fiber Bundle Codes

In this section we define the code. As remarked earlier, the code results from a chain complex, and thus we focus on constructing chain complexes. We proceed from a general possible construction to our specific instantiation. We begin with reviewing the product of two chain complexes in Section 2.1, and recall the (untwisted) homological product in Section 2.2. We next explain in Section 2.3 that general fiber bundles are obtained by twisting boundary maps. Deferring specific choices of base and fiber complexes and twists, we study algebraic aspects of fiber bundle complexes to establish homology and cohomology isomorphisms at dimension 1 in Section 2.4. We finally define our code in Section 2.5 by specifying the fiber complex; the base complex will be a random classical code with polylogarithmic weight parity checks, whose combinatorial properties will be studied in the next sections. In Sections 3.2 and 3.4 we lower bound the weight of cohomology and homology representatives for this choice, using probabilistic methods.

2.1 Products of chain complexes

Let us begin by recalling the definition of a homological product. Given a *base* complex $\mathcal{B}$ and a *fiber* complex $\mathcal{F}$,[4] we construct their product $\mathcal{E}$, a *bundle*, as follows. We take tensor products of component chain vector spaces, which inherit the boundary maps from the constituent complexes $\mathcal{B}$ and $\mathcal{F}$:

$$\begin{array}{ccccccc}
 & & & \xleftarrow{\partial_j\otimes \mathrm{I}} & \mathcal{B}_j\otimes\mathcal{F}_k & & \\
 & & & & \downarrow{\scriptstyle \mathrm{I}\otimes\partial_k} & & \\
\mathcal{B}_0\otimes\mathcal{F}_1 & \xleftarrow{\partial_1\otimes\mathrm{I}} & \mathcal{B}_1\otimes\mathcal{F}_1 & \xleftarrow{} & \vdots & & \\
\downarrow{\scriptstyle \mathrm{I}\otimes\partial_1} & & \downarrow{\scriptstyle \mathrm{I}\otimes\partial_1} & & \downarrow & & \\
\mathcal{B}_0\otimes\mathcal{F}_0 & \xleftarrow{\partial_1\otimes\mathrm{I}} & \mathcal{B}_1\otimes\mathcal{F}_0 & \xleftarrow{} & \vdots & &
\end{array} \tag{1}$$

Then, the chain space $\mathcal{E}_r$ of the bundle is defined to be the direct sum

$$\mathcal{E}_r = \bigoplus_{p+q=r} \mathcal{E}_{p,q} \quad \text{where} \quad \mathcal{E}_{p,q} = \mathcal{B}_p \otimes \mathcal{F}_q$$

along the diagonal line $p+q=r$ in the diagram for each $r \geq 0$. A preferred basis of the chain vector space $\mathcal{E}_r$ is also inherited from constituent cells; every r-cell of $\mathcal{E}$ is a pair (b^p, f^q) with $p+q=r$ where $b^p \in \mathcal{B}_p$ is a p-cell of the base and $f^q \in \mathcal{F}_q$ is a q-cell of the fiber. We will refer to such an r-cell of the bundle as a (p,q)-cell. If $\mathcal{B}$ and $\mathcal{F}$ are b- and f-complexes, respectively, then $\mathcal{E}$ is a $(\mathsf{b}+\mathsf{f})$-complex.

[4] The untwisted, usual homological product does not distinguish between base and fiber.

2.2 Trivial bundles

We have only defined the chain spaces $\mathcal{E}_r$ above, but not yet the boundary maps $\partial^{\mathcal{E}}$. There are in fact many ways to define $\partial^{\mathcal{E}}$ given the diagram in (1), and this diversity will be realized in the discussion of twisted bundles below. Before we show such diversity, we recall the untwisted boundary map. It suffices to specify how the boundary map acts on each direct summand $\mathcal{E}_{p,q}$ of $\mathcal{E}_r$:

$$\partial^{\mathcal{E}}_r|_{(p,q)} = \mathrm{I} \otimes \partial^{\mathcal{F}}_q \ + \ \partial^{\mathcal{B}}_p \otimes \mathrm{I}, \tag{2}$$

where $\partial^{\mathcal{B}}$ and $\partial^{\mathcal{F}}$ are the boundary maps of $\mathcal{B}$ and $\mathcal{F}$, respectively. One may verify that $\partial^{\mathcal{E}}_r \partial^{\mathcal{E}}_{r+1} = 0$ for all r; here we use that the vector spaces are over $\mathbb{F}_2$.[5]

This definition of homological product via eq. (2) to build a trivial bundle is standard in topology, where the chain complexes are obtained from cell decompositions of two manifolds. The product of the chain complexes corresponds to the cellulation of the product of two manifolds. However, the algebraic construction of trivial bundles above does not have to come from topological spaces. For example, it has been applied [BH14] to input chain complexes which represent random quantum codes.

From now on we will usually drop superscripts and subscripts from boundary maps; the meaning of ∂ will be obvious from the context.

2.3 Twisted bundles

We assume that the fiber admits some *automorphism* group G, which is a collection of permutation actions on the set of q-cells for each q such that boundary operator commutes with this permutation. Such an automorphism naturally extends by linearity to each chain vector space $\mathcal{F}_q$. The requirement of a fiber automorphism group then reads that for each q,

$$g\partial f = \partial g f \qquad \text{for all } g \in G,\ f^q \in \mathcal{F}_q. \tag{3}$$

Definition 2.1. Given a fiber automorphism group G obeying eq. (3), a *connection* φ of a bundle is an arbitrary assignment of a automorphism group element, a *twist*, for each pair of a base cell and one of its boundary cell:

$$\{(b,a) \ : \ b,a \text{ are cells such that } a \in \partial b\} \xrightarrow{\varphi} G. \tag{4}$$

where we have identified ∂b with its support (the collection of cells with nonzero coefficients in ∂b). We define a *twisted boundary map* $\partial^{\mathcal{E}}$ by φ:[6]

$$\begin{aligned} \partial^{\mathcal{E}}_{(0,q)}(b^0 \otimes f) &= b^0 \otimes \partial f, \\ \partial^{\mathcal{E}}_{(1,q)}(b^1 \otimes f) &= b^1 \otimes \partial f \ + \sum_{a^0 \in \partial b^1} a^0 \otimes \varphi(b^1, a^0) f. \end{aligned} \tag{5}$$

Proposition 2.2. *If the base is a 1-complex, then the twisted boundary map satisfies* $\partial^{\mathcal{E}}_r \partial^{\mathcal{E}}_{r+1} = 0$ *for all* $r \geq 0$.

[5] With a general coefficient group, an extra sign is needed: $\partial^{\mathcal{E}}_{(p,q)} = (-1)^p \mathrm{I} \otimes \partial^{\mathcal{F}}_q + \partial^{\mathcal{B}}_p \otimes \mathrm{I}$.

[6] With a general coefficient group, the first term of eq. (5) has sign -1.

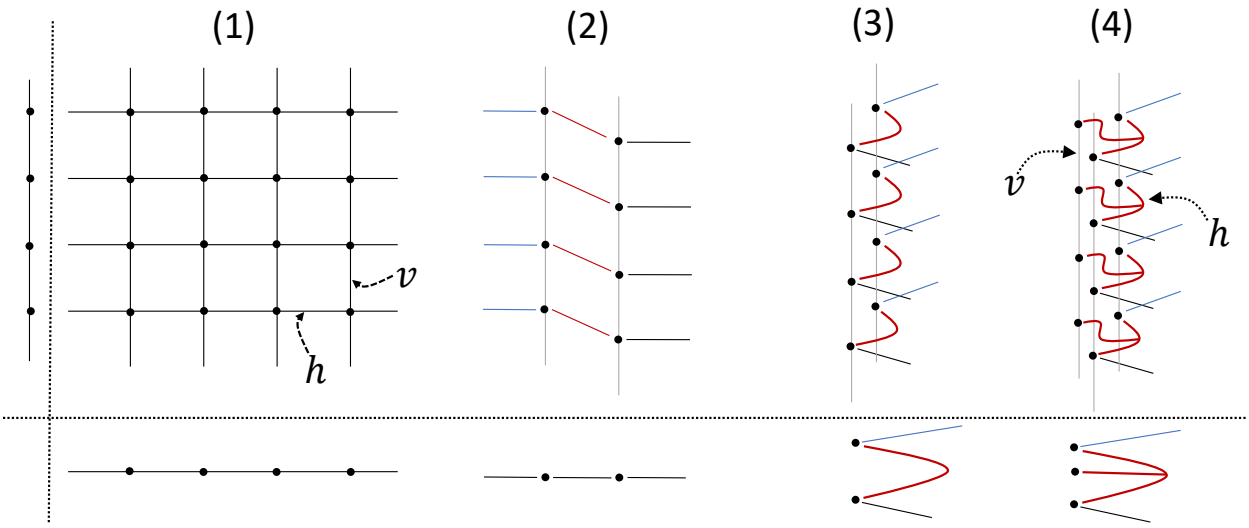

Figure 1: Fiber bundles over base 1-complexes. (1) depicts a trivial bundle built from two 1-complexes, each representing a long line or a circle. Any bundle 1-cell that is a lift of a base 1-cell is referred to as a *horizontal* 1-cell, indicated by h in the figure. Any bundle 1-cell that vanishes upon projection onto the base is referred to as a *vertical* 1-cell, indicated by v in the figure. The projection is defined in Definition 2.4. (2) depicts some twisting. Since a fiber is acted on by an automorphism, the entire fiber over a base cell is shifted. Note that the shown twist can be removed using gauge redundancy. To draw a nonremovable twisting, we must have had a cycle in the base. (3) is the same as (2) but base 1-cells are positioned on the right-hand side and base 0-cells on the left-hand side. (4) introduces a "1"-cell (red) of the base that has three boundary 0-cells.

Proof. It suffices to check the claim for basis elements. It is obvious that $\partial^{\mathcal{E}}_{q-1}\partial^{\mathcal{E}}_{q}(b^0 \otimes f^q) = b^0 \otimes \partial^{\mathcal{F}}_{q-1}\partial^{\mathcal{F}}_{q} f^q = 0$ for any q. If $b = b^1$ is a base 1-cell and f is a fiber q-cell, by eq. (5) we see

$$\begin{aligned}\partial^{\mathcal{E}}_{q}\partial^{\mathcal{E}}_{q+1}(b^1 \otimes f^q) &= \partial^{\mathcal{E}}_{q}(b^1 \otimes \partial f^q) + \partial^{\mathcal{E}}_{q} \sum_{a^0 \in \partial b^1} a^0 \otimes \varphi(b^1, a^0) f^q \\ &= \left(b^1 \otimes \partial\partial f^q + \sum_{a^0 \in \partial b^1} a^0 \otimes \varphi(b^1, a^0)\partial f^q \right) + \sum_{a^0 \in \partial b^1} a^0 \otimes \partial\varphi(b^1, a^0) f^q \\ &= 0\end{aligned}$$

where the second equality is because a^0 is a 0-cell and the third is because of eq. (3). □

The definition of twisted boundary map can be generalized to any higher-dimensional base complex. This generalization, however, requires certain conditions so that $\partial^{\mathcal{E}}\partial^{\mathcal{E}} = 0$ is fulfilled. For example, if the base is a 2-complex, then we may need extra terms in the boundary map:

$$\partial^{\mathcal{E}}_{(2,q)}(b \otimes f) = b \otimes \partial f + \sum_{e \in \partial b} e \otimes \varphi(b, e) f + \sum_{v \in \partial e : e \in \partial b} v \otimes f^{+}_{v,e,b} \tag{6}$$

where $f^{+}_{v,e,b}$ is some $(q+1)$-cell of the fiber. For such extra terms to exist, the twists and the fiber complex should jointly obey certain conditions. We do not pursue in this generalization further, and from now on the base will always be a 1-complex. With the restriction that the base is a 1-complex, we will find it convenient to distinguish bundle 1-cells as follows.

Definition 2.3. Members of $\mathcal{E}_{1,0}$ are *horizontal*. Members of $\mathcal{E}_{0,1}$ are *vertical*.

Every bundle 1-chain is a sum of a horizontal chain and a vertical chain, and such a decomposition is always unique. See Figure 1.

While the twists described in eq. (4) are completely arbitrary, not all choices of twists give different "geometry" for the bundle. This is known as gauge redundancy. Here the geometry refers to an equivalence class of bundles where the equivalence relation is given by an isomorphism between chain complexes such that it commutes with the twisted bundle boundary maps and sends cells to cells (preferred bases). For example, one can transform the fiber by an automorphism h and change the twists as $\varphi(b, a) \mapsto h\varphi(b, a)h^{-1}$. Even more flexibly, one can transform a fiber over a particular base cell, and simultaneously change the twists that connect the base cell with others.

This gauge redundancy can be so rich that if, for example, the base is a cyclic graph (a circle), the set of all twists can be simplified so that there is a non-identity assignment only for one pair of a 1-cell and its boundary 0-cell. We will avoid this simplification by having a complicated base, and this is part of reason that we will take a random code for the base.

2.4 Isomorphisms on (co)homology groups

Under conditions we use later, the first homology $H_1(\mathcal{E})$ and cohomology $H^1(\mathcal{E})$ of the bundle will be isomorphic to $H_1(\mathcal{B})$ and $H^1(\mathcal{B})$ of the base, respectively. In particular, the first Betti numbers agree: $b_1(\mathcal{E}) = b_1(\mathcal{B})$. The isomorphism will be induced by the bundle projection:

Definition 2.4. A linear map called the *bundle projection* $\Pi_r : \mathcal{E}_r \to \mathcal{B}_r$ is defined as

$$\begin{aligned} b^r \otimes f^0 &\mapsto b^r, \\ b^{r-j} \otimes f^j &\mapsto 0 \quad \text{if } j > 0 \end{aligned}$$

for all r-cells b^r and $(r-j)$-cells b^{r-j} of the base, and 0-cells f^0 and j-cells f^j of the fiber.

Lemma 2.5. *The bundle projection induces vector space isomorphisms* $\Pi_* : H_1(\mathcal{E}) \to H_1(\mathcal{B})$ *and* $\Pi^* : H^1(\mathcal{B}) \to H^1(\mathcal{E})$ *if all of the following are true:*

(i) $\mathcal{B}$ *is a* 1*-complex.*

(ii) The boundary ∂f^1 *of any fiber* 1*-cell* f^1 *has even weight.*

(iii) Every fiber 0*-chain of even weight is a boundary.*

(iv) $H_0(\mathcal{B}) = 0$*, i.e. the zeroth Betti number* $b_0(\mathcal{B})$ *vanishes.*

(v) Every fiber automorphism acts trivially on $H_1(\mathcal{F})$*.*

The conditions (ii) and (iii) are redundant in a topological setting where a 1-cell is always a line segment.

Proof. The proof consists of the propositions below.

Proposition 2.6. *Assume (i) and (ii). Then the following diagram commutes:*

$$\begin{array}{ccccc} \mathcal{E}_2 & \xrightarrow{\partial} & \mathcal{E}_1 & \xrightarrow{\partial} & \mathcal{E}_0 \\ \downarrow{\scriptstyle 0} & & \downarrow{\scriptstyle \Pi} & & \downarrow{\scriptstyle \Pi} \\ 0 = \mathcal{B}_2 & \xrightarrow[0]{} & \mathcal{B}_1 & \xrightarrow[\partial]{} & \mathcal{B}_0 \end{array} \tag{7}$$

Proof. For the left square, we need to show that $\Pi\partial : \mathcal{E}_2 \to \mathcal{B}_1$ is zero. For a base 1-cell b and fiber 1-cell f, we have $\Pi\partial(b \otimes f) = \Pi b \otimes \partial f = |\partial f| b = 0$ by (ii). For the right square, take any 1-chain of the bundle, and decompose it as $h^1 + v^1$ where $h^1 \in \mathcal{E}_{1,0}$ is horizontal and $v^1 \in \mathcal{E}_{0,1}$ is vertical. It is obvious that $\Pi\partial h^1 = \partial\Pi h^1$. By (ii), we have $\Pi\partial v^1 = 0$, and clearly $\partial\Pi v^1 = 0$. □

Proposition 2.7. *Assume (i) and (ii). Then the induced map* $\Pi_* : H_1(\mathcal{E}) \to H_1(\mathcal{B})$ *is well-defined.*

Proof. We have to show that (1) any closed 1-chain becomes closed, and (2) any 1-chain that is a boundary becomes a boundary.

Decompose a bundle 1-chain as $h^1 + v^1$ as before. If $h^1 + v^1$ is closed, then $\partial h^1 = \partial v^1$. By (ii), we see $\Pi\partial v^1 = 0$. Hence, $\partial\Pi h^1 = \Pi\partial h^1 = 0$. This shows (1).

For (2), it suffices to examine boundary of $(1,1)$-cells, by (i). We examine $\partial(b^1 \otimes f^1) = b^1 \otimes \partial f^1 + \sum_{a^0 \in \partial b^1} a^0 \otimes \varphi(b^1, a^0) f^1$. Under Π, the sum vanishes by definition of Π and so does the first term by (ii). Hence, the projection of a boundary is zero. □

Proposition 2.8. *Assume (i)—(iii). Then* $\Pi_* : H_1(\mathcal{E}) \to H_1(\mathcal{B})$ *is onto.*

Proof. Given a homology representative cycle b^1 of the base, we choose an arbitrary fiber 0-cell f^0 so $b^1 \otimes f^0$ projects down to b^1. Observe that $\partial(b^1 \otimes f^0) = \sum_{a^0 \in \partial b^1} a^0 \otimes \varphi(b^1, a^0) f^0 = \sum_j a_j^0 \otimes f_j'$ where the a_j^0 are distinct 0-cells of the base and the f_j' are some fiber 0-chains. Since $\Pi\partial(b^1 \otimes f^0) = \partial b^1 = 0$, we have $\sum_{a^0 \in \partial b^1} \Pi(a^0 \otimes \varphi(b^1, a^0) f^0) = \sum_j |f_j'| a_j^0 = 0$. This means that $|f_j'| = 0 \bmod 2$ for all j. By (iii), we have $f_j' = \partial s_j^1$ for some fiber 1-chain s_j^1. Now, $b^1 \otimes f^0 + \sum_j a_j^0 \otimes s_j^1$ is closed and projects down to b^1. □

Proposition 2.9. *Assume (i)—(v). Then* $\Pi_* : H_1(\mathcal{E}) \to H_1(\mathcal{B})$ *is one-to-one.*

Proof. Suppose $\Pi_*(v^1+h^1)=0$ where v^1+h^1 is a decomposition of a 1-cycle of $\mathcal{E}$ into vertical ($\mathcal{E}_{0,1}$) and horizontal ($\mathcal{E}_{1,0}$) 1-chains. The projection eliminates vertical 1-cells by definition, and takes the mod-2 sum of horizontal 1-cells. So, the vanishing projection means that over any base 1-cell, there are an even number of horizontal 1-cells. That is, we can write the horizontal chain h^1 as $h^1=\sum_j b_j^1\otimes f_j^0$ where b_j^1 are distinct base 1-cells and each f_j^0 is an even-weight 0-chain of the fiber. By (iii), $f_j^0=\partial s_j^1$ for some 1-chain s_j^1 in the fiber, and thus $\partial\sum_j b_j^1\otimes s_j^1=h^1+u^1$ for some vertical 1-chain u^1. Then, we have $v^1+h^1=v^1+u^1+\partial\sum_j b_j^1\otimes s_j^1$. Since v^1+h^1 is closed, v^1+u^1 is also closed. Therefore, v^1+h^1 is homologous to v^1+u^1, a vertical 1-cycle.

But now we can show that any vertical 1-cycle of the form $a^0\otimes s^1$ with $\partial s^1=0$ is a boundary: The assumption (iv) gives a base 1-chain c^1 such that $a^0=\partial c^1$. Then, $\partial(c^1\otimes s^1)=\sum_{t^0\in\partial c^1} t^0\otimes\varphi(c^1,t^0)s^1$. Since $t^0\otimes\varphi(c^1,t^0)s^1$ is homologous to $t^0\otimes s^1$ by (v), we see $\partial(c^1\otimes s^1)$ is homologous to $(\partial c^1)\otimes s^1=a^0\otimes s^1$. □

This completes the proof of Lemma 2.5 for homology. The cohomology isomorphism is straightforward by abstract nonsense using the commutative diagram (7). The details are as follows.

If w_1 is a 1-cocycle of the base, then $(\partial^\mathsf{T}\Pi^* w_1)(e^2)=w_1(\Pi\partial e^2)=w_1(\partial\Pi e^2)=(\partial^\mathsf{T}w_1)(\Pi e^2)=0$ for any bundle 2-chain e^2. If $w_1=\partial^\mathsf{T}w_0$ is a chain of the base, then $(\Pi^*w_1)(e^1)=w_1(\Pi e^1)=w_0(\partial\Pi e^1)=w_0(\Pi\partial e^1)=(\partial^\mathsf{T}\Pi^*w_0)(e^1)$, and so Π^*w_1 is a coboundary. This shows that Π^* is a well-defined map from $H^1(\mathcal{B})$ to $H^1(\mathcal{E})$. We claim that Π^* is one-to-one. This will finish the proof of Lemma 2.5 because the vector space dimensions are the same for homology and cohomology. To show the claim, suppose that Π^*w_1 is a coboundary for a base 1-cohomology representative w_1. That is, Π^*w_1 vanishes on any bundle 1-cycle. Now by the homology isomorphism, any bundle 1-homology cycle is a lift of a base 1-cycle. This means that w_1 vanishes on all base 1-cycles. Since the dual vector space of $H_1(\mathcal{B})$ can be identified with $H^1(\mathcal{B})$, we see w_1 is a coboundary. □

Under the assumptions of Lemma 2.5, it will be instructive to have a more elementary description of the homology and cohomology representatives. A lift of a base homology representative c^1 is illustrated in the proof above. To recap, a lift consists of horizontal 1-cells that project onto c^1 together with vertical 1-cells that cap the boundary of these horizontal 1-cells through a 1-chain in the fiber. Because of the twists, the boundary of the horizontal 1-cells can be "far apart" within each fiber.

Regarding cohomology, since $\mathcal{B}$ is a 1-complex, there is no restriction on the 1-cocycles; every 1-chain is a 1-cocycle. That is, any 1-cell b of the base represents a 1-cohomology class. Its lift functional Π^*b as a cohomology representative must evaluate to $1\in\mathbb{F}_2$ for every horizontal cell over b, so the functional Π^*b can be identified with $\sum_{f^0} b\otimes f^0$ where the sum is over all 0-cells f^0 of the fiber. The cohomology representative gives an upper bound on d_X when the quantum code's X logical operators are 1-cohomology. We summarize this as a proposition for later reference.

Proposition 2.10. *Assume all conditions (i)—(v) of Lemma 2.5. If $\{[b_1],[b_2],\dots\}$ is a basis of $H^1(\mathcal{B})$ where b_j are some base 1-cocycles, the following is a complete basis of representatives for $H^1(\mathcal{E})$:*

$$\{b_1\otimes F_0, b_2\otimes F_0,\dots\},$$

where $F_0\in\mathcal{F}_0$ denotes the sum of all fiber 0-cells. In particular, there exists a nontrivial representative of $H^1(\mathcal{E})$ of weight equal to the number of 0-cells in the fiber.

2.5 Circle bundle over classical codes

We choose the fiber to be a cycle graph, i.e., a circle. As a chain complex, the fiber is a 1-complex with $m_\mathcal{F}$ 0-cells and $n_\mathcal{F}$ 1-cells where $m_\mathcal{F}=n_\mathcal{F}>1$. This circle admits an automorphism group

that is the dihedral group of order $2n_{\mathcal{F}}$; however, we do not use the reflection symmetry, but only the rotation symmetry. The circle fulfills all the conditions related to the fiber, namely (ii), (iii), and (v) of Lemma 2.5; any automorphism of the circle leaves the fundamental homology cycle invariant.

Definition 2.11. Our fiber bundle code is a quantum CSS code whose logical operators are associated with homology and cohomology at dimension 1 of the twisted bundle complex $\mathcal{E}_2 \to \mathcal{E}_1 \to \mathcal{E}_0$ built from the circle fiber $\mathcal{F}_1 \to \mathcal{F}_0$ and a base $\mathcal{B}_1 \to \mathcal{B}_0$.

This definition leaves room for the base complex to be any classical code and for the twists to be completely arbitrary members of $\mathbb{Z}_{n_{\mathcal{F}}}$.

We will pick $n_{\mathcal{F}} = m_{\mathcal{F}} = \ell^2$ for some integer ℓ (and it will be convenient, though not strictly necessary, to assume ℓ is odd, hence $n_{\mathcal{F}}$ is odd). In fact, all twists will be multiples of ℓ so we only use a subgroup of order $n_{\mathcal{F}}/\ell = \ell$ of the rotation group. Roughly speaking, this is done so that if some weight needs to "move through the fiber" to join two cells that differ by a twist, we have some lower bound on how far it needs to move.

We will use a random classical code for the base $\mathcal{B}$. There will be $n_{\mathcal{B}}$ bits (variables) in this classical code, considered to be 1-cells of the base, and there will be $m_{\mathcal{B}}$ parity checks in the code, considered to be 0-cells in the base. We will represent this classical code by its Tanner graph, a bipartite graph B with $m_{\mathcal{B}}$ left-vertices and $n_{\mathcal{B}}$ right-vertices. We will later choose $m_{\mathcal{B}} = (3/4)n_{\mathcal{B}}$, and all vertices will have degree very close to $\Delta = \Theta(\log^2 n_{\mathcal{B}})$. In this way the random classical code will have minimum distance $\Omega(n_{\mathcal{B}})$ and all of its parity checks will be linearly independent (with high probability); the latter condition is equivalent to $H_0(\mathcal{B}) = 0$.

Thus the bundle $\mathcal{E}$ will have $N = n_{\mathcal{B}} \cdot m_{\mathcal{F}} + m_{\mathcal{B}} \cdot n_{\mathcal{F}}$ 1-cells corresponding to qubits of the resulting quantum code, and the total number of cells in the bundle will be $(n_{\mathcal{B}} + m_{\mathcal{B}}) \cdot (n_{\mathcal{F}} + m_{\mathcal{F}})$. Given that $H_0(\mathcal{B}) = 0$, i.e., $b_0(\mathcal{B}) = 0$, it then follows by construction that $b_1(\mathcal{B}) = (1/4)n_{\mathcal{B}}$ so that the fiber bundle code has $\Theta(n_{\mathcal{B}})$ logical qubits. We will prove that the resulting quantum code has distances $d_X = \Omega(m_{\mathcal{F}}/\log^2 n_{\mathcal{B}})$ (see Lemma 3.7) and $d_Z = \Omega(n_{\mathcal{B}} \cdot {m_{\mathcal{F}}}^{1/2}/\log^2 n_{\mathcal{B}})$ (see Lemma 3.11) provided $n_{\mathcal{B}} \geq m_{\mathcal{F}}$. We then choose $n_{\mathcal{B}} \sim m_{\mathcal{F}}$, giving $N = \Theta({n_{\mathcal{B}}}^2)$ and giving distances $d_X = \Omega(N^{1/2}/\log^2 N)$, $d_Z = \Omega(N^{3/4}/\log^2 N)$. Together, these facts prove Theorem 1.1.

Remark. It is possible to slightly improve the polylogs in Theorem 1.1 by adjusting our choices for $\ell, n_{\mathcal{F}}, n_{\mathcal{B}}$ by polylogarithmic factors. Specifically, a minor improvement arises from choosing $m_{\mathcal{F}} = n_{\mathcal{B}}/\Delta$ and $\ell = \sqrt{m_{\mathcal{F}}/\Delta}$, where recall $\Delta = \Theta(\log^2 n_{\mathcal{B}})$. However, these polylog improvements deteriorate again after passing through the weight reduction process leading to Corollary 1.2. Thus we have chosen to make slightly non-optimal parameter choices so as to simplify the presentation.

3 The random base code, with twists

Throughout this section and the next section we write $n = n_{\mathcal{B}}$ and $m = m_{\mathcal{B}}$, for brevity.

3.1 A random base code

The base code B is identified with its Tanner graph, having variable vertices $[n]$ and check vertices $[m]$. Our construction will require $\frac{1}{2} < m/n < 1$; for simplicity we fix

$$m = \tfrac{3}{4}n, \tag{8}$$

assuming that m is an integer. We will also fix a parameter $\Delta = \Delta(n)$ representing the average degree of the check vertices. Eventually we will choose

$$\Delta = \Theta(\log^2 n),$$

but for now we only assume

$$\beta \ln n \le \Delta \le n^{o(1)}, \tag{9}$$

where β is a large universal constant to be chosen later. We will choose a *random* base code $\boldsymbol{B}$,[7] with the neighborhood $\partial^{\mathsf{T}} a$ of each check $a \in [m]$ independently being a random density-$\frac{\Delta}{n}$ subset of $[n]$. By this we mean that each variable $i \in [n]$ is included into $\partial^{\mathsf{T}} a$ independently with probability $\frac{\Delta}{n}$.

It is well known that such a randomly constructed code $\boldsymbol{B}$ will have various expansion-type properties. We collect here some standard results along these lines.

Proposition 3.1. *Except with probability at most $O(1/n^{100})$, all check vertices in $\boldsymbol{B}$ have degree between $.99\Delta$ and 1.01Δ and all variable vertices have degree between $.74\Delta$ and $.76\Delta$.*

Proof. This can be achieved by a standard Chernoff + union bound argument, taking the constant β in eq. (9) large enough and using $m \le n$. □

Proposition 3.2. *The bipartite graph $\boldsymbol{B}$ has the following property, except with probability at most $O(1/n^{100})$: For every $S \subseteq [m]$ with $|S| \le \frac{1}{10^5 \Delta} m$, the neighborhood of S in $[n]$ has cardinality at least $.9\Delta|S|$.*

Proof. By Proposition 3.1, it suffices to prove this conditioned on the assumption that every check vertex in $\boldsymbol{B}$ has degree between $.99\Delta$ and 1.01Δ. Under this conditioning, the neighborhoods of each check vertex remain independent and have a certain distribution on their cardinality; conditioned on their cardinality, they are uniformly random subsets of $[n]$. By ignoring edges (which only hurts us), we may therefore assume that the neighborhoods of the check vertices are independent random subsets of $[n]$ of cardinality $d := \lceil .99\Delta \rceil$. Thus we have reduced to the d-regular model of random bipartite graphs, where the claim we want to prove is standard, relying on the inequality

$$.99\Delta > \frac{h_2(\frac{1}{10^5\Delta}) + h_2(\frac{.8\Delta}{10^5\Delta})}{h_2(\frac{1}{10^5\Delta}) - \frac{.8}{10^5} h_2(\frac{1}{.8\Delta})},$$

(here $h_2(\cdot)$ is the binary entropy function), and on the assumption from eq. (9) that $\Delta \ge \beta \ln n$ for large constant β; see, e.g., [Chu79, Bas81]. □

Corollary 3.3. *Assume B satisfies the conclusion of Propositions 3.1 and 3.2. Let $S \subseteq [m]$ have $0 < |S| \le \frac{1}{10^5\Delta} m$. Say a variable vertex $j \in [n]$ is a* counique neighbor of S *if it neighbors exactly one vertex in S. Then, the number of non-counique neighbors of S is at most $.09\Delta|S|$ and there exists some $a^\bullet \in S$ for which more than $.81\Delta$ of its neighbors are counique neighbors. In particular, $a^\bullet$ has at least a .8 fraction of its neighbors (a strict majority) being counique.*

Proof. Let every $a \in S$ give a token to each of its neighbors in $[n]$. By Proposition 3.1, at least $.99\Delta|S|$ tokens are given out. Every vertex in the set C of counique neighbors gets one token, and every vertex in the set C' of non-unique neighbors gets at least two. Thus $.99\Delta|S| \ge |C| + 2|C'|$. But $|C| + |C'| \ge .9\Delta|S|$, by Proposition 3.2. Thus $|C'| \le .09\Delta|S|$, and so $|C| \ge .81\Delta|S|$. We conclude that even on average, a vertex $a \in S$ gives out at least $.81\Delta$ tokens to counique neighbors. □

[7] In this section, boldface denotes random variables/objects.

Proposition 3.4. *Except with probability $O(1/n^{100})$, the parity check matrix of $\boldsymbol{B}$ is of full rank.*

Proof. Let $\boldsymbol{v}_a \in \mathbb{F}_2^n$ be the indicator for the neighborhood of check vertex $a \in [m]$. We wish to show for all $\emptyset \neq A \subseteq [m]$ that $\sum_{a\in A} \boldsymbol{v}_a \neq 0$. So fix such an A of cardinality $w \neq 0$. For $i \in [n]$, the ith coordinate of $\sum_{a\in A} \boldsymbol{v}_a$ is distributed as Binomial$(w, \frac{\Delta}{n})$ modulo 2. The event that this is 0 has probability

$$q_w \coloneqq \tfrac{1}{2} + \tfrac{1}{2}(1 - \tfrac{2\Delta}{n})^w$$

and these events are independent across $i \in [n]$. For $w \leq \frac{n}{\Delta}$ it holds that $q_w \leq \exp(-\frac{w\Delta}{2n})$, and hence q_w^n (which is the probability of $\sum_{a\in A} \boldsymbol{v}_a = 0$) is at most $\exp(-\frac{\Delta}{2})^w \leq 1/n^{101w}$, the last inequality provided the constant β in eq. (9) is large enough. On the other hand, if $w \geq \frac{n}{\Delta}$ then we have $q_w \leq \frac{1}{2} + \frac{1}{2}\exp(-\frac{2\Delta w}{n}) \leq \frac{1}{2} + \frac{1}{2}e^{-2} \leq 2^{-.8}$, and hence $q_w^n \leq 2^{-.8n}$. Taking a union bound over all A, we conclude that the probability of $\boldsymbol{B}$'s parity check matrix having a nontrivial linear dependence is at most

$$\sum_{1\leq w\leq \frac{n}{\Delta}} \tbinom{m}{w}/n^{101w} + \sum_{\frac{n}{\Delta}\leq w\leq m} 2^{-.8n} \leq \left((1+1/n^{101})^m - 1\right) + 2^m 2^{-.8n} = O(1/n^{100}),$$

where the last inequality used $m = \frac{3}{4}n$. □

Proposition 3.5. *Except with probability $O(1/n^{100})$, the code $\boldsymbol{B}$ has minimum distance at least $.2n$.*

Proof. Indeed, the number .2 can be replaced with any δ such that $h_2(\delta) \leq \frac{3}{4}$; in other words, with high probability $\boldsymbol{B}$ achieves the Gilbert–Varshamov bound. This is a standard property of random LDPC codes with $\Delta \to \infty$. Gallager showed it in a slightly different model of Δ-regular random LDPC codes; the proof in our case is a standard exercise along the lines of Proposition 3.4 and Lemma 4.6. □

3.2 Cohomology representative weight

We have not yet specified the twists, but we are already able to lower-bound the weight of a cohomology representative. Recall Proposition 2.10 implies that the least weight cohomology represenative has its weight *upper*-bounded by $m_{\mathcal{F}}$. We start with a definition, and then give the lower bound.

Definition 3.6. In brief, the *shadow* of a chain $e \in \mathcal{E}$ is the set of base cells on which e has support. More precisely, given a bundle 1-chain e, write it as a sum $h + v$ of horizontal and vertical chains, with $h = \sum_b b \otimes f_b^0$ and $v = \sum_a a \otimes f_a^1$ where a runs over base 0-cells and b over base 1-cells. Then the shadow of the vertical part of e is the set $\{a : f_a^1 \neq 0\}$, and the shadow of the horizontal part of e is the set $\{b : f_b^0 \neq 0\}$. Similarly given a bundle 0-chain e with $e = \sum_a a \otimes f_a^0$, its shadow is $\{a : f_a^0 \neq 0\}$. We write $|e|_{\text{vsw}}$ for the *vertical shadow weight* of a bundle 1-chain e; i.e., the cardinality of its shadow of the vertical part. For a 0-chain e, we simply write $|e|_{\text{sw}}$ for the shadow weight.

Lemma 3.7. *Assume the base code B satisfies the conclusions of Propositions 3.1 and 3.2. Then for any choice of twists, the following holds: Let r be any nontrivial representative of $H^1(\mathcal{E})$ and write it as a sum $h + v$ of horizontal and vertical chains. Then either $|h| \geq m_{\mathcal{F}}/2$, or $|v|_{\text{vsw}} \geq \Omega(n/\Delta)$. In particular, $|r| = |h| + |v| \geq \Omega(m_{\mathcal{F}} + n/\Delta)$, which is $\Omega(m_{\mathcal{F}}/\Delta)$ assuming $n \geq m_{\mathcal{F}}$.*

Proof. Let r be any nontrivial representative of $H^1(\mathcal{E})$. Using the basis of Proposition 2.10, we may write $r = x \otimes F_0 + \partial^{\mathsf{T}} u$ where x is a nontrivial representative of $H^1(\mathcal{B})$ and where u is a

bundle 0-chain. This expression is not unique; if we toggle $u \mapsto u + a \otimes F_0$ for any base 0-cell a and simultaneously change $x \mapsto x + \partial^\mathsf{T} a$, then the overall change in $x \otimes F_0 + \partial^\mathsf{T} u$ is $(\partial^\mathsf{T} a) \otimes F_0 + \partial^\mathsf{T}(a \otimes F_0)$, which vanishes regardless of twists because F_0 is invariant under any fiber automorphism. Thus we may assume that in the expansion $u = \sum_a a \otimes f_a$ (where each a is a base 0-cell and each f_a is a fiber 0-chain), each f_a has Hamming weight at most $m_\mathcal{F}/2$. Having done this, the shadow of the vertical part of r and the shadow of u coincide (as no f_a equals F_0). Writing S for this shadow, we have

$$r = x \otimes F_0 + \partial^\mathsf{T} u = \underbrace{x \otimes F_0 + \sum_{a \in S, b \in \partial^\mathsf{T} a} b \otimes \varphi(b,a)^{-1} f_a}_{h} + \underbrace{\sum_{a \in S} a \otimes \partial^\mathsf{T} f_a}_{v} \tag{10}$$

where $h \in \mathcal{E}_{1,0}$ is horizontal and $v \in \mathcal{E}_{0,1}$ is vertical. Here $\varphi(b,a)$ are some twists. If $|S| \geq \frac{1}{10^5 \Delta} m$ then we are done easily: for every $a \in S$ we have $0 \neq |f_a| \neq m_\mathcal{F}$ and hence $|\partial^\mathsf{T} f_a| \geq 2$; thus $|r| \geq |v| \geq 2|S| \geq \Omega(m_\mathcal{F}/\Delta)$, as needed.

Thus it remains to handle the case that $|S| \leq \frac{1}{10^5 \Delta} m$. In this case we claim that in fact $|h| \geq m_\mathcal{F}/2$, which is more than sufficient to complete the proof. First, if $S = \emptyset$ then $|r| = |h| = |x| m_\mathcal{F} \geq m_\mathcal{F}$, using the fact that r is a *nontrivial* cohomology representative, and the claim is established.

Otherwise, $0 < |S| \leq \frac{1}{10^5 \Delta} m$, and since B satisfies the conclusion of Proposition 3.2, the number of neighbors (base 1-cells) of S is at least $.8\Delta|S|$. As every $a \in S$ has between $.99\Delta$ and 1.01Δ neighbors (Proposition 3.1), it follows that there must be some base 0-cell $a^\bullet \in S$ such that more than half of the 1-cells in the coboundary of $a^\bullet$ are not in the coboundary of any other 0-cell of S. (This relies on $.8 > 1.01 \cdot \frac{3}{4}$ and also $|S| \neq 0$.) Let C be this set of *counique*-neighbor base 1-cells: $C = \partial^\mathsf{T} a^\bullet \setminus \bigcup_{a \in S \setminus \{a^\bullet\}} \partial^\mathsf{T} a$. We now consider whether or not C overlaps with x.

If $C \cap x$ contains a base 1-cell b, then the number of all horizontal cells of r over b is already at least $m_\mathcal{F}/2$, because $b \otimes \varphi(b,a)^{-1} f_a$ has at most $m_\mathcal{F}/2$ cells.

Otherwise, if $C \cap x = \emptyset$, then dropping $a^\bullet$ from S would reduce $|h|$, since less than half of the horizontal part of $\partial^\mathsf{T} a^\bullet$ was canceling in eq. (10) by the choice of $a^\bullet$. This dropping yields a cohomologous representative r' of $H^1(\mathcal{E})$ with a lighter horizontal part and where the shadow S of the vertical part still satisfies $|S| \leq \frac{1}{10^5 \Delta} m$. Repeating this argument, we either come to a representative whose horizontal weight is at least $m_\mathcal{F}/2$, or else we reduce to the case of $S = \emptyset$ where it was already shown that the horizontal part has weight at least $m_\mathcal{F}$. Either way, we have established the claim that the original r had $|h| \geq m_\mathcal{F}/2$. □

3.3 Twists

We now choose the twists. It is worth making a mental shift at this point. The twist is defined by an automorphism $\varphi(b^1, v)$ which is a function of a 1-cell b^1 and some 0-cell $v \in \partial b^1$. Mentally, up to this point, we have tended to think of it as "for each 1-cell b^1, for each v in ∂b^1" there is a twist, but now it is worth thinking instead "for each 0-cell v, for each b^1 in $\partial^\mathsf{T} v$" there is a twist. This of course is no difference mathematically but is an easier mental picture. In the language of the base code, it means that for each check of the base code, for each bit in the check, there is a twist.

All twists will be chosen to be integer multiples of ℓ. Indeed, we will only have $k = \Theta(\log n)$ distinct choices of twists (where the precise k will be specified later). Informally, we will partition the checks of the base code into k "types"; for each type, we will pick a single twist by some random multiple of ℓ, and for each bit in each check, we will either twist by that multiple or by 0, the choice of which again being random.

Formally, let us make the following definition:

Definition 3.8. Given a base code B, we say it is *partitioned* if:

- its check vertices are partitioned into k sets $T_1, \ldots, T_k$ of equal size m/k (assumed to be an integer), one for each "type";
- and, the neighborhood $\partial^{\mathsf{T}} a$ of each check $a \in [m]$ is partitioned into two sets, $\textsc{Heads}_a$ and $\textsc{Tails}_a$.

In our construction, in addition to assuming that the base code $\boldsymbol{B}$ has random density-$\frac{\Delta}{n}$ checks as in Section 3.1, we also assume:

- the partition into types is the trivial one, $T_1 = \{1, \ldots, m/k\}$, $T_2 = \{m/k+1, \ldots, 2m/k\}$, etc. (in fact this doesn't matter since $\boldsymbol{B}$ is random in the first place);
- and, the neighborhood $\partial^{\mathsf{T}} a$ of each check $a \in [m]$ is partitioned uniformly at random into $\textsc{Heads}_a$ and $\textsc{Tails}_a$.

Finally, for each set T_τ we choose the twists φ_τ so that a certain graph defined later is a good spectral expander. It will be shown that choosing the φ_τ uniformly at random from $\ell, 2\ell, \ldots, (\ell-1)\ell$ will give the desired spectral expansion with high probability (which can then certified if desired).

Then, given this partitioning we define for any pair b^1, a with $a \in \partial b^1$, the twist $\varphi(b^1, a)$ to be given as follows: check node a is in some set T_τ. If b^1 is in $\textsc{Heads}_a$ then $\varphi(b^1, a) = 0$. Else if b^1 is in $\textsc{Tails}_a$ then $\varphi(b^1, a) = \varphi_\tau$.

3.4 Homology representative weight

We now prove bounds on the least weight representative of homology. We will reduce this problem to proving certain properties of a classical code which we study in Section 4.

We use the assumption that the twists are multiples of ℓ. Given a horizontal 1-cell $b \otimes f$, we say that $f \in \{0, 1, \ldots, m-1\}$ is the *fiber position* of $b \otimes f$. Let us first simplify homology representatives:

Lemma 3.9. *For any 1-homology representative of weight w, there exists a homologous representative r of weight at most w such that the fiber position of any nonzero horizontal cell in the support of r is 0 mod ℓ.*

Proof. To prove this, we use a method we call "sliding." Consider a graph[8] $\textsc{Horz}$ whose nodes correspond to the horizontal cells of a given homology representative s and with a link between two horizontal cells if and only if their boundaries overlap. The fiber positions of a pair of linked horizontal cells differ by 0 mod ℓ because the twists are multiples of ℓ.

If $\textsc{Horz}$ is connected, then we can translate s along the fibers, i.e., all fiber cells f_j^0 and f_i^1 of $s = \sum_j b_j \otimes f_j^0 + \sum_i a_i \otimes f_i^1$ are replaced by shifted fiber cells $f_j^0 + y$ and $f_i^1 + y$ with a common $y = \pm 1 \in \mathbb{Z}_{n_{\mathcal{F}}} = \mathbb{Z}_{m_{\mathcal{F}}}$. If we keep translating the representative until the fiber position of any horizontal cell is 0 mod ℓ, then we have the claim of the lemma.

If $\textsc{Horz}$ has more than one connected cluster (a maximal connected subset of nodes), then we "slide" any one cluster C of horizontal cells in two steps as follows. First, we translate all the horizontal cells h in C by shifting their fiber components by ± 1 as above. Second, we add vertical cells on the fibers over base 0-cells where the boundary $\partial \sum_{h \in C} h$ is supported, so that the overall chain is still closed. We need one and only one vertical cell around each 0-cell of $\partial \sum_{h \in C} h$. This

[8] This graph $\textsc{Horz}$ has nothing to do with any other graph in this paper.

sliding is clearly a modification of the original representative s by the boundary of a 2-chain (a Z-stabilizer). Every added vertical cell in the second step contributes to ± 1 to the weight of vertical part of the homology representative. In fact, the total weight is a piecewise linear function of the amount that the cluster is slid, with the slope of this function constant until the cluster becomes connected to another cluster, at which point we modify the graph HORZ by adding links.

Hence, we end up with a single cluster that contains all the horizontal cells of the slid homology representative, and we finish by an overall translation as before. □

An intuitive picture for this sliding is as follows: there are 1-chains in the fiber whose endpoints are such as to cancel the boundary of h; we can think of these 1-chains as "strings" that are "pulling" on h; we slide in the direction in which the strings pull most strongly (or pick a direction arbitrarily if there is no preferred direction). Once the cluster becomes connected to another cluster, we slide that combined cluster, and so on. Continue this until there is only a single connected cluster. All the vertices of the cluster must the same fiber position mod ℓ; finally, slide that cluster until the claim is obeyed.

The sliding simplifies the problem of the weight of homology cycles as follows. Note that the base code's minimum distance is $\Theta(n)$ with high probability; see Proposition 3.5.

Lemma 3.10. *Assume that the base code has distance $\Theta(n)$. The weight of a nontrivial homology representative is lower bounded by the minimum of the following problem.*

Consider a chain $h = \sum_b b \otimes f_b$ consisting of horizontal cells whose fiber positions are 0 mod ℓ. *For any* 0*-cell a in the base, say that a* has an error *if ∂h is nonvanishing in the fiber over a. Then, minimize $|h| + \ell |\partial h|_{\mathrm{sw}}$, subject only to the requirement that there are $\Theta(n)$ different b such that $f_b \neq 0$.*

Proof. Write a nontrivial 1-homology cycle r as $r = h + v$ where h is the horizontal part and v the vertical part. Let $h = \sum_b b \otimes f_b$, where the sum is over base 1-cells b and f_b is a fiber 0-chain. The bundle projection $\Pi(h)$ is a nontrivial element of $H_1(\mathcal{B})$ by Lemma 2.5. Since the base code has distance $\Theta(n)$, then the set of b such that $f_b \neq 0$ has cardinality $\Theta(n)$.

Slide as above. Then, the weight $|v|$ of the vertical part of $r = h + v$ is at least ℓ times the number of base 0-cells that have an error; here we use that after sliding all the nonzero "strings" in v must have length at least ℓ. □

This lemma implies a significant simplification of the problem: it suffices to consider just horizontal chains. Theorem 4.9 proven below implies that any horizontal 1-chain h of a sufficiently small weight $|h|$ compared to $n\ell/\Delta$ will have at least $\Omega(|h|)$ errors. Hence for any nontrivial homology representative $r = h + v$ with the horizontal part h and the vertical part v we have either $|h| \geq n\ell/\Delta$ or $|v| \geq \Omega(n\ell)$ where the latter case is because we must have $|h| = \Omega(n)$. Hence:

Lemma 3.11. *Assume $m = \Theta(n)$. With high probability, the weight of any nontrivial representative of $H_1(\mathcal{E})$ is $\Omega(n\ell/\Delta)$.*

Theorem 4.9 is however phrased in terms of a classical code that we call a *twist graph code*. This is the error correcting code whose bits are hoizontal cells of $\mathcal{E}$ and whose checks are obtained from 0-cells of $\mathcal{E}$ in the obvious way. However, we will find it useful to explicitly redefine this code in terms of a graph that we call the twist graph in order to use expansion properties.

4 Coding Bounds for Twist Graph Code

We now define the twist graph code. This is simply a restatement of the checks on horizontal 1-chains in more graph theoretic terms. We are concerned with checks on horizontal 1-chains which obey the condition of Lemma 3.10 that $h = \sum_j j \otimes f_j$ with all $(f_j)_i = 0$ unless $i = 0 \bmod \ell$. So, throughout this section, we will regard these 1-chains as bit strings of length $n\ell$ rather than of length $n \cdot n_F$.

4.1 Twist graph and assumptions on twists

We define the twist graph $\vec{S}$ as follows. This is a directed graph, possibly with multi-edges (but without self-loops). We have vertices chosen from $[\ell]$. For each $t \in [k]$, there is an edge from each vertex i to $i + \varphi_t/\ell \pmod \ell$. Note: the twist φ_t is a multiple of ℓ so $\varphi_t/\ell \in \{1, 2, \ldots, \ell - 1\}$.

While the twist graph is obtained as described in the above paragraph, we will in this section only use the following more general assumptions on $\vec{S}$:

Definition 4.1. It is assumed we have a directed graph $\vec{S}$, with vertex set U of cardinality ℓ, and k "types" of directed edges. This graph may have multi-edges but no self-loops. We assume each vertex in $\vec{S}$ has one in-edge of each type, and also one out-edge of each type.

Notation 4.2. We write S for the undirected version of $\vec{S}$, which is a $2k$-regular graph. Letting $1 = \kappa_1 \geq \kappa_2 \geq \cdots \geq \kappa_\ell \geq -1$ denote the eigenvalues of the normalized adjacency matrix of S, we write $\kappa_S = \max\{\kappa_2, |\kappa_\ell|\}$ for the second-largest in magnitude.

We will need expansion properties of this graph S. From [AR94], for any $\kappa > 0$, for $k = O(\log(\ell)/\kappa^2)$, with probability at least .999 we have $\kappa_S \leq \epsilon$. We will ultimately choose ϵ to be a very small universal constant (see Assumption 4.8); thus $k = O(\log \ell) = O(\log m_{\mathcal{F}})$.

4.2 Twist graph code

Definition 4.3. Given $\vec{S}$ and a partitioned base code B, we define a new $\mathbb{F}_2$-linear code $B(\vec{S})$ as follows: The block length of $B(\vec{S})$ is $\ell \cdot n$, and we write a received word $w \in \mathbb{F}_2^{U \times [n]}$ as $w = (w_u)_{u \in U}$, where $w_u \in \mathbb{F}_2^n$. The number of parity checks in $B(\vec{S})$ will be $\ell \cdot m$, and they are defined as follows. Given a type $\tau \in [k]$ and a pair $(y, z) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$, we introduce the notion of the *type-τ checks on* (y, z), meaning all m/k parity checks of the form

$$\sum_{i \in \textsc{Tails}_a} y_i + \sum_{j \in \textsc{Heads}_a} z_j = 0 \pmod 2 \quad \text{for } a \in T_\tau. \tag{11}$$

Now $B(\vec{S})$ consist of imposing, for each directed edge (u, v) in $\vec{S}$ of type τ, all type-τ checks on (w_u, w_v).

Note that for a fixed a, each of $\textsc{Heads}_a$ and $\textsc{Tails}_a$ is a random density-$\frac{\Delta}{2n}$ subset of $[n]$, but these two sets are not quite independent (since any $j \in [n]$ is in at most one of them). We will use the following fact:

Proposition 4.4. *Let $(y, z) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$. For a randomly chosen and randomly partitioned check imposed on (y, z), the probability it is violated is*

$$\frac{1}{2} - \frac{1}{2}\left(1 - \frac{\Delta}{n}\right)^{|y+z|}\left(1 - \frac{2\Delta}{n}\right)^{|y \cap z|} \geq \frac{1}{2} - \frac{1}{2}\exp\left(-(|y| + |z|)\frac{\Delta}{n}\right).$$

Proof. Call the check a. We use the notation $\chi_J(x) = \prod_{j\in J}(-1)^{x_j}$ and write $\boldsymbol{I} = \textsc{Tails}_a$, $\boldsymbol{I}' = \textsc{Heads}_a$ for brevity. From eq. (11), the probability of violation is

$$\mathbf{E}\big[\tfrac{1}{2} - \tfrac{1}{2}\chi_{\boldsymbol{I}}(y)\chi_{\boldsymbol{I}'}(z)\big] = \tfrac{1}{2} - \tfrac{1}{2}\prod_{j=1}^{n}\mathbf{E}[\chi_{\boldsymbol{I}_j}(y_j)\chi_{\boldsymbol{I}'_j}(z_j)],$$

where we used independence of the pairs $(\boldsymbol{I}_j, \boldsymbol{I}'_j)$ across $j \in [n]$. By first considering whether or not $\partial^{\mathsf{T}} a \ni j$, and if so, whether $j \in \boldsymbol{I}$ or $j \in \boldsymbol{I}'$, we compute

$$\mathbf{E}[\chi_{\boldsymbol{I}_j}(y_j)\chi_{\boldsymbol{I}'_j}(z_j)] = \begin{cases} 1 & \text{if } y_j = z_j = 0, \\ 1 - \frac{\Delta}{n} & \text{if } y_j + z_j = 1, \\ 1 - \frac{2\Delta}{n} & \text{if } y_j = z_j = 1. \end{cases}$$

Thus our expression for the exact probability of violation follows. As for the inequality, it uses

$$\left(1 - \frac{\Delta}{n}\right)^{|y+z|}\left(1 - \frac{2\Delta}{n}\right)^{|y\cap z|} \le \exp(-|y+z|\Delta/n)\exp(-2|y\cap z|\Delta/n)$$

and the fact that $|y| + |z| = |y+z| + 2|y\cap z|$. □

Below we establish a property of $\boldsymbol{B}(\vec{S})$ that is minor variant of a standard property of random LDPC codes, that words of small Hamming weight violate a proportionate number of parity checks. We first need to upgrade the assumption eq. (9):

Assumption 4.5. We now make the stronger assumption

$$\beta k \ln n \le \Delta = \Delta(n) \le n^{o(1)},$$

where β is a large universal constant to be fixed later. As a remark, since k will eventually be $\Theta(\log n)$, the above constraint is the reason for our eventual choice of $\Delta = \Theta(\log^2 n)$.

Lemma 4.6. *For random $\boldsymbol{B}(\vec{S})$ as described, except with probability at most $O(1/n^{100})$ the following holds: For any type $\tau \in [k]$ and any pair $(y,z) \in \mathbb{F}_2^n \times \mathbb{F}_2^n$ with $|y|, |z| \le \frac{4}{\Delta}n$, the number of violated type-τ checks on (y,z) is at least $.01\frac{\Delta}{k}(|y| + |z|)$.*

Proof. Fix any type τ and any pair (y,z) with $|y|+|z| = t$, where $0 < t \le \frac{8}{\Delta}n$. Using Proposition 4.4, the probability that a given constraint $a \in T_\tau$ is violated is at least

$$\tfrac{1}{2} - \tfrac{1}{2}\exp(-\Delta t/n) \ge \tfrac{1}{2} - \tfrac{1}{2}(1 - .1\Delta t/n) > .04\Delta t/n,$$

where the first inequality used $t \le \frac{8}{\Delta}n$. Thus if we consider all $m/k = (3/4)n/k$ randomly chosen checks in T_τ, the expected number that are violated is at least $\mu := .03\Delta t/k$. A Chernoff bound therefore implies that

$$\mathbf{Pr}[(y,z) \text{ violates fewer than } \tfrac{1}{3}\mu = .01\tfrac{\Delta}{k}(|y| + |z|) \text{ checks from } T_\tau] \le \exp(-\tfrac{2}{9}\mu).$$

Taking a union bound over all $\tau \in [k]$ and all pairs (y,z) with $|y| + |z| = t$ (of which there are at most $\binom{2n}{t} \le (2en/t)^t$) yields a failure probability of at most

$$k(2en/t)^t \exp(-\tfrac{2}{9}\mu) \le n^{o(1)} \exp(\ln(2en/t) - (.06/9)\Delta/k)^t \le 1/n^{100t},$$

provided the constant β in Assumption 4.5 is large enough. Now taking a union bound over all $t \ge 1$ completes the proof. □

Now we come to a subtler property of the code $\boldsymbol{B}(\vec{S})$: Even if a received word $w = (w_u)_{u\in U}$ has some w_u that does *not* have low Hamming weight (it may even have Hamming weight $n/2$), still there will be $\Omega(|w_u|)$ violated checks among *all* the checks involving w_u. It is crucial here is that in eq. (8) we ensured $2m > n$; in this way, when the $2m$ checks involving w_u are chosen at random, there is a chance for the union bound to overcome the 2^n possibilities for w_u. (More precisely, when the m checks are chosen at random and then split into HEADS/TAILS pairs, the probability of at least one violation in the pair is close to .75 as shown in ineq. (12) below, so the probability of no violation in all pairs is roughly $.25^m = 2^{-2m}$.)

Lemma 4.7. *For random $\boldsymbol{B}(\vec{S})$ as described, except with probability at most $2^{-\Omega(n)}$ the following holds: Let $x \in \mathbb{F}_2^n$ be any word with $|x| \geq \frac{4}{\Delta}n$. Let $\Upsilon \subseteq [k]$ have cardinality at least $.98k$. Assume we have a sequence of pairs $(x, y_\tau)_{\tau\in\Upsilon}$ and $(z_\tau, x)_{\tau\in\Upsilon}$ in $\mathbb{F}_2^n \times \mathbb{F}_2^n$, where $|y_\tau|, |z_\tau| \leq \frac{2}{\Delta}n$ for all $\tau \in \Upsilon$. Then, aggregating the type-τ checks on (x, y_τ) and (z_τ, x) for all $\tau \in \Upsilon$, there are at least $.01n$ parity check violations (out of a total of $2|\Upsilon|m/k$).*

Proof. There are at most $2^n \cdot 2^k \cdot (2^{h_2(2/\Delta)n})^{2k} \leq 2^{(1+o(1))n}$ choices for x, Υ, the y_τ's, and the z_τ's (where the inequality used Assumption 4.5). Let us fix any such choices and consider the random partitioned code $\boldsymbol{B}$. For a given type $\tau \in \Upsilon$, we have m/k randomly chosen and randomly partitioned checks that get imposed on (x, y_τ) and also on (z_τ, x). Consider one such check $a \in T_\tau$. Using the notation χ_J and $\boldsymbol{I}, \boldsymbol{I}'$ from Proposition 4.4, the probability that the two checks a imposes on (x, y_τ) and (z_τ, x) are *both satisfied* is

$$\begin{aligned}\mathbf{E}[(\tfrac{1}{2} + \tfrac{1}{2}\chi_{\boldsymbol{I}}(x)\chi_{\boldsymbol{I}'}(y_\tau))(\tfrac{1}{2} + \tfrac{1}{2}\chi_{\boldsymbol{I}}(z_\tau)\chi_{\boldsymbol{I}'}(x))] \\ = \tfrac{1}{4} + \tfrac{1}{4}\mathbf{E}[\chi_{\boldsymbol{I}}(x)\chi_{\boldsymbol{I}'}(y_\tau)] + \tfrac{1}{4}\mathbf{E}[\chi_{\boldsymbol{I}}(z_\tau)\chi_{\boldsymbol{I}'}(x))] + \tfrac{1}{4}\mathbf{E}[\chi_{\boldsymbol{I}}(x + z_\tau)\chi_{\boldsymbol{I}'}(x + y_\tau)].\end{aligned}$$

Using the reasoning from Lemma 4.6, we have

$$\mathbf{E}[\chi_{\boldsymbol{I}}(x)\chi_{\boldsymbol{I}'}(y_\tau)] \leq \exp(-(|x| + |y_\tau|)\Delta/n) \leq \exp(-4),$$

where we used $|x| \geq \frac{4}{\Delta}n$. Similarly $\mathbf{E}[\chi_{\boldsymbol{I}}(z_\tau)\chi_{\boldsymbol{I}'}(x))] \leq \exp(-4)$, and also

$$\mathbf{E}[\chi_{\boldsymbol{I}}(x + z_\tau)\chi_{\boldsymbol{I}'}(x + y_\tau)] \leq \exp(-(|x + z_\tau| + |x + y_\tau|)\Delta/n) \leq \exp(-4),$$

where we used $|x + y_\tau|, |x + z_\tau| \geq \frac{2}{\Delta}n$. We may therefore conclude

$$\mathbf{Pr}[\text{at least one of } a\text{'s checks on } (x, y_\tau) \text{ and } (z_\tau, x) \text{ is } \textit{unsatisfied}] \geq 1 - (\tfrac{1}{4} + \tfrac{3}{4}\exp(-4)) \geq .73. \tag{12}$$

As a consequence, when the (at least $.98m$) parity checks in $(T_\tau)_{\tau\in\Upsilon}$ are chosen at random, the number of violations among those imposed on the (x, y_τ)'s and the (z_τ, x)'s stochastically dominates a Binomial$(.98m, .73)$ random variable. A short calculation[9] shows that such a binomial random variable is smaller than $.01 \cdot \frac{4}{3}m = .01n$ with probability at most $O(2^{-1.2n})$. Thus the theorem holds by a union bound, since $2^{(1+o(1))n} \cdot O(2^{-1.2n}) = 2^{-\Omega(n)}$. □

4.3 Low-weight words violate many checks

Recall that κ_S is the second largest eigenvalue of the undirected graph S.

Assumption 4.8. We assume $\kappa_S \leq .00002$.

[9] For a Binomial(t, p) random variable $\mathbf{X}$, we know $\mathbf{Pr}[\frac{1}{t}\mathbf{X} < p - \epsilon] < \exp(-tD(p - \epsilon\|p))$ for $0 < \epsilon < p$, where $D(x\|y) = x\ln(x/y) + (1 - x)\ln((1 - x)/(1 - y))$.

We recall the following form of the Expander Mixing Lemma [AC88, Vad12, Lem. 4.15]:

Expander Mixing Lemma. *If A_1 and A_2 are subsets of vertices in S, and $(\boldsymbol{u}, \boldsymbol{v})$ is a uniformly random edge (with orientation), then*

$$|\mathbf{Pr}[\boldsymbol{u} \in A_1 \text{ and } \boldsymbol{v} \in A_2] - \alpha_1\alpha_2| \leq \kappa_S \sqrt{\alpha_1(1-\alpha_1)\alpha_2(1-\alpha_2)},$$

where α_i denotes $|A_i|/|U|$ for $i = 1, 2$.

Our goal for the remainder of this section is to prove the following:

Theorem 4.9. *Suppose B satisfies the conclusions of Lemma 4.6 and Lemma 4.7. Then every $w \in \mathbb{F}_2^{U \times [n]}$ of relative Hamming weight at most $\epsilon := .0002/\Delta$ (i.e., with $|w| \leq \epsilon \ell n$) violates at least $.004|w|$ parity checks in $B(\vec{S})$.*

Proof. Let B and w be as given, and write $w = (w_u)_{u \in U}$ for $w_u \in \mathbb{F}_2^n$. For a given $u \in U$, the subword w_u participates in $2m$ parity checks from $B(\vec{S})$; we will define $\mathrm{Viol}(u)$ to be the number of these checks that are violated. By double-counting, we need to show that

$$\sum_{u \in U} \mathrm{Viol}(u) \geq .008|w| = .008 \sum_{u \in U} |w_u|. \tag{13}$$

Let us say that $u \in U$ is

$$\textit{light} \text{ if } |w_u|/n \leq 2/\Delta, \quad \textit{heavy} \text{ if } |w_u|/n \geq 4/\Delta, \quad \textit{medium} \text{ if } 2/\Delta < |w_u|/n < 4/\Delta.$$

We write $L \subseteq U$ (respectively H, M) for the subset of light (respectively heavy, medium) vertices, and we write $\theta_L = |L|/|U|$ (respectively θ_H, θ_M) for their fractional size. By Markov's inequality we have

$$\theta_H \leq \epsilon\Delta/4 \leq .00005 \qquad \text{and} \qquad \theta_L \geq 1 - \epsilon\Delta/2 \geq .9999. \tag{14}$$

Say that $u \in L \cup M$ is "bad" if $\mathbf{Pr}_{\boldsymbol{v} \sim u}[\boldsymbol{v} \in H] \geq .9$, where $\boldsymbol{v} \sim u$ denotes that $\boldsymbol{v}$ is a uniformly random neighbor of u in S. By the Expander Mixing Lemma, if $\boldsymbol{u} \sim L \cup M$ is chosen uniformly at random, and then $\boldsymbol{v} \sim \boldsymbol{u}$ is a randomly chosen neighbor, the probability of $\boldsymbol{v} \in H$ is at most

$$\theta_H + \kappa_S \theta_H (1 - \theta_H) \leq 1.1\theta_H$$

where we used Assumption 4.8 (with much room to spare). It thus follows from Markov's inequality that at most a $\frac{1.1\theta_H}{.9} \leq 1.25\theta_H$ fraction of $u \in L \cup M$ are bad. Now for a *good* $u \in L \cup M$, which has at least a .1-fraction of its neighbors also in $L \cup M$, the conclusion of Lemma 4.6 tells us that

$$\mathrm{Viol}(u) \geq .1 \cdot 2k \cdot .01 \tfrac{\Delta}{k} |w_u| = .002\Delta|w_u|.$$

Thus

$$\sum_{u \in L \cup M} \mathrm{Viol}(u) \quad \geq \quad \underbrace{.002\Delta \sum_{u \in L \cup M} |w_u|}_{\textbf{Main}} \quad - \quad .01|H|n, \tag{15}$$

where the justification for the subtracted term is that

$$\sum_{\text{bad } u \in L \cup M} .002\Delta|w_u| \leq 1.25\theta_H |L \cup M| \cdot .002\Delta(4/\Delta)n \leq .01|H|n.$$

We now divide into two cases.

Case 1: $|H| \leq .001|M|$. In this case, the subtracted term in ineq. (15) is at most $.00001|M|n$, whereas

$$\textbf{Main} \geq .002\Delta \cdot |M|(2/\Delta)n = .004|M|n.$$

Thus the subtracted term is at most a $\frac{.00001}{.004} \leq .003$ fraction of **Main**, and hence the right-hand side of ineq. (15) is at least $.997 \cdot \textbf{Main}$. Furthermore,

$$\sum_{u \in H} |w_u| \leq |H|n \leq .001|M|n \leq .25 \cdot .004|M|n \leq .25 \cdot \textbf{Main}.$$

Subtracting this from ineq. (15) we conclude that

$$\sum_{u \in L \cup M} \mathrm{Viol}(u) - \sum_{u \in H} |w_u| \geq (.997 - .25) \cdot \textbf{Main} \geq .001\Delta \sum_{u \in L \cup M} |w_u|,$$

which certainly implies ineq. (13) (as we may assume $\Delta \geq 8$).

Case 2: $|H| > .001|M|$. In this case, by the Expander Mixing Lemma, if $\boldsymbol{u} \sim H$ is chosen uniformly at random, and $\boldsymbol{v}$ is a random neighbor of $\boldsymbol{u}$ in S, the probability of $\boldsymbol{v} \in L$ is at least

$$\theta_L - \kappa_S \sqrt{(1-\theta_H)\theta_L(1-\theta_L)/\theta_H} \geq .9999 - \kappa_S \sqrt{(\theta_M + \theta_H)/\theta_H} \geq .9999 - \kappa_S \sqrt{1001} \geq .999,$$

where the first inequality used ineq. (14), then we used $(\theta_M + \theta_H)/\theta_H \leq 1001$ from Case 2, and finally the last inequality used Assumption 4.8. Now say that vertex $u \in H$ is "bad" if $\mathbf{Pr}_{\boldsymbol{v} \sim u}[\boldsymbol{v} \in L] < .99$. A Markov argument now implies that at most a $\frac{1-.999}{1-.99} = .1$ fraction of $u \in H$ are bad. As for the at least $.9|H|$ "good" $u \in H$, each has k out-neighbors and k in-neighbors in $\vec{S}$; by the definition of "goodness", for at least a .98-fraction of the $\tau \in [k]$ we have that both the τth out-neighbor *and* the τth in-neighbor are in L. The conclusion of Lemma 4.7 then tells us that $\mathrm{Viol}(u) \geq .01n$ for each good $u \in H$. Thus

$$\sum_{u \in H} \mathrm{Viol}(u) \geq .9|H| \cdot .01n = .009|H|n.$$

Adding this to .1 times ineq. (15) yields

$$.1 \sum_{u \in L \cup M} \mathrm{Viol}(u) + \sum_{u \in H} \mathrm{Viol}(u) \geq .0002\Delta \sum_{u \in L \cup M} |w_u| + .008|H|n \geq .008 \sum_{u \in U} |w_u|$$

(as we may assume $\Delta \geq 40$), and this implies ineq. (13) as needed. □

5 Decoding

In these sections, we discuss decoding the code constructed for Theorem 1.1. The code of Corollary 1.2 is obtained from this code by a weight-reduction operation followed by a distance-balancing operation. We believe (but leave it to future work) that it is fairly straightforward to show that given a decoder for some code one can also find a decoder for the weight-reduced code with similar efficiency. To decode the distance-balanced code, we can rely on the result of [EKZ20] which gave a general means to decode a distance-balanced quantum code.

5.1 Decoding cohomology

In this section we consider the problem of decoding against X errors. Recall that we have associated Z stabilizers with the 2-cells of $\mathcal{E}$. So, the question of decoding against X errors means reconstructing an error pattern e (up to stabilizers of the code) on 1-cells from its coboundary s on 2-cells, where s is referred to as the *error syndrome*, in coding theory terminology. To express this in a language that is independent of our arbitrary choice to associate Z stabilizers with 2-cells, we refer to this as decoding cohomology. We assume throughout that $n_{\mathcal{F}} = m_{\mathcal{F}} = \Theta(n_{\mathcal{B}}) = \Theta(m_{\mathcal{B}})$. Our fiber bundle code has $N = n_{\mathcal{B}} \cdot m_{\mathcal{F}} + m_{\mathcal{B}} \cdot n_{\mathcal{F}} = \Theta(n_{\mathcal{B}}{}^2)$ qubits.

We give a polynomial time decoder (more precisely, $\mathrm{poly}(N)$ time) and show, under the same assumptions as in Lemma 3.7, that it decodes arbitrary errors up to a polylogarithmic fraction of d_X, meaning that if some error pattern e_0 occurs, with

$$|e_0| \leq \frac{1}{20} \frac{m_{\mathcal{B}}}{10^5 \Delta^2} \tag{16}$$

which is a sufficiently small polylogarithmic fraction of $d_X = \Omega(m_{\mathcal{F}}/\Delta)$, then the algorithm decodes correctly, computing e_0 up to stabilizers. Without loss of generality, we may assume that e_0 is a minimal weight chain with the given coboundary. We also recall the "shadow" terminology from Definition 3.6.

Constructing e_{arb}. Given some syndrome $s = \partial^{\mathsf{T}} e_0$, the first stage of the algorithm is devoted to constructing some arbitrary 1-chain e_{arb} satisfying

$$s = \partial^{\mathsf{T}} e_{\mathrm{arb}} \quad \text{and} \quad |e_{\mathrm{arb}}|_{\mathrm{vsw}} \leq |s|.$$

In aid of this, we first define a linear map K from p-chains of $\mathcal{E}$ to $(p-1)$-chains of $\mathcal{B}$ (for arbitrary p) by specifying $K(b^{p-1} \otimes f^1) = b^{p-1}$ and $K(b^p \otimes f^0) = 0$. This map is in a sense "dual" to the bundle projection map Π, as K is nonvanishing on $(p,1)$-cells while Π is nonvanishing on $(p,0)$-cells. In words, K retains the base cells where the chain has an odd number of fiber 1-cells. Note that $K(s) = \partial^{\mathsf{T}} K(e_0)$.

The main challenge in constructing e_{arb} will be constructing a base 0-chain e_b such that

$$\partial^{\mathsf{T}} e_b = K(s) \quad \text{and} \quad |e_b| \leq |s|.$$

Having constructed this e_b, we may fix any 1-cell f^1 in $\mathcal{F}$, and then $\partial^{\mathsf{T}}(e_b \otimes f^1) = s$ up to a coboundary of some horizontal chain that can easily be constructed. Further, $|e_{\mathrm{arb}}|_{\mathrm{vsw}} = |e_b|$. So it remains to construct e_b.

Note that if we didn't require any bound on $|e_b|$, obtaining a solution to $\partial^{\mathsf{T}} e_b = K(s)$ would be a simple matter of linear algebra: the affine subspace defined by $\partial^{\mathsf{T}} e_b = K(s)$ is nonempty (since $\partial^{\mathsf{T}} K(e_0) = K(s)$) and we would just have to find any solution. To efficiently get the desired low-weight solution, we can run a Sipser–Spielman (belief propagation) type of decoding algorithm (see [SS96, Thm. 10], [HLW06, Thm. 12.9]). The setup is slightly nonstandard, in that the roles of the "code bits" and "check bits" are reversed: We think of the "code" as all solutions of $\partial^{\mathsf{T}} e_b = K(s)$, with each base 0-cell j enforcing the "affine parity check" $\sum_{a \in \partial j} (e_b)_a = 0/1$, the right-hand side depending on whether $j \in K(s)$. The algorithm begins with the "faulty" solution $e_b = 0$, which has Hamming distance at most $|K(e_0)|$ from a true solution, and repeatedly seeks to toggle a 0-cell in e_b so as to decrease the number of violated affine parity checks. Using the fact (Proposition 3.2) that we have expansion factor exceeding $\frac{3}{4}\Delta$ from the 0-cells to the 1-cells in the base (in fact, using Corollary 3.3), the Sipser–Spielman argument directly shows that, so long as

$|K(e_0)|+|K(s)| \le \frac{1}{10^5\Delta} m_{\mathcal{B}}$, the algorithm will terminate after at most $|K(s)|$ toggles, yielding an e_b with $\partial^{\mathsf{T}} e_b = K(s)$ and $|e_b| \le |K(s)| \le |s|$. Since $|K(e_0)| \le |e_0|$ and $|K(s)| = |\partial^{\mathsf{T}} K(e_0)| \le 1.01\Delta|e_0|$, we see the algorithm will succeed provided $|e_0|$ is bounded as in ineq. (16).

Amending e_{arb}. The second step of the algorithm is again a greedy decoder. We know by Proposition 2.10 there exists a bundle 0-chain w and a base 1-chain x such that

$$e_0 = e_{\mathrm{arb}} + \partial^{\mathsf{T}} w + x \otimes F_0.$$

For *arbitrary* w, x, let the *horizontal weight* denote the Hamming weight of the horizontal part of $e = e_{\mathrm{arb}} + \partial^{\mathsf{T}} w + x \otimes F_0$. We initialize $w = 0$. We greedily choose x to minimize the horizontal weight; this is linear time, given w. We then search for a *fixable* base 0-cell a:

Definition 5.1. For an arbitrary bundle 1-chain e, a base 0-cell a is *amended* if all of the following are true; otherwise, a base 0-cell is *fixable* for e.

(i) e contains at most $m_{\mathcal{F}}/2$ horizontal cells over any base 1-cell $b \in \partial^{\mathsf{T}} a$.

(ii) e contains at most half of all horizontal cells of $\partial^{\mathsf{T}}(a \otimes f^0)$ for any fiber 0-cell f^0.

(iii) Let $A = (\partial^{\mathsf{T}} a) \otimes F_0$. For any 1-cocycle $z \subseteq A + a \otimes F_1$, where $a \otimes F_1$ consists of all vertical cells over a, it holds that $|e \cap A| \le .8|(e+z) \cap A| + .2|A|$.

Remark that the vertical weight or the shadow weight of the vertical part of e has nothing to do with this amendableness. The condition (iii) means that e has horizontal weight on A which is fairly close to the minimum possible. Indeed, all three conditions are trivial if e has the minimum weight in A upon adding cocycles "near a" (i.e., supported on $A + a \otimes F_1$). The constant .8 is chosen because it is close to the approximation ratio achievable by the Goemans-Williamson algorithm. We will elaborate on this shortly.

Our algorithm will repeatedly find any fixable 0-cell a and amend it by changing w and x in $e_{\mathrm{arb}} + \partial^{\mathsf{T}} w + x \otimes F_0$ until no further fixable 0-cells can be found. At this point, the algorithm terminates and declares the final chain to be the decoding. Note that the algorithm terminates after at most N fixes, since the horizontal weight always decreases.

Testing if a is fixable. We have to explain how to test whether a base 0-cell a is fixable. A simple method is by bruteforce optimization. As remarked above, all we have to do is to optimize the horizontal weight by adding various cocycles. This can be done in quasipolynomial time $2^{O(\Delta)} \operatorname{poly}(m_{\mathcal{F}})$ as follows. There are $2^{O(\Delta)}$ base 1-chains supported on $\partial^{\mathsf{T}} a$. For each such 1-chain x, we consider $e + \partial^{\mathsf{T}}(a \otimes y) + x \otimes F_0$ with a fiber 0-chain y varying. Since we only care about the horizontal weight, each bit (a fiber 0-cell) of y can be independently optimized by a "majority vote" over the horizontal cells in its coboundary.

The bruteforce optimization above is more than necessary. In fact, our definition of fixable cells is designed to adopt approximate optimization. Let us first consider the condition (iii). Suppose a cocycle $z = z^{\mathrm{opt}}$ supported on $A + a \otimes F_1$ minimizes $|(e+z) \cap A|$. Then, the condition (iii) is equivalent to demanding that

$$|A| - |A \cap e| \ge .8\big(|A| - |A \cap (e + z^{\mathrm{opt}})|\big).$$

Let us call $|A| - |A \cap e'|$ the *local satisfaction* of e' for any bundle 1-chain e'. Let us think of a base 1-chain x supported on $\partial^{\mathsf{T}} a$ as a collection of binary variables $x_1, \dots, x_i, \dots, x_{|\partial^{\mathsf{T}} a|}$. Similarly,

a fiber 0-chain y is a collection of binary variables $y_1, \ldots, y_j, \ldots, y_{m_{\mathcal{F}}}$. Then a horizontal cell $x_i \otimes \varphi(x_i, a)^{-1} y_j \in \partial^{\mathsf{T}}(a \otimes y_j)$ in A can be given coordinates (i, j). This horizontal cell at (i, j) over $\partial^{\mathsf{T}} a$ is unoccupied in $e + \partial^{\mathsf{T}}(a \otimes y) + x \otimes F_0$ if and only if $x_i + y_j = e_{i,j} \bmod 2$ where $e_{i,j}$ is the occupancy of e on (i, j). Thus, the local satisfaction is the number of these satisfied equations. The derandomized Goemans-Williamson algorithm [MR99] gives an approximately optimal solution x, y such that the local satisfaction of $e + \partial^{\mathsf{T}}(a \otimes y) + x \otimes F_0$ is higher than .878 times the optimum local satisfaction.

Once the condition (iii) is met, we can alternatingly optimize x or y while withholding the other. Each round of this optimization takes time $\mathrm{poly}(m_{\mathcal{F}}, \Delta)$. This alternation terminates before $O(\Delta m_{\mathcal{F}})$ rounds because the local satisfaction must increase. The conditions (i) and (ii) are then fulfilled. Overall, it takes $\mathrm{poly}(N)$ time to test if a is fixable using the described approximate optimization.

Overview of the analysis. Let $w(\tau), x(\tau)$ denote the states of w, x after τ steps of the algorithm. Let

$$e(\tau) = e_{\mathrm{arb}} + \partial^{\mathsf{T}} w(\tau) + x(\tau) \otimes F_0.$$

The proof of correctness will take two parts. In the first part, we assume that at the end of the algorithm the shadow weight of $w(\tau)$ is sufficiently small. Under this assumption, we show that when the algorithm terminates it has correctly decoded; i.e., the final $e(\tau)$ is equal to e_0 up to a coboundary.[10] In the second part we show that indeed the shadow weight of $w(\tau)$ indeed remains sufficiently small throughout the algorithm, using the fact that $|e_{\mathrm{arb}}|_{\mathrm{vsw}} \leq |s|$.

We will find something unusual in the second part of the proof: the proof that the shadow weight of $w(\tau)$ remains small does not just use the fact that the algorithm terminates after a certain number of steps, but rather uses graph-theoretic expansion properties. Indeed, nothing we show rules out the algorithm running for $\gg n_{\mathcal{B}}$ steps.

First part of the analysis. Here we show correctness assuming $|w(\tau)|_{\mathrm{sw}}$ is small. Note that this implies $|e(\tau)|_{\mathrm{vsw}}$ is also small.

Lemma 5.2. *Suppose that the algorithm terminates after τ steps, and suppose that $|e(\tau)|_{\mathrm{vsw}} + |e_0| \leq \frac{1}{10^5 \Delta} m_{\mathcal{F}}$. Then $e(\tau)$ is cohomologous to e_0.*

Proof. Say a 1-chain g in $\mathcal{E}$ is a *background* if $\partial^{\mathsf{T}} g = \partial^{\mathsf{T}} e(\tau)$. As per the discussion in Lemma 3.7, for any background g there is a unique decomposition

$$e(\tau) = g + \partial^{\mathsf{T}} w + x \otimes F_0,$$

where the 0-chain $w = \sum_a a \otimes y_a$ has $|y_a| < n_{\mathcal{F}}/2$ for all a and where x is a 1-chain in $\mathcal{B}$. We call w the *stabilizer* associated to g, and we call x the *logical*. It will also be important for us to keep track of the *stabilizer shadow* S of g (i.e., S is the shadow of w).

The proof will consider a sequence $g_0, g_1, g_2, \ldots$ of backgrounds, with the initial background g_0 being e_0. We will establish the following properties:

1. *The logical x_t of each g_t will be the same for all t.*
2. *The stabilizer shadow S_t decreases in size by 1 at each step, until it becomes empty.*

[10] Note that even with a maximum likelihood decoder it would not be possible to decode e_0 exactly in general since there may be different error patterns of the same weight and with the same coboundary.

3. *The logical x_t vanishes outside the coboundary of S_t, where $\partial^\mathsf{T} S_t = \bigcup_{a \in S_t} \partial^\mathsf{T} a$.*

4. *The horizontal weight of g_t on $\partial^\mathsf{T} S_t$ is smaller than $m_\mathcal{F}/100$.*

Using only Items 1 to 3, we observe that when the sequence ends at some t, we have an empty S_t, the logical x_t must therefore be everywhere zero, and hence the logical x_0 of e_0 also vanishes. This implies that e_0 is cohomologous to $e(\tau)$, as claimed.

Let us handle the base case of g_0, and then describe how $g_1, g_2, \ldots$ are inductively formed and why Items 1 to 4 hold.

Base case. Regarding the base case $g_0 = e_0$, we need to establish Items 3 and 4. The latter is immediate since the total horizontal weight of e_0 is already small by the beginning assumption of the decoder. As for Item 3, from $e(\tau) = e_0 + \partial^\mathsf{T} w_0 + x_0 \otimes F_0$ we see that on every base 1-cell b outside of $\partial^\mathsf{T} S_0$, we have agreement between $e(\tau)$ and e_0 up to the addition of $b \otimes F_0$, this addition being governed by whether b is in x_0. But e_0 has horizontal weight less than $n_\mathcal{F}/2$ on every base 1-cell, including b. Thus x_0 must vanish on b or else $e(\tau)$ would have horizontal weight more than $n_\mathcal{F}/2$ there, contradicting the condition (i) of amended cells (any cell in the neighborhood of b would be trivially fixable). This establishes Item 3, and thus the base case.

The inductive construction. Suppose we have formed g_t. If the stabilizer shadow S_t is empty, we are done. Otherwise, we apply Corollary 3.3 to S_t, obtaining some $a^\bullet \in S_t$. (Note that $|S_0| \leq \frac{1}{10^5 \Delta} m_\mathcal{F}$ by the hypotheses of the lemma, and so all subsequent stabilizer shadows also satisfy this bound, by Item 2.) Writing $w_t = \sum_a a \otimes y_a$, we form the next background by taking $g_{t+1} = g_t + \partial^\mathsf{T}(a^\bullet \otimes y_{a^\bullet})$ and $w_{t+1} = \sum_{a \neq a^\bullet} a \otimes y_a$; that is, we simply "shift" the $a^\bullet \otimes y_{a^\bullet}$ part of w_t to the background. Now Items 1 and 2 clearly hold, and it remains to verify Items 3 and 4.

Write the neighborhood of $a^\bullet$ in B as $C \cup D$, where the cells C are counique neighbors of S_t and the cells D are non-counique; Corollary 3.3 tells us that $|D| \leq \frac{1}{4}|C|$.

Inductively verifying Item 3. This is equivalent to showing that $x_{t+1} = x_t$ vanishes on C. To do this, we begin by observing that when the decoding algorithm terminated with $e(\tau)$, the cell $a^\bullet$ was not fixable. Let us evaluate this fact in the context of the decomposition $e(\tau) = g_t + \partial^\mathsf{T} w_t + x_t \otimes F_0$. Fixing cell $a^\bullet$ amounts to arbitrarily altering $y_{a^\bullet}$ in this decomposition, and then optimizing x_t to achieve minimal horizontal weight.

A first observation is that while we always have $|y_{a^\bullet}| \leq m_\mathcal{F}/2$, we claim that unfixability of $a^\bullet$ implies that in fact $|y_{a^\bullet}| \leq .4 m_\mathcal{F}$. Otherwise, $\partial^\mathsf{T} w_t$ puts horizontal weight between $.4 m_\mathcal{F}$ and $.5 m_\mathcal{F}$ on each cell of C, and from Item 4 we know that g_t modifies this by at most $.01 m_\mathcal{F}$ (collectively, even). Thus even after optimizing x_t, the contribution to $e(\tau)$'s horizontal weight from C is at least $.39 n_\mathcal{F}|C|$. On the other hand, if $y_{a^\bullet}$ were fixed to 0, the contribution to $e(\tau)$'s horizontal weight from $C \cup D$ would be at most $.5 m_\mathcal{F}|D| \leq .125 m_\mathcal{F}|C|$ from D, and at most $.01 m_\mathcal{F}$ from C (the possible contribution from g_t, recalling Item 3), for a total of at most $.135 m_\mathcal{F}|C|$. Then, we have $.8(.135 m_\mathcal{F}|C|) + .2 m_\mathcal{F}(|C| + |D|) \leq .358 m_\mathcal{F}|C| < .39 m_\mathcal{F}|C|$. This means $a^\bullet$ *was* fixable by violating the condition (iii) of the definition of amended cells, a contradiction. Thus we have established the claim $|y_{a^\bullet}| \leq .4 n_\mathcal{F}$.

But now we deduce that $g_t + \partial^\mathsf{T} w_t$ has horizontal weight at most $.41 m_\mathcal{F}$ on every cell of C. This indeed implies that x_t vanishes on C, since $e(\tau)$ also has horizontal weight less than $m_\mathcal{F}/2$ on each cell of C (indeed, as mentioned earlier $e(\tau)$ has horizontal weight less than $m_\mathcal{F}/2$ on every base 1-cell, else it would be trivially fixable).

Inductively verifying Item 4. Let H_{t+1} denote the horizontal weight of g_{t+1} on $\partial^{\mathsf{T}} S_{t+1}$, and H_t the horizontal weight of g_t on $\partial^{\mathsf{T}} S_t$. We will in fact show $H_{t+1} \leq H_t$, which is sufficient to inductively verify Item 4. We may write $H_{t+1} - H_t = \text{NEW} - \text{LOSS}$, where LOSS equals the horizontal weight of g_t on C, and where NEW is the weight gain in D when g_t is replaced by g_{t+1}. In turn, we can write LOSS as the sum of contributions LOSS_u from each fiber vertex u, and similarly write NEW as a sum of contributions NEW_u. Our goal will be to show

$$\text{NEW}_u \leq \text{LOSS}_u \quad \forall u. \tag{17}$$

For a fixed fiber vertex u, consider the bit value of the chain $y_{a^\bullet}$ on u, call it $(y_{a^\bullet})_u$. Since $a^\bullet$ is not fixable for $e(\tau)$, this value must be locally optimal (in terms of minimizing horizontal weight) given g_t, x_t, and given w_{t+1}, i.e. given the y_a for $a \neq a^\bullet$. That is, this value equals the "majority vote" — across all $j \in C \cup D$ — of the bits $z_{u,j} \coloneqq (g_t + \partial^{\mathsf{T}} w_{t+1} + x_t \otimes F_0)_{\varphi(j,a^\bullet)^{-1}u}$. Recall we already established that x_t vanishes on C and by construction C is not in the coboundary of the shadow of w_{t+1}; thus for $j \in C$ we simply have $z_{u,j} = (g_t)_{\varphi(j,a^\bullet)^{-1}u}$. Hence we precisely have $\text{LOSS}_u = |\{j \in C : z_{u,j} = 1\}|$.

As for NEW_u, it is zero if $(y_{a^\bullet})_u = 0$, in which case Ineq. (17) certainly holds. Suppose instead $(y_{a^\bullet})_u = 1$. Then, since the majority vote of $z_{u,j}$ across $j \in C \cup D$ is 1, we must have

$$\begin{aligned} |\{j \in C : z_{u,j} = 1\}| + |\{j \in D : z_{u,j} = 1\}| &\geq (|C| + |D|)/2 \\ \implies \quad \text{LOSS}_u + |D| &\geq (|C| + |D|)/2 \\ \implies \quad \text{LOSS}_u &\geq (|C| - |D|)/2, \end{aligned}$$

so, since $|C| \geq 3|D|$ we have $\text{LOSS}_u \geq |D|$. At the same time, trivially $\text{NEW}_u \leq |D|$, verifying Ineq. (17). □

Second part of the analysis. It remains to show that for any step τ of the decoding algorithm, $|w(\tau)|_{\text{sw}}$ is sufficiently small compared to $n_{\mathcal{B}}/\Delta$. Define Q to be the shadow of the horizontal part of e_{arb}, i.e., Q is the set of all base 1-cells over which there is some horizontal cell of e_{arb}. From the construction of e_{arb} we have

$$|Q| \leq |\partial^{\mathsf{T}} e_b| \leq 1.01\Delta |e_b| \leq 1.01\Delta |\partial^{\mathsf{T}} e_0| \leq 1.01\Delta(1.01\Delta + 2)|e_0| \leq 2\Delta^2 |e_0| \leq \tfrac{1}{10}\tfrac{1}{10^5} m_{\mathcal{B}}$$

Define $P(\tau)$ to be the set of all base 0-cells that were amended in the first τ steps of the algorithm. We have that the shadow of $w(\tau)$ is contained in $P(\tau)$; we may not have equality as it is possible that a cell could be amended multiple times. We are going to bound $|P(\tau)|$. Note that $P(0) = \emptyset$.

Suppose a base 0-cell a is first included in $P(\tau)$ at step $\tau' \leq \tau$, then since $e(0), e(1), \ldots, e(\tau' - 1)$ have no support on base 1-cells outside $Q \cup \partial^{\mathsf{T}} P(\tau' - 1)$, more than half of $\partial^{\mathsf{T}} a$ must be in $Q \cup \partial^{\mathsf{T}} P(\tau' - 1)$ for a to be amended. Hence, every cell $a \in P(\tau)$ must have at least half the cells in its coboundary either in Q or in the coboundary of some other cell in $P(\tau)$. This implies that the number $n_1(a)$ of counique neighbors (a base 1-cell) of a that are not in Q is at most the half of the degree of a in the base Tanner graph. That is, for any $a \in P(\tau)$ we have $n_1(a) \leq \frac{1}{2} \cdot 1.01\Delta$.

On the other hand, the number of noncounique neighbors of $P(\tau)$ cannot exceed $.1\Delta|P(\tau)|$ by Corollary 3.3.

By counting counique and noncounique neighbors of $P(\tau)$ that are not in Q, we have

$$\begin{aligned} |\partial^{\mathsf{T}} P(\tau)| &\leq |Q| + |\partial^{\mathsf{T}} P(\tau) \setminus Q| \\ &\leq |Q| + .1\Delta|P(\tau)| + \sum_{a \in P(\tau)} n_1(a) \\ &\leq |Q| + .1\Delta|P(\tau)| + .6\Delta|P(\tau)|. \end{aligned}$$

But Proposition 3.2 implies that so long as $|P(\tau)| \leq \frac{1}{10^5\Delta} m_{\mathcal{B}}$, we know the left-hand side $|\partial^{\mathsf{T}} P(\tau)|$ is at least $.9\Delta|P(\tau)|$; we deduce that

$$|P(\tau)| \leq \tfrac{1}{10^5\Delta} m_{\mathcal{B}} \quad \implies \quad |P(\tau)| \leq \tfrac{1}{.2\Delta}|Q|.$$

Using the assumption that $|Q| \leq 10^{-6} m_{\mathcal{B}}$, we see for all sufficiently large $m_{\mathcal{B}}$ that

$$|P(\tau)| \leq \tfrac{1}{2}\tfrac{m_{\mathcal{B}}}{10^5\Delta} \implies |P(\tau+1)| \leq |P(\tau)| + 1 \leq \tfrac{m_{\mathcal{B}}}{10^5\Delta} \implies |P(\tau+1)| \leq \tfrac{1}{.2\Delta}|Q| \leq \tfrac{1}{2}\tfrac{m_{\mathcal{B}}}{10^5\Delta}.$$

We conclude that $|P(\tau)|$ is always bounded by $\frac{1}{2}\frac{1}{10^5\Delta} m_{\mathcal{B}}$.

5.2 Decoding homology

We now give a proposed algorithm for decoding homology. We conjecture, but do not prove, that this algorithm decodes errors of weight up to a polylogarithmic fraction of d_Z.

The algorithm takes as input a 0-chain s_0 in $\mathcal{E}$ which contains the syndrome. It initializes a 1-chain u in $\mathcal{E}$ to 0. It initializes some 0-chain s in $\mathcal{E}$ to s_0. As the algorithm proceeds, it modifies the chain u by a sequence of local updates explained below. After each update, the algorithm then updates s so that $s = s_0 + \partial u$. The algorithm attempts by these local updates to reduce $|s|$.

Define a *fiber string* of length at most r to be a vertical 1-chain of Hamming weight at most r whose boundary consists of exactly 2 0-cells. The string can be thought of as stretching between between these two 0-cells. Here we assume that $r < \ell < n_F$.

The algorithm has a counter r, initialized at $r = 0$. The algorithm loops over $r = 0$ to $r = \ell/\operatorname{polylog}(n_F)$. For update r, the algorithm performs a greedy search, trying to reduce $|s|$ by either adding to u a fiber string of length at most r or by adding to u some horizontal cell plus some sum of fiber strings of length at most r.

The algorithm performs this greedy search for the given r until it is not possible to reduce $|s|$ further, at which point it increments r and continues the loop if $r < \ell/\operatorname{polylog}(n_F)$.

After the loop terminates, the algorithm then takes the given error chain s and bundle projects it to the base, giving a 0-chain in $\mathcal{B}$. It then attempts to find some 1-chain w in $\mathcal{B}$ such that ∂w is equal to the given 0-chain in $\mathcal{B}$ and so that $|w|$ is small compared to the distance of the base code. (We conjecture that the base code is such that, if the weight of the bundle projected s is small enough, then such a w can be found by a greedy search; this seems easier to prove than some other properties we need also.)

Finally, given w, one may find some horizontal 1-chain x in $\mathcal{E}$ whose bundle projection is equal to w, and with $|x| = |w|$; for example, one may simply take $x = w \otimes f^0$ for any fiber 0-cell f^0. Then, the chain $u + x$ has the property that its boundary, after bundle projection, is equal to s_0 after bundle projection, and so one can add some vertical 1-chain y to $u + x$ to obtain a 1-chain whose boundary is s_0; The algorithm then outputs this 1-chain.

Let us sketch why we conjecture this works. Suppose the true error pattern was some 1-chain e with $\partial e = s_0$. It is possible that e might, for example, even be a sum of stabilizers, in which case perhaps $s_0 = 0$ even though $e \neq 0$. Without loss of generality, we may assume that e is a minimal weight error pattern with $\partial e = s_0$. After the loop terminates, we have that $e + u$ is given by some sum $h + v$ of horizontal and vertical chains. We expect that $|h|$ will still be small compared to d_Z and so $|\partial h|$ will be large, indeed proportional to $|h|$ in some way depending on Δ. We expect however that many of the 0-cells in h will be attached to fiber strings of length $> r$ in v, where "attached" means that one of the two cells in the boundary of that string is the given 0-cell in h. However, assuming the greedy algorithm does not increase the weight of v too much compared to the weight of vertical cells in e, then for $r = \ell/\operatorname{polylog}(n_B)$, the weight of h must

then be small compared to n_B. So, at this point, the algorithm has (assuming these conjectures are correct) computed e up to some 1-chain $h+v$ with $|h|$ small. The algorithm then returns some other horizontal chain x, with $|x|$ small, and so computes e up to $h+v+x+y$, with h, x horizontal and having small Hamming weight and v, y vertical, i.e., it computes e up to a closed 1-chain such that the Hamming weight of this chain on horizontal cells is small compared to the distance of the base code. However, any such closed 1-chain is homologically trivial.

5.3 Decoding cohomology against erasure errors in almost linear time

With *erasure errors*, we are given syndrome bits and a specific set D of qubits (erased qubits) on which there are potential errors.[11] The location of the true errors is unknown and it is the decoder's goal to determine the true errors up to stabilizers. Here we present an algorithm for this erasure decoding problem where the actual errors are X, denoted as a bundle 1-chain $x \in \mathcal{E}_1$, under the assumption that $|\mathsf{D}|$ is less than a polylogarithmic fraction of $m_{\mathcal{F}}$, which is smaller than d_X by Lemma 3.7.

If we find any chain x' that reproduces the given syndrome, i.e., $\partial^{\mathsf{T}} x' = \partial^{\mathsf{T}} x$, with the constraint that $x' \subseteq \mathsf{D}$, then the combination $x + x'$ of the actual errors and a correction is coclosed and has weight less than d_X, and hence is a coboundary. It is thus obvious that this erasure decoding problem is solved in time $|\mathsf{D}|^3 \operatorname{poly}(\Delta)$ since x' is a solution of an inhomogeneous system of linear equations of $|\mathsf{D}|$ variables that participate in $|\mathsf{D}| \operatorname{poly}(\Delta)$ equations. It is however not so obvious whether such x' can be found in linear time in $|\mathsf{D}|$.

Lemma 5.3. *Assume the supposition of Lemma 3.7. If $|\mathsf{D}| < m_{\mathcal{F}}/(10^5\Delta^2)$, then in time $|\mathsf{D}| \operatorname{poly}(\Delta, \log|\mathsf{D}|)$ we can find a chain x' such that $x' + x$ represents the zero cohomology class.*

Proof. The algorithm is based on belief propagation that deforms D to decrease $|\mathsf{D}|$ down to zero. Let us define two sets $\mathsf{C} \subset \mathsf{D}$ and $\mathsf{P} \subset \mathcal{E}_{0,0} \times \mathcal{E}_{0,1}$. We will consider the time complexity to compute these sets after we show the correctness of the algorithm in the following.

C consists of all cells $e \in \mathsf{D}$ such that some 2-cell touches no other cells of D but e. The error on e is thus determined. By removing C from D and recording the determined errors and iterating these, we can sometimes eliminate all of D. For example, if $\mathsf{D}(0)$ at time step 0 is a collection of consecutive horizontal cells over a single base 1-cell, occupying an interval in the fiber, then $\mathsf{C}(0)$ would be just two end cells and the shrunk $\mathsf{D}(1) = \mathsf{D}(0) - \mathsf{C}(0)$ will have two end cells in $\mathsf{C}(1)$. This continues until $\mathsf{D}(t)$ becomes empty. If this iteration does not eliminate all of D, then we are left with $\mathsf{D}(t_1) \neq \emptyset$ where every 2-cell on the coboundary of some cell of $\mathsf{D}(t_1)$ meets at least two 1-cells of $\mathsf{D}(t_1)$. Let $\mathsf{D}(t_j)$ be any configuration such that $\mathsf{C}(t_j) = \emptyset$.

We use X-stabilizers (associated with bundle 0-cells) to rescue the situation. An X-stabilizer is $\partial^{\mathsf{T}} u$ for some bundle 0-cell u, and $\partial^{\mathsf{T}} u$ contains exactly two vertical cells, "between" which there are $(1 \pm .01)\Delta$ horizontal cells. Suppose that (i) $\mathsf{D}(t_j)$ contains at least one of the the vertical cells, say v, of $\partial^{\mathsf{T}} u$, and that (ii) $\mathsf{D}(t_j)$ contains more than half of the horizontal cells of $\partial^{\mathsf{T}} u$ which would belong to $\mathsf{C}(t_j)$ if $v \in \partial^{\mathsf{T}} u$ were absent from $\mathsf{D}(t_j)$. Given $\mathsf{D}(t_j)$, we collect all pairs (u, v) satisfying (i),(ii) to form $\mathsf{P}(j)$. For some $(u, v) \in \mathsf{P}(j)$, we set $\mathsf{D}(t_j + \frac{1}{2}) = (\mathsf{D}(t_j) - v) \cup (\partial^{\mathsf{T}} u - v)$. We will show that $\mathsf{P}(j)$ is nonempty shortly using Proposition 3.2, but let us see first why this is a rescue.

It is important to note that any error (no error or X) on v can be expressed by errors on $\partial^{\mathsf{T}} u - v$ up to X-stabilizers (coboundaries of 0-cells). So, for any correction on $\mathsf{D}(t_j)$ there is an equivalent correction on $\mathsf{D}(t_j + \frac{1}{2})$. Seeking a correction on $\mathsf{D}(t_j + \frac{1}{2})$, rather than on $\mathsf{D}(t_j)$, potentially increases overall weight of the correction, but the increment is not big as we show: $\mathsf{D}(t_j + \frac{1}{2})$ lacks v, so the

[11] One may consider an equivalent setting in which the erased qubits are simply lost with no errors on other qubits; in this case, one may initialize those qubits arbitrarily and then measure stabilizers.

condition (ii) for $\mathsf{P}(j)$ implies that $\mathsf{C}(t_j + \frac{1}{2})$ has more than half of the horizontal cells of $\partial^{\mathsf{T}} u$, i.e., $|\mathsf{C}(t_j + \frac{1}{2})| > \frac{1}{2}(|\partial^{\mathsf{T}} u| - 2)$. Since $\mathsf{D}(t_j)$ contains more than half of the horizontal cells of $\partial^{\mathsf{T}} u$ and also v, we see $|\mathsf{D}(t_j + \frac{1}{2})| < |\mathsf{D}(t_j)| + \frac{1}{2}(|\partial^{\mathsf{T}} u| - 2)$. Therefore, removing $\mathsf{C}(t_j + \frac{1}{2})$ from $\mathsf{D}(t_j + \frac{1}{2})$, we obtain $\mathsf{D}(t_j + 1)$ whose cardinality is strictly less than $|\mathsf{D}(t_j)|$.

Hence, if we employ $\mathsf{P}(j)$ whenever $\mathsf{C}(t_j) = \emptyset$, we can always decrease $|\mathsf{D}(t)|$ until $\mathsf{D}(t)$ becomes empty. The final correction x' has weight bounded by $|\mathsf{D}|(1+0.6\Delta)$ because each transition $\mathsf{D}(t_j) \to \mathsf{D}(t_j + \frac{1}{2})$ can enlarge the support of x' by the number of added cells which is less than $1 + \frac{1}{2}1.01\Delta$. It follows that $x+x'$ has weight less than $|\mathsf{D}|\Delta$ which is less than d_X, implying that $x+x'$ represents the zero cohomology class.

It remains to show that $\mathsf{P}(j) \neq \emptyset$ whenever $\mathsf{D}(t_j) \neq \emptyset$ but $\mathsf{C}(t_j) = \emptyset$. Consider the *shadow* of the vertical cells of $\mathsf{D}(t_j)$ onto the base. Here by shadow we mean as before the set of all base 0-cells above which there is at least one vertical cell in $\mathsf{D}(t_j)$. Since the cardinality of the shadow is less than $m/(10^5\Delta)$ (which follows by induction in j with the assumption that $\Delta|\mathsf{D}(0)| < m/(10^5\Delta)$), Proposition 3.2 implies that there is a base 0-cell a such that more than half of its coboundary 1-cells are counique neighbors. Let us go to the fibers over a and its counique neighbors $b_1, \ldots, b_q$. Any vertical cell $v = a \otimes f^1 \in \mathsf{D}(t_j)$ above a must share every 2-cell $b \otimes f^1$ in its coboundary with some other 1-cells of $\mathsf{D}(t_j)$. If b here is a counique neighbor of a, then the 2-cell $b \otimes f^1$ cannot be shared with any other vertical cell of $\mathsf{D}(t_j)$, so it must be shared with a horizontal cell of $\mathsf{D}(t_j)$. Conversely, if any horizontal cell over a counique neighbor b of a shares a 2-cell with a vertical cell in their coboundary, it must do with a vertical cell over a. Hence, within $\mathsf{D}(t_j)$ all the vertical cells over a and all the horizontal cells over the counique neighbors of a, together form a graph $\mathfrak{P}$ whose every node is linked to some other node by sharing a 2-cell.

Consider the *spine* of $\mathfrak{P}$; the spine is the union of the images of all nodes of $\mathfrak{P}$ under the map $a \otimes f^1 \mapsto a \otimes f^1$ and $b \otimes \varphi(b,a)^{-1} f^0 \mapsto a \otimes f^0$ for any counique neighbor $b \in \partial^{\mathsf{T}} a$. The spine has (far) less than $n_{\mathcal{F}}$ elements, and hence divided into consecutive clusters. Any cluster cannot have a 0-cell (that comes from a horizontal cell of $\mathfrak{P}$) at either end; if it did, there would be a horizontal cell that is exposed to a bundle 2-cell that only sees this horizontal cell. Let $v = a \otimes f^1$ be the 1-cell (that comes from a vertical cell of $\mathfrak{P}$) at the bottom end of a cluster. Any horizontal cell linked to v in $\mathfrak{P}$ must be $b \otimes f^0_u$ for some counique neighbor $b \in \partial^{\mathsf{T}} a$ where f^0_u is the upper end of f^1 because v is at the bottom of a cluster in the spine. Therefore, $\mathsf{D}(t_j)$ contains the horizontal cells $b_i \otimes \varphi(b_i,a)^{-1} f^0_u$ but not $b_i \otimes \varphi(b_i,a)^{-1} f^0_d$ for all $i = 1, \ldots, q$. If we deleted v from $\mathsf{D}(t_j)$, then all $b_i \otimes \varphi(b_i,a)^{-1} f^0_u$ would belong to $\mathsf{C}(t_j)$. Now, a bundle 0-cell $u = a \otimes f^0$ has coboundary that contains $v = a \otimes f^1$ and all $b_i \otimes \varphi(b_i,a)^{-1} f^0_u$ for $i = 1, \ldots, q$. Since q is more than half of the degree of a, we see that $(u, v) \in \mathsf{P}(j)$.

We now analyze the time complexity. One should not compute C and P every time D is updated. Instead, they should be initially computed once by going over all cells of D and small neighborhoods, and every time a cell is removed or added to D, the sets C and P should be updated accordingly. For each cell of D it takes time $O(\Delta)$ to determine its membership to C and P. When removing a cell of C from D, there are $O(\Delta^2)$ cells to examine, so it takes time $\tilde{O}(\Delta^3)$ to remove a cell and update the sets. In the transition $\mathsf{D}(t) \to \mathsf{D}(t + \frac{1}{2})$, we alter $O(\Delta)$ cells so it takes $\tilde{O}(\Delta^4)$ to complete this transition. The total number of removals is bounded by $|\mathsf{D}(0)|\Delta$, yielding overall time complexity $\tilde{O}(|\mathsf{D}(0)|\Delta^5)$. □

Acknowledgments

MBH thanks Mike Freedman for explaining spectral sequences; they weren't needed since the homology was computed in a more elementary way, but they helped in understanding the homology

of fiber bundles. RO thanks Venkat Guruswami for coding theory discussions, and thanks Microsoft Quantum for hosting him throughout the time this work was completed.

References

[AC88] Noga Alon and Fan R. K. Chung. Explicit construction of linear sized tolerant networks. *Discrete Math.*, 72(1-3):15–19, 1988. 4.3

[AR94] Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Structures Algorithms*, 5(2):271–284, 1994. 4.1

[Bas81] Leonid Bassalygo. Asymptotically optimal switching circuits. *Problems of Information Transmission*, 17(3):206–211, 1981. 3.1

[BH14] Sergey Bravyi and Matthew B. Hastings. Homological product codes. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 273–282. ACM, New York, 2014. 1, 1.1, 2.2

[Chu79] Fan R. K. Chung. On concentrators, superconcentrators, generalizers, and nonblocking networks. *Bell System Tech. J.*, 58(8):1765–1777, 1979. 3.1

[EKZ20] Shai Evra, Tali Kaufman, and Gilles Zémor. Decodable quantum LDPC codes beyond the $\sqrt{n}$ distance barrier using high dimensional expanders. Technical Report 2004.07935, arXiv, 2020. 1, 1.1, 5

[FH14] Michael H. Freedman and Matthew B. Hastings. Quantum systems on non-k-hyperfinite complexes: a generalization of classical statistical mechanics on expander graphs. *Quantum Inf. Comput.*, 14(1-2):144–180, 2014. 1

[FM01] Michael H. Freedman and David A. Meyer. Projective plane and planar quantum codes. *Found. Comput. Math.*, 1(3):325–332, 2001. 1.1

[FML02] Michael H. Freedman, David A. Meyer, and Feng Luo. Z_2-systolic freedom and quantum codes. In *Mathematics of quantum computation*, Comput. Math. Ser., pages 287–320. Chapman & Hall/CRC, Boca Raton, FL, 2002. 1

[Has17a] Matthew B. Hastings. Quantum codes from high-dimensional manifolds. In *Proceedings of the 8th Annual Innovations in Theoretical Computer Science*, volume 67 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 25, 26. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2017. 1.1

[Has17b] Matthew B. Hastings. Weight reduction for quantum codes. *Quantum Inf. Comput.*, 17(15-16):1307–1334, 2017. 1, 1.1

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561, 2006. 5.1

[Kit03] Alexei Yu. Kitaev. Fault-tolerant quantum computation by anyons. *Ann. Physics*, 303(1):2–30, 2003. 1, 1.1

[LTZ15] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum expander codes. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, pages 810–824. IEEE Computer Soc., Los Alamitos, CA, 2015. 1

[MR99] Sanjeev Mahajan and H. Ramesh. Derandomizing approximation algorithms based on semidefinite programming. *SIAM J. Comput.*, 28(5):16411663, 1999. 5.1

[RU08] Tom Richardson and Rüdiger Urbanke. *Modern Coding Theory*. Cambridge University Press, 2008. 1

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1710–1722, 1996. 5.1

[TZ14] Jean-Pierre Tillich and Gilles Zémor. Quantum LDPC codes with positive rate and minimum distance proportional to the square root of the blocklength. *IEEE Trans. Inform. Theory*, 60(2):1193–1202, 2014. 1

[Vad12] Salil P. Vadhan. *Pseudorandomness*. Now, 2012. 4.3

A Notation

Here we present a brief list of some notation used in the paper. The notation is not in alphabetical order but rather is in a rough order of when the concepts are introduced.

$\mathcal{B}$: the base of the bundle

$\mathcal{F}$: the fiber of the bundle

$\mathcal{E}$: the fiber bundle. Qubits of a quantum code are associated with 1-cells of this bundle while checks of the code are associated with 0- and 2-cells.

$n_{\mathcal{B}}$: the number of 1-cells in the base.

$m_{\mathcal{B}}$: the number of 0-cells in the base.

$n_{\mathcal{F}} = m_{\mathcal{F}}$: the number of 0-cells in the fiber which is the same as the number of 1-cells in the fiber.

B: a bipartite graph defining the base. The right vertices represent bits (variables) of a code which correspond to 1-cells of the base, and the left vertices represent checks of a code which correspond to 0-ells of the base.

F: a cycle graph.

$[n]$: shorthand notation for the set of integers $\{1, 2, \ldots, n\}$.

ℓ: an integer chosen so that $n_{\mathcal{F}} = m_{\mathcal{F}} = \ell^2$. All twists are integer multiples of ℓ.

10^5: a large universal constant related to bounding expansion in the base graph; see Proposition 3.2.

U: the set of used twists: $U = [m_{\mathcal{F}}/\ell] = [\ell]$.

n, m: shorthand notation used sometimes for $n_{\mathcal{B}}, m_{\mathcal{B}}$.

Δ: the average check-degree of the base code. Ultimately, Δ is chosen to be $\Theta(\log^2 n)$.

k: the number of distinct twists. Ultimately, k is chosen to be $\Theta(\log n)$.