Quantum LDPC Codes:
An exposition of recent results

Tushant Mittal

Abstract

LDPC CSS codes is a class of quantum error correcting codes, and a long standing open problem was to construct such codes with distance as close to the number of qubits, N, as possible.

Until recently, the best construction had distance slightly better than $\sqrt{N}$ but a recent breakhthrough one by Pantaleev and Kalachev'21 achieves linear distance (and rate).

This result can be seen as the culmination of a sequence of constructions based on tensor product and its modifications. In 2014, Tillich and Zémor defined the first tensor product based construction to achieve constant rate codes with distance $\sqrt{N}$. This was improved (by polylog(N) factors) upon by Evra, Kaufman and Zémor '20 and shortly later by Kaufman and Tessler'20 by using high-dimensional expanders(HDX).

Hastings, Haah and O'Donnell introduced the idea of a "quotiented tensor product" which utilizes the symmetry of the constituent complexes to quotient the tensor product which leads to a boost in parameters. They constructed a code using $\mathbb{Z}_l$ symmetry to achieved a distance of $N^{3/5}/\text{polylog}(N)$. Panteleev and Kalachev'20 also used $\mathbb{Z}_l$-symmetric complexes but managed to achieve an almost linear distance of $N/\log N$.

Breuckmann and Eberhardt'20 defined the quotiented tensor product for any group G and conjectured that G-quotiented tensor product of expander codes based on a certain Cayley graph $Cay(G, S)$ (LPS graph) should yield codes of linear distance and rate. Not much later, Panteleev and Kalachev'21 proved the conjecture resolving the problem at least from the perspective of distance and rate.

In this work, we provide a unified exposition of these results focusing primarily on the property of distance and decodability.

Contents

1 Preliminaries **6**
- 1.1 Notation table . . . 6
- 1.2 Linear Codes . . . 6
- 1.3 Tanner Codes . . . 7
- 1.4 Chain Complexes . . . 7
- 1.5 CSS Codes . . . 8

2 A Plethora of Products **9**
- 2.1 Tensor Product . . . 9
  - 2.1.1 Hypergraph Product . . . 10
  - 2.1.2 HDX Codes . . . 10
- 2.2 Symmetric tensor codes . . . 11
  - 2.2.1 Group Action and Quotients . . . 12
  - 2.2.2 Balanced Product . . . 14
  - 2.2.3 Twisted Product . . . 15
  - 2.2.4 Lifted Product . . . 16
  - 2.2.5 Summary of the products . . . 18
  - 2.2.6 An alternate perspective . . . 18

3 Structure **20**

4 Distance **22**
- 4.1 Tensor Product lower bound . . . 22
- 4.2 Upper bound . . . 23
- 4.3 The [HHO21] distance bound . . . 24
- 4.4 The [PK21] distance bound . . . 27

5 Decoding **29**
- 5.1 Classical Decoding . . . 29
- 5.2 Quantum Decoding . . . 30
- 5.3 Tensor Reduction . . . 31
- 5.4 Twisted Product . . . 34
  - 5.4.1 The algorithm . . . 34
  - 5.4.2 The proof . . . 36

Introduction

Information is prone to corruption which necessitates the construction of error-correcting codes that can enable computation and communication in the presence of errors. Since the era of Shannon, classical coding theory has made great strides and we have various constructions of error-correcting codes that are widely used in computing devices. On the other hand, we are far from building large-scale quantum computers and one of the (many) barriers to doing so is the lack of quantum error-correction that can enable fault-tolerant computation. Moreover, such codes also have theoretical applications in complexity theory, cryptography, pseudorandomness etc...(see [Tre04] for a survey). In a similar vein, one could expect connections of quantum error-correcting codes to quantum complexity.

A priori codes are merely a set of strings but adding structure can be very useful. Classically, one of the most well-studied structured families is that of *linear codes*, i.e., codes that are linear subspaces of $\mathbb{F}_q^n$. If the subspace has dimension k, then it defines a $[n, k, d]$-code where its *distance*, d, is the Hamming weight of the smallest non-zero vector in it. Varshamov [Var57] extended the existential bound of Gilbert [Gil52] to linear codes to show that over $\mathbb{F}_q$, $[n, k, d]$- linear codes exist with *rate* $k/n < 1 - H_q(d/n)$. Gallager in his PhD thesis [Gal60] defined the notion of a *low density parity check* (LDPC) code as one which has a parity check matrix with constant row and column sparsity. He showed that [Gal62] random binary LDPC codes also attain the GV bound with high probability[1].

Generalizing the GV bound, Calderbank and Shor [CS96], and Steane [Ste96], defined CSS codes and proved that such codes exist with $k/n < 1 - 2H_q(d/n)$. Bravyi, Terhal and Leemhuis [BTL10] proved that any $[[n, k, d]]$ stabilizer code can be mapped to a $[[4n, 2k, 2d]]$ CSS code which preserves sparsity upto a constant factor. Thus, if we are interested in asymptotically optimal constructions, there is no loss of generality in considering CSS codes.

A CSS code is defined by a pair of linear codes C_X, C_Z such that $C_Z^\perp \subseteq C_X$. Let H_x, H_z be parity check matrices[2] of C_X, C_Z. A family of CSS codes is LDPC if both H_x, H_z have constant sparsity. The X-distance, d_x, is defined as the minimum weight vector in $C_X \backslash C_Z^\perp$. Similarly, one defines d_z. The resulting CSS code is then a $[[n, k, d]]$-code where $d = \min(d_x, d_z)$, and $k = \dim(C_X) - \dim(C_Z^\perp) = \dim(C_Z) - \dim(C_X^\perp)$. A code is *good* if $d, k = \Theta(n)$. Clearly, the quantum GV bound shows that good CSS codes exist and the question now is whether Gallager's result holds in the quantum setting , that is,

Does there exist an infinite family of good LDPC CSS codes?

[1] For a modern proof that works over any $\mathbb{F}_q$, see [MRR+20].

[2] This means that $\ker(H_x) = C_X$ and $\ker(H_z) = C_Z$

The best construction to date is the one by Bravyi and Hastings [BH14] who showed that the tensor product (they termed it *homological product*) of two random linear codes yields good CSS codes with weight being $\Theta(\sqrt{n})$. As a first step, we ignore the rate and ask if we can build codes with linear distance. This is still open but there has been significant progress on this recently with constructions pushing the distance from $\sqrt{n}$ all the way to $\frac{n}{\log n}$. The focus of this work is to survey and give an exposition of these works.

Given the close relation to linear codes, it is only natural that insights from constructing those can be helpful to building CSS codes that achieve linear distance. This perspective is what separates these product constructions from earlier ones like surface codes, for example–Kitaev's toric code, which primarily use algebraic topological methods. Specifically, we see the use of two key properties (i) expansion (in [TZ14, EKZ20, KT21]) and (ii) symmetry (in [HHO21, PK21, BE21a]).

The idea of constructing codes using graphs was proposed by Gallager in 1963 and generalized by Tanner [Tan81]. Sipser and Spielman [SS96] combined this construction with that of expander graphs to get the first explicit family of "good" binary LDPC codes. This led to the insight that constructing and analyzing codes using expansion-like properties of the *factor graph* which is a 1-complex, $C_X = C_1 \xrightarrow{H_x} C_0$ can be very useful. Its generalization for a CSS code is a 2-complex $\mathcal{C} = C_2 \xrightarrow{H_z^T} C_1 \xrightarrow{H_x} C_0$ which is a chain complex by the orthogonality condition, i.e., $H_z^T H_x = 0$.

Tillich and Zémor [TZ14] defined the *hypergraph product* of two expander-based codes as in [SS96] yielding a constant rate CSS code with distance $O(\sqrt{n})$. When viewed as 1-complexes this is the same as the usual tensor product of complexes which can be performed on larger complexes. This was done by Evra, Kaufman and Zémor [EKZ20] by replacing one of the expanders with an expanding simplicial 2-complex, i.e., *high-dimensional expander* (HDX) yielding a distance of $O\left(\sqrt{n \log n}\right)$. Kaufman and Tessler [KT21] showed that one could take iterated tensor products if the base complexes had some nice properties. Using the Ramanujan complexes as in [EKZ20], they could obtain codes with distance $O(\sqrt{n \log^k n})$ for any k. However, it can be shown that the tensor product of two codes of sub-linear distance cannot yield a code of linear distance[3]. This restriction can be overcome by utilizing symmetry.

Symmetry, that is, invariance under the action of a group, is a useful property of a code. Such symmetries help in proving properties about the code and give rise to natural operations like quotienting that can lead to quantitative improvements by reducing redundancy. One of the largest family of symmetric codes that are studied are cyclic or quasi-cyclic codes which have symmetries of $\mathbb{Z}_l$.

Hastings, Haah and O'Donnell [HHO21] introduced the *twisted product* and showed that for a suitable construction of a $\mathbb{Z}_l$-symmetric graph $\mathcal{B}$, the twisted product of $\mathcal{B}$ and the l-cycle graph yields a code with distance $\Theta(N^{3/5})$ upto polylogarithmic factors. Panteleev and Kalachev [PK21] gave a simplified construction of a graph $\mathcal{B}$ with an elegant proof that improved the bound significantly obtaining codes of distance $N/\log N$. The work of Breuckmann and Eberhardt [BE21a] abstracted out the constructions to define a generalized quotiented product for complexes symmetric under the action of any group.

[3] See Lemma 4.2.1 for a formal statement.

Moving beyond distance, we need decodable algorithms to be able to perform error-correction. The notion of decoding of CSS codes is similar to decoding the classical codes C_X, C_Z upto *(co)boundaries*. A decoder for the hypergraph product [TZ14] was given by Leverrier, Tillich and, Zémor [LTZ15], based on a generalization of the iterative algorithm of Sipser and Spielman [SS96]. Evra, Kaufman and Zémor [EKZ20] give a decoding algorithm for their construction via a reduction to decoding the constituent codes. The construction of Hastings, Haah and O'Donnell [HHO21] has a Z-decoding which utilizes vertex expansion which only holds for the cocomplex. The constructions of [PK21, BE21a] do not have a decoding algorithm yet.

About the document In this exposition, we will focus on the product constructions and specifically on the results of [TZ14, EKZ20, HHO21, PK21, BE21a]. The goal of the document is to serve as a self-contained introduction to the recent progress made in constructions of quantum LDPC codes. For a broader perspective, the reader is referred to the excellent survey by Breuckmann and Eberhardt [BE21b] which goes beyond LDPC CSS codes and also discusses issues of practical relevance.

Chapter 1 gives basic definitions and sets up the notation. In Chapter 2, we define the various generalizations of tensor products and state the main distance results which we prove in Chapter 4. We prove a structural result on the (co)homology which is a generalization of the Künneth formula in Chapter 3 which immediately lets us compute the rate and is also very useful in proving distance and decoding.

We begin Chapter 5 by generalizing the decoder in [EKZ20] to give a black-box decoder for tensor product of complexes using their individual decoders. We then proceed to discuss the decoder in [HHO21]. In **??**, we sketch connections to two well-studied problems in theoretical computer science – (i) construction of explicit expanders (ii) construction of explicit CSP gap instances.

Chapter 1

Preliminaries

1.1 Notation table

We summarize the key definitions here for easy reference. These will be explained in detail in the rest of the chapter.

Notation	Definition
n-complex $\mathcal{C}$	A chain complex of $\mathbb{F}_2$-vector spaces of length n
$\mathcal{C}^*$	The co-complex(dual complex) of $\mathcal{C}$
$(\mathcal{C})_d$	$(\mathcal{C})_d = C_d \to C_{d-1} \to C_{d-2}$
$n_j(\mathcal{C})$	$\dim(C_j)$
$r_j(\mathcal{C})$	$\dim(H_j(\mathcal{C}))$
$d_j(\mathcal{C})$	$\min\{\lvert x\rvert \mid x \in \ker(\partial_j) \setminus \operatorname{im}(\partial_{j+1})\}$
$\lambda(G)$	$\lambda_2(A)$ where A is the adjacency matrix
$H : \mathbb{F}_2^m \to \mathbb{F}_2^n$ is(α, β)-expanding	If for every x such that $\lvert x\rvert \leqslant \alpha m$, $\lvert Hx\rvert \geqslant \beta\lvert x\rvert$
$\mathbb{F}_2[S]$, for a finite set S	$\{\sum_{s\in S} a_s s \mid a_s \in \mathbb{F}_2\} \cong \mathbb{F}_2^{\lvert S\rvert}$.
$T(G, C_0)$	Tanner code on graph G, code C_0

1.2 Linear Codes

A binary linear code of blocklength n is a vector subspace of $\mathbb{F}_2^n$. The elements of this space are called *codewords*. The *dimension* of the code is the dimension of the subspace and its *distance* is the smallest Hamming weight of a non-zero codeword. A code of length n, dimension k and distance d is written as an $[n, k, d]$ code. The *rate* of a code is the normalized dimension, i.e., k/n and is a measure of how efficient the code is. A code is specified either as as the kernel of a *parity check matrix* or as the rowspace of a *generator matrix*. Given a parity check matrix H for a code $C = \ker(H)$, we associate to it a bipartite graph called the *Tanner graph*. The vertices of this graph are the set of columns (called the bits) and the set of rows (the parity checks) of H. There is an edge between row i and column j if $H(i, j) = 1$. A more intuitive way to interpret the graph is that we have the bits of $x \in \mathbb{F}_2^n$ placed on the vertices representing the columns. A vertex representing row j computes the j^{th} parity check as the sum of all neighbouring bits. Thus, $x \in C$ if and only if every parity check

computes to zero. The dual code of C represented as $C^\perp = \{y \mid \langle x, y\rangle = 0 \ \ \forall x \in C\}$[1]. The generator matrix of $C^\perp$ is the parity check matrix of C. Thus, if $C = \ker(H),\ C^\perp = \mathrm{Im}(H^T)$. Here's an example to illustrate all these terms.

Example 1.2.1. Let $C = \ker \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$ It is a $[5, 2, 2]$ code as the equations are independent and the smallest weight codeword is $[0, 0, 1, 0, 1]^T$. It's Tanner graph is the following

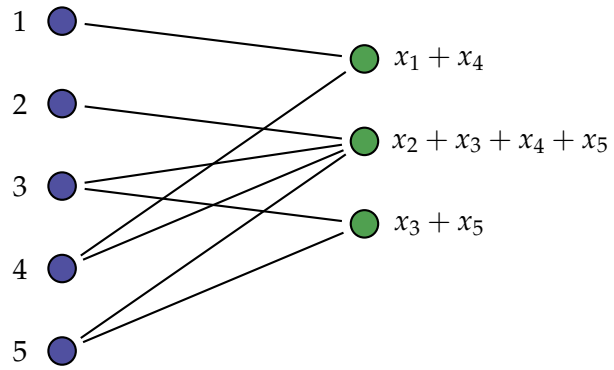

Figure 1.1: Tanner graphs of C

1.3 Tanner Codes

Let G be a d-regular graph on $2n$ vertices and let C be a code of length d. An *ordering* on G is a collection of bijective functions $\{\mathrm{ord}_v : [d] \to N(v) \mid v \in V\}$ where $N(v)$ denotes the set of neighbors of v. For $f : E(G) \to \{0, 1\}$, and a vertex v, let f_i^v denote $f(v, \mathrm{ord}_v(i))$. Then,

$$T(G, C) = \mathrm{span}(f \mid (f_1^v, \cdots, f_d^v) \in C \ \ \forall v)$$

This defines the code as a subspace of the dual space but by a standard isomorphism, we have $T(G, C) \subseteq \mathbb{F}_2[E(G)]^* \cong \mathbb{F}_2^{nd}$.

1.4 Chain Complexes

A chain complex $\mathcal{C}$ over $\mathbb{F}_2$ is a sequence $(C_i)_i$ of $\mathbb{F}_2$-vector spaces and *boundary maps* $\partial_i : C_i \to C_{i-1}$ such that any two consecutive maps are zero, i.e., $\partial_{i-1}\partial_i = 0$. An equivalent way to say this is $\mathrm{Im}(\partial_i) \subseteq \ker(\partial_{i-1})$. We will use the phrase n-complex to specify the length of the sequence. For a vector space V, we have its associated dual space, V^*, which is the space of all linear functions on V. For every, map $\phi : V \to W$, we have a map $\phi^* : W^* \to V^*$ which is given by $\phi^*(f)(v) = f(\phi(v))$ for $f \in W^*, v \in V$. Given a basis $\{e_i\}$, we have an associated basis of the dual $\{e_i^*\}$ where the function $e_i^*(e_j) = \delta_{ij}$ is the Kronecker delta function which is 1 when $i = j$ and 0 otherwise. Using this basis, it is easy to check that if the map ϕ is given by a matrix A, then ϕ^* is given by A^T. Thus, we

[1]Here $\langle y, x\rangle = \sum_i y_i x_i$ where x_i are the coefficients of x obtained by representing as a sum over the fixed basis. This is not an inner product as it is over $\mathbb{R}$ because $\langle x, x\rangle = 0$ if x has even Hamming weight.

can define the dual of a chain complex (called a *cocomplex*) as $\mathcal{C}^*$ which has the sequence of vector spaces $(C^*_{n-i})_i$ and the *coboundary maps* $\partial^*_i : C^*_i \to C^*_{i+1}$. We will usually identify $C_i \cong C^*_i$ using this identification[2] and not mention the dual very explicitly. The i^{th} *homology* of $\mathcal{C}$ is defined as $H_i(\mathcal{C}) = \ker(\partial_i)/\operatorname{im}(\partial_{i+1})$. Similarly, the j^{th} *cohomology* is defined as $H^j(\mathcal{C}) = \ker(\partial^*_j)/\operatorname{im}(\partial^*_{j-1})$. It is easy to see that $H^j(\mathcal{C}) = H_{n-j}(\mathcal{C}^*)$.

Since, distance is a notion that is sensitive to the choice of a basis, we will assume that each C_i comes equipped with a basis[3] and we will misuse notation to refer to both the basis and the space spanned by it as C_i. Let us look at the most common examples of chain complexes we will encounter.

Example 1.4.1 (Bipartite complex). Let H be the parity check matrix of a code C. Then, $\mathcal{C} = \mathbb{F}_2^n \xrightarrow{H} \mathbb{F}_2^m$ is a 1-chain complex such that $H_1(\mathcal{C}) = \ker(H) = C$. In general, for a bipartite graph it's adjacency matrix is of the form $\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}$ and we can build this complex by taking A to be the boundary map. This is what we will refer to when we mention the complex of a bipartite graph. This is well-defined upto duality, i.e., taking A or A^T.

Example 1.4.2 (Simplicial complex). Given a ground space of elements, say, $[n]$. A set of subsets $X \subseteq 2^{[n]}$ defines a simplicial complex if it is downward closed, i.e., if $x \in X$ then for any $y \subseteq x$, $y \in X$. Let $X_j = \{x \in X \mid |x| = j+1\}$, i.e., the set of all subsets of size $j+1$. We build a chain complex with $C_j := \mathbb{F}_2[X_j]$ and the maps $\partial_j(x) = \sum_{e \in x} (x \backslash \{e\})$. The, dual map is also very natural, $\partial^*_j(x) = \sum (x \cup \{e\})$ where the sum is over e such that $x \cup \{e\} \in X_{j+1}$. A graph $G = (V, E)$ can be seen naturally as a simplicial 1-complex where $X_0 = V, X_1 = E$. We will refer to the complex $E \to V$ where the map is given by the edge vertex incidence matrix, as the *graph complex* associated to G. [4]

1.5 CSS Codes

We will focus on a particular class of quantum codes called Calderbank-Shor-Steane (CSS) codes which were first defined by [CS96, Ste96]. These are a subset of a larger family of codes called *stabilizer codes*. CSS codes are a defined by a pair of codes C_X, C_Z such that $C_X \subseteq C_Z^\perp$ and $C_Z \subseteq C_X^\perp$. This is in bijection with 2-chain complexes over $\mathbb{F}_2$. Let $\mathcal{C} = C_2 \xrightarrow{H_z^T} C_1 \xrightarrow{H_x} C_0$ be a complex which defines a CSS code where H_x, H_z are the parity checks of C_X, C_Z. The parameters of the code are length - $n_1(\mathcal{C})$, dimension - $r_1(\mathcal{C})$ and distance $d = \min(d_X, d_Z)$ where $d_X = d_1(\mathcal{C}), d_Z = d_1(\mathcal{C}^*)$. We also require that H_x, H_z are low-density, i.e., the weight of their rows and columns are bounded by a constant. Analogously to linear codes, these are represented as a $[[n, k, d]]$ or $[[n, k, d_X, d_Z]]$ code.

[2] The choice of the dual basis is such that the isomorphism preserves Hamming distances.

[3] Such a complex is sometimes referred to as a *based chain complex* in the literature. A priori the basis is unordered but we will impose an order in some cases, especially when dealing with Tanner codes.

[4] There are multiple ways of creating a complex from a graph. The earlier example itself was one where we restricted to bipartite graphs. Another one could be via the adjacency matrix. However, these two examples are the only ones we will use.

Chapter 2

A Plethora of Products

The main object we are interested in constructing and analyzing is chain complexes of vector spaces over $\mathbb{F}_2$. A topological way of constructing these would be by applying a (co)homology theory to some topological space. This has been the traditional way of doing things especially from a physics perspective as the topology of the code matters during implementation. One of the earliest examples of a quantum CSS code is Kitaev's toric code which gives the complex obtained by cellulating the torus and applying cellular homology. Apart from using the usual simplicial/cellular homology, there have been constructions using more exotic homology theories like the Khovanov homology [Aud14]. Another approach, which has been more fruitful recently, is to start out with two or more simple chain complexes, typically graphs or simplicial complexes, and combine them using some notion of a product. While there have been a plethora of such products, all of them are essentially a tensor product with some modifications. In this chapter, we will try to define and study these various products in a unified manner.

2.1 Tensor Product

Tensor product is the most canonical product of a pair of chain complexes and forms the backbone of all the new product constructions.

Definition 2.1.1 (Tensor Product)**.** Let $\mathcal{C}, \mathcal{D}$ be chain complexes. The complex $\mathcal{C} \otimes \mathcal{D}$ is the complex such that $(\mathcal{C} \otimes \mathcal{D})_k = \bigoplus_{i=0}^{k} \mathcal{C}_i \otimes \mathcal{D}_{k-i}$ and $\partial_k(x \otimes y) = \partial_i^{\mathcal{C}}(x) \otimes y + (-1)^i x \otimes \partial_{k-i}^{\mathcal{D}}(y)$ if $x \in \mathcal{C}_i, y \in \mathcal{D}_{k-i}$.

Let $\mathcal{E} = \mathcal{C} \otimes \mathcal{D}$. Clearly, from the definition $n_j(\mathcal{E}) = \sum_{i=0}^{k} n_i(\mathcal{C}) n_{k-i}(\mathcal{D})$. Now, analysis of the (co)homology of the product is standard in algebraic topology and the main result here is the Kunneth's formula. We will prove a more general version of this later (Lemma 3.0.1).

Theorem 2.1.2 (Künneth)**.** $H_k(\mathcal{E}) \cong \oplus_i H_i(\mathcal{C}) \otimes H_{k-i}(\mathcal{D})$.

The distance properties are trickier to analyze and these are computed individually for each construction. However, we do have some general lower and upper bounds that let's us prove non-trivial results and equips us with general-purpose procedures.

Theorem 2.1.3. *[ZP19] Let $\mathcal{D}$ be a 1-complex and $\mathcal{C}$ be any complex and $\mathcal{E} = \mathcal{C} \otimes \mathcal{D}$. Then,*

$d_k(\mathcal{E}) \geqslant \min(d_k(\mathcal{C}), d_{k-1}(\mathcal{C})d_1(\mathcal{D}))$. *Moreover, if the map* $\partial_1^{\mathcal{D}}$ *is surjective, then* $d_k(\mathcal{E}) \geqslant d_{k-1}(\mathcal{C})d_1(\mathcal{D})$.

This theorem is very useful to prove distance bounds for quantum CSS codes formed by tensor products. Say the code is given by a short chain $E_2 \to E_1 \to E_0$, $d_X = d_1(\mathcal{E})$ and $d_Z = d_1(\mathcal{E}^*)$. We have seen earlier that $\mathcal{E}^* = (\mathcal{C} \otimes \mathcal{D})^* \cong \mathcal{C}^* \otimes \mathcal{D}^*$. Thus, all we need to do is to apply this theorem twice, once for $(\mathcal{C}, \mathcal{D})$ and once for $(\mathcal{C}^*, \mathcal{D}^*)$. We will use it in this section and prove a general version of it in Chapter 4.

2.1.1 Hypergraph Product

The specific case when $\mathcal{C}$ is also a 1-complex was proved in [TZ14]. Let C be the code given by $H_1(\mathcal{C}) = \ker(\partial_1)$. In other words, we have ∂_1 as the parity check matrix and the basis of C_0 are the parity check vertices. The term *hypergraph* refers to the one whose vertices are the basis of C_0, C_1 and the hyperedges are the collection of $\mathrm{supp}(\partial_1^*(v))$ where v is a parity check vertex.

The main result in this paper can be obtained by applying the one-dimensional version of this lower bound to a complex $\mathcal{E} = \mathcal{C} \otimes \mathcal{C}^*$.

Theorem 2.1.4. *[TZ14][Theorem 1] Let* $\mathcal{A}$ *be 1-complex such that* $\partial_1^{\mathcal{A}}$ *given by* $H \in \mathbb{F}_2^{n-c\times n}$ *is surjective. Let,* $\mathcal{E} = \mathcal{A} \otimes \mathcal{A}^*$. *Then* $d_1(\mathcal{E}) \geqslant d_1(\mathcal{A})$ *and* $d_1(\mathcal{E}^*) \geqslant d_1(\mathcal{A})$. *Moreover,* $n_1(\mathcal{E}) = n^2 + (n-c)^2$ *and* $r_1(\mathcal{E}) = c^2$.

Proof. The complex is self-dual so we only need to prove it for $\mathcal{E}$. As, $\mathcal{A}^* = A_0^* \to A_1^* \to 0$, we have that $H_0(\mathcal{A}^*) = A_1^* / \mathrm{im}(\partial_1^*)$. Since $c > 0$, $\mathrm{im}(\partial_1^*) \neq A_1^*$ and thus, there exists some basis vector of A_1^* that is not in the image. Thus, $d_0(\mathcal{A}^*) = 1$. Now we use Theorem 2.1.3 with $\mathcal{C} = \mathcal{A}^*, \mathcal{D} = \mathcal{A}$. Since the map ∂_1 is surjective, we get that,

$$d_1(\mathcal{E}) \geqslant d_0(\mathcal{A}^*)d_1(\mathcal{A}) = d_1(\mathcal{A}).$$

$$n_1(\mathcal{E}) = n_1(\mathcal{A}^*)n_0(\mathcal{A}) + n_0(\mathcal{A}^*)n_1(\mathcal{A}) = \dim(A_0)^2 + \dim(A_1)^2 = n^2 + (n-c)^2.$$

$$\begin{aligned} r_1(\mathcal{E}) &= r_1(\mathcal{A}^*)r_0(\mathcal{A}) + r_0(\mathcal{A}^*)r_1(\mathcal{A}) \\ &= r_0(\mathcal{A}^*)r_1(\mathcal{A}) && \text{(Surjectivity implies } r_0(\mathcal{A}) = 0) \\ &= (\dim(A_1^*) - \dim \mathrm{im}(\partial_1^*)) \dim \ker(\partial_1) \\ &= c^2. \qquad \blacksquare \end{aligned}$$

Taking $c = \Theta(n)$ yields a constant rate quantum LDPC code with distance growing as $O(\sqrt{n})$. These are achievable by using bipartite expanders of constant degree.

2.1.2 HDX Codes

A natural way to generalize the earlier idea is to use a high-dimensional expander (HDX) instead of an expander and this is the approach taken in a recent work of Evra, Kaufman, and Zémor [EKZ20] which achieves a family of codes with a distance of $\sqrt{N \log N}$.

High dimesional expanders are simplicial complexes with “expansion" properties and there are multiple notions of expansion that are not known to be equivalent (as for graphs).

We will not define what they are as the only property we will need is of bounded degree i.e. , the boundary maps ∂_i are all sparse.

The particular construction uses what is called the LSV *Ramanujan complex* which is the generalization of the much studied LSV Ramanujan graph. Let $\mathcal{X}$ be the d-dimensional Ramanujan complex and consider the clipped complex $\mathcal{X}_d = X_d \to X_{d-1} \to X_{d-2}$. It is shown that

Theorem 2.1.5. *[EKZ20, Thm 2.2] For d-dimensional LSV complexes $\mathcal{X}$, we have that $d_j(\mathcal{X}) = \Omega(\log n_j(\mathcal{X}))$ and $d_j(\mathcal{X}^*) = \Omega(n_j(\mathcal{X}^*))$ for $1 \leqslant j \leqslant d-1$. Moreover, $r_j(cX) > 0$ for $j = 1, 2$.*

For $r = 1, 2$, we get, $\mathcal{Q}_r = [[n, 1, O((\log n)^r), O(n)]]$. One issue is that the X-distance (and thus, the overall distance) is very small. This is remedied by a generic balancing procedure of tensoring with a good classical code.

Theorem 2.1.6. *[EKZ20, Thm 2.3] Let $\mathcal{X}$ be an 2-complex and let $\mathcal{C}$ be a 1-complex with a surjective boundary map. For $\mathcal{E} = \mathcal{X}_d \otimes \mathcal{C}$ we have, $d_2(\mathcal{E}) \geqslant d_1(\mathcal{X}) d_1(\mathcal{C})$ and $d_1(\mathcal{E}^*) \geqslant d_1(\mathcal{X}^*)$*

Proof. The claim for $\mathcal{E}$ follows directly from Theorem 2.1.3 for $(\mathcal{X}, \mathcal{C})$ for using surjectivity. For $\mathcal{E}^*$, we use the theorem for $(\mathcal{X}^*, \mathcal{C}^*)$ to get get $d_1(\mathcal{E}^*) \geqslant \min(d_1(\mathcal{C}^*), d_0(\mathcal{C}^*) d_1(\mathcal{X}^*))$. Since, $d_1(\mathcal{C}^*) = \infty$ by convention and $d_0(\mathcal{C}^*) = 1$, we get the claim. ∎

Let us rewrite this in coding theory terminology as this is a useful general procedure which is used in many constructions.

Corollary 2.1.7 (Distance balancing)**.** *Let $\mathcal{Q}$ be a $[n, K, d_x, d_z]$ code and $\mathcal{C}$ be a classical code defined by a surjective map with parameters $[m, k, d]$. Let $\mathcal{Q}' = (\mathcal{Q} \otimes \mathcal{C})_3$ [1]. Then, $\mathcal{Q}'$ has parameters $[O(nm), kK, dd_X, d_Z]$.*

To get the construction in [EKZ20], we tensor $\mathcal{Q}_r$ obtained from Theorem 2.1.6 with a classical expander code $\mathcal{C}$ which is a $[m, O(m), O(m)]$ code where $m = n/(\log n)^r$. Using the corollary, we get a quantum code with final parameters $[nm, O(m), O(n), O(n)]$ and thus, the distance is $O(\sqrt{N(\log N)^r})$ where $N = nm = n^2/(\log n)^r$ is the length of the code.

Another application of this theorem was used in [PK21] to boost the rate of the code at the cost of the distance. This is via the following lemma.

Corollary 2.1.8 (Dimension boosting)**.** *Let $\mathcal{Q}, \mathcal{C}$ be as before and let $\mathcal{Q}' = ((\mathcal{Q} \otimes \mathcal{C})_3^* \otimes \mathcal{C})_3$. Then, $\mathcal{Q}'$ has parameters $[O(n^2 N), k^2 K, dd_Z, dd_X]$.*

Proof. The dual simply switches the X, Z codes and thus flips d_X and d_Z. The rest is just applying the previous corollary twice. ∎

2.2 Symmetric tensor codes

The codes we have seen are applications of tensor product directly to expanders and HDXs. However, there have been a set of constructions recently that perform modifications of the tensor product. They improve upon the tensor product by avoiding a direct upper bound (see Lemma 4.2.1 and discussion thereafter). These seem different from the

[1] Recall that for a complex $\mathcal{X}$, $(\mathcal{X})_d = X_d \to X_{d-1} \to X_{d-2}$

definition, but as we will show, are all equivalent, at least in the case for which useful constructions exist. In particular, we will look at the *twisted homological product* from [HHO21], *lifted product* from [PK21] and *balanced product* from [BE21a]. We will show that each of these are equivalent to each other in some restricted setting. This equivalence is mentioned in [BE21a] but a complete proof is not presented. Here we will give a complete proof of the equivalence. Each of these products work with complexes that have a symmetry i.e., an action of a group on it.

2.2.1 Group Action and Quotients

Definition 2.2.1 (Group Representation)**.** A representation of a group H is a pair (ρ, V) where V is a vector space over some field $\mathbb{F}$ and ρ is group homomorphism $\rho : H \to GL(V)$ where $GL(V)$ is the mutiplicative group of automorphisms of V to itself. In other words if $V \cong \mathbb{F}^n$, ρ maps group elements to $n \times n$ invertible matrices(over $\mathbb{F}$).

There are two main representations that we will encounter.

- *Natural representation* - For $\mathrm{Sym}(l)$, $(\rho_{nat}, \mathbb{F}^l)$ is defined by $\rho_{nat}(\sigma)e_i = e_{\sigma \cdot i}$ where $\{e_1, \cdots, e_l\}$ is a basis of $V = \mathbb{F}^l$.
- *Regular representation* - For any finite group group H, let $R := \mathbb{F}_2[H]$. The set $\{g \mid g \in H\}$ forms a basis of R. The left regular representation is given by $\rho_{reg}(h)g = hg$, while the right variant is, $\rho_{reg}(h)g = gh^{-1}$.

Definition 2.2.2 (H-action on a complex)**.** Let $\mathcal{C}$ be a chain complex and H be a finite group. The action of H is given by a collection of homomorphisms $\{\rho_i\}$ where $\rho_i : H \to GL(C_i)$. For $x \in C_i$, we write[2] $h \cdot x = \rho_i(h)x$. The key condition is that the action must be compatible with the boundary map, i.e., $g \cdot (\partial a) = \partial(g \cdot a)$.

While the above definition works for any representation, we will impose the restriction that eachρ_i actually maps into the set of permutation matrices[3].This lets H permute the basis elements and prohibits actions like $h \cdot e_1 = e_2 + e_3$.

Example 2.2.3 (Action on a graph). In the case when $\mathcal{F}$ is a graph complex, this compatibility condition is equivalent to requiring the action to be a graph isomorphism. To see this let $e = (u, v)$ be an edge. For any $g \in H$, $g \cdot (\partial e) = g \cdot (u + v) = g \cdot u + g \cdot v = \partial(g \cdot e)$. Thus, $g \cdot e = (g \cdot u, g \cdot v)$

Definition 2.2.4 (Quotient Complex)**.** For a chain complex $\mathcal{C}$ with the action of H, define the chain complex $\mathcal{C}/H$ where $(\mathcal{C}/H)_i = C_i/H := C_i/\langle v - h \cdot v \mid v \in C_i, h \in H \rangle$.

In other words, we are shrinking orbits of every vector to a single point. Observe that for any $v \in C_i$, $\partial(v - h \cdot v) = \partial v - \partial(h \cdot v) = \partial v - h \cdot \partial(v) = 0 \in C_{i-1}/H$. Thus, the boundary maps are well-defined on the quotients because of the compatibility.

To do this, we focus on a nice class of actions called *free actions*.

Definition 2.2.5 (Free action)**.** An action is called *free* if for any i and any basis $v \in C_i$, $h \cdot v = v$, then $h = e$.

[2]Note that we overload notation to similarly represent the action irrespective of the space C_i.

[3]This forces $\rho_i = \rho_{nat} \circ \rho_i'$ where $\rho_i' : H \to \mathrm{Sym}(\dim(C_i))$.

We will first show that the action reduces to a collection of regular representations and we thus get the structure of V/H. We will then observe the structure of the quotiented boundary maps.

Lemma 2.2.6. *Let H act on a complex $\mathcal{C}$ such that the action on each C_i is free. Then for each i,*

i) $C_i \cong C_i/H \otimes \mathbb{F}_2[H]$.

ii) The action decomposes as $\rho_i(h) = I \otimes \rho_{reg}(h)$.

iii) The boundary maps $\partial_i : C_i \to C_{i-1}$ *are of the form* $\partial_i' \otimes \rho_{reg}(s_i)$ *where* $\partial_i' : C_i/H \to C_{i-1}/H$.

Proof. Let x be any element. Then $h{\cdot}x = g{\cdot}x$ implies that $(g^{-1}h){\cdot}x = x$. Since, the action is free, we get that $g = h$. Thus, the orbit of x i.e. $\{h{\cdot}x \mid h \in H\}$ has size $|H|$. This also means that dimension of C_i/H = $\dim C_i/|H|$. Let b be one of the basis vectors of C_i/H. Since, the orbit of b (when considered an element of V) has size $|H|$ we have distinct basis vectors $b \otimes h := h{\cdot}b$. Now, $g{\cdot}(b \otimes h) = g(h{\cdot}b) = b \otimes gh$. Thus, the action is identity on the first component and regular on the second. Moreover, if $\partial_i(v \otimes e) = \sum_w (w \otimes s_i(v,w))$, $s_i(v,w) \in H$ then for any $h \in H$,

$$\partial_i(v \otimes h) = \partial_i\left(h{\cdot}(v \otimes e)\right) = h\partial(v \otimes e) = \sum_w (w \otimes hs_i(v,w)).$$

Thus, if we order the basis of C_i in a lex order, then the matrix of ∂_i has $\rho_{reg}(s_i(v,w))$ in the block (v,w). Thus, we define $\partial_i'(v) = \sum_w g_w w$, and the claim follows. ∎

Definition 2.2.7 (Signed matrices)**.** Let H be a finite group and let $R = \mathbb{F}_2[H]$. We call any matrix over R[4] a signed matrix.

Definition 2.2.8 (Lifted matrix)**.** Let $R = \mathbb{F}_2[H]$ and let $X = \sum_{i,j} x_{ij}E_{ij}$ be a signed matrix with $x_{ij} \in R$. Given a representation (ρ, V) over $\mathbb{F}_2$ of H we can define the lifted matrix $\rho(X) = \sum_{i,j} E_{ij} \otimes \rho(x_{ij})$.

Using the property $\rho(gh) = \rho(g)\rho(h)$, one can show that $\rho(XY) = \rho(X)\rho(Y)$.

Constructing graphs with free actions - Lifting

Let $G = (V,E)$ be an undirected graph with an ordering on V such that by convention we have $(u,v) \in E$ if $u \leqslant v$. Let H be a subgroup of $\mathrm{Sym}(l)$.

Definition 2.2.9 ((H,l)-lift of a graph)**.** An (H,ℓ)-labelling of an undirected graph $G = (V,E)$ is a function $s : E \to H \subseteq \mathrm{Sym}(l)$. The lifted graph $G(s) = (V',E')$ is a graph on l copies of the vertices $V' = V \times [l]$ where for every edge $(u,v) \in E$ we have an edge between (u,i) and $(v, s(u,v){\cdot}i)$ in E'.

Note that if a graph is a (H,l)-lift of a smaller graph then it has a natural action of H on the vertices $h{\cdot}(u,i) = (u, h{\cdot}i)$. This extends to an action on edges when H is commutative. as $e' = ((u,i),(v,s(e)i))$ be an edge. Then h must act as

$$h{\cdot}e' = ((u,h{\cdot}i),(v,h{\cdot}s(e)i)) = ((u,h{\cdot}i),(v,s(e)(h{\cdot}i))) \in E'.$$

[4]We now view R as a ring by defining the multiplication operation using the group multiplication of H. R is called the *group algebra* of H

Example 2.2.10 (Graph lifting). Let A_G be the adjacency matrix of G. Given an (H, l)- labelling, we define its signed adjacency matrix which is obtained replacing 1 with $s(u, v)$ in the $(u, v)^{th}$ entry of the adjacency matrix.

$$A_G(s) = \sum_{(u,v)\in E(G)} s(u, v)E_{u,v}$$

where $E_{u,v}$ is a matrix with the only non-zero value being in $(u, v)^{th}$ entry with the value 1. It is easy to see that the adjacency matrix of the lifted graph, $G(s)$, is $A_{G(s)} = \rho_{nat}(A_G(s))$.

We now prove that a free action on a graph induces one on any Tanner construction. Let G be a d-regular graph. If the graph has a group H acting on it, we define the notion of H-ordering which is an ordering [5] such that for every v, h, i, we have $ord_{hv}(i) = h \cdot ord_v(i)$.

Lemma 2.2.11. *If we have a free action of H on L along with an H-ordering, then for any C_0, the graph $T(L, C_0)$ has a free H-action.*

Proof. $T(L, C_0)$ is a bipartite graph with vertices on the left being indexed by edges of L and on the other side by (v, q) where v is a vertex of L and q denotes one of the constraints of the code C_0.

The action of h is very natural as $h((v, q)) := (h(v), q)$ and the action of h on the left vertex marked by $e = (v, v')$ is same as its action on L i.e $h(e) = (h(v), h(v'))$. This is a valid vertex in $T(L, C_0)$ as action of h is a graph homomorphism (on L).

If the edge (v, v') participates in the q^{th} constraint then, it means that $ord_v(k) = v'$ for some k which lies in the support of the constraint. Since the ordering is an H-ordering, $ord_{hv}(k) = h(v')$ and thus, the edge $(hv, hv') = h(v, v')$ participates in the q^{th} constraint at $h(v)$. Thus, $(hv, q) \sim h(v, v')$ is a valid edge in $T(L, C_0)$. Hence, it is a graph homomorphism.

Since the action of h is free on L, it is so on the vertices of $T(L, C_0)$. Moreover, the homomorphsims respect the bipartiteness and therefore the action on edges is free as well (a switch cannot happen). ∎

2.2.2 Balanced Product

With all the group action terminology in place, we can quite easily define the balanced product introduced in [BE21a]. Let $\mathcal{C}, \mathcal{D}$ be two complexes with an action of H. Technically we require the action to be a left action on $\mathcal{C}$ and a right action on $\mathcal{D}$ but we will focus on the case when H is commutative and thus this won't matter. Define $C_i \otimes D_j/H := C_i \otimes D_j/\langle h\cdot v \otimes w - v \otimes h\cdot w\rangle$. This is a direct generalization of the previous definition of quotient as we can consider $C_i/H = C_i \otimes \mathbb{F}_2/H$ with the action of H on $\mathbb{F}_2$ being trivial. Now, we define $\mathcal{C} \otimes_H \mathcal{D}$ as just the usual tensor product but replace each $C_i \otimes D_j$ by $C_i \otimes D_j/H$. It is

[5] Recall that an ordering is a collection of bijective functions $\{ord_v : [d] \to N(v) \mid v \in V\}$

the same check as we did for C_i/H to see that the boundary maps are well-defined.

$$\begin{aligned}
\partial(h{\cdot}v \otimes w) &= \partial(hv) \otimes w + v \otimes \partial(w) \\
&= h\partial(v) \otimes w + hv \otimes \partial(w) && \text{By compatibility of action} \\
&= \partial(v) \otimes h{\cdot}w + v \otimes h\partial(w) && \text{By quotient on target space} \\
&= \partial(v) \otimes h{\cdot}w + v \otimes \partial(h{\cdot}w) && \text{By compatibility of action} \\
&= \partial(v \otimes h{\cdot}w)
\end{aligned}$$

2.2.3 Twisted Product

The twisted product defined in [HHO21] draws its inspiration from the notion of fiber bundles in differential geometry and hence the name. We can however define it directly at the level of chain complexes without worrying about the topology. To do this, we require a setup in which we have chain complexes $\mathcal{B}, \mathcal{F}$ and a group H that *acts* on $\mathcal{F}$.

The *twist map* φ is a function that maps the set $E = \{(b, a) \mid a \in \text{supp}(\partial b)\ b \in B_1, a \in B_0\}$ to H. If B is the bipartite graph corresponding to the 1-complex $\mathcal{B}$, then the set E is the set of edges of B. Note that this is the same as a labeling of the graph.

With this setup in place, we define the *twisted product* $\mathcal{B} \otimes_\varphi \mathcal{F}$ as follows. The spaces are the same as that in the usual tensor product but the boundary maps are twisted.

The map from $B_iF_1 \to B_iF_0$ is untwisted i.e. it is just $\text{id} \otimes \partial_\mathcal{F}$ and the map from $B_1F_i \to B_0F_i$ is twisted. Let f be a basis element of F_i. Then the map would be $\partial(b \otimes f) = \sum_{a \in \partial b} a \otimes (\varphi(b, a) \cdot f)$.

Putting this together for instance we have $\partial_2 : B_1F_1 \to B_1F_0 \oplus B_0F_1$, defined by $\partial_2(b \otimes f) = b \otimes \partial(f) + \sum_{a \in \partial b} a \otimes (\varphi(b, a) \cdot f)$. We can similarly write down ∂_1. Note that when the map φ is trivial, i.e. everything is mapped to the identity, then this map becomes the usual boundary map for the tensor product. In this sense, it generalizes the tensor product. However, the definition given above only works when $\mathcal{B}$ is a 1-complex.

Lemma 2.2.12. *[HHO21, Prop. 2.2] The above definition defines a complex i.e. $\partial_1\partial_2 = 0$*

Proof. Let b, f be basis elements of B_1, F_1.

$$\begin{aligned}
\partial_1\partial_2(b \otimes f) &= \partial_1 \left(b \otimes \partial(f)\right) + \partial_1 \left(\sum_{a \in \partial b} a \otimes (\varphi(b, a) \cdot f)\right) \\
&= \sum_{a \in \partial b} a \otimes (\varphi(b, a) \cdot \partial(f)) + \left(\sum_{a \in \partial b} a \otimes \partial\left(\varphi(b, a) \cdot f\right)\right) \\
&= \sum_{a \in \partial b} a \otimes (\varphi(b, a) \cdot \partial(f) + \partial\left(\varphi(b, a) \cdot f\right)) \\
&= \sum_{a \in \partial b} a \otimes 0
\end{aligned}$$

where the last step is due to the compatibility of the action of H on $\mathcal{F}$. ∎

Let $\mathcal{B}(\varphi)$ denote the lifted 1-complex using the twist map φ as its labelling. Explicitly, the lifted complex has the space $B(\varphi)_i = B_i \times H$ and the map $\partial_i(b, h) = \sum_{a \in \partial b}(a, \varphi(b, a)h)$.

Lemma 2.2.13. *The map $\eta : B_iF_j \to B(\varphi)_iF_j/H$ defined as $\eta(b \otimes f) = (b, e) \otimes f$ is an isomorphism of complexes. That is, $\mathcal{B} \otimes_\varphi \mathcal{F} \cong (\mathcal{B}(\varphi) \otimes \mathcal{F})/H$.*

Proof. On each space, we can define $\eta^{-1}((b, h) \otimes f) = b \otimes hf$. Clearly $\eta^{-1}\eta$ is the identity. Now, the basis of $B(\varphi)_iF_j$ before quotienting consists of $(b, h) \otimes f$ but we see that $(b, h) = h \cdot (b, e)$ and thus $(b, h) \otimes f = h(b, e) \otimes f = (b, e) \otimes hf$ where the last equality comes from the quotient. Thus, elements of the form $((b, e) \otimes f)$ can be taken as basis for $B(\varphi)_iF_j/H$ for which $\eta\eta^{-1}((b, e) \otimes f) = ((b, e) \otimes f)$.

To check commutativity, we need to check the following diagram,

$$\begin{array}{ccc} b \otimes f & \xrightarrow{\partial} & b \otimes \partial f + \sum_{a \in \partial b} a \otimes \varphi(b, a)f \\ \downarrow \eta & & \downarrow \eta \\ (b, e) \otimes f & \xrightarrow{\partial'} & (b, e) \otimes \partial f + \partial(b, e) \otimes f \end{array}$$

The first summand is equal by definition.

$$\begin{aligned} \partial(b, e) \otimes f &= \sum_{a \in \partial b} (a, \varphi(b, a)) \otimes f && \text{By definition of lift} \\ &= \sum_{a \in \partial b} (a, e) \otimes \varphi(b, a)f && \text{By identification in quotient} \\ &= \eta(\sum_{a \in \partial b} a \otimes \varphi(b, a)f) \end{aligned}$$

Thus, the squares commute.

■

2.2.4 Lifted Product

Let C, D be two graphs each with an edge-labeling from a subgroup H. Their signed adjacency matrices $\mathcal{C} = C_1 \to C_0$ represents a map $M : R^{n_1} \to R^{n_0}$ where $R = \mathbb{F}_2[H]$.

For a vector space $V \cong R^n$, we denote by $\rho_{reg}(V) := V \otimes_{\mathbb{F}_2} R \cong \mathbb{F}_2^{n|H|}$. To see this isomorphism, observe that $v \otimes r = (\sum_i p_i e_i) \otimes r = \sum_i e_i \otimes p_i r$. Expressing $p_i r \in R$ as a linear sum over H we get $\sum_i e_i \otimes p_i r = \sum_i e_i \otimes (\sum_h a_h^i h)$. Thus, $\{e_i \otimes h \mid i \in [n], h \in H\}$ forms a basis.

Now, we can define $\rho_{reg}(\mathcal{C})$ as the complex where each space C_i is replaced by $\rho_{reg}(C_i)$ and the boundary maps are $\rho_{reg}(\partial_i)$. The dimensions match and thus the boundary maps make sense. Also, as ρ_{reg} is a representation, $\rho_{reg}(\partial_{i-1})\rho_{reg}(\partial_i) = \rho_{reg}(\partial_{i-1}\partial_i) = \rho_{reg}(0) = 0$ so we do get a chain complex.

We define the lifted product as follows. First we form the tensor product of the corresponding chain complexes which are now spaces over R. The definition of the product is identical to the $\mathbb{F}_2$-case and it produces a chain complex as long as R is element-wise

commutative, i.e. H is commutative. We denote tensor products over any ring R as $\otimes_R$. The lifted product can then be stated succinctly as $LP(\mathcal{C},\mathcal{D}) := \mathbb{B}(\mathcal{C}/H \otimes_R \mathcal{D}/H)$.

Theorem 2.2.14. *Let $\mathcal{C},\mathcal{D}$ be 1-complexes with a free action of an abelian group H. Then,*

$$LP(\mathcal{C},\mathcal{D}) = \rho_{reg}(\mathcal{C}/H \otimes_{\mathbb{F}_2[H]} \mathcal{D}/H) \cong \mathcal{C}/H \otimes_\varphi \mathcal{D} \cong (\mathcal{C}\otimes\mathcal{D})/H.$$

Proof. We will show the first isomorphism, i.e., the one between lifted and twisted product. The second one was already shown in Lemma 2.2.13.

Now, by definition,

$$\begin{aligned}
\rho_{reg}(\mathcal{C}/H \otimes_{\mathbb{F}_2[H]} \mathcal{D}/H)_j &= (\mathcal{C}/H \otimes_{\mathbb{F}_2[H]} \mathcal{D}/H)_j \otimes \mathbb{F}_2[H] \\
&= (\mathcal{C}/H \otimes_{\mathbb{F}_2} \mathcal{D}/H)_j \otimes \mathbb{F}_2[H] && \text{By definition} \\
&= \mathcal{C}/H \otimes (\mathcal{D}/H)_j \otimes \mathbb{F}_2[H]) \\
&\cong \mathcal{C}/H \otimes D_j && \text{Using } \textit{Lemma } 2.2.6.
\end{aligned}$$

Thus, if $\nu_j : (D/H)_j \otimes \mathbb{F}_2[H] \to D_j$ is the isomorphism from Lemma 2.2.6 we have our isomorphism as $\eta_i : \oplus_{j\leqslant i} \mathrm{id}_{\mathcal{C}/H} \otimes \nu_j$.

To check commutativity, we restrict to $(C/H)_{i-j} \otimes D_j$ and check the following diagram,

$$\begin{array}{ccc}
c\otimes d\otimes e & \xrightarrow{\partial'_i} & c\otimes\rho_{reg}(\partial'_j d\otimes e) + \rho_{reg}(\partial'_{i-j}c\otimes d\otimes e) \\
\downarrow{\scriptstyle \eta_i} & & \downarrow{\scriptstyle \eta_{i-1}} \\
c\otimes\nu_i(d\otimes e) & \xrightarrow{\partial_i} & c\otimes\partial_j(\nu_i(d\otimes e)) + \partial_{i-j}c\otimes\nu_{i-1}(d\otimes e)
\end{array}$$

The first term is equal by Lemma 2.2.6 as,

$$\begin{aligned}
\eta_{i-1}(c\otimes\rho_{reg}(\partial' d\otimes e)) &= c\otimes\nu_{j-1}(\rho_{reg}(\partial'_j d\otimes e) && \text{Definition of } \eta_{i-1} \\
&= c\otimes\nu_{i-1}(\partial_j(d\otimes e)) && \text{By claim iii) of } \textit{Lemma } 2.2.6 \\
&= \partial\nu_i(d\otimes e) && \nu \text{ is a morphism of complexes}
\end{aligned}$$

For the second one,

$$\begin{aligned}
\eta_{i-1}\rho_{reg}(\partial'_{i-j}c\otimes d\otimes e) &= \eta_{i-1}\rho_{reg}\left(\sum_{a\in\partial c}\varphi(c,a)a\otimes d\otimes e\right) && \text{By definition of lift} \\
&= \eta_{i-1}\left(\sum_{a\in\partial c} a\otimes d\otimes\rho_{reg}(\varphi(c,a))e\right) && \text{As the tensor is over } \mathbb{F}_2[H] \\
&= \sum_{a\in\partial c} a\otimes\nu_i\left(d\otimes\varphi(c,a)\right) && \text{By definition} \\
&= \sum_{a\in\partial c} a\otimes\varphi(c,a)\nu_i(d\otimes e) && \text{Isomorphism and the action} \\
&= \partial_i\left(c\otimes\nu_i(d\otimes e)\right) && \text{Definition of twisted product}
\end{aligned}$$

Thus, the square commutes. ∎

2.2.5 Summary of the products

Product	Conditions	Definition
Lifted Product	Both are 1-complexes with free action of abelian H.	$\rho_{reg}(\mathcal{C}/H \otimes_{\mathbb{F}_2[H]} \mathcal{D}/H)$
Twisted Product	$\mathcal{C}$ is a 1-complex with a free action of H. $\mathcal{D}$ has any action of H.	$(\mathcal{C} \otimes \mathcal{D})/H$
Balanced Product	$\mathcal{C}, \mathcal{D}$ both have any action of H.	$(\mathcal{C} \otimes \mathcal{D})/H$

Construction of [HHO21]

Construct a set $|B_1|$ of size n and B_0 of size $m = 3n/4$. For every $a \in B_0$, pick we have an edge (b, a) for $b \in B_1$ with probability Δ/n where $\Delta = \Theta(\log^2 n)$. Divide B_0 into $\Theta(\log n)$ buckets of equal size and for bucket j, sample φ_j uniformly from $\{0, \cdots, \sqrt{l} - 1\}$ where $l = \Theta(n)$ is odd. Construct the twist map by having $\varphi(b, a) = 0$ with probability $1/2$ and $\varphi(b, a) = \varphi_j$ with probability $1/2$ if a belongs to j^{th} bucket. The final code is $\mathcal{B} \otimes_\varphi \mathcal{F}$ where $\mathcal{F} : \mathbb{Z}_l \to \mathbb{Z}_l$ is the graph complex of the cycle on l vertices.

Construction of [PK21]

Take a d-regular expander G on $2n$ vertices and pick a "good" code C_0[6]. Construct a random $\mathbb{Z}_l$-lift of G with $l = 2^n - 1$. Here, a random lift means that for each $e \in E(G)$, $s(e)$ is chosen uniformly at random from $\mathbb{Z}_l$. Since, $\mathbb{Z}_l$ is abelian, $G(s)$ has a free action of $\mathbb{Z}_l$ and by Lemma 2.2.11, so does $\mathcal{B} := T(G(s), C_0)$. Let $\mathcal{F}$ be the graph complex associated to the cycle graph on l-vertices. Then, the natural action of $\mathbb{Z}_l$ on $\mathcal{F}$ is free if l is odd. The final code is given by $LP(\mathcal{B}, \mathcal{F}) = \rho_{reg}(\mathcal{B}/H \otimes_{\mathbb{F}_2[\mathbb{Z}_l]} \mathcal{F}/H)$.

2.2.6 An alternate perspective

We can view all these three products also as lifts in a slightly different way without speaking much about group actions. This viewpoint is similar to defining the tensor product as a hypergraph product of the Tanner graph as in [TZ14].

Let $\mathcal{B}, \mathcal{F}$ both denote 1-complexes with labellings $s_\mathcal{B}$, $s_\mathcal{F}$ on the bipartite graph associated to the boundary maps. Let $\hat{\mathcal{B}}, \hat{\mathcal{F}}$ denote the lifted complex i.e. we lift the bipartite graph and consider the boundary map corresponding to that. The tensor product $\mathcal{B} \otimes \mathcal{F}$ can be represented as a layered graph with the first layer corresponding to $B_1 \times F_1$, the second layer corresponding to $B_1 \times F_0 \cup B_0 \times F_1$ and the final layer corresponding to $B_0 \times F_0$.

Lifted Product can be seen as taking a tensor $\mathcal{B} \otimes \mathcal{F}$ and then lifting it to $\mathcal{B} \hat{\otimes} \mathcal{F}$. Here, the the signing of the edges from $B_1F_0 \to B_0F_0$ and $B_1F_1 \to B_0F_1$ are the ones from the edges of $\mathcal{B}$ whereas those on the edges from $B_0F_1 \to B_0F_0$ and $B_1F_1 \to B_1F_0$ are the ones from the edges of $\mathcal{F}$. The twisted tensor product on the other hand is $\mathcal{B} \otimes \hat{\mathcal{F}}$ and the balanced product is $\mathcal{E}$ where $\hat{\mathcal{E}} = \hat{\mathcal{B}} \otimes \mathcal{F}$ where we have the equivalence $[(u, h), (v, g)] \sim [(u, e), (v, gh^{-1})]$

[6]The good property is that it is a code at the Gilbert-Varshamov (GV) bound and has constant fractional rate and distance.

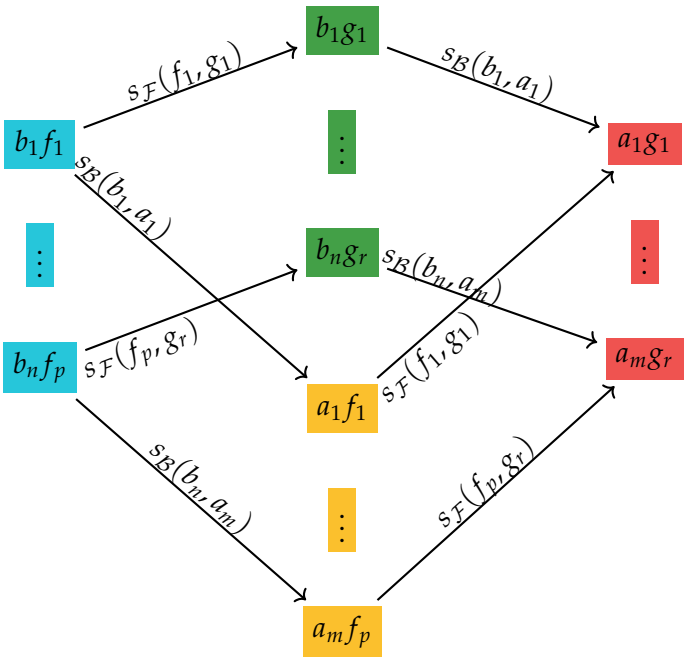

Figure 2.1: Tensor Product of the Tanner graphs

Chapter 3

Structure

Now that we have seen a bunch of constructions, it is time to analyze these codes in detail. The first step in understanding such a linear code will be to compute its dimension i.e. its rate. Since the constructions are algebraic and structured, we will be able to decompose the code i.e. the (co)homology in terms of its constituent parts. This will not only let use compute the dimension easily but will be of much help later when we wish to bound the distance and decode these codes. In particular, we will be proving a general version of Künneth formula which applies not just to the usual tensor product but also to the much more general balanced product.

Künneth formula for twisted product

We will first state this for the case of the twisted product. We treat this case separately for two reasons : one, that is good in the sense that the theorem is unchanged from the tensor case, and two, we will be using this later for decoding. An elementary proof of this is given in [HHO21, Prop. 2.1] while a cleaner and more general version is in [BE21a]. We present the latter with some added explanations.

Lemma 3.0.1. *[HHO21, BE21a] Let $\mathcal{E} = \mathcal{B} \otimes_\varphi \mathcal{F}$ and let $H_0(\mathcal{F}), H_1(\mathcal{F})$ have representatives such that φ acts by identity on these. Then, $H_1(\mathcal{E}) \cong H_1(\mathcal{B})H_0(\mathcal{F}) \oplus H_0(\mathcal{B})H_1(\mathcal{F})$*

Proof. The assumption on the twists means that we have $H_0(\mathcal{F}) = \text{span}\{[v_i]\}$ such that $\phi(b,a)v_i = v_i$ for every a, b, i. Similarly, $H_1(\mathcal{F}) = \text{span}\{[u_i]\}$.

Let $y \in H_1(\mathcal{E})$ and suppose $y = \sum_b b \otimes g_b + \sum_a a \otimes f_a \in E_1$. Express $g_b = \partial(f_b) + z_b$ where $z_b \in \text{span}\{v_i\}$. Then,

$$\begin{aligned} y' &= y + \partial\left(\sum_b b \otimes f_b\right) \\ &= \sum_b b \otimes (g_b + \partial f_b) + \sum_a a \otimes f_a + \sum_b \sum_{a \in \partial b} a \otimes \varphi(b,a) f_b \\ &= \sum_b b \otimes z_b + \sum_a a \otimes f'_a \end{aligned}$$

Now,

$$\begin{aligned}\partial(y) = \partial(y') &= \sum_b \partial\left(b \otimes z_b\right) + \sum_a a \otimes \partial(f'_a) \\ &= \sum_b \sum_{a \in \partial b} a \otimes \varphi(b,a) z_b + \sum_a a \otimes \partial(f'_a) \\ &= \sum_b \sum_{a \in \partial b} a \otimes z_b + \sum_a a \otimes \partial(f'_a) \\ &= \sum_b \partial b \otimes z_b + \sum_a a \otimes \partial(f'_a)\end{aligned}$$

Since, $y \in \ker(\partial_1)$ the final expression must be zero. However, $\partial(f'_a) \in \mathrm{im}(\partial_1)$ whereas z_b by definition is not in the image. Therefore, they can't cancel each other. So, for each a, $\partial(f'_a) = 0$ i.e. $f'_a \in H_1(\mathcal{F})$. If $a \in \mathrm{im}(\partial_1^{\mathcal{B}})$ i.e $a = \partial(w_b)$ then,

$$y + \partial(w_b \otimes f'_a) = y + w_b \otimes \partial(f'_a) + \partial(w_b) \otimes f'_a = y + a \otimes f'_a.$$

Here we used that $f'_a \in H_1(\mathcal{F})$ is unaffected by twists. Hence, we can erase such terms from y. and the a that are left belong to $H_0(\mathcal{B})$. Thus, the second term is in $H_0(\mathcal{B})H_1(\mathcal{F})$.

We rewrite the first term by writing each z_b in terms of v_i's. We get $\sum_i b_i \otimes v_i$ and applying the map we get that $\sum_i \partial(b_i) \otimes v_i = 0$ and thus $b_i \in H_1(\mathcal{B})$. Therefore, the first term lies in $H_1(\mathcal{B})H_0(\mathcal{F})$. ∎

While the notion of the homology is enough for us to compute rate, it is unhelpful in computing distance which (unlike rate) is not invariant under isomorphism. Thus, we write the following corollary for later use.

Corollary 3.0.2. *Fix any set of linearly independent representatives of $H_i(\mathcal{B}), H_i(\mathcal{F})$ for $i \in \{0,1\}$ such that the representatives for $H_i(\mathcal{F})$ are invariant under the set of twists φ. Then,* $\ker(\partial_1^{\mathcal{E}}) = \{x_1 \otimes y_0 + x_0 \otimes y_1 + \mathrm{im}(z)\}$ *where x_i (resp. y_i) range over the linear span of the representatives of $H_i(\mathcal{B})$ (resp. $H_i(\mathcal{F})$) and v is a non-trivial kernel vector iff when written in the above way $x_1 \otimes y_0 + x_0 \otimes y_1 \neq 0$*

Chapter 4

Distance

This section gets into the trickiest property to analyze for these constructions, i.e., distance. We will first show how to lower bound distance in terms of the distance of the constituent parts. We will prove the result as in [ZP19] but the proof will be much simpler as a result of the structure from Künneth theorem. We will then look at an upper bound for the distance which displays the limitation of the tensor product. This will also shed light on how twists can circumvent the upper bound which leads us to the results of [HHO21, PK21] which prove better bonds by leveraging twists and also imposing expansion like properties on these which are very much inspired from classical coding theory.

4.1 Tensor Product lower bound

Let $\mathcal{B}, \mathcal{F}$ be 1-complexes[1]and let $\mathcal{E} = \mathcal{B} \otimes \mathcal{F}$. Let $B_1' \subseteq B_1$ be a subspace such that B_1' is spanned by a subset of the basis of B_1. Define the complex $\mathcal{B}' : B_1' \to B_0$ where the boundary map is the restriction of $\partial_1(\mathcal{B})$. Similarly define $\mathcal{F}'$. Then, $\mathcal{E}' = \mathcal{B}' \otimes \mathcal{F}'$ is well-defined. In other words, we are selecting a subset of columns from the matrices of the maps $\partial_1^{\mathcal{B}}, \partial_1^{\mathcal{F}}$ and then forming the tensor product. It is direct to see that the complex is a subcomplex of $\mathcal{E}$ i.e the natural inclusion maps $E_i' \to E_i$ which commutes with the boundary maps.

Theorem 4.1.1. *[ZP19] Let $\mathcal{B}, \mathcal{F}$ be 1-complexes and let $\mathcal{E} = \mathcal{B} \otimes \mathcal{F}$. Then,*

$$d_1(\mathcal{E}) \geqslant \min(d_1(\mathcal{B})d_0(\mathcal{F}), d_0(\mathcal{B})d_1(\mathcal{F})).$$

Proof. Let v be a kernel vector such that $|v| < \min(d_1(\mathcal{B})d_0(\mathcal{F}), d_0(\mathcal{B})d_1(\mathcal{F}))$ and let $v = \sum_b b \otimes g_b + \sum_a a \otimes f_a$.

Assume now that neither of the zero homologies are trivial i.e. $d_0(\mathcal{B}) = d_0(\mathcal{F}) = 1$ [2]. Let $I = \{b \mid g_b \neq 0\}$, $J = \bigcup_a \text{supp}(f_a)$. then, by the weight assumption on v, we have that $|I| \leqslant d_1(\mathcal{B})$, $|J| \leqslant d_1(\mathcal{F})$. Let $B_1' = \text{span } I$, $F_1' = \text{span } J$. Thus, $H_1(\mathcal{B}') = H_1(\mathcal{F}') = 0$ which from Künneth's theorem implies that $H_1(\mathcal{E}') = 0$.

[1]We just do this for ease of notation but none of the arguments need $\mathcal{B}$ to be a one-complex. To analyze $d_p(\mathcal{E})$ we only need the 1- complex $\mathcal{B}' : B_p \to B_{p-1}$

[2]Note that either $H_0(\mathcal{B}) = 0$ and thus $d_0(\mathcal{B}) = \infty$ or there is some basis element not in the image and thus $d_0(\mathcal{B}) = 1$

We modify the construction if we have surjective maps, i.e. ($H_0 = 0$). Both cannot be zero else the tensor code is trivial . Since the argument is symmetric, let's assume $d_0(\mathcal{B}) = \infty$ i.e. $H_0(\mathcal{B}) = 0$. We take $F'_1 = F_1$ and take B'_1 as above and add more basis elements such that the restricted map becomes surjective again i.e. $H_0(\mathcal{B}') = 0$ without adding a kernel vector. This is possible as we are essentially taking a subset of columns and adding the linearly independent ones. This ensures that $H_1(\mathcal{E}') = H_1(\mathcal{B}')H_0(\mathcal{F}) \oplus H_0(\mathcal{B}')H_1(\mathcal{F}) = 0$.

By construction $v \in \mathcal{E}'$ and moreover is in $\ker(\partial_1^{\mathcal{E}'})$. But since $H_1(\mathcal{E}') = 0$, we have that $v \in \text{im}(\partial_2^{\mathcal{E}'}) \subseteq \text{im}(\partial_2^{\mathcal{E}})$. This shows that v is a boundary element. ∎

This proof does not generalize to twisted product because F'_1 need not be closed under the action of the automorphism group of F_1. If we do have that F'_1 is indeed closed, then the argument goes through. As discussed above, if $H_0(\mathcal{B}) = 0$, then the above observation gives the following corollary.

Corollary 4.1.2. *Let $\mathcal{B}, \mathcal{F}$ be 1-complexes and let $\mathcal{E} = \mathcal{B} \otimes_\varphi \mathcal{F}$. If $H_0(\mathcal{B}) = 0$, then, $d_1(\mathcal{E}) \geqslant d_1(\mathcal{B})$.*

This setting is important because, as we've seen, the construction in both [EKZ20] and [HHO21] have that $H_0(\mathcal{B}) = 0$. This can always be done by shrinking B_0 to the image i.e. $B'_0 := \text{im}(\partial_1)$. However, the bound above isn't sufficient unless $\dim(F_i)$ is tiny compared to $\mathcal{B}$ and even then, the construction is good only on side as the other distance would depend on $d_1(\mathcal{F}^*)$. [3]

To go beyond $\sqrt{n}$, we want something like $d_1(\mathcal{E}) \geqslant d_1(\mathcal{B})d_1(\mathcal{F})^c$ for some constant c . We will see below that the usual (untwisted) tensor product cannot achieve that and this is where the importance of twists comes in. While we do not have generic bounds yet, for the specific case of the cycle map, $\mathcal{F} : \mathbb{Z}_l \to \mathbb{Z}_l$, and clever choices of the twists, [HHO21] prove a bound of roughly $O(d_1(\mathcal{B})\sqrt{l})$ and [PK21] prove an optimal linear bound of $O(\min(d_1(\mathcal{B}), d_0(\mathcal{B})l)$.

4.2 Upper bound

Lemma 4.2.1. *[ZP19] Let $\mathcal{E} = \mathcal{C} \otimes \mathcal{D}$. Then, $d_k(\mathcal{E}) \leqslant \min_i d_i(\mathcal{C})d_{k-i}(\mathcal{D})$.*

Proof. The construction is very simple. Let j be the argmin of the above quantity and let $x \in C_j,\ y \in D_{k-j}$ be such that $|x| = d_j(\mathcal{C})\ \ |y| = d_{k-j}(\mathcal{D})$. Now, $x \otimes y \in H_j(\mathcal{C})H_{k-j}(\mathcal{D})$. Pick x, y to be representatives of $H_j(\mathcal{C}), H_{k-j}(\mathcal{D})$ and extend it to a complete set arbitrarily. By Corollary 3.0.2 of Künneth's theorem, $x \otimes y$ is a non-trivial kernel vector and thus clearly, $d_k(\mathcal{E}) \leqslant |x \otimes y| = d_j(\mathcal{C})d_{k-j}(\mathcal{D})$. ∎

Now we can see why this upper bound breaks down for the twisted products. Most of the argument still honds and so does the generalized Künneth but we can't choose a representative of $H_i(\mathcal{F})$ arbitrarily as they must be invariant under φ. In the untwisted case, all twists are identity and thus this condition is trivial but in general it is not. For example, when $\mathcal{F} = \mathbb{Z}_l \to \mathbb{Z}_l$ then the only vector invariant under a permutation is the all-ones eigenvector. Therefore, while $d_i(\mathcal{F}) = 1$, the only invariant representative is $\mathbb{1}_i$ which has a

[3] This is why in the hypergraph product construction [TZ14], one takes $\mathcal{F} = \mathcal{B}^*$ so that the code is symmetric and this assumption is true on both sides giving a $\sqrt{n}$-distance.

large weight i.e., l. Thus, the bound merely gives an upper bound of $l \cdot \min(d_1(\mathcal{B}), d_0(\mathcal{B}))$. The bound isn't weak as this is actually achieved (upto constant factors) by [PK21].

Let us analyze in a more detailed way where the upper bound discussed above comes from in the untwisted case and how that barrier can be crossed.

We start with $v \in H_1(\mathcal{B})$ and $g \in H_0(\mathcal{F})$. We can assume that $|g| = 1$ i.e. g is a basis vector. In the untwisted case, $v \otimes g \in H_1(\mathcal{E})$ and thus we have a non-trivial element of weight $d_1(\mathcal{B})$. This no longer holds in the twisted case.

$$\partial(v \otimes g) = \sum_{b \in v} \sum_{a \in \partial(b)} a \otimes \varphi(b,a) g \tag{4.1}$$

$$= \sum_{a} \sum_{b \in v \cap a \in \partial(b)} \varphi(b,a) g \tag{4.2}$$

$$= \sum_{a} \partial_1^{\mathcal{F}}(f_a) = \partial_1^{\mathcal{E}} \left(\sum_a a \otimes f_a \right) \tag{4.3}$$

The last line follows because v is a kernel vector and so for every a the vector $g_a := \sum_{b \in v \cap a \in \partial(b)} \varphi(b,a) g \in F_0$ has even weight. Thus, we have a preimage, $f_a \in F_1$. This says that $v \otimes g + \sum_a a \otimes f_a$ is in a non-trivial kernel vector where we can think of the second term as a "correction" term which is non-zero in the twisted case.

If every twist is identity, $g_a = \sum g = 0$ and we recover our earlier observation. Assume that every twist is small, say, $\varphi : E \to \{0, 1, \cdots, r\}$. Then $|f_a| \leqslant r$ and thus, $d_1(\mathcal{E}) \leqslant d_1(\mathcal{B}) + r n_0(\mathcal{B})$.

Proof strategy of PK and HHO Generalizing the example above, we can say that for any $h \in B_1 F_0$ such that $h + v \in \ker(\partial_1^{\mathcal{E}})$, there is a unique such low-weight v which depends solely on ∂h. Both the proofs have two main parts.

1. (Expansion) Both use expansion, albeit in different forms, to argue that if $|h|$ is too small, then, $|\partial h| = \Omega(|h|)$.
2. (Structure) HHO uses structural constraints on twists to show that $|v| = \tilde{\Omega}(\sqrt{l}|\partial h|)$ whereas PK uses the inherent symmetry of $\mathbb{Z}_l$ to argue that $|v| = \Omega(l|\partial h|)$.

4.3 The [HHO21] distance bound

We start our analysis by observing that the specific design choices of the twists in [HHO21] are quite natural once we look at the expression for g_a in Eq. (4.2). As analyzing the sum can be cumbersome, we make some intuitive simplifications -

- For each a, limit the number of $\{\varphi(b,a) \mid b \in \partial^T(a)\}$. The extreme case is that we just have one twist but in that case $g_a = 0$. The next best thing is to have just two twists. One being identity and the other being non-trivial.
- Observe that if every twist was a multiple of k, then the support of g_a has points spaced apart by a multiple of k and thus if $g_a \neq 0$, then $|f_a| \geqslant k$.

- These are exactly the two simplifications made, whereby in the second one, k is taken to be $\sqrt{l}$.
- [HHO21] actually goes a bit further and reduces the number of twists by partitioning B_0 into $k = \Theta(\log n)$ equal-sized buckets and having the same non-trivial twist for each a in that bucket.

Now that we have motivated the twist design in [HHO21], let's move beyond this one example to more general cases. Let $h = \sum_b b \otimes g_b \in B_1 F_0$ be the horizontal component of the smallest non-trivial kernel vector. The first thing we notice is that the second point above no longer holds. Indeed if we have some g_b with its support spaced apart by 1, then φg_b also has the same. This can be remedied if we can ensure that for every b we have $\text{supp}(g_b) \subseteq \{0, \sqrt{l}, 2\sqrt{l}, \cdots l - \sqrt{l}\}$. This seems like a serious restriction but one of the most interesting aspects of the proof is that this is without loss of generality. Such a g_b ensures that each g_a is also similarly supported and thus, $|f_a| \geqslant \frac{|g_a|}{2}\sqrt{l}$ as the shortest chain between any two points has length that is a multiple of $\sqrt{l}$. Therefore, $|v| \geqslant \frac{\sqrt{l}}{2}|\partial h|$ which establishes part (2).

Lemma 4.3.1. *[HHO21, Lemma 3.10] Consider all chains $h = \sum_b b \otimes g_b$ such that $|h|_{\mathcal{B}} \geqslant d_1(\mathcal{B})$ and $\text{supp}(g_b) \subseteq \{0, \sqrt{l}, 2\sqrt{l}, \cdots l - \sqrt{l}\}$. Then, $d_1(\mathcal{E}) \geqslant \min_h |h| + \frac{\sqrt{l}}{2}|\partial h|$*[4].

The condition on $|h|$ is because $h + v = z \otimes \mathbb{1}_0 + \partial(x)$ with $z \in H_1(\mathcal{B})$ from our structure of the kernel. Note that we use the fact that $H_0(\mathcal{B}) = 0$. Since, $\partial(x) = \sum_b b \otimes y_b$ with $|y_b|$ being even, it cannot cancel out the support of z and thus, $|h| \geqslant |z| \geqslant d_1(\mathcal{B})$.

This handles the part(2) of the proof and now we focus on the first one.

Lemma 4.3.2. *Let h be as above such $|h| < c\frac{n_1(\mathcal{B})\sqrt{l}}{\Delta}$, then $|\partial h| > 2\frac{c}{\Delta}|h|$.*

Theorem 4.3.3. *Let $\mathcal{E} = \mathcal{B} \otimes \varphi\mathcal{F}$ be the construction as in [HHO21]. Then,*

$$d_1(\mathcal{E}) \geqslant \Omega(d_1(\mathcal{B})\sqrt{l}, \;\; d_1(\mathcal{E}^*) \geqslant \Omega(d_1(\mathcal{B})).$$

Proof. Either $|h| > c\frac{n_1(\mathcal{B})\sqrt{l}}{\Delta}$ or from Lemma 4.3.2,

$$\frac{\sqrt{l}}{2}|\partial h| \geqslant \frac{c}{\Delta}\sqrt{l}|h| \geqslant \frac{c}{\Delta}\sqrt{l}d_1(\mathcal{B}).$$

In either case, we are done by Lemma 4.3.1. For the distance of the dual, we have surjectivity of the boundary map and thus use Corollary 4.1.2. ∎

In the rest of this subsection will sketch the proof of Lemma 4.3.2. We start by rewriting $h = \sum_b \otimes g_b = \sum_{i=1}^{l} w_i \otimes i$. This is just expressing h in the basis of F_0. The above lemma says that for every i not a multiple of $\sqrt{l}$, we can assume that $w_i = 0$. Thus, we relabel and write $h = \sum_{k=0}^{\sqrt{l}-1} w_{k\sqrt{l}} \otimes k\sqrt{l} =: \sum_{i=0}^{\sqrt{l}-1} w_i \otimes i$ and so, w_i actually denotes $w_{i\sqrt{l}}$ from now on.

The image, $\partial(h) \in B_0 F_0$, can be written as $\sum u_i \otimes i$. Since every twist is a multiple of $\sqrt{l}$, φg_b is supported on multiples of $\sqrt{l}$ and therefore we can similarly write $\partial h = \sum_{i=0}^{\sqrt{l}-1} u_i \otimes i$

[4]This is a slight variant of the version n the paper as they have $\min_h |h| + \sqrt{l}|\partial h|_{sw}$ where $|\partial h|_{sw}$ is the weight of the projection of h to B_0. Since, $|\partial h| \geqslant 2|\partial h|_{sw}$, the version we state is stronger.

where $u_i = \sum_a c_{i,a} a$. Let $\{\tau_1, \cdots, \tau_k\}$ represent the k buckets and let φ_j be the non-trivial map of j^{th} bucket. For every $a \in \tau_j$, $\varphi(b,a) \in \{\mathrm{id}, \varphi_j\}$. We can write the expression for $c_{i,a}$ very explicitly as:

$$c_{i,a} = \sum_{b \in \partial^T a} (g_b)_{\varphi^{-1}(b,a)i} = \sum_{\varphi(b,a)=\mathrm{id}} (g_b)_i + \sum_{\varphi(b,a)=\varphi_j} (g_b)_{\varphi_j^{-1} i}$$

Thus for each $a \in \tau_j$, $c_{i,a}$ only depends on the two vectors. $w_i, w_{\varphi_j^{-1} i}$. On the other hand, each w_i influences only $c_{i,a}$ or $c_{\varphi_j i,a}$. Let $f(i,j) := \sum_{a \in \tau_j} c_{i,a}$ and $g(i,j) := |w_i| + |w_{\varphi_j^{-1} i}|$. Then, the final goal is to show that

$$\begin{aligned}
|\partial h| &\geqslant \frac{2c}{\Delta}|h| \\
\sum_i \sum_a c_{i,a} &\geqslant \frac{2c}{\Delta}|w_i| \\
\sum_i \sum_{j \in [k]} \sum_{a \in \tau_j} c_{i,a} &\geqslant \frac{2c}{\Delta}\frac{1}{2k} \sum_{j \in [k]} \left(|w_i| + |w_{\varphi_j^{-1} i}| \right) \\
\sum_i \sum_{j \in [k]} f(i,j) &\geqslant \frac{c}{k\Delta} \sum_i \sum_{j \in [k]} g(i,j)
\end{aligned}$$

This relationship is very naturally modeled on a graph on $\sqrt{l}$ vertices indexed by i and we have an edge $(i, \varphi_j(i))$for every $j \in [k]$. This is called the *twist graph* in the paper and is equivalent to the Cayley graph on $\mathbb{Z}_{\sqrt{l}}$ with the set of twists $\{\varphi_j\}$ being the generators. Each term in the above inequality corresponds to an edge in the twist graph

Since, the proof uses the fact that this graph is an expander, we are forced to have at least $O(\log \sqrt{l}) = \theta(\log n)$ buckets and this explains yet another of their construction parameters.

We will term a vector $w \in B_1$ *light* if $|w| \leqslant \frac{2|B_1|}{\Delta}$, *medium* if $\frac{2|B_1|}{\Delta} \leqslant |w| \leqslant \frac{4|B_1|}{\Delta}$ and *heavy* otherwise. We say that a vertex i in the graph is light,medium or heavy if w_i is light,medium or heavy.

The proof proceeds by proving that each of these statements is true with large probability.

1. (Both vectors not heavy) If $|w_i|, |w_{\varphi_j^{-1} i}| \leqslant \frac{4|B_1|}{\Delta}$, then, $f(i,j) \geqslant \frac{0.01}{k\Delta} g(i,j)$
2. Let H, M be the set of heavy and medium vertices respectively. Then, if $|H| < 0.001|M|$, one can just ignore the heavy vertices and the above lemma suffices. Else, by expander mixing lemma, most of neighours of a heavy vertex are light. Now we we need the next lemma.
3. One vector heavy and other light. Let $|w_i| > \frac{4|B_1|}{\Delta}$. Let $S \subseteq [k]$ such that $|S| > 0.98k$ and for every $j \in S$, each of $|w_{\varphi_j^{-1} i}|, |w_{\varphi_j i}|$ is *light*. Then, $\sum_{j \in S} f(i,j) + f(\varphi_j(i), j) \geqslant 0.01|B_1|$

4.4 The [PK21] distance bound

We have seen in Theorem 2.2.14 that the twisted product construction of [HHO21] is equivalent to the lifted product when the group acting is abelian. Both these constructions have currently been analyzed in the case when the group is $\mathbb{Z}_l$ but these constructions (and their proofs) are quite different.

The only assumption on the twists in [PK21] is that the lifted graph is expanding i.e. the second largest eigenvalue is bounded by εd. This makes it quite easy to prove part (1) i.e. the expansion part of the proof.

Lemma 4.4.1. *[PK21, Lemma 3] Let G be a d-regular graph, $s : E(G) \to \mathbb{Z}_l$ be a labelling such that $\lambda(G(s)) < \varepsilon d$. Let $C_0 \subseteq \mathbb{F}_2^d$ be a linear code such that $d(C_0), d(C_0^\perp) \geqslant \varepsilon d$. Let $\mathcal{B} = T(G(s), C_0)$, $\mathcal{F} : \mathbb{Z}_l \to \mathbb{Z}_l$ and $\mathcal{E} = \mathcal{B} \otimes_{\mathbb{Z}_l} \mathcal{F}$. Then, there exists constants α, β depending on $(\varepsilon, d, \lambda)$ such that*

- *For any $h \in B_1F_0$ such that $|h| \leqslant \alpha ln$, $|\partial_{\mathcal{E}} h| \geqslant \beta |h|$.*
- *For any $h \in B_0F_1$ such that $|h| \leqslant \alpha ln$, $|\partial^*_{\mathcal{E}} h| \geqslant \beta |h|$.*

The above lemma is easy to prove using the expander mixing lemma and we therefore, omit its proof. Now, we focus on the structure part of the argument to show that $|v| \geqslant \Omega(l|\partial h|)$. Clearly, $|v| \geqslant \frac{|\partial h|}{2} \geqslant \frac{\beta |h|}{2}$ but this does not suffice when $|h|$ is too small. The trick is to work with a modified vector, say h', which has a larger weight but $|\partial h'| \leqslant c|\partial h|$.

By the extension of Künneth's formula to twisted products, Lemma 3.0.1, we get,

$$z = x \otimes \mathbb{1}_0 + y \otimes \mathbb{1}_1 + \partial(\sum_b b \otimes f_b) \text{ where } x \in H_1(\mathcal{B}),\ y \in H_0(\mathcal{B}).$$

In the HHO construction, we could assume that $H_0(\mathcal{B}) = 0$ and thus, for z to be non-trivial, $x \neq 0$. But here, that is no longer the case and we will have two consider both cases.

In this section, we write the group $\mathbb{Z}_l$ multiplicatively being generated by γ. The correspondence is thus, $\gamma^r \leftrightarrow r$ and $\gamma^l = \gamma^0 = \text{id}$. Recalling the definition, $F_1 \cong F_0 = \mathbb{F}_2[e_0, \cdots, e_{l-1}]$. We denote by $\iota : F_1 \to F_0$, the map $\iota(e_i) = e_i$. Clearly, $|\iota(f)| = |f|$ for any $f \in F_1$. It also follows from the definition of the map that $\partial(f) = (1+\gamma)\iota(f)$. This simple re-characterization of the map will be very helpful for norm computations[5].

Lemma 4.4.2 (Case 1)**.** *Let $z = x \otimes \mathbb{1}_0 + \partial_2(w) + y \otimes \mathbb{1}_1 \in H_1(\mathcal{E})$. If $x \neq 0$, then, $|z| \geqslant \frac{\beta \alpha nl}{4}$.*

Proof. Let $z = h + v$ where $h \in B_1F_0,\ v \in B_0F_1$. We will assume $|h| \leqslant \frac{\alpha nl}{4}$ because otherwise the claim is directly true. Denote $h_t := \left(1 + \gamma + \cdots + \gamma^t\right) h$. We first show that there exists $1 \leqslant t \leqslant l$ such that $\frac{\alpha ln}{2} \leqslant |h_t| \leqslant \alpha nl$.

Let $h = \sum_b b \otimes g_b$. Since, $x \neq 0$, there are at least $|x|$ many b such that $|g_b| = 1 \mod 2$. This is because $\partial_2(b \otimes f_b)|_{B_1F_0} = b \otimes \partial(f_b)$ and $\partial(f_b)$ only has even weight. Now, $h_l = \sum_b |g_b| b \otimes \mathbb{1}_0$ and thus

$$|h_l| = l \cdot \#\{b \text{ such that} |g_b| \text{ is odd.}\} \geqslant l d_1(\mathcal{B}) \geqslant \alpha ln.$$

[5]The paper does not make the identification explicit and instead writes $\partial(f) = (1+\gamma)f$. We prefer to do so to prevent confusion between the different spaces.

From the definition, $h_t = h_{t-1} + \gamma^t h$ and thus, $|h_t| \leqslant |h_{t-1}| + |h| \leqslant |h_{t-1}| + \alpha l n/8$. Thus, there exists some t for which $\alpha n l/2 \leqslant |h_t| \leqslant \alpha n l$.

$$\begin{aligned}
\partial\left(\left(1+\gamma+\cdots+\gamma^t\right)h\right) &= \left(1+\gamma+\cdots+\gamma^t\right)\partial h && \text{(Compatibility of action)}\\
&= \left(1+\gamma+\cdots+\gamma^t\right)\partial v && (\partial(h+v)=0)\\
&= \left(1+\gamma+\cdots+\gamma^t\right)(1+\gamma)\iota(v) && \text{(Definition of } \partial_1^{\mathcal{F}})\\
&= (1+\gamma^{t+1})\iota(v).
\end{aligned}$$

Thus, $|\partial(h_t)| = |(1+\gamma^{t+1})\iota(v)| \leqslant 2|v|$. From the expansion property, $|\partial(h_t)| \geqslant \beta|h_t| \geqslant \beta\frac{\alpha l n}{2}$ and thus, the result follows. ∎

We will use the following simple lemma which is easy to prove by a probabilistic argument but we omit it.

Lemma 4.4.3. *[PK21, Lemma 5] Let $w = \sum_b b \otimes f_b$ such that $|f_b| \leqslant l/2$ for every b. Then, there exists $t > 0$ such that $|(1+\gamma^t)w| \geqslant |w|$.*

Lemma 4.4.4 (Case 2)**.** *Let $z = \partial_2(w) + y \otimes \mathbb{1}_1 \in H_1(\mathcal{E})$ and let y be such that w is lowest weight. Then, $|z| \geqslant \frac{\beta l}{6d}$.*

Proof. Let $w = \sum_b b \otimes f_b$ and assume that $|f_b| > l/2$. Then, we consider $w' = w + b \otimes \mathbb{1}_1$ which reduces the weight of f_b. As, $\partial(b \otimes \mathbb{1}_1) = \partial(b) \otimes \mathbb{1}_1$ we can take $y' = y + \partial(b)$ and this reduces weight of w. The low weight assumption therefore implies that $|f_b| \leqslant l/2$ for every b.

Let $z = h + v$ where $h \in B_1F_0$, $v \in B_0F_1$ as before. $v = \partial(w)|_{B_0F_1} + y \otimes \mathbb{1}_1$ and thus,

$$|w| \geqslant \frac{|\partial w|_{B_0F_1}}{d} \geqslant \frac{|y|l - |v|}{d} \geqslant \frac{l - |v|}{d}.$$

If $|h| \geqslant \frac{l-|v|}{6d}$, then, $|z| \geqslant |h| + \frac{|v|}{6d} > \frac{l}{6d}$ and we are done. So we assume otherwise.

Let $w_t = (1+\gamma^t)w$. By Lemma 4.4.3, there is a t' for which $|w_{t'}| \geqslant |w|$. Since, $w_1 = (1+\gamma)w$ and $h = \iota(w_1)$, we have, $|w_1| = |\iota(w_1)| = |h|$. Moreover, $w_t = w_1 + \gamma w_{t-1}$and thus, $|w_t| \leqslant |w_{t-1}| + |h|$. Therefore, we can find, $t \leqslant t'$ such that $\frac{l-|v|}{2d} \leqslant |w_t| \leqslant \frac{l-|v|}{d} \leqslant l$.

$$\begin{aligned}
2|z| \geqslant |(1+\gamma^t)z| &= |\partial(1+\gamma^t)w|\\
&\geqslant \beta|\partial(w_t)|\\
&\geqslant \beta\frac{l-|v|}{2d}\\
&\geqslant \frac{\beta l}{2d} - |z|.
\end{aligned}$$

∎

Chapter 5

Decoding

In this chapter, we will look at the recent decoding algorithms for quantum LDPC codes. The first one by [EKZ20] is a general-purpose reduction which reduces the problem of decoding the tensor product of two codes to decoding the codes individually. The second one we will study is the algorithm from [HHO21] which gives a decoder for the twisted product. Let's begin with the classical definition of the decoding problem and look at what its quantum generalization is.

5.1 Classical Decoding

The classical setup is this: you have a code-word x that you wish to transmit across some communication channel. The process of communication introduces errors and the final output is $x + e$ for some error e. The problem of unique decoding is to recover x given $x + e$ or equivalently, to compute e. There are different error models which try to capture how such errors occur. For example, the errors could be random or adversarial. In the random case, we ask for decoders that work with high probability. A classic example is the Shannon's binary symmetric channel where a code-word is a boolean string and each bit is flipped with a probability p. The one we will stick to is the Hamming model in which the errors can be adversarial but we have an upper bound on the number or such errors. Let's stick to our model of binary linear codes and see what precisely the question is. The code is a linear subspace $\mathcal{C} \subseteq \mathbb{F}_2^n$ with the property that for every non-zero code-word $x \in \mathcal{C}$, $|x| \geqslant d$. The (unique)decoding radius is the maximum amount of error upto which unique decoding is possible. Since, any two code-words are distance d apart, if the error is $< d/2$, then there is a unique code-word closest to the perturbed word. Decoding thus can be seen as computing i.e. $\arg\min_{y \in \mathcal{C}} |y + x'|$ where $x' = x + e$ is our input with a guarantee that $|e| < d/2$ and the minimizer is the unique code-word x. While this is an information theoretic bound, achieving this algorithmically is challenging and we are usually satisfied to decode errors upto a radius that is any constant factor of the distance.

Classical decoding For a code $\mathcal{C} \subseteq \mathbb{F}_2^n$, let $x \in \mathcal{C}$ and $e \in \mathbb{F}_2^n$ such that $|e| < d/2$

Input	$s := x + e$
Output	x or equivalently, $y = \arg\min_{y \in \mathcal{C}} \lvert y + s \rvert$

5.2 Quantum Decoding

As a quantum CSS code is a set of two codes, it is no surprise that we need to solve two decoding tasks. But both are symmetric so we just need to understand one of them. Let $\mathcal{C} : C_2 \to C_1 \to C_0$ be a quantum CSS code. We have two vector spaces $\mathcal{C}_X = \ker(\partial_1)$, $\mathcal{C}_Z^\perp = \text{im}(\partial_2)$ with the relation, $\ker(\partial_1) = \text{im}(\partial_2) \oplus W$. We also know that for every non-zero vector $w \in W$ and any $y \in \text{im}(\partial_2)$, we have $|w + y| \geqslant d_1(\mathcal{C})$. The code is W and we have a perturbed vector $w + e$ where $w \in W$, $e \in C_1$ and $|e| < d/2$. The situation would be identical to the classical case if we were given $w + e$ as the input. However, that is prohibited in the quantum setting due to the no cloning theorem. This is because codewords now represent quantum states and once you "see" them you have destroyed them and hence, you cannot directly read off the perturbed state as that would destroy the original information you wish to correct! However, what one can do is make syndrome measurements.

One can imagine the quantum setting as one in which we have a black-box around the perturbed quantum state $x + e$ that prevents us from seeing it but lets us make apply certain unitary operators and make *syndrome* measurements. Using the syndrome $s = \partial_1(e)$, we want to compute the error e using which one can compute a unitary matrix U_e which is such that $U_e(x + e) = x$ and thus the state is recovered in a black-box fashion. [1] We will first reformulate the classical decoding problem discussed earlier which will make the generalization to the quantum case very natural.

Classical decoding (reformulated) For a 1-complex $\mathcal{C}$, let $e \in C_1$.

Input	$s := \partial_1(e)$
Output	e or equivalently (if $\|e\| < d_1(\mathcal{C})/2$),
	y such that $\partial_1(y) = s$, $\|y\| < d_1(\mathcal{C}) - \|e\|$

This is indeed a reformulation because given $x + e$ as earlier, we can compute $s = \partial_1(x + e) = \partial_1(e)$. If we can decode s to compute e, then we get $x = s + e$. In the other direction, given s, we can use Gaussian elimination to construct $x' + e$ for some $x' \in \mathcal{C}$. Now, we can run the earlier decoder, to compute x' and hence e. So, these two problems are identical. The quantum generalization is now direct,

Quantum decoding For a 2-complex $\mathcal{E}$, let $e \in E_1$.

Input	$s := \partial_1^{\mathcal{E}}(e)$
Output	$y = e + \partial_2(w)$ for any $w \in E_0$, or equivalently,
	y such that $\partial_1(y) = s$, $\|y\| < d_1(\mathcal{E}) - \|e\|$

The first condition ensures that y and e differ only by a kernel vector while the second forces this difference to be a coboundary because of the distance property.

The terms $X(Z)$-decoding refer to decoding $\mathcal{E}$ and $\mathcal{E}^*$ respectively.

[1] The mapping between quantum states and their error correction which are vectors in $\mathbb{C}^{2^n}$ to $\mathbb{F}_2^n$ is quite interesting and detailed information can found in any textbook or course lecture notes .

5.3 Tensor Reduction

Now, we will discuss the reduction in [EKZ20] that proves that if we can decode the individual codes $\mathcal{Q}, \mathcal{C}$ then we can decode the tensor product code. Although, they state their result in a slightly restricted setting and give different proofs for X, Z decoding, the core idea is the same and we will present here a simple unified proof from which deriving their results would be straightforward.

The setting is as follows. Let $\mathcal{X}, \mathcal{Y}$ be chain complexes and $\mathcal{E} := \mathcal{X} \otimes \mathcal{Y}$. For any low-weight error $e \in E_k$, given the syndrome, $\partial_k(e)$, we wish to compute $e + l$ for any $l \in \mathrm{im}(\partial^{\mathcal{E}}_{k+1})$. The goal of the section is to show that if we have decoding algorithms for the complexes, $\mathcal{X}, \mathcal{Y}$, we can decode $\mathcal{E}$. For a vector $x \in E_k$, let x_j denote its projection to $X_j Y_{k-j}$.

Uncoupled case

We start with the case when the error e is supported only on one of the subspaces $X_j Y_{k-j}$. The general case follows directly by reducing this case using coboundary elements.

Lemma 5.3.1. *Let $e_j \in X_j Y_{k-j}$ such that $|e_j| \leqslant \delta^{\mathcal{X}}_j \delta^{\mathcal{Y}}_{k-j}$. Then, e_j can be decoded given $\partial^{\mathcal{E}}_k(e_j)$.*

Proof. Let $e_j = \sum_y v_y \otimes y$ where y runs over the basis of Y_{k-j}. Let $\partial(e_j) = s_j + s_{j-1}$, and $s_{j-1} = \sum_y \partial^{\mathcal{X}}_j(v_y) \otimes y$. Thus, we can simply read off $\partial^{\mathcal{X}}_j(v_y)$ and feed this to the X_j-decoder to obtain v'_y. Let $u = \sum_y v'_y \otimes y$.

If $|v_y| \leqslant \delta^{\mathcal{X}}_j$, then the decoder's output v'_y is equal to v_y. Let $S = \{y \mid |v_y| > \delta^{\mathcal{X}}_j\}$. Since, $|e_j| \leqslant \delta^{\mathcal{X}}_j \delta^{\mathcal{Y}}_{k-j}$, we get that $|S| \leqslant \delta^{\mathcal{Y}}_{k-j}$. Therefore,

$$e_j + u = \sum_{y \in S} (v_y + v'_y) \otimes y = \sum_x x \otimes w_x.$$

If we visualize the tensor product space $X_j Y_{k-j}$ as a matrix, then the change of basis from Y to X amounts to looking at rows instead of columns. As a consequence, for every x, $|w_x|$ is bounded by the number of non-zero columns which are supported on S. Thus, $|w_x| \leqslant |S| \leqslant \delta^{\mathcal{Y}}_{k-i}$. We can thus, compute $s_j + \partial(u)_j = \partial(e+u)_j = \sum_x x \otimes \partial^{\mathcal{Y}}_{k-j}(w_x)$ and decode each $\partial^{\mathcal{Y}}_{k-j}(w_x)$ by $\mathcal{Y}$-decoding. ∎

Normal Form

To handle the general case, we need to first have a kind of *normal* form for the error e such that we can decouple the syndrome and decode it individually using Lemma 5.3.1. To do so, we start by decomposing our spaces as a union of the images of the boundary map and a "residue". Let $\mathcal{X}, \mathcal{Y}$ be chain complexes and let $I_i := \mathrm{im}(\partial^{\mathcal{X}}_{i+1})$, $J_i := \mathrm{im}(\partial^{\mathcal{Y}}_{i+1})$. Then, [2]

$$\begin{aligned} X_i &= I_i \oplus X'_i, \quad Y_{k-i} = J_{k+1-i} \oplus Y'_{k-i} \\ X_i \otimes Y_{k+1-i} &= (I_i \oplus X'_i) \otimes \left(J_{k-i} \oplus Y'_{k-i}\right) \\ &= I_i J_{k+1-i} \oplus X'_i J_{k-i} \oplus I_i Y'_{k+1-i} \oplus X'_i Y'_{k-i} \end{aligned}$$

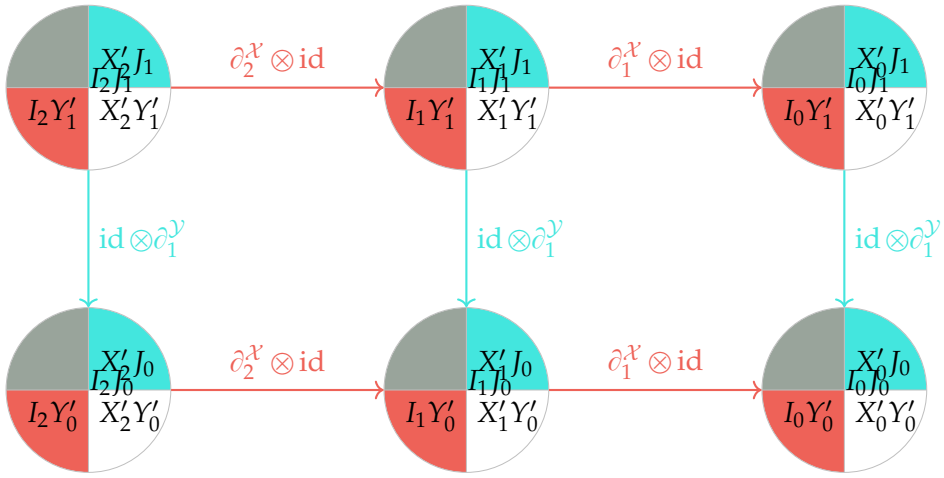

Figure 5.1: A visual depiction of the decomposition

We say that $e \in \mathcal{E}_k$ is in *normal form* if given its syndrome, $\partial_k(e)$, we can compute the syndrome of each component $\partial_k(e_j)$. The following lemma states that we can always obtain such a form by adding a nice coboundary.

Lemma 5.3.2. *For any $e \in E_k$, there exists a vector $z \in E_{k+1}$ such that, $e' = e + \partial_{k+1}(z)$ is in normal form. Hence, given $\partial_k(e)$, we can compute $\partial_k(e'_j)$ for every $j \leqslant k$.*

Proof. Since, $X'_jJ_{k-j} \subseteq \mathrm{im}(\mathrm{id} \otimes \partial_1^{\mathcal{Y}})$, there exists a z_j such that the restriction of $\partial_{k+1}z_j + e_j$ to X'_jJ_{k-j} is zero. Let $z = \sum_j z_j$ and $e' = e + \partial_{k+1}(z)$. Then,

$$e'_j = e_j + (\mathrm{id} \otimes \partial_1^{\mathcal{Y}})(z_j) + (\partial_1^{\mathcal{X}} \otimes \mathrm{id})(z_{j+1}).$$

The restriction of e'_j to X'_jJ_{k-j} is still zero as $\mathrm{im}(\partial_1^{\mathcal{X}} \otimes \mathrm{id})$ is disjoint from X'_jJ_{k-j}. For any j, we look at $s_j = (\mathrm{id} \otimes \partial_1^{\mathcal{Y}})(e_j) + (\partial_1^{\mathcal{X}} \otimes \mathrm{id})(e_{j+1})$. By definition, the first term is contained in X_jJ_{k-j-1}. Since, e_{j+1} has a zero $X'_{j+1}J_{k-j-1}$ component, $(\partial_1^{\mathcal{X}} \otimes \mathrm{id})(e_{j+1})$ is contained in $I_jY'_{k-j-1}$ and hence disjoint from first term. Thus, by projecting s_j to the relevant subspace we can compute each term individually and hence compute $\partial_k(e'_j)$. ∎

Theorem 5.3.3. *Let $\delta_i^{\mathcal{X}}, \delta_j^{\mathcal{Y}}$ be the number of errors that the decoding algorithm for the maps $\partial_i^{\mathcal{X}}, \partial_j^{\mathcal{Y}}$ can correct. If $\partial_i = 0$ we let $\delta_i = 1$. Then, Algorithm 1 corrects $\min_i \delta_i^{\mathcal{X}} \delta_{k-i}^{\mathcal{Y}}$ errors for the map $\partial_k^{\mathcal{E}}$ where the minimum is over the indices i such that $\dim(X_iY'_{k-i}) > 0$.*

Proof. Let the error be e and assume it is in normal form. Using Lemma 5.3.2, we can compute $\partial_k(e_j)$ for each j. Using Lemma 5.3.1, we can decode e_j and thus we obtain $e = \sum_j e_j$. ∎

[2]**Note** - There are situations in which we already have a fixed basis which we do not wish to alter. We therefore do not assume that we have a basis that respects this decomposition.

Algorithm 1: Decoding TensorProduct

input : $s = \partial_k^{\mathcal{E}}(e)$
output: $e + l,\ l \in \mathrm{im}(\partial_{k+1}^{\mathcal{E}})$
$u \leftarrow 0$;
for $j \in \{k, \cdots, 0\}$ **do**
  if $\dim(X_j Y'_{k-j}) > 0$ **then**
    if $\partial_j^{\mathcal{X}} \neq 0$ **then**
      Write $s_{j-1} = \sum_y \partial_j^{\mathcal{X}}(v_y) \otimes y$;
      Decode each $\partial_j^{\mathcal{X}}(v_y)$ using X_j-decoding algorithm to obtain v'_y;
      $u_j \leftarrow \sum_y v'_y \otimes y$;
      $u \leftarrow u + u_j$;
      $s \leftarrow s + \partial_k(u_j)$;
    if $\partial_{k-j}^{\mathcal{Y}} \neq 0$ **then**
      Let P be the projection matrix $P : X_j \to X'_j$
      Obtain $(P \otimes I)s_j = \sum_x x \otimes \partial_{k-j}^{\mathcal{Y}}(w_x)$
      Decode each $\partial_{k-j}^{\mathcal{Y}}(w_x)$ using Y_{k-j}-decoding algorithm to obtain w'_x;
      $u'_j \leftarrow \sum_x x \otimes w'_x$;
      $u \leftarrow u + u'_j$;
      $s \leftarrow s + \partial_k(u'_j)$;
return u

We can now use this to get the results from [EKZ20] as a special case.

Theorem 5.3.4. *[EKZ20, Thm. 2.5] Let $\mathcal{X}$ be a Ramanujan 2-complex and $\mathcal{C}$ be a 1-complex such that $\partial_1^{\mathcal{C}}$ is surjective. Let $\mathcal{E} = (\mathcal{X} \otimes \mathcal{C})_3$.*

i) Suppose the classical LDPC code $\mathcal{C}$ comes with a polynomial-time decoding algorithm that corrects any pattern of less than α fraction of errors. Then, there is a polynomial time algorithm for $\mathcal{E}$ that corrects all X-errors of weight smaller than $\alpha|C_1|\frac{d_1(\mathcal{X})}{2}$.

ii) Suppose there is a polynomial time decoding algorithm for the component quantum code $\mathcal{X}$ that corrects any pattern of Z-errors of weight smaller than w. Then there exists a polynomial time algorithm for $\mathcal{E}$ that corrects any pattern of Z- errors of weight smaller than w.

Proof. i) We have to find an error in the space $\mathcal{E}_1 = X_2C_0 \oplus X_1C_1$. Since, $\partial_1^{\mathcal{C}}$ is surjective, $\dim(C'_0) = 0$ and hence $\dim(X_2C'_0) = 0$. Now, from Theorem 5.3.3, we get that we can do decode $\mathcal{E}_1$ upto $\delta_1^{\mathcal{X}}\delta_1^{\mathcal{Y}}$ errors. From the classical LDPC assumption, $\delta_1^{\mathcal{Y}} = \alpha|C_1|$ and the decoder for $\partial_1^{\mathcal{X}}$ is the cyclic decoder which can decode all errors upto $\frac{d_1(\mathcal{X})}{2}$ in polynomial time. Thus, $\delta_1^{\mathcal{X}} = \frac{d_1(\mathcal{X})}{2}$. The result directly follows.

ii) Now, we will apply the theorem to $\mathcal{E}^*$. Clearly, $\partial_0^{\mathcal{X}^*} = \partial_0^{\mathcal{C}^*} = 0$ and by convention $\delta_0^{\mathcal{X}^*} = \delta_0^{\mathcal{Y}^*} = 1$. Also, $(\partial_1^{\mathcal{C}})^*$ is injective and thus, $\partial_1^{\mathcal{Y}^*}$ is completely invertible and thus, $\min(\delta_1^{\mathcal{X}^*}, \delta_1^{\mathcal{Y}^*}) = \delta_1^{\mathcal{X}^*}$. Z-error decoding of $\mathcal{X}$, corresponds to decoding $\partial_1^{\mathcal{X}^*}$ and thus $\delta_1^{\mathcal{X}^*} = w$. Therefore, from Theorem 5.3.3, we can correct w errors.

■

5.4 Twisted Product

We denote by $\mathcal{E} = \mathcal{B} \otimes_\varphi \mathcal{F}$, the twisted product construction as described earlier in Section 2.2.5. The paper only gives us a Z-decoding algorithm because it relies on unique expansion which is a stronger form of (α, β)-expansion. We call a map $H : \mathbb{F}_2^m \to \mathbb{F}_2^n$, (α, γ) - unique expanding if for every $x \in \mathbb{F}_2^m$ such that $|x| \leqslant \alpha m$, there are at least $\gamma|x|$ elements $b \in \text{supp}(Hx)$ such that $b \in \text{supp}(\partial(a))$ for a unique $a \in \text{supp}(x)$.

Lemma 5.4.1. *[HHO21, Prop. 3.1] If $\mathcal{B}$ is constructed randomly as described, then, except with probability at most $O(1/n^{100})$, the map $\partial_1^{\mathcal{B}^*} : B_0 \to B_1$ is $(\frac{1}{10^5\Delta}, 0.81\Delta)$ unique-expanding.*

Thus, we will always use the dual maps and to avoid notational clutter, we now redefine our complex to be the dual of the actual one,

$$\mathcal{B} := B_0 \to B_1,\ \mathcal{E} := B_0F_0 \to B_0F_1 \oplus B_1F_0 \to B_1F_1.$$

Given $s \in B_1F_1$ as input, the goal is to recover a vector y which is equal to the error upto coboundaries. The only certificate that y is indeed of the above form is to show that $\partial_1(y) = s$ and that y is low-weight. The algorithm will also proceed along the same lines. We first construct a y such that $\partial_1(y) = s$ and then iteratively add elements of $\ker(\partial_1)$ such that $|y|$ becomes lesser than $d_1(\mathcal{E}) - |e^*|$.

5.4.1 The algorithm

Let us denote by e^*, the error such that the input provided is $s = \partial_1(e^*)$. We will use $x|_v$, $x|_h$ to denote the restriction of $x \in E_1$ to the spaces B_1F_0, B_0F_1 respectively.

The first step is to build a pre-image of the syndrome which is computationally easy using the following lemma which gives us a vector e_0 such that $e^* + e_0 \in \ker(\partial_1)$.

Lemma 5.4.2. *Given $s = \partial_1(e)$, we can compute a vector e_0 such that $\partial_1(e_0) = s$, in deterministic polynomial time.*

Proof. Let $e^*|_h = e_b \otimes \mathbb{1}_1 + \sum_a a \otimes f_a$ where $|f_a|$ is even for every a. Similarly, let $s = s_p \otimes \mathbb{1}_1 + \sum_b b \otimes f_b$. By direct computation and a parity argument , $\partial_1^{\mathcal{B}}(e_b) = s_p$. Now, if we we assume that $\partial_1^{\mathcal{B}^*} : B_1 \to B_0$ was surjective, $\partial_1^{\mathcal{B}}$ is injective and we can find the exact pre-image e_b by linear algebra. [3]

$$\begin{aligned}
e^* &= e^*|_h + \sum_b b \otimes g_b \\
&= e_b \otimes \mathbb{1}_1 + \sum_a a \otimes f_a + \sum_b b \otimes g_b \\
&= e_b \otimes f_0 + \sum_a a \otimes f'_a + \sum_b b \otimes g_b \\
\partial(e^*) &= \partial(e_b \otimes f_0) + \partial\left(\sum_a a \otimes f'_a\right) + \sum_b b \otimes \partial(g_b) \\
s &= \partial(e_b \otimes f_0) + \sum_b b \otimes \left(\sum_{a : b \in \partial a} \varphi(b,a) f'_a\right) + \sum_b b \otimes \partial(g_b)
\end{aligned}$$

[3] [HHO21] uses Sipser-Spielman to compute e_b.

$$= \partial(e_b \otimes f_0) + \sum_b b \otimes (f_b + \partial(g_b))$$
$$= \partial(e_b \otimes f_0) + \sum_b b \otimes \partial(g'_b)$$

The main thing is that $|f'_a|$ is even for each a and thus, $|f_b|$ and $|f_b + \partial(g_b)|$ are also even. Since, every vector of even weight is in the image, we have such a g'_b . Since, s is known and $\partial(e_b \otimes f_0)$ can be computed, we can compute $\partial(g'_b)$. Computing g'_b is then easy as the boundary map here is the one for the circle. ■

This completes part one of the algorithm which builds some pre-image of the syndrome. In the HHO construction, we have $H_0(\mathcal{B}) = 0$, $H_i(\mathcal{F}) = \text{span}\{\mathbb{1}_i\}$ and any twist preserves the all-ones vector. Applying Lemma 3.0.1 then gives us,

$$e^* + e_0 = \partial_0^{\mathcal{E}}(w) + x \otimes \mathbb{1}_0.$$

The task now is to remove the part not in the coboundary, i.e., x. This will be done by a sequence of local changes that will try to reduce $|e_0|_h|$. Let us look at the overall algorithm now and then focus on the iterative step.

Algorithm 2: Iterative HHO decoder

input : $s = \partial_1^{\mathcal{E}}(e^*)$
output: $y = e^* + l,\ l \in \text{im}(\partial_2^{\mathcal{E}})$
Let $s = \sum_{b \in S_o} b \otimes \mathbb{1}_1 + \sum_{b \in S_e} b \otimes f_b$;
$s_p \leftarrow \sum_{b \in S_o} b$;
Compute e_b such that $\partial_1^{\mathcal{B}}(e_b) = s_p$;
Compute $z = \sum_b b \otimes g_b \in B_1 F_0$ such that $\partial_1(z) = \sum_{b \in S_e} b \otimes f_b$;
$y \leftarrow e_b \otimes f_0 + z$;
while *There is some progress* **do**
  for $a \in B_0$ **do**
    Compute $y_a \in F_0, x \in B_1$ such that $|\,(y + \partial_2(a \otimes y_a) + x \otimes \mathbb{1}_0)_h\,|$ is minimized;
    $y \leftarrow y + \partial_2(a \otimes y_a) + x \otimes \mathbb{1}_0$;
  end
end
return y

Line 8 can be done in two ways. The key is to notice that this is a system of 2-XOR equations and we need to satisfy as many equations as possible. The brute force method is to go over all x supported over $\partial_1(a)$ and then we can readily compute y. This takes time $O(2^{\Delta}|F_0|)$ which is doable in polynomial time if the degree is at most $O(\log n)$. In [HHO21], the degree is $O(\log^2 n)$ and thus they appeal to the MAX-CUT algorithm by Goemans-Williamson to approximately compute such a solution. We now mention a couple observations from such an update that will be helpful later.

1. Observation 1 - We can assume that $|y_a| \leqslant |F_0|/2$. It is easy to see that we can toggle the x to compensate for it
2. Observation 2 - The change is non-trivial i.e $y_a \neq 0$ only if $y \cap \text{supp}(\partial(a)) > \frac{\Delta}{2}$. This is because if half of the neighbours of a are currently unsupported i.e., empty, then

adding this adds a weight of at least $\frac{\Delta|y_a|}{2}$. The decrease on the supported neighbors can be at most $\frac{\Delta|y_a|}{2}$. Therefore, for the y_a to reduce weight more than half its neighbors must already be in the support of y.

5.4.2 The proof

As mentioned earlier, we need to argue that once the decoder halts, i.e., once we reach a "local minima", we will obtain a sufficiently low weight y. Let e_0 be the initial y before the start of weight reduction (line 6).

Denote by e_t the vector y after t iterations and let e_τ be the final output. The proof outline is as follows.

1. We start with $e_0 = e_b \otimes f_0 + z,\ z \in B_1F_0$ such that $|e_b| \leqslant |e^*|$
2. We say that a is *fixed* if in some iteration we added a non-trivial $a \otimes y_a$. We first show that the set A of fixed a is small i.e. $|A| \leqslant c|B_0|$.
3. Thus, the output is $e_\tau = e_b \otimes f_0 + z + \partial(\sum_{a\in A} a \otimes y_a) + x_\tau \otimes F_0$. Rewrite this as $e_\tau = e_b \otimes f_0 + \sum_{a\in A} a \otimes \partial(y_a) + z_h$ where $z_h \in B_1F_0$ is the horizontal part.
4. Let $e_\tau = e^* + \partial(w) + x \otimes \mathbb{1}_0$. Comparing the vertical (i.e. B_0F_1) components, we get that if $w = \sum_{a\in W} a \otimes w_a$, then $|W| \leqslant |A| + |e_b|$.
5. We rearrange and compare horizontal parts to get,

$$(e_\tau + e^* + \partial(w))_h := \sum_b b \otimes g_b = x \otimes \mathbb{1}_0.$$

The crux of the entire proof is to show that for any b, $|g_b| < |F_0|$ and thus $x_b = 0$. This is achieved by using that $|W|$ is small (which enables using vertex expansion of $\partial^{\mathcal{B}}$), and that e_τ is a local minima.

Few fixes

In this part, we will prove the claim that the number of elements that are *fixed* are few. Let A_t be the set of all a that have been fixed till the t^{th} iteration. Note that $|A_t| \leqslant t$ as the same a could be *fixed* multiple times.

Going back to the proof of Lemma 5.4.2, we have $s = \partial(e_b \otimes f_0) + \partial(z)$ where $z = \sum_{b\in Q} b \otimes g'_b$. Here, the set Q is contained in the support of s and $\partial(e_b)$. But e_b was constructed such that $\partial(e_b) = s_p$ whose support is a subset of that of s. Thus, $Q \subseteq \text{supp}(s)$ and hence, $|Q| \leqslant \Delta|e^*|$.

Lemma 5.4.3. *Let $\partial(A_t) = \bigcup_{a\in A_t} \partial(a)$. Then, $\gamma|A_t| \leqslant |\partial(A_t)| \leqslant |Q| + \frac{\Delta}{2}|A_t|$. Therefore, $|A_t| \leqslant \frac{2|Q|}{(2\gamma-\Delta)} \leqslant \frac{2\Delta|e^*|}{(2\gamma-\Delta)}$ for any time t.*

Proof. By (α, γ) expansion of ∂ and the assumption that $|A_t| \leqslant \alpha|B_0|$, we have that $|\partial(A_t)| \geqslant \gamma|A_t|$. We now upper bound this to prove a bound on $|A_t|$.

Let a_t be the last new element that is *fixed* in the set A_t. Let C_t are the co-unique elements added to the support, i.e, $C_t = \{b \in \partial(a_t) \mid b \notin \partial(A_{t-1})\}$. Let $D_t = \partial(a_t) \backslash C_t$. We can see that $\partial(A_t) = \bigcup_t C_t$.

From observation 2, we know that at least half of neighbors of a_t must be present, i.e., $|\partial(a_t) \cap (Q \cup \partial A_{t-1})| \geqslant \Delta/2$. Now, D_t is already present and the only elements that can be present in C_t are those in Q. Thus, $|C_t \cap Q| \geqslant \Delta/2 - |D_t|$ and therefore,

$$|C_t \cap \overline{Q}| \leqslant |C_t| - (\Delta/2 - |D_t|) = |C_t| + |D_t| - \Delta/2 = \Delta/2.$$

Now, $|(\partial(A_t) \cap Q)| \leqslant |Q|$ and $\partial(A_t) \cap \overline{Q} = \bigcup_t (C_t \cap \overline{Q})$. Thus,

$$\gamma|A_t| \leqslant \partial(A_t)| = |Q| + \sum_t |C_t \cap \overline{Q}| \leqslant |Q| + \frac{\Delta|A_t|}{2}. \qquad \blacksquare$$

Small total weight

This is an ingenious inductive argument and the the crux of the entire proof[4]. As mentioned before, let

$$e_\tau = e^* + \partial(w) + x \otimes \mathbb{1}_0$$

where $w = \sum_{a \in W} a \otimes y_a$. We have noted that $|W| \leqslant |A| + |e_b|$ and from the earlier part we have $|W| \leqslant c|e^*| \leqslant \alpha|B_0|$ where c is a constant. The inductive procedure is to slowly unravel $\partial(w)$, i.e. , we order the elements of W and define $W_t = W \backslash \{a_1, \cdots, a_{t-1}\}$. Thus, $W_1 = W$ and we inductively define W_{t+1} by picking $a_t \in W_t$ as follows.

Inductive Ordering Let $S_t = \bigcup_{a \in W_t} \partial(a)$ and decompose $S_t = C_t \cup D_t$ which like earlier denotes the sets of unique and non-unique neighbors i.e. C_t is that subset of b which has a unique neighbor in W_t. Since, $|W_t| \leqslant |W| \leqslant \alpha|B_0|$, we can use Lemma 5.4.1 to get that $|C_t| \geqslant \gamma|W_t|$. By an averaging argument, we have that there exists an $a \in W_t$ with at least γ neighbors in C_t. We define this to be a_t. If there are many such, pick any.

Theorem 5.4.4. *[HHO21, Lemma 5.2] Let $e_1 = e^*$ and recursively define $e_t = e_{t-1} + \partial(a_{t-1} \otimes y_{a_{t-1}}) := \sum_b b \otimes g_b^t$ where the $a_i \in W$ are defined as above. If the expansion parameter $\gamma > \frac{3\Delta}{4}$, then for every time step t the following statements hold,*

1. *For every $b \in B_1$ we have $|g_b^t| < \frac{|F_0|}{2}$*
2. *$\sum_{b \in S_t} |g_b^t| \leqslant \frac{|F_0|}{100}$ where $S_t = \bigcup_{i \geqslant t} \partial(a_i)$*
3. *For $t > 1$, $|y_{a_{t-1}}| < \frac{|F_0|}{2}\left(\frac{\Delta}{\gamma} - \frac{96}{100}\right)$*

Before we prove it, we see how this suffices to show that x is 0 and hence, decoding is achieved.

Corollary 5.4.5. *The vectors e_τ, e^* differ by a coboundary, i.e., $e_\tau = e^* + \partial(w)$.*

Proof. At the last step, i.e., $t = |W| + 1$, we have that $e_{|W|+1} = e^* + \partial(w) = e_\tau + x \otimes \mathbb{1}_0$. We compare the horizontal parts on either sides. On the LHS, for any b we are guaranteed that $|g_b| < \frac{|F_0|}{2}$ and we also have that same for any g_b^τ in e_τ as it is a local minima and if a b violates it, then $e_\tau + b \otimes \mathbb{1}_0$ would reduce the horizontal weight and therefore the algorithm wouldn't have terminated. Now, $b \otimes g_b = b \otimes g_b^\tau + b \otimes x_b \mathbb{1}_0$ and since $|g_b| + |g_b^\tau| < |F_0|$ we must have $x_b = 0$. As it holds for every b, $x = 0$. $\blacksquare$

[4] According to me, of course.

Proof Overview The reason that W is sorted according to co-unique neighbors is that the counique neighbors at time t don't change after time t as they have no other neighbors. Thus, for such b we can reason about $g_b^t = g_b^\tau$ by leveraging that e_τ is a local minima. For other b, we have to resort to a worst case trivial bound. From local minimality, $|e_\tau + \partial(a \otimes z)|_h \geqslant |e_\tau|_h$ for any z. Using this we can glean two facts about $y_{a_{t-1}}$. Let the unique neighbors of a_{t-1} be $C := C_{t-1} \cap \partial(a_{t-1})$ and let $D := \partial(a_{t-1}) \backslash C$.

- Applying it with $z = y_{a_{t-1}}$, we get that $|y_{a_{t-1}}|$ is not too large (i.e. claim 3) if the weight on $C \subseteq S_{t-1}$ is small (claim 2).
- For any $i \in \mathrm{supp}(y_{a_{t-1}})$, apply it with $z = g_i$ which is a vector in F_0 only supported on i. We get that $\sum_{b \in C} |(g_b)_{\varphi(b,a_{t-1})^{-1} i}| \geqslant \frac{\Delta}{2} - |D|$. This readily proves claim 2.

What follows is merely a more detailed exposition of the above idea.

Proof of Theorem 5.4.4. **Base Case**. $e_1 = e^*$. Thus, $\sum_b |g_b| = |e^*| \leqslant \frac{|F_0|}{100}$ and therefore both the conditions are satisfied. The third claim is valid only for $t > 1$.

Inductive Step - Let the inductive claim hold at time $t-1$. We first prove the third claim. Since, e_τ is a local minima,

$$|e_\tau|_h \leqslant |e_\tau + \partial(a_{t-1} \otimes y_{a_{t-1}})|_h.$$

Clearly, the vectors only differ in g_b for $b \in \partial(a_{t-1})$. Let the unique neighbors of a_{t-1} be $C := C_{t-1} \cap \partial(a_{t-1})$. Since $a_{t-1} \in W_{t-1}$, we have that $\partial(a_{t-1}) \subseteq S_{t-1}$. By the recursive definition we have,

$$g_b^t = g_b^{t-1} + \varphi(b, a_{t-1})^{-1} y_{a_{t-1}}$$

and thus,

$$|g_b^t| = |g_b^{t-1} + \varphi(b, a_{t-1})^{-1} y_{a_{t-1}}| \geqslant |y_{a_{t-1}}| - |g_b^{t-1}| \tag{5.1}$$

For any subsequent any $t' \geqslant t$, $g_b^{t'} = g_b^t$ because if $b \in C$ then it has no other neighbor in W_t. Thus, this y_b is the same as that in e_τ.

The weight added must be greater than the weight reduced so we have,

$$\begin{aligned}
\sum_{b \in C} |y_b| + \sum_{b \in D} |y_b| &\leqslant \sum_{b \in C} |y'_b| + \sum_{b \in D} |y'_b| \\
\sum_{b \in C} |y_b| &\leqslant \sum_{b \in C} |y'_b| + \sum_{b \in D} (|y_b| - |y'_b|) \\
\sum_{b \in C} |g_b^t| &\leqslant \sum_{b \in C} |g_b^{t-1}| + \sum_{b \in D} \tfrac{|F_0|}{2} \\
|C||y_{a_{t-1}}| - \sum_{b \in C} |g_b^{t-1}| &\leqslant \sum_{b \in C} |g_b^{t-1}| + |D| \tfrac{|F_0|}{2} && \text{(Using Eq. (5.1))} \\
|C||y_{a_{t-1}}| &\leqslant 2 \sum_{b \in C} |g_b^{t-1}| + |D| \tfrac{|F_0|}{2} \\
|C||y_{a_{t-1}}| &\leqslant 2 \frac{|F_0|}{100} + |D| \tfrac{|F_0|}{2} && \text{(By inductive claim 2)} \\
|y_{a_{t-1}}| &\leqslant \frac{|F_0|}{2} \left(\frac{\Delta - |C|}{|C|} + \frac{4}{100} \right) && (|C| + |D| = \Delta)
\end{aligned}$$

$$|y_{a_{t-1}}| \leqslant \frac{|F_0|}{2}\left(\frac{\Delta}{\gamma} - \frac{96}{100}\right) \qquad \text{(a is such that } |C| \geqslant \gamma\text{)}.$$

This shows claim 3 and using this we can easily show claim 1 for the inductive step t as the only b that were changed were those present in $\partial(a_{t-1}) \subseteq S_{t-1}$. Thus, $|g_b^t| \leqslant |g_b^{t-1}| + |y_{a_{t-1}}| \leqslant \frac{|F_0|}{100} + \frac{|F_0|}{2}\left(\frac{\Delta}{\gamma} - \frac{96}{100}\right)$. This is less than $\frac{|F_0|}{2}$ if $\gamma > \frac{100}{194}\Delta$.

Now we prove claim 2. First note that $S_t = S_{t-1}\backslash C$. So, we need to account for all the weight added on D minus the weight already present on C. We write $y_{a_{t-1}} = \sum_{i\in T} g_i$ where g_i are unit vectors in F_0 supported only on i and $T = \text{supp}(y_{a_{t-1}})$. Clearly, total weight added is at most $|D||T|$. The weight removed is the sum of weight removed for each such i. Let i_b denote $\varphi(b, a_{t-1})^{-1}i$. In precise terms, the weight removed is $\sum_{i\in T}\sum_{b\in C}|(g_b^{t-1})|_{i_b}$. We will now show that the inner sum is at least $\frac{|C|-|D|}{2}$ for every i. This implies that total difference is $|T|\left(|D| - \frac{|C|-|D|}{2}\right) \leqslant 0$ if $\gamma > \frac{3\Delta}{4}$.

From local minimality, we have $|e_\tau + a_{t-1}\otimes g_i|_h \geqslant |e_\tau|_h$. For $b \in C$, $g_b^\tau = g_b^t$ and $(e_\tau + a_{t-1}\otimes g_i)_b = g_b^{t-1}$. For $b \in D$ we don't make any claims and use the trivial bounds.

Thus,

$$\begin{aligned}
\sum_{b\in C}|(g_b^{t-1})_{i_b}| + \sum_{b\in D}|(z_b)_{i_b}| &\geqslant \left|\sum_{b\in C}(g_b^t)_{i_b}\right| + \sum_{b\in D}|(z_b + \varphi(b,a_{t-1})^{-1}g_i)_{i_b}| \\
\sum_{b\in C}|(g_b^{t-1})_{i_b}| &\geqslant \left|\sum_{b\in C}(g_b^{t-1} + \varphi(b,a_{t-1})^{-1}g_i)_{i_b}\right| + \sum_{b\in D}\left(|(\overline{z_b})_{i_b}| - |(z_b)_{i_b}|\right) \\
\sum_{b\in C}|(g_b^{t-1})_{i_b}| &\geqslant \sum_{b\in C}\left(|\varphi(b,a_{t-1})^{-1}g_i)_{i_b}| - |g_b^{t-1}|\right) - |D| \\
2\sum_{b\in C}|(g_b^{t-1})_{i_b}| &\geqslant |C| - |D|.
\end{aligned}$$

This concludes the proof of claim 2 and, therefore, of the entire theorem. ∎

Bibliography

[ACKM19] Naman Agarwal, Karthekeyan Chandrasekaran, Alexandra Kolla, and Vivek Madan. On the Expansion of Group-Based Lifts. *SIAM J. Discret. Math.*, 33(3):1338–1373, 2019. arXiv:1311.3268, doi:10.1137/17M1141047.

[AMN98] Yossi Azar, Rajeev Motwani, and Joseph (Seffi) Naor. Approximating probability distributions using small sample spaces. *Combinatorica*, 18(2):151–171, February 1998. doi:10.1007/pl00009813.

[Aud14] Benjamin Audoux. An application of Khovanov homology to quantum codes. *Annales de l'Institut Henri Poincaré D*, 1(2), 2014. arXiv:1307.4677, doi:10.4171/AIHPD/6. 9

[BE21a] Nikolas P. Breuckmann and Jens N. Eberhardt. Balanced Product Quantum Codes. *IEEE Transactions on Information Theory*, 67(10):6653–6674, 2021. arXiv:2012.09271, doi:10.1109/TIT.2021.3097347. 4, 5, 12, 14, 20

[BE21b] Nikolas P. Breuckmann and Jens N. Eberhardt. Quantum Low-Density Parity-Check Codes. *PRX Quantum*, 2:040101, Oct 2021. arXiv:2103.06309, doi:10.1103/PRXQuantum.2.040101. 5

[BH14] Sergey Bravyi and Matthew B. Hastings. Homological Product Codes. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 273–282. ACM, 2014. arXiv:1311.0885, doi:10.1145/2591796.2591870. 4

[BL06] Yonatan Bilu and Nathan Linial. Lifts, discrepancy and nearly optimal spectral gap. *Combinatorica*, 26(5):495–519, October 2006.

[BMvT78] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Trans. Inf. Theory*, 24(3):384–386, 1978. doi:10.1109/TIT.1978.1055873.

[BS01] Eli Ben-Sasson. *Expansion in Proof Complexity*. PhD thesis, Hebrew University, 2001.

[BTL10] Sergey Bravyi, Barbara M Terhal, and Bernhard Leemhuis. Majorana fermion codes. *New Journal of Physics*, 12(8):083039, Aug 2010. URL: http://dx.doi.org/10.1088/1367-2630/12/8/083039, doi:10.1088/1367-2630/12/8/083039. 3

[CS96] A. R. Calderbank and Peter W. Shor. Good quantum error-correcting codes exist. *Phys. Rev. A*, 54:1098–1105, Aug 1996. doi:10.1103/PhysRevA.54.1098. 3, 8

[CT12] Eden Chlamtac and Madhur Tulsiani. *Convex Relaxations and Integrality Gaps*, pages 139–169. Springer US, Boston, MA, 2012. doi:10.1007/978-1-4614-0769-0_6.

[DFHT21] Irit Dinur, Yuval Filmus, Prahladh Harsha, and Madhur Tulsiani. Explicit SoS lower bounds from high-dimensional expanders. In *12th Innovations in Theoretical Computer Science Conference, ITCS 2021*, 2021. doi:10.4230/LIPIcs.ITCS.2021.38.

[DMS99] Ilya Dumer, Daniele Micciancio, and Madhu Sudan. Hardness of approximating the minimum distance of a linear code. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99,17-18 October, 1999, New York, NY, USA*, pages 475–485. IEEE Computer Society, 1999. doi:10.1109/SFFCS.1999.814620.

[EKZ20] Shai Evra, Tali Kaufman, and Gilles Zémor. Decodable quantum LDPC codes beyond the square root distance barrier using high dimensional expanders. In *Proceedings of the 61st IEEE Symposium on Foundations of Computer Science*, pages 218–227. IEEE, 2020. arXiv:2004.07935, doi:10.1109/FOCS46700.2020.00029. 4, 5, 10, 11, 23, 29, 31, 33

[Gal60] Robert G Gallager. *Low density parity check codes*. PhD thesis, Massachusetts Institute of Technology, 1960. URL: https://dspace.mit.edu/handle/1721.1/11804. 3

[Gal62] R. Gallager. Low-density parity-check codes. *IRE Transactions on Information Theory*, 8(1):21–28, 1962. doi:10.1109/TIT.1962.1057683. 3

[Gil52] E. N. Gilbert. A comparison of signalling alphabets. *Bell System Technical Journal*, 31(3):504–522, 1952. doi:10.1002/j.1538-7305.1952.tb01393.x. 3

[HHO21] Matthew B. Hastings, Jeongwan Haah, and Ryan O'Donnell. Fiber bundle codes: breaking the $n^{1/2}$polylog(n) barrier for quantum LDPC codes. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1276–1288. ACM, 2021. arXiv:2009.03921, doi:10.1145/3406325.3451005. 2, 4, 5, 12, 15, 18, 20, 22, 23, 24, 25, 27, 29, 34, 35, 37

[JMT21] Fernando Granha Jeronimo, Tushant Mittal, and Madhur Tulsiani. Explicit Abelian Lifts and Quantum LDPC Codes. Personal communication, 2021.

[KT21] Tali Kaufman and Ran J. Tessler. New cosystolic expanders from tensors imply explicit quantum LDPC codes with $\Omega(\sqrt{n}\log^k n)$ distance. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1317–1329. ACM, 2021. arXiv:2008.09495, doi:10.1145/3406325.3451029. 4

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, September 1988. doi:10.1007/bf02126799.

[LTZ15] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zémor. Quantum Expander Codes. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science*, pages 810–824. IEEE, 2015. arXiv:1504.00822, doi:10.1109/FOCS.2015.55. 5

[MOP20] Sidhanth Mohanty, Ryan O'Donnell, and Pedro Paredes. Explicit near-ramanujan graphs of every degree. In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 510–523. ACM, 2020. arXiv:1909.06988, doi:10.1145/3357713.3384231.

[MRR+20] Jonathan Mosheiff, Nicolas Resch, Noga Ron-Zewi, Shashwat Silas, and Mary Wootters. LDPC codes achieve list decoding capacity. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 458–469. IEEE, 2020. arXiv:1909.06430, doi:10.1109/FOCS46700.2020.00050. 3

[PK21] Pavel Panteleev and Gleb Kalachev. Quantum LDPC Codes with Almost Linear Minimum Distance. *IEEE Transactions on Information Theory*, December 2021. arXiv:2012.04068, doi:10.1109/TIT.2021.3119384. 2, 4, 5, 11, 12, 18, 22, 23, 24, 27, 28

[Rao19] Shravas Rao. A Hoeffding inequality for Markov chains. *Electronic Communications in Probability*, 24:1 – 11, 2019. arXiv:1806.11519, doi:10.1214/19-ECP219.

[SS96] M. Sipser and D.A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996. doi:10.1109/18.556667. 4, 5

[Ste96] Andrew Steane. Multiple-particle interference and quantum error correction. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1954):2551–2577, Nov 1996. arXiv:quant-ph/9601029v3, doi:10.1098/rspa.1996.0136. 3, 8

[Tan81] R. Tanner. A recursive approach to low complexity codes. *IEEE Transactions on Information Theory*, 27(5):533–547, 1981. doi:10.1109/TIT.1981.1056404. 4

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *Quaderni di Matematica*, 13:347–424, 2004. arXiv:cs/0409044v1. 3

[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 Transactions on Information Theory*, 60(2):1193–1202, Feb 2014. URL: http://dx.doi.org/10.1109/TIT.2013.2292061, doi:10.1109/tit.2013.2292061. 4, 5, 10, 18, 23

[Var57] RR Varshamov. Estimate of the number of signals in error correcting codes. *Dokl. Akad. Nauk SSSR*, 117:739–741, 1957. 3

[Var97a] Alexander Vardy. Algorithmic complexity in coding theory and the minimum distance problem. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 92–109. ACM, 1997. doi:10.1145/258533.258559.

[Var97b] Alexander Vardy. The intractability of computing the minimum distance of a code. *IEEE Trans. Inf. Theory*, 43(6):1757–1766, 1997. doi:10.1109/18.641542.

[ZP19] Weilei Zeng and Leonid P. Pryadko. Higher-Dimensional Quantum Hypergraph-Product Codes with Finite Rates. *Physical Review Letters*, 122(23):230501, June 2019. arXiv:1810.01519, doi:10.1103/PhysRevLett.122.230501. 9, 22, 23