

# Understanding Quantumness And Testability.

David Ponarovsky

August 9, 2023



# Contents

<b>1</b>	<b>Introduction</b>	<b>7</b>
<b>2</b>	<b>Codes</b>	<b>9</b>
2.1	Introduction . . . . .	9
2.2	Codes in General. Notations and Definitions. . . . .	9
2.3	Singleton Bound . . . . .	10
2.4	Tanner Code . . . . .	11
2.5	Expander Codes . . . . .	13
2.6	Randomized Constructions. . . . .	14
2.7	Locally Testable Codes. . . . .	15
2.8	Polynomial Code. . . . .	15
<b>3</b>	<b>Quantum Error Correction Codes.</b>	<b>19</b>
3.1	Introduction. . . . .	19
3.2	Quantum Noise. . . . .	20
3.3	CSS Codes. . . . .	21
3.4	qLDPC Codes. . . . .	24
3.5	Quantum Expander Codes. . . . .	27
<b>4</b>	<b>Good qLDPC Codes and LTC.</b>	<b>29</b>
4.1	Quantum Tanner Codes. . . . .	29
4.2	Decoding and Testing . . . . .	33
<b>5</b>	<b>First Attempt For Good qLTC.</b>	<b>35</b>
5.1	The Polynomial-Code Is Not $w$ -Robust. . . . .	35
<b>6</b>	<b>Local Majority <math>\neq</math> Local Testability.</b>	<b>37</b>
6.0.1	Overcoming The Vanishing Rate. . . . .	39



# List of Figures

2.1	Peterson Graph. . . . .	12
2.2	The plot $x \mapsto (x - 1)(x - 2)$ and $x \mapsto (x - 1)(x - 4)$ presents the extension of the polynomials . . . . .	16
3.1	On the left is the Toric Graph. On the right are cross and face checks. . . .	24



## Chapter 1

# Introduction





# Chapter 2

## Codes

### 2.1 Introduction

Coding theory has emerged due to the need to transfer information in noisy communication channels. By embedding a message in a higher-dimensional space, one can guarantee robustness against possible faults. The ratio of the original content length to the transmitted message *length* is the *rate* of the code, and it measures how consuming our communication protocol is. Additionally, the *distance* of the code quantifies how many faults the scheme can absorb such that the receiver can recover the original message. We can consider the code as a collection of all strings that satisfy specified restrictions.

Non-formally, a code is good if its distance and rate scale linearly with the encoded message length. In practice, one is also interested in implementing these checks efficiently. We say that a code is an LDPC if any bit is involved in a constant number of restrictions, each of which is a linear equation, and if any restriction contains a fixed number of variables.

Moreover, another characteristic of the code is its testability, which is the complexity of the number of random checks one must do to verify that a given candidate is in the code. Besides being considered efficient in terms of robustness and overhead, good codes are also vital components in establishing secure multiparty computation [BGW19] and have a deep connection to probabilistic proofs.

In Section 2, we state the notations, definitions, and formal theorem. Then, in Sections 3 and 4, we review past results and provide their proofs to make this paper self-contained. Readers familiar with the basic concepts of LDPC, Tanner, and Expanders codes construction may consider skipping directly to Section 5, in which we provide our proof. Readers familiar with the basic concepts of LDPC, Tanner, and Expanders codes construction may skip Sections 2, 3, and 4 and proceed directly to Section 5, where we provide our proof.

### 2.2 Codes in General. Notations and Definitions.

Here we focus only on linear binary codes, which one could think about as linear subspaces of  $\mathbb{F}_2^n$ . A common way to measure resilience is to ask how many bits an evil entity needs to flip such that the corrupted vector will be closer to another vector in that space than the original one. Those ideas were formulated by Hamming [Ham50], who presented the following definitions.

**Definition 1.** Let  $n \in \mathbb{N}$  and  $\rho, \delta \in (0, 1)$ . We say that  $C$  is a **binary linear code** with parameters  $[n, \rho n, \delta n]$ . If  $C$  is a subspace of  $\mathbb{F}_2^n$ , and the dimension of  $C$  is at least  $\rho n$ . In

addition, we call the vectors belong to  $C$  codewords and define the distance of  $C$  to be the minimal number of different bits between any codewords pair of  $C$ .

From now on, we will use the term code to refer to linear binary codes, as we don't deal with any other types of codes. Also, even though it is customary to use the above parameters to analyze codes, we will use their percent forms called the relative distance and the rate of code, matching  $\delta$  and  $\rho$  correspondingly.

**Definition 2.** A **family of codes** is an infinite series of codes. Additionally, suppose the rates and relative distances converge into constant values  $\rho, \delta$ . In that case, we abuse the notation and call that family of codes a code with  $[n, \rho n, \delta n]$  for fixed  $\rho, \delta \in [0, 1)$ , and infinite integers  $n \in \mathbb{N}$ .

Notice that the above definition contains codes with parameters attending to zero. From a practical view, it means that either we send too many bits, more than a constant amount, on each bit in the original message. Or that for big enough  $n$ , adversarial, limited to changing only a constant fraction of the bits, could disrupt the transmission. That distinction raises the definition of good codes.

**Definition 3.** We will say that a family of codes is a **good code** if its parameters converge into positive values.

## 2.3 Singleton Bound

To get a feeling of the behavior of the distance-rate trade-of, Let us consider the following two codes; each demonstrates a different extreme case. First, define the repetition code  $C_r \subset \mathbb{F}_2^{n \cdot r}$ , In which, for a fixed integer  $r$ , any bit of the original string is duplicated  $r$  times. Second, consider the parity check code  $C_p \subset \mathbb{F}_2^{n+1}$ , in which its codewords are only the vectors with even parity. Let us analyze the repetition code. Clearly, any two  $n$ -bits different messages must have at least a single different bit. Therefore their corresponding encoded codewords have to differ in at least  $r$  bits. Hence, by scaling  $r$ , one could achieve a higher distance as he wishes. Sadly the rate of the code decays as  $n/nr = 1/r$ . In contrast, the parity check code adds only a single extra bit for the original message. Therefore scaling  $n$  gives a family which has a rate attends to  $\rho \rightarrow 1$ . However, flipping any two different bits of a valid codeword is conversing the parity and, as a result, leads to another valid codeword.

To summarize the above, we have that, using a simple construction, one could construct the codes  $[r, 1, r]$ ,  $[r, r - 1, 2]$ . Each has a single perfect parameter, while the other decays to the worst.

Besides being the first bound, Singleton bound demonstrates how one could get results by using relatively simple elementary arguments. It is also engaging to ask why the proof yields a bound that, empirically, seems far from being tight.

**Theorem** (Singleton Bound.). *For any linear code with parameter  $[n, k, d]$ , the following inequality holds:*

$$k + d \leq n + 1$$

*Proof.* Since any two codewords of  $C$  differ by at least  $d$  coordinates, we know that by ignoring the first  $d - 1$  coordinate of any vector, we obtain a new code with one-to-one corresponding to the original code. In other words, we have found a new code with the

same dimension embedded in  $\mathbb{F}_2^{n-d+1}$ . Combine the fact that dimension is, at most, the dimension of the container space, we get that:

$$\dim C = 2^k \leq 2^{n-d+1} \Rightarrow k + d \leq n + 1$$

□

It is also well known that the only binary codes that reach the bound are:  $[n, 1, n]$ ,  $[n, n-1, 2]$ ,  $[n, n, 1]$  [AF22]. In particular, there are no good binary codes that obtain equality (And no binary code which get close to the equality exists). Let's review the polynomial code family [RS60], which is a code over none binary field that achieve the Singleton Bound.

Next, we will review Tanner's construction, that in addition to being a critical element to our proof, also serves as an example of how one can construct a code with arbitrary length and positive rate.

## 2.4 Tanner Code

The constructions require two main ingredients: a graph  $\Gamma$ , and for simplicity, we will restrict ourselves to a  $\Delta$  regular graph, Yet notice that the following could be generalize straightforwardly for graphs with degree at most  $\Delta$ . The second ingredients is a ;small' code  $C_0$  at length equals the graph's regularity, namely  $C_0 = [\Delta, \rho\Delta, \delta\Delta]$ . We can think about any bit string at length  $\Delta$  as an assignment over the edges of the graph. Furthermore, for every vertex  $v \in \Gamma$ , we will call the bit string, which is set on its edges, the local view of  $v$ . Then we can define, [Tan81]:

**Definition 4.** Let  $C = \mathcal{T}(\Gamma, C_0)$  be all the codewords which, for any vertex  $v \in \Gamma$ , the local view of  $v$  is a codeword of  $C_0$ . We say that  $C$  is a **Tanner code** of  $\Gamma, C_0$ . Notice that if  $C_0$  is a binary linear code, So  $C$  is.

**Example 1.** Consider the Petersen graph  $\Gamma$ , which is a regular graph with degree 3. Let  $C_0$  be the set of all words with even parity. It follows that  $C_0$  contains all even-length binary strings of length 3: 000, 110, 101, and 011. However, the size of  $\mathcal{T}(\Gamma, C_0)$  is significantly larger, as shown in Figure fig. 2.1. Specifically, any rotation of the inner and outer cycles simultaneously gives rise to another valid codeword, so any assignments that are not invariant under these rotations would produce five additional valid codewords.



Figure 2.1: Peterson Graph.

**Lemma 1.** *Tanner codes have a rate of at least  $2\rho - 1$ .*

*Proof.* The dimension of the subspace is bounded by the dimension of the container minus the number of restrictions. So assuming non-degeneration of the small code restrictions, we have that any vertex count exactly  $(1 - \rho)\Delta$  restrictions. Hence,

$$\dim C \geq \frac{1}{2}n\Delta - (1 - \rho)\Delta n = \frac{1}{2}n\Delta(2\rho - 1)$$

Clearly, any small code with rate  $> \frac{1}{2}$  will yield a code with an asymptotically positive rate  $\square$

Based on Lemma 1, we can obtain a recipe for constructing codes with a almost non-vanishing rate for arbitrarily large lengths and dimensions. This recipe involves concatenating a series of Tanner codes over complete graphs. To be more precise, we can define a family of codes as follows:

$$C_{i+1} = \mathcal{T}(K_{n(C_i)+1}, C_i)$$

$$C_0 = \text{Some simple } \Delta[1, \rho_0, \delta_0] \text{ code.}$$

Where  $n(C_i)$  represents the code length of the  $i$ th code. Repeating the process described above  $\log_{\Delta}^*(n)$  times allows us to extend the initial code  $\Delta[1, \rho_0]$  to  $n[1, \sim 2\rho^{\log_{\Delta}^*(n)}]$ . Interestingly, any family of finite groups generated by a constant-size generator set can define a family of codes by utilizing their Cayley graphs as a basis for Tanner codes.

Once we have seen that Tanner codes enable us to achieve rates, the next natural question to ask is about the distance of the codes. Achieving a linear distance requires a little bit more from the graphs, but to understand this idea better, let us return to the repetition code. For instance, the repetition code can be presented as a Tanner code over the cycle graph.

**Example 2.** *In this representation, each vertex checks if the bits on its edges are equal. A valid codeword is an assignment in which all the bits are equal, since otherwise, there would be an edge with no supporting vertex. An illustration of a legal assignment is provided in ??.*

Recall that the distance of a linear code is the minimal weight of the non-zero codewords. Consider a codeword  $c \in C$  and group the vertices by four sets  $V_i$  such that  $V_i$  is the set of vertices that see  $i \in \{0, 1\}^2$ . Since  $c \in C$ , we have that  $|V_{10}| = |V_{01}| = 0$ . Additionally, any vertex in  $V_{00}$  is not connected to  $V_{11}$ , which gives us two possible cases: either all the vertices in  $V_{11}$  are isolated, or the graph is not connected. Hence, the distance of the code is equal to  $\frac{1}{2} \sum |V_i| \cdot |i| = \frac{1}{2} 2 \cdot n = n$ .

Analyzing the repetition code gives a clue of how, in certain cases, one might prove a lower bound on the code distance. We would like to say that, if the weight of the code word is below the distance, then it must be that is at least one vertex see non trivial local view which is not a codeword in  $C_0$ . Put it differently, we can't spread a small weight codeword over  $\{V_i\}$ , defined above, without to expanse into subsets corresponds to low  $|i|$ . Next we are going to present the Expander codes, which are Tanner codes constructed by graph with good algebraic expansion.

## 2.5 Expander Codes

We saw how a graph could give us arbitrarily long codes with a positive rate. We will show, Sipser's result [SS96] that if the graph is also an expander, we can guarantee a positive relative distance.

**Definition 5.** *Denote by  $\lambda$  the second eigenvalue of the adjacency matrix of the  $\Delta$ -regular graph. For our uses, it will be satisfied to define expander as a graph  $G = (V, E)$  such that for any two subsets of vertices  $T, S \subset V$ , the number of edges between  $S$  and  $T$  is at most:*

$$|E(S, T) - \frac{\Delta}{n} |S| |T| | \leq \lambda \sqrt{|S| |T|}$$

This bound is known as the Expander Mixing Lemma. We refer the reader to [HLW06] for more detailed survey.

**Theorem.** *Theorem, let  $C$  be the Tanner Code defined by the small code  $C_0 = [\Delta, \delta\Delta, \rho\Delta]$  such that  $\rho \geq \frac{1}{2}$  and the expander graph  $G$  such that  $\delta\Delta \geq \lambda$ .  $C$  is a good LDPC code.*

*Proof.* We have already shown that the graph has a positive rate due to the Tanner construction. So it's left to show also the code has a linear distance. Fix a codeword  $x \in C$  and denote by  $S$  the support of  $x$  over the edges. Namely, a vertex  $v \in V$  belongs to  $S$  if it connects to nonzero edges regarding the assignment by  $x$ . Assume towards contradiction

that  $|x| = o(n)$ . And notice that  $|S|$  is at most  $2|x|$ , Then by The Expander Mixing Lemma we have that:

$$\begin{aligned} \frac{E(S, S)}{|S|} &\leq \frac{\Delta}{n}|S| + \lambda \\ &\leq_{n \rightarrow \infty} o(1) + \lambda \end{aligned}$$

Namely, for any such sublinear weight string,  $x$ , the average of nontrivial edges for the vertex is less than  $\lambda$ . So there must be at least one vertex  $v \in S$  that, on his local view, sets a string at a weight less than  $\lambda$ . By the definition of  $S$ , this string cannot be trivial. Combining the fact that any nontrivial codeword of the  $C_0$  is at weight at least  $\delta\Delta$ , we get a contradiction to the assumption that  $v$  is satisfied, videlicet,  $x$  can't be a codeword  $\square$

**Setting  $C_0$  To Be The Polynomial Code.**

$$\begin{aligned} \log \Delta \dim C &\geq \frac{1}{2}n\Delta - (1 - \rho)\Delta n = \frac{1}{2}n\Delta(2\rho - 1) \\ \Rightarrow \dim C &\geq \frac{1}{\log \Delta} \frac{1}{2}n\Delta(2\rho - 1) \end{aligned}$$

## 2.6 Randomized Constructions.

**Claim 1.** *Let  $A$  be a random matrix in  $M(\mathbb{F}_2^{k \times n})$ . For any non-zero  $x \in \mathbb{F}$ , we have that  $Ax$  is distributed uniformly.*

*Proof.* By the fact that  $x \neq 0$ , there exists at least one coordinate  $i \in [k]$  such that  $x_i \neq 0$ . Thus, we have

$$\begin{aligned} (Ax)_j &= \sum_k A_{jk}x_k = \sum_{i \neq k} A_{jk}x_k + A_{ji}x_i \\ &= \sum_{i \neq k} A_{jk}x_k + A_{ji} \end{aligned}$$

Notice that due to the fact that  $\mathbb{F}_2$  is a field, there is exactly one assignment that satisfies the equation conditioned on all the values  $A_{jk}$  where  $j \neq k$ .

$$\begin{aligned} \Pr[(Ax)_j = 1] &= \sum_{A_{jk}; k \neq i} \Pr[(Ax)_j = 1 \mid A_{jk}; k \neq i] \Pr[A_{jk}; k \neq i] \\ &= \frac{1}{2} \end{aligned}$$

Therefore, any coordinate of  $Ax$  is distributed uniformly  $\Rightarrow Ax$  is distributed uniformly.  $\square$

By the uniformity of  $Ax$ , we obtain that the expected Hamming weight of  $Ax$  is:

$$\mathbf{E}[|Ax|] = \mathbf{E}\left[\sum_i^n (Ax)_i\right] = \frac{1}{2}n$$

As the coordinates of  $A_x$  are independent (each row of  $A$  is sampled separately), we can use the Hoff's bound to conclude that:

$$\Pr\left[||Ax| - \mathbf{E}[|Ax|]| \geq \left(\frac{1}{2} - \delta\right)n\right] \leq e^{-n(\frac{1}{2} - \delta)^2}$$

Now, we will use the union bound to show that any  $x \in \mathbb{F}_2^k$ ,  $Ax$  is of weight at least  $\delta$ .

$$\Pr \left[ |Ax| \geq \delta : \forall x \in \mathbb{F}_2^k \right] \geq 1 - |\mathbb{F}_2^k| \cdot e^{-n(\frac{1}{2}-\delta)^2}$$

Denote  $k = \rho n$  and notice that the above holds when  $\rho \geq (\frac{1}{2} - \delta)^2$ .

## 2.7 Locally Testable Codes.

In addition to distance and rate, we are also interested in ensuring that the checking process is robust. Specifically, we want to guarantee that a single missed check will only have a tiny probability of compromising the entire computation, even in the presence of significant errors.

**Definition 6.** Consider a code  $C$  a string  $x$ , and denote by  $\xi(x)$  the fraction of the checks in which  $x$  fails.  $C$  will be called a **local-testable**  $f(n)$  if there exists  $\kappa > 0$  such that

$$\frac{d(x, C)}{n} \leq \kappa \cdot \xi(x) f(n)$$

Next, we will introduce the polynomial code, a testable code. We will then demonstrate a proof that emphasizes the concept of restriction degeneration, meaning that if more than a certain number of restrictions are satisfied, then almost all the remaining restrictions must also be satisfied. It is also worth mentioning that the polynomial code has a nice quantum variant which we will present in the next chapter.

## 2.8 Polynomial Code.

Consider the field  $\mathbb{F}_q$  for an arbitrary prime power  $q$  greater than  $n$ . The polynomial codes rely on the fact that any two different polynomials in the ring  $\mathbb{F}_q[x]$  of degree at most  $d$  differ by at least  $q - d + 1$  points. For example, consider a pair of polynomials of degree 1, namely two linear straight lines. If they are not identical, then they have at most one intersection point, and they disagree on each of the  $n - 1$  remaining points. By defining the code to be the subspace containing all polynomials of degree at most  $d$ , such that any codeword is an image of such a polynomial encoded by  $n$  numbers, we can guarantee a lower bound on the code's distance. Formally, we define:

**Definition 7** (Polynomial Code. [RS60]). Fix  $m > n$  to be a prime power and let  $a_0, a_1, a_2, \dots, a_n$  distinct points of the field  $\mathbb{F}_q$  and define the code  $C$  as follows:

$$C = \{p(a_0), p(a_1), p(a_2), \dots, p(a_n) : p \text{ is polynomial at degree at most } d\}$$

Observe that  $C$  is a linear code of length  $n$  over the alphabet  $\mathbb{F}_q$ . The following Lemma states the relation between the maximal degree of the polynomials and the properties of the code.

**Lemma 2.** Fix the degree of the polynomial code to be at most  $d$ . Then the parameters of the code are  $[n, d + 1, n - d]$ .

*Proof.* The dimension of the code is equal to the dimension of the polynomial space of degree at most  $d$ , which is spanned by the monomial basis  $e_0, e_1, e_2, \dots, e_d = 1, x, \dots, x^d$ , and is therefore  $d + 1$ . Furthermore, suppose that  $f$  and  $g$  are two different polynomials, i.e.  $f \neq g$ . Thus,  $h = f - g$  is a non-zero polynomial of degree at most  $d$ , and therefore has at most  $d$  roots. This implies that there are at most  $d$  points in which  $f$  and  $g$  are equal, and at least  $n - d$  points in which they disagree. In other words, the distance between any two different codewords of the code is at least  $n - d$ .  $\square$



Figure 2.2: The plot  $x \mapsto (x - 1)(x - 2)$  and  $x \mapsto (x - 1)(x - 4)$  presents the extension of the polynomials

Now we are about to prove that the polynomial code is a testable code. The next claim will enable us to design a random test to certify polynomials.

**Claim 2.** Consider the finite field  $\mathbb{F}_q$  for prime  $q$ , denote by  $w$  the  $d$  root of 1, namely  $w^d = 1$  then for any polynomial  $f$  with degree at most  $d$  it holds:

$$f(x) = \frac{1}{d} \sum_t^d f(x + w^t y) \quad \forall x, y \in \mathbb{F}_q \quad (2.1)$$

*Proof.* Denote by  $a_j$  the  $j$ th coefficient of  $f$ . And recall that for any fixed  $l \in \mathbb{F}_q$  the summation  $\sum_t^d w^{t \cdot l}$  equals:

$$\sum_t^d w^{t \cdot l} = \begin{cases} \frac{w^{l \cdot d} - 1}{w^l - 1} = 0 & l \neq 0 \\ d & l = 0 \end{cases}$$

Thus:

$$\begin{aligned} \sum_t f(x + w^t y) &= \sum_{t,j} a_j (x + w^t y)^j = \sum_{t,j} a_j \left( \sum_l x^{j-l} w^{t \cdot l} y^l \binom{j}{l} \right) \\ &= \sum_j a_j \sum_l x^{j-l} y^l \binom{j}{l} \sum_t w^{t \cdot l} \\ &= \sum_j a_j x^j \cdot d = f(x) \cdot d \end{aligned}$$

$\square$



**Example 3.** For example consider a linear function over  $\mathbb{F}_q$ , then it  $w^2 = 1 \Rightarrow w = -1$  and it clears that  $f(x) = \frac{1}{2} (f(x-y) + f(x+y))$ .

Hence, we can consider the following tester: randomly select  $x, y \in \mathbb{F}_q$  and accept if eq. (2.1) is satisfied for a given function  $f$ .

**Claim 3.** Suppose that  $f : \mathbb{F}_q \rightarrow \mathbb{F}_q$  satisfies eq. (2.1) then, either  $f$  is polynomial at degree at most  $d$  or that  $f$  is the zero function.

*Proof.* Assume by contradiction that  $f$  has more than  $d+1$  zeros (w.l.g). Denote them by  $x_1, x_2, \dots, x_{d+1}$ . We will show that for any  $x \in \mathbb{F}_q$  one can find  $y \in \mathbb{F}_q$  such that  $x + w^t y = x_t$ . Those equations compose a linear diophantine system which can be reduced to the following system:

$$\begin{bmatrix} w^0 & 0 & \dots & 0 & 0 \\ w^1 & 1 & \dots & 0 & 0 \\ \vdots & \ddots & \ddots & & \\ w^{d-1} & 0 & \dots & 1 & 0 \\ w^d & 0 & \dots & 0 & 1 \end{bmatrix} \begin{bmatrix} y \\ k_2 \\ k_3 \\ \vdots \\ k_d \end{bmatrix} = \begin{bmatrix} x_1 - x \\ x_2 - x \\ \vdots \\ x_d - x \\ x_{d+1} - x \end{bmatrix}$$

The determinant of the above matrix is  $w \neq 0$  and therefore there exists a  $y$  satisfies the equations. Hence for any  $x$  we have  $f(x) = \frac{1}{d} (f(x_1) + f(x_2) + \dots + f(x_{d+1})) = 0$ .  $\square$

Consider the following decoding process. Given a candidate  $f : \mathbb{F}_q \rightarrow \mathbb{F}_q$ , we will set for any point  $x \in \mathbb{F}_q$  the plurality of eq. (2.1) over the  $y \in \mathbb{F}_q$ . Denote the output by  $g$ , and we have that:

$$g(x) = \arg_{z \in \mathbb{F}_q} \max \Pr_y \left[ \sum_i^d f(x + w^i y) = z \right]$$

**Claim 4.** If  $f$  pass the test with probability  $\geq 1 - \varepsilon$  then the relative distance between  $f$  and  $g$  is at most  $2\varepsilon$ .

*Proof.* Notice that for any  $x$  for which there exists  $z$  such  $\Pr_y \left[ \sum_i^d f(x + w^i y) = z \right] \geq \frac{1}{2}$  it must hold that  $g(x) = f(x)$ . Denote by  $A$  that set and by  $\alpha$  the probability to hit such  $x$  (namely  $|A|/|\mathbb{F}_q|$ ). The probability to pass the test can be bounded by:

$$\begin{aligned} 1 - \varepsilon &\leq \Pr[f \text{ pass}] = \Pr[f \text{ pass} \cap x \in A] + \Pr[f \text{ pass} \cap x \notin A] \\ &\leq 1 \cdot \alpha + \frac{1}{2} \cdot (1 - \alpha) \\ &\Rightarrow \alpha \geq 1 - 2\varepsilon \end{aligned}$$

Thus, at most  $2\varepsilon$  of the points do not agree with  $g$ .  $\square$

**Claim 5.** Let  $f$  be a function that pass the test with probability  $1 - \varepsilon$  and fix  $x \in \mathbb{F}$  then:

$$\Pr_{u,v} \left[ \sum_i^d f(x + w^i u) = \sum_i^d f(x + w^i v) \right] \geq 1 - 2\varepsilon \cdot d$$

*Proof.* Recall that for any  $i$  the random variable  $x + w^i u$  is distributed uniformly random regardless  $x$ . Therefore we have that for any  $i$  the following equation holds with probability at least  $1 - \varepsilon$ :

$$f(x + w^i u) = \sum_j^d f(x + w^i u + w^j v)$$

Thus, by the union bound, those equations satisfied simultaneously for all the  $i$  with probability  $1 - d \cdot \varepsilon$ . By repeating on the same arguments but swapping  $u$  and  $v$  we have that

$$f(x + w^i v) = \sum_j^d f(x + w^i v + w^j u)$$

also happens with probability at least  $1 - d \cdot \varepsilon$  and therefore, again by the union bound, both of the equations sets hold with probability at least  $1 - 2d \cdot \varepsilon$ . But that is exactly the event in which:

$$\sum_i^d f(x + w^i u) = \sum_i^d \sum_j^d f(x + w^i u + w^j v) = \sum_j^d f(x + w^j v)$$

□

**Claim 6.** For any  $x$  the probability over  $v$  that  $g(x) = \sum_i f(x + w^i v)$  is greater than  $1 - \Theta(d \cdot \varepsilon)$ .

*Proof.* Fix arbitrary  $x$  and denote by  $\alpha$  the probability to hit  $v$  such that  $g(x) = \sum_i^d f(x + w^i v)$  and by  $A$  the set of  $v$ 's satisfy the relation. Notice that the probability to pick a pair  $v, u$  such that  $\sum_i^d f(x + w^i v) = \sum_i^d f(x + w^i u)$  is lower than the probability that either  $v$  and  $u$  are both belong to  $A$  or they both belong to his complement. Namely:  $\alpha^2 + (1 - \alpha)^2 \geq 1 - 2\varepsilon \cdot d$ . Hence we get  $\alpha \geq \frac{1}{2} (1 + \sqrt{1 - 4\varepsilon d}) = 1 - \Theta(\varepsilon \cdot d)$ . □

**Claim 7.** For  $\varepsilon = \Theta(1/d^2)$   $g$  is a polynomial at degree at most  $d$ .

*Proof.* Fix  $x, y \in \mathbb{F}_q$ . From claim 6 with probability at least  $1 - \Theta(d \cdot \varepsilon)$  it holds that  $g(x + w^j y) = \sum_i^d f(x + w^i v + w^j y)$ . Moreover, as  $v + w^j y$  distributed uniformly for any  $j$ , it follows that one can fix  $j$  and has with probability at least  $1 - \Theta(\varepsilon d)$  that  $g(x) = \sum_i^d (f(x + w^i (v + w^j y)))$ . So, the relation holds for all the  $j$ 's with probability  $1 - \Theta(\varepsilon d^2)$  combining all together obtains:

$$g(x) = \sum_{i,j}^d f(x + w^i (v + w^j y)) = \sum_j^d \sum_i^d f(x + w^i v + w^j y) = \sum_j^d g(x + w^j y)$$

With probability  $1 - \Theta(\varepsilon \cdot d^2)$ . However, the probability is about sampling  $v$  such those transimitions are valid. But as the value of  $g$  is independent on the choice of  $v$  it's enough to have just a single  $v$  to guarantee  $g(x) = \sum_j^d g(x + w^j y)$  namely that the probability is positive. So if  $\varepsilon = \Theta(1/d^2)$  we have that for any  $x, y$  there exists  $v$  such section 2.8 is satisfied and therefore  $g$  is satisfies eq. (2.1) for any  $x, y$ . □

## Chapter 3

# Quantum Error Correction Codes.

### 3.1 Introduction.

It is widely believed that quantum machines have a significant advantage over classical machines in a wide range of computational tasks [Gro96], [AK99]. Simple algorithms which can be interpreted as the quantum version of scanning all the options, reducing the running time by the square root of the classical magnitude. Nevertheless, Shor has demonstrated a polynomial-depth quantum circuit that solves the hidden abelian subgroup problem [Sho97], which is considered a breakthrough, as it made the computer science community believe that a quantum computer could offer an exponential advantage.

Despite a consensus on the superiority of the ideal quantum computing model, it is still uncertain whether it is possible to implement such a machine in a noisy environment. Still, simply pointing out the existence of noise is not sufficient to negate the feasibility of computation. Evidence of this is that classical computers also experience a certain rate of errors. Therefore, to fully comprehend the difficulty, let us compare two main factors that made it a challenging task.

First is the magnitude of the error rate; classical computers also have errors, and sometimes we witness system failures (e.g. the blue screen). The error rate of modern computers is so low that the probability of errors propagating stays negligible, even if the length of the computation is polynomial in the scale of what is considered a reasonable input size. It's worth mentioning that in exascale computing, when supercomputers perform around  $10^{18}$  operations per second, it is difficult to overlook faults. In quantum computing, we become aware of their existence much earlier.

The second difference, which is a subtle point, is that quantum states are susceptible to additional types of errors. In addition to the possibility of bit-flip errors, a quantum state may also experience a change in phase. For example, consider the initial state  $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ , and suppose that due to noise the state is transformed into  $\frac{1}{\sqrt{4}}(\sqrt{3}|0\rangle + |1\rangle)$ . Classical circuits are oblivious to such faults, meaning that their operation would remain unchanged as no error has occurred. Quantum circuits, however, are usually affected and may fail. Furthermore, when designing a decoder for quantum error correction codes, one must ensure that the decoding process does not introduce bit-flip errors if a classical code is used to protect against phase flips.

### 3.2 Quantum Noise.

**Definition 8** (Bit and phase flip.). *Consider a quantum state  $|\psi\rangle$  encoded in the computation base. We will say that a bit-flip occurs in a scenario the operator Pauli  $X$  is applied on one of our state's qubits. The bit-flip event could be considered as exactly as the standard bit-flip error in the classical regime. Similarly, phase-flip occurs when the Pauli  $Z$  is applied on one of the qubits.*

However, even though quantum noise is so violent, it has been proven that any ideal circuit of polynomial depth can be transformed into a robust circuit at poly-logarithmic cost [AB99]. In other words, there is a threshold: if physicists provide qubits and a finite gate set with a noise rate below that threshold, then  $BQP$ , the class of polynomial-time ideal quantum computation, is feasible and can be computed on a realistic machine.

The basic idea in [AB99] was to show the existence of quantum error correction codes, which would enable logic operations to be performed in a way that prevents errors from propagating. This process involves separating the operation into two stages: the operation itself and an error correction stage. This comes with an additional cost in terms of space and time, but it can reduce the probability of the final state being faulty. The trade-off between the resources needed and the rate of decrease defines the threshold. If the balance is positive, then the process can be repeated in a recursive manner, and after log-log iterations, the failure probability will decay to zero. At the same time, the circuit will scale to a maximum of poly-logarithmic width and depth.

Let's return to the repetition code presented in Chapter 2. We would like to have an analog; a first and natural attempt might consider duplicating copies of the state. Unfortunately, copying a general state is not a linear operation and therefore cannot be done in the circuit model (or any other believed to be feasible). In particular, there is no circuit  $U$  which can simultaneously duplicate the states  $|0\rangle, |1\rangle, |+\rangle, |-\rangle$ .

To overcome the issue, Shor came up with the nine-qubit code [Sho95], which at first glance might seem a naive straightforward implementation of "duplication", but instead uses a clever insight about quantumness in general. Any operation can be seen as a linear (and even unitary) operation over a subspace embedded in a large enough dimension. The encoding is given as follows:

$$\begin{aligned} |\bar{0}\rangle &= \frac{1}{2\sqrt{2}} (|000\rangle + |111\rangle)^{\otimes 3} \\ |\bar{1}\rangle &= \frac{1}{2\sqrt{2}} (|000\rangle - |111\rangle)^{\otimes 3} . \end{aligned}$$

For convenience, let us use the notation  $|\mathbf{GHZ}^\pm\rangle = |0^m\rangle \pm |1^m\rangle$ . We can also consider the Shor code over  $m^2$  qubits, which is defined as above, such that any logical state contains  $m$  products over  $m$  qubits. Therefore, the state  $|\bar{0}\rangle$  over  $m^2$  qubits can be written as  $|\mathbf{GHZ}^+\rangle^m$ . We are now ready to prove a statement regarding the robustness.

**Lemma 3.** *The Shor code over 9 qubits enable to correct a single either bit or phase flip.*

It is evident that a single bit-flip error can be handled in the same way as in the conventional case. The decoder will check if any of the triples have the same value, and if not, it will correct it by majority. To create a decoder that can also correct a phase-flip error, we need the following statement. In this chapter, we denote the Hadamard gate over  $m$  qubits as  $H^m$ .

**Claim 8.**  $H^m |\mathbf{GHZ}^\pm\rangle = \sum_{x \cdot \mathbf{1} = 2^\pm} |x\rangle$

*Proof.*

$$\begin{aligned} H^m |\mathbf{GHZ}^\pm\rangle &= H^m |0^m\rangle \pm H^m |1^m\rangle = \sum_{x \in \mathbb{F}_2^m} |x\rangle \pm \sum_{x \in \mathbb{F}_2^m} (-1)^{x \cdot \mathbf{1}} |x\rangle \\ &= \sum_{x \in \mathbb{F}_2^m} \left(1 \pm (-1)^{x \cdot \mathbf{1}}\right) |x\rangle = \sum_{x \cdot \mathbf{1} = 2^\pm} |x\rangle \end{aligned}$$

□

Now it is clear how to correct a phase flip. One can apply the Hadamard transform and compute the parity of each triple. By the assumption that only a single phase flip may occur, either all the triples have the same parity or the faulted one has an opposite parity and needs to be corrected. Thus, we obtain an  $[[9, 1, 3]]$  quantum error correction code. Asymptotically, this is an  $[[m^2, 1, m]]$  code.

### 3.3 CSS Codes.

The Shor code is a specific case of the more general CSS (Calderbank-Shor-Steane) code [CS96]. A family composed by two binary codes  $C_X, C_Z$  such that  $C_Z^\perp \subset C_X$ .

**Definition 9** (CSS Code). *Let  $C_X, C_Z$  classical linear codes such that  $C_Z^\perp \subset C_X$  define the  $Q(C_X, C_Z)$  to be all the code words with following structure:*

$$|\mathbf{x}\rangle := |x + C_Z^\perp\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} |x + z\rangle$$

Clearly, the codewords are all the codewords in  $C_X$  which don't belong to  $C_Z^\perp$  and therefore the dimension of the quantum code is  $\dim Q(C_X, C_Z) = \dim C_X - \dim C_Z^\perp = \dim C_X + \dim C_Z - n$ . Yet, it's not stems immediately how one can correct faults. Next, we are going to repeat the decoding process of the Shor code in the general setting of CSS codes.

**Lemma 4.** *Let  $C_X, C_Z$  classical codes such  $Q(C_X, C_Z)$  is a CSS code. Let  $d_X$  be the minimal weigh of codeword in  $C_X$  which is not in  $C_Z^\perp$ , and define by the same way  $d_Z$  to be the minimal weight of codeword in  $C_Z$  which doesn't belong to  $C_X^\perp$ . Then the distance of  $Q(C_X, C_Z)$  equals to  $\min d_X, d_Z$ . Moreover there is a decoder which correct any fault with weight at most  $d/2$ .*

*Proof.* First let us prove the following claim:

**Claim 9.** *Denote by  $H^{\otimes n}$  the Hadamard gate over  $n$  qubits. Then for any code  $C$  it holds that:  $H^n |C^\perp\rangle = |C\rangle$*

*Proof.*

$$\begin{aligned} H^n |C^\perp\rangle &= \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} H^n |z\rangle = \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \sum_{y \in \mathbb{F}_2^n} (-1)^{\langle z, y \rangle} |y\rangle \\ &= \frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \left( \sum_{y \in C_Z} |y\rangle + \text{other terms} \right) \end{aligned} \tag{3.1}$$

Since the columns of matrix  $H_Z$  form a basis for the complementary space  $C_Z^\perp$ , and due to the dimensional theorem and the equivalence between the row rank and column rank of a matrix, we can deduce that  $\dim \text{rank } H_Z^\top + \dim \ker H_Z = n$ , which implies that  $|C_Z^\perp| |C_Z| = 2^n$ . Thus the norm of

$$\frac{1}{\sqrt{2^n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} \sum_{y \in C_Z} |y\rangle = \sqrt{\frac{|C_Z^\perp|}{2^n}} \sum_{y \in C_Z} |y\rangle \quad (3.2)$$

equals 1 and the summation over the vectors  $y \notin C_Z$  in the inner closure in equation 3.1 must cancel. So we left only with a uniform superposition over the codewords in  $C_Z$ . Or in other words  $H^n |C^\perp\rangle = |C\rangle$ .  $\square$

Claim 9 states that, when considering CSS codes, pauli  $X$  operators can be seen as pauli  $Z$  operators in the rotated frame. That it,

$$H^n X^f |C_Z^\perp\rangle = \overbrace{H^n X^f H^n}^{Z^f} H^n |C_Z^\perp\rangle = Z^f |C\rangle$$

That insight hints a description to decoder for the quantum code. If one knows how to correct errors for each of the classical code  $C_X, C_Z$  than he can start by correct the bit flips in using the decoder of  $C_X$ , rotate the state by applying the Hadamard transform and then correct, what was before the transformation phase flips and now a bit flips by using the decoder of  $C_Z$ . Then in the end applying the Hadamard transform again for backing to the initial computation space. Indeed that decoder correct  $\min\{d(C_X)/2, d(C_Z)/2\}$  errors. Yet more work is needed to show that this decoder also correct  $d(Q(C_X, C_Z))/2$  errors.

Let us assume the existences of decoders of the classical codes  $C_X$  and  $C_Z$ , Denoted  $D_X : \mathbb{F}_2^n \rightarrow C_X$  and  $D_Z$ . In particular for any  $\xi \in \mathbb{F}_2^n$ ,  $D_X(\xi) = \arg \min_{x \in C_X} |x + \xi|$ .

**Claim 10.** For any  $x_0 \in C_X$  and  $z_1 \neq z_2 \in C_Z^\perp$ ,  $D_X$  correct  $|x + z_1 + f\rangle, |x + z_2 + f\rangle$  into two different words in  $C_X$ .

*Proof.* Suppose not, namely there exists  $y \in C_X$  such that  $D_X$  correct  $|x + z_1 + f\rangle, |x + z_2 + f\rangle$  into  $|y\rangle$ . Then we have that for both  $i \in \{1, 2\}$  it holds that  $d(x + z_i + f, y) \leq d(C_Z^\perp/2)$  and therefore  $d(x + z_1 + f, x + z_2 + f) \leq d(C_Z^\perp)$ . But

$$\begin{aligned} d(x + z_1 + f, x + z_2 + f) &= |x + z_1 + f - x - z_2 - f| \\ &= |z_1 - z_2| = d(z_1, z_2) \end{aligned}$$

contradiction for the assumption that  $z_1, z_2 \in C_Z^\perp$ .  $\square$

We are ready to show step by step the decoding process. Let  $P = X^f Z^e$  be an error such that  $e, f < d/2$  act on the state  $|\mathbf{x}\rangle$ . Denote by  $H_X, H_Z$  the parity check matrices of  $C_X, C_Z$ . Using the commute relation  $[X^f, Z^e] = (-1)^{\langle e, f \rangle}$  we have that:

$$|\mathbf{x}\rangle \mapsto^P \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^f Z^e |x + z\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle$$

Now the decoder computes and stores the syndrome relative to the bits code using the parity check matrix  $H_X$ . And apply the inverse gate.

$$\begin{aligned}
& \mapsto^{H_X} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle |H_X(x + z + f)\rangle \\
& = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} (-1)^{\langle e, f \rangle} Z^e |x + z + f\rangle |H_X f\rangle \\
& \mapsto^{X^f} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} Z^e |x + z\rangle |H_X f\rangle
\end{aligned}$$

Then rotating into the phases base:

$$\begin{aligned}
& \mapsto^{H^{\otimes n}} \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e H^{\otimes n} |x + z\rangle \\
& = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e H^{\otimes n} X^x |z\rangle = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} X^e Z^x H^{\otimes n} |z\rangle \\
& = \frac{1}{\sqrt{|C_Z^\perp|}} X^e Z^x H^{\otimes n} \sum_{z \in C_Z^\perp} |z\rangle = \frac{1}{\sqrt{|C_Z|}} X^e Z^x \sum_{z \in C_Z} |z\rangle \\
& = \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} X^e |z\rangle = \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} |z + e\rangle
\end{aligned}$$

So now we have back into the begging. Only now the phase flips are playing the role of bit flips relative to the code  $C_Z$ .

$$\begin{aligned}
& = \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} |z + e\rangle |H_Z e\rangle \\
& \mapsto \frac{1}{\sqrt{|C_Z|}} (-1)^{\langle x, e \rangle} Z^x \sum_{z \in C_Z} |z\rangle |H_Z e\rangle \mapsto^{H^n} \frac{1}{\sqrt{|C_Z^\perp|}} X^x \sum_{z \in C_Z^\perp} |z\rangle \\
& = \frac{1}{\sqrt{|C_Z^\perp|}} \sum_{z \in C_Z^\perp} |z + x\rangle = |\mathbf{x}\rangle
\end{aligned}$$

□

There is still one big difference between the classic repetition code and the Shor code. While each parity check of the Shor code examines a square root number of qubits, any check of the repetition code touches no more than a constant number of qubits; that is, any check just tests if any two adjacent bits are equal. That brings us to ask whether the Shors code is really the quantum analogy for the repetition code?

For getting an hint before formally presenting a quantum LDPC code, let's take another look on the general structure of the CSS codes. The decoding procedure the proof above teach us an additional point about CSS code, the task of finding a good code quantum code, could be reduce for finding a two classic binary linear codes which their parity check matrices ortogonal to each other. Furthermore, if one is willing to has an qLDPC code,

then  $H_X$  and  $H_Z$  can't be parity check matrices of good classical code as any column of  $H_Z^\top$  is a codeword of  $C_X$ .

$$C_Z^\perp \subset C_X \Rightarrow H_X H_Z^\top = 0$$

And by being an LDPC code, the rows weights of  $H_Z$  is bounded by constant. Therefore there is a codeword  $\in C_X$  which is also a row of  $H_Z$  that has a constant weight.

### 3.4 qLDPC Codes.

As exactly as in the classic case, qLDPC codes are codes in which any check act non trivially on at most a constant number of qubits, It was proved that using a good Quantum LDPC code one can achieve a fault tolerance threshold theorem at the cost of only constant overhead<sup>1</sup> [Got14]. We are now about to embark on a detailed review of the first quantum LDPC code [Den+02].

Recall that one way to present a code is by define the parity check matrix, Consider the  $l \times l$  Tours, namely the Cayley graph of the group product  $\mathbb{Z}_l \times \mathbb{Z}_l$ . Associate any coordinate (bit/qubit) with an edge on the Tours. And consider the following two restrictions:

1. Each vertex requires from its local view, the bits lay on its supported edges, To has an even parity. We will refer to this type of check as *cross check*.
2. Similarly, each face requires the same from its supported edges, but computes the parity in a different (specific) base. That it, the face first rotates the qubits by applying the Hadamard transform on them, and then computes their XOR. Finally, the qubits are rotated back to the computation base. We shall refer to this type of check as *face check*.



Figure 3.1: On the left is the Toric Graph. On the right are cross and face checks.

<sup>1</sup>under the assumption of holding an efficient decoder.



For example consider some vertex  $v$  on the Torus, and let  $|\psi\rangle = \sum_x |\cdots x_{e_0} x_{e_1} x_{e_2} x_{e_3} \cdots\rangle$  when  $e_0, e_1, e_2, e_3$  are the edges compose the local view of  $v$ . Then in any ket can be in the support of  $|\psi\rangle$  only if the parity of  $e_0, e_1, e_2, e_3$  is even.

**Claim 11.** *The  $l \times l$  Toric code is a CSS code, with dimension 2 and distance  $\Theta(l)$ .*

*Proof.* Consider a pair of cross and a face checks. If they are not intersect (share edges) then, obviously, they commute. So suppose they share an edge. Now there are a finite number of cases (4) in which a cross check can intersect a face, and for all of them we have they must to intersect in an exactly two edges. Therefore the crosses checks commute with all the faces checks.

Now, denote by  $C_X$  and  $C_Z$  the linear codes defined by the crosses and faces checks. Observes that  $C_X$  contain all the subgraph of the tours which have only evens degree, namely all the loops. The following claim will be used to show that all the low weight loops (squeezeable loops) are in  $C_Z^\perp$ .

**Claim 12.** *Let  $c$  be an assignment of ones on a unite square, namely a closed loop at length 4, then  $c \in C_Z^\perp$ .*

*Proof.* Denote by  $c'$  a codeword of  $C_Z$ , therefore the parity sum induced by  $c'$  on any square equals zero, Particularly the induced parity on the square supporting  $c$  is zero. But that parity is exactly  $c \cdot c'$ . As it true for any  $c' \in C_Z$  we obtain that  $c \in C_Z^\perp$ .  $\square$

**Claim 13** (Veblen's theorem). *The set of even subgraph is a linear space spanned by simple cycles (vertices degree equals exactly 2).*

*Proof.* Let  $P \subset E$  be an even subgraph then it must to have a simple cycle denote it by  $P'$ , now notice that  $P/P'$  is also an even subgraph. To see that consider a vertex  $v$ , As  $P'$  is simple cycle, substituting  $P'$  is either not effect the degree of  $v$  in  $P/P'$  or it decrees  $v$  degree by exactly 2 so  $d_{P/P'}(v) = d_P(v) - 2$ . In both cases  $d_{P/P'}(v)$  is even. By repeating recursively until  $P/P' = \{\emptyset\}$  we get a decomposition of  $P$  into a sum of simple cycles.  $\square$

**Claim 14.** *Associate with any vertex of the Torus a coordinate in  $\mathbb{Z}_l \times \mathbb{Z}_l$ . Consider a simple cycle  $P$  subset of the Torus. Denote  $P$  by the vertices composing it  $v_0 v_1 \dots v_k$  arranged in order. Consider a vertex  $v_i \in P$  such that  $\{v_{i-1}, v_i\}, \{v_i, v_{i+1}\}$  are in the same direction and denote  $v = (x, y) \in \mathbb{Z}_l \times \mathbb{Z}_l$ . Then there exists a vertex  $u \in P$ ,  $u \neq v_{i-1} v_{i+1}$  that shares one of the coordinates of  $v$ . Put differently, there exists  $z$  such that either  $u = (z, y)$  or  $u = (x, z)$ .*

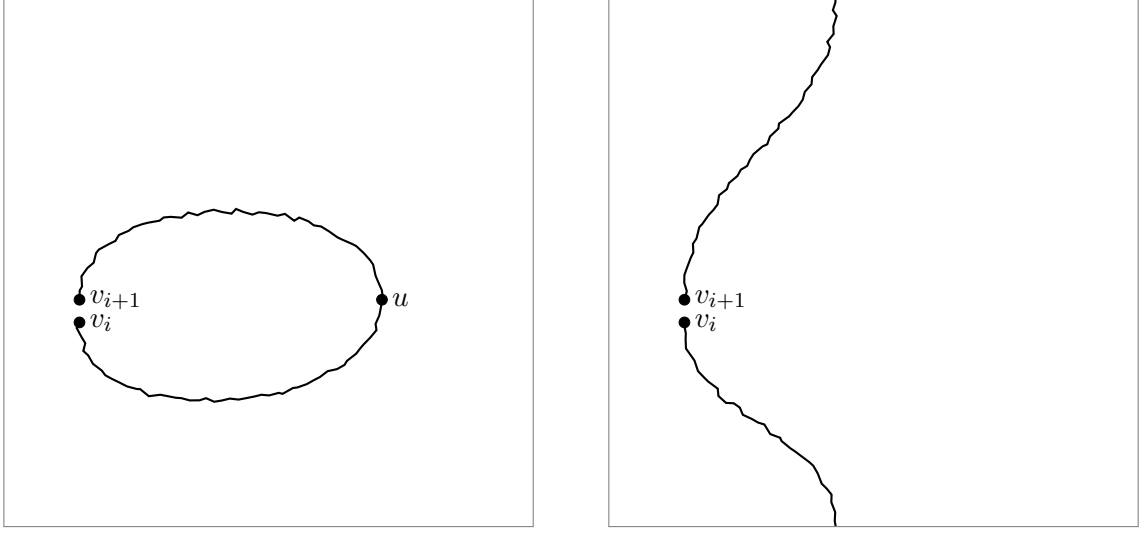
*Proof.* Assume without loss of generality that  $v_{i-1} = (x, y - 1)$  and  $v_{i+1} = (x, y + 1)$ . Now denote by  $f$  the projection on the second coordinate,  $f(a, b) = b$ . and observe that for any  $j$  the distance  $f(v_{j+1}) - f(v_j)$  satisfies:

$$f(v_{j+1}) - f(v_j) = \begin{cases} \pm(l-1) & v_j = (\cdot, l-1), v_{j+1} = (\cdot, 0) \\ \in \{\pm 1, 0\} & \text{else} \end{cases}$$

So if there is no  $u \neq v_i$  such that  $f(u) = f(v_i) = f(v_{i-1}) + 1$  then there must to be an edge  $\{v_j, v_{j+1}\}$ ,  $v_j = (\cdot, l-1)$ ,  $v_{j+1} = (\cdot, 0)$  such  $P$  pass trough it. So

$$\begin{aligned} f(v_{i-1}) &\leq f(v+1) + |f(v+2) - f(v+1)| + \dots + -(l-1) \\ &\leq f(v+1) - |P| - (l-1) \end{aligned}$$

But  $|P| \leq l$ . □



**Definition 10.** *Horizontal and Vertical diameters.* Let  $P$  be defined again as exactly as in claim 14. We will say that the horizontal diameter of  $P$  is:

$$\max_{v,u \in P} \min \{|v_x - u_x|, |v_x + l + u_x|\}$$

Similarly, we define the vertical diameter to be:

$$\max_{v,u \in P} \min \{|v_y - u_y|, |v_y + l + u_y|\}$$

**Claim 15.** *If  $P$  is a non empty simple cycle with vertical and horizontal diameters  $d_1, d_2$ , then it can either be a square or can be decomposed into two simple cycles  $P_1, P_2$  with either the vertical diameter of both of them being strictly less than  $d_1$  and their horizontal diameters being at most  $d_2$ , or the horizontal diameter of both of them being strictly less than  $d_2$  and their vertical diameters being at most  $d_1$ .*

*Proof.* If  $P$  is not a square, it must have either a vertical or horizontal diameter greater than 1. We will assume, without loss of generality, that the vertical diameter is greater than 1. Let  $v_0 v_1 \dots v_k$  be the vertices that compose  $P$ , and let  $v_i, v_j$  be the vertices at maximal distance. We will pick the first vertex  $v_{i'}$  from the left of  $v_i$  such that  $v_{i',y} \neq v_{i,y}$ . By claim 14, there must be at least one vertex  $v_l$  in  $P$  such that  $v_{i,y} = v_{l,y}$  and is the closest in horizontal distance to  $v_i$ . We will denote by  $P^\uparrow$  the path  $v_i, v_{i+1} \dots v_{l-1}, v_l$  and by  $P^\downarrow$  the path  $v_l, v_{l+1} \dots v_{i-1}, v_i$ . We will also denote by  $L$  the straight line  $v_i, (v_{i,x} + 1, v_{i,y}), (v_{i,x} + 2, v_{i,y}), \dots, v_l$ .

Observe that the vertical diameter will be lower by exactly 1 than the vertical diameter of  $P$ , and the horizontal distance between any point on  $L$  to other point on  $P$  will be lower than the horizontal distance of between the end of  $L$  (namely  $\{v_i, v_l\}$ ), to the same point. So if  $P^\uparrow \cup L, P^\downarrow \cup L$  are simple cycles then we get the desire. Suppose that  $P^\uparrow \cup L$  is not a simple cycle. This means that the path  $v_i, v_{i+1} \dots v_{l-1}, v_l \dots (v_{i,x} + 2, v_{i,y}), (v_{i,x} + 1, v_{i,y}), v_i$  must contain an inner loop. This loop cannot be supported on  $P^\uparrow$  alone, because  $P^\uparrow \subset P$  so the inner loop would have to be contained also in  $P$ , which is a contradiction. Therefore, the inner loop is also supported on  $L$ , so it follows that there is a vertex in  $L$  with degree

$> 2$  in  $P^\dagger \cup L$ , namely a vertex  $u \neq v_i, v_l$  such that  $u_y = v_{i,y}$ , and also  $u \in P$ . But by the construction of  $L$ ,  $|u_x - v_{i,x}| < |v_{l,x} - v_{i,x}|$ , which is a contradiction to the fact that  $v_l$  was chosen to minimize the horizontal distance. Thus, we have our claim.  $\square$

We have now established that  $c \in C_X$  with  $|c| < l$  can be decomposed into simple cycles, each of which has a weight less than  $l$ . Applying claim 15 recursively, we can further decompose each of these cycles into a sum of unit squares. However, claim 12 states that unit squares are in  $C_Z^\perp$ , so  $c \in C^\perp$ .  $\square$

### 3.5 Quantum Expander Codes.

As similar to the classical case, the next natural question to ask is whether there are codes with positive rates. The quantum expanders were the first quantum LDPC codes to achieve a square-root distance and positive rate [TZ14; LTZ15]. The leading insight was the idea that the Toric code could be represented as a variant product of the repetition code. For example, consider the cross restriction in Figure fig. 3.1; that restriction can be obtained by gluing two vertices of two different cycle graphs.

**Definition 11.** For any two matrices  $A, B$ , with the same number of rows, denote by  $[A, B]$  the matrix obtained by attach  $B$  next to  $A$  from right. Let  $H_1, H_2 \in \mathbb{F}_2^{n \times r}$  be the parity check matrices. Define the bit and the phase parity checks matrices to be:

$$\begin{aligned} H_X &= \begin{bmatrix} H_1 \otimes I_r & I_n \otimes H_2^\top \end{bmatrix} \\ H_Z &= \begin{bmatrix} I_r \otimes H_2 & H_1^\top \otimes I_n \end{bmatrix} \end{aligned}$$

The matrices are orthogonal to each other as  $H_X H_Z^\top = H_1 \otimes H_2^\top + H_1 \otimes H_2^\top = 0$  and therefore the pair define a valid CSS code. We will call to that code the Hyperproduct and denote it by  $Q(H_1 \times H_2)$ .

Obliviously, if  $H_1, H_2, H_1^\top, H_2^\top$  are parity checks matrices of an LDPC codes, so are  $H_X, H_Z$  as their maximal row weight is at most two times larger.

**Example 4.** The Toric code could be thought as the Hyperproduct of the repetition code with himself. The parity check matrices of the codes are given follow. The left  $3 \times 3$  matrix corresponds to the repetition code while the right  $18 \times 9$  corresponds to the vertices check of the Toric code.

$$\begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

**Claim 16.** Let  $A, B \in \mathbb{M}_{n \times r_1}, \mathbb{M}_{n \times r_2}$  then  $\dim \ker[A, B]$  is  $\dim \ker A + n$ .

*Proof.* The proof is omitted.  $\square$

**Claim 17.** Let  $k_1, k_2$  be the dimension of the codes with the full rank parity check matrices  $H_1, H_2$ . Then the dimension of the Hyperproduct code is  $\geq k_1 k_2$ .

*Proof.* We will find the dimensions of each of the classical codes defined by  $\ker H_X$  and  $\ker H_Z$ . Notice that the length of the  $H \otimes I_n$  equals  $n \times n = n^2$ , And assuming the fullness of the ranks, the length of  $I_{r_1} \otimes H_2^\top$  is  $r_1 \cdot r_2$ . Thus the length of  $\ker H_X$  is  $n^2 + r_1 r_2$ . Now, recall that for any matrix  $A$  it holds that  $\dim \ker(A \otimes I_l) = l \cdot \dim \ker A$ . Therefore using claim 16 we obtain that the dimension of  $\ker H_X$  is  $k_1 n + r_1 r_2$ .

By the same arguments we have that  $\dim C_Z = k_2 n + r_1 r_2$ . Thus the dimension of the quantum code is:

$$\begin{aligned} \dim Q(C_X, C_Z) &= \dim C_X + \dim C_Z - (n^2 + (n - k_1)(n - k_2)) \\ &= (k_1 + k_2)n + 2(n - k_1)(n - k_2) - (n^2 + (n - k_1)(n - k_2)) \\ &= k_1 k_2 \end{aligned}$$

$\square$

**Remark 1.** Let  $H'_1$  be a parity check matrix obtained by puncturing columns from  $H_1$ , denote by  $k'_1$  the dimension of that code. Then the Hyperproduct  $Q(H'_1 \times H_2)$  is a CSS code with dimension  $k'_1 k_2$ . Moreover if the number of columns left after the puncturing is less than the distance of  $\ker H_1$  then it must hold that  $k'_1 = 0 \Rightarrow \dim Q(H'_1 \times H_2) = 0$ . Otherwise, one can take a non trivial codeword of  $\ker H'_1$  and extending it to a valid codeword of  $\ker H_1$  by set any punctured coordinate of it to zero. The yielded codeword has weight less than  $d$  which is contradiction.

**Claim 18.** Denote by  $d$  the minimal distance of  $\ker H_1$ . Any codeword  $x$  of  $C_X = \ker H_X$  with edge at most  $d$  belongs to  $C_Z^\perp$ .

*Proof.* Define by  $H'_1$  the matrix which obtained by puncturing from  $H_1$  the columns associated with the coordinates  $e_i$  such that subspace corresponding to  $e_i \otimes I_n$  doesn't support  $x$ . Denote by  $S$  the set of the remaining coordinates. For example if  $H_1 \otimes I_n = I_2 \otimes I_2$  and  $x = [1, 0, 0, 0]$  then  $H'_1$  is the unit matrix  $[1, 0]^\top$  obtained by puncturing the second column of  $H_1 = I_2$ , and  $S = \{e_1\}$ .

As  $|x| < d$  we have that  $|S| < d$ , Namely  $H'_1$  supported on less than  $d$  coordinates and therefore  $\dim Q(H'_1 \times H_2) = k'_1 k_2 = 0$ . Thus, by the fact that for any CSS code  $\dim C = \dim C_X - \dim C_Z^\perp$  it follows that  $\dim \ker H'_X = \dim \ker H_Z^\perp \Rightarrow \ker H'_X = \ker H_Z^\perp$ . Denote by  $x'$  the restriction of  $x$  to the columns of  $H'_X$  and clearly, by the definition of the construction,  $x'$  is a codeword of  $\ker H'_X$ . Thus  $x'$  is also codeword of  $\ker H_Z^\perp$  and by the same argument,  $x$  is also a codeword of  $\ker H_Z^\perp$ .  $\square$

Immediately from claim 18 we obtain that existences of quantum LDPC codes with positive rate and  $\Theta(\sqrt{n})$  distance by taking the Hyperproduct of two classical expander codes.

**Theorem 1.** There exists an infinity family of QLDPC codes with positive rate and  $\Theta(\sqrt{n})$  distance.

## Chapter 4

# Good qLDPC Codes and LTC.

The existence of good QLDPC codes and LTC were considered open problems for roughly two decades. Even though they seemed to be related only by containing the word "code" in their title, they were proven to exist by the same construction. They first appeared in [Din+22] as good locally testable codes and not long after that in [PK21], in which they also extended and derived the result to obtain the quantum code. Here, we follow the [LZ22] work, which simplifies the original proof and doesn't rely on any concept more complicated than what we already saw in the previous chapters. They also coined the term "Quantum Tanner Codes".

### 4.1 Quantum Tanner Codes.

Recall our insight that for a pair of LDPC codes to define a good CSS code, it must hold that neither of them is a good code, and moreover they both have a constant distance. Therefore, we understand that any codeword in  $C_X$  that has small weight belongs to  $C_Z^\perp$ . To prove it, we are going to come up with a construction such that if  $x \in C_X$  and  $|x|$  is small then there is some small codeword  $z \in C_Z^\perp$  such that  $|x + z| < |x|$ ; by repeating this process recursively it follows that  $x \in C_Z^\perp$ . To formulate this theorem we will need the following definitions:

**Definition 12.** Let  $C = \mathcal{T}(G, C_0)$ . We say that  $x \in C_\oplus$  is **reducible** if there exists a vertex  $v$  and a small codeword  $c_v$ , for which, adding the assignment of  $c_v$  over the  $v$ 's edges to  $x$  decreases the weight. Namely,  $|x + c_v| < |x|$ . If  $x \in C_\oplus$  is not a reducible codeword then we say that  $x$  is **irreducible**.

The next two definitions are concerned with the properties of the small code that will be set on the edges. Using them, one can characterize cases in which a local view can be reduced by subtracting a codeword from the dual code.

**Definition 13** ( $w$ -Robustness). Let  $C_0$  be a code of length  $\Delta$  with minimum distance  $\delta_0 \Delta$ .  $C = (C_A^\perp \otimes C_B^\perp)^\perp$  will be said to be  $w$ -robust if for any codeword  $c \in C$  of weight less than  $w$ , it follows that  $c$  can be decomposed into a sum of  $c = t + s$  such that  $t \in C_A \otimes \mathbb{F}^B$  and  $s \in \mathbb{F}^A \otimes C_B$ , where  $s$  and  $t$  are each supported on at most  $\frac{w}{\delta_0 \Delta}$  rows and columns. For convenience, we will denote by  $B'$  ( $A'$ ) the rows (columns) supporting  $t$  ( $s$ ) and use the notation  $t \in C_A \otimes \mathbb{F}^{B'}$ .

The definition we gave for  $w$ -Robustness is identical to the one stated by Zemor and Leiverar, but we also included the decomposition property in the definition. We refer the readers to the appendix section in [Leiverar] for an existence proof by random construction. We mention that the random construction also gives the Gilbert-Varshamov bound.

**Definition 14** ( $p$ -Resistance to Puncturing.). *Let  $p, w$  be integers. We will say that the dual tensor code  $C_A \otimes \mathbb{F} + \mathbb{F} \otimes C_B$  is  $w$ -robust with  $p$ -resistance to puncturing, if the code obtained by removing (puncturing) a subset of at most  $p$  rows and columns is  $w$ -robust.*

**Definition 15** (Quantum Tanner Code.). *Let  $\Gamma$  be a group at size  $n$ . And let  $A, B$  be a two generator set of  $\Gamma$  such that if  $a \in A$  ( $B$ ) then also  $a^{-1} \in A$  ( $B^{-1}$ ) and that for any  $g \in \Gamma, a \in A, b \in B$  it holds that  $g \neq agb$ . Define the left-right Cayley complex to be the graph  $G = (\Gamma, E)$  obtained by taking the union of the two Cayley graphs generated by  $A$  and  $B$ . So the vertices pair  $u, v$  are set on a square diagonal only if there are  $a \in A$  and  $b \in B$  such that  $u = avb$ . We can assume that  $G$  is a bipartite graph (otherwise just take  $\Gamma' = \Gamma \times \mathbb{Z}_2$  and define the product to be  $a(u, \pm) = (au, \mp)$ ).*

Now divide the graph into positive and negative vertices according to their coloring  $V^-$  and  $V^+$ . And define the positive graph to be  $G^+ = (V^+, E)$  and by  $G^- = (V^-, E)$  the negative graph, where  $E$  denotes the squares, put differently there is an edge between  $v$  and  $u$  in  $G^+$  if both vertices are positive and they are laid on the ends of a square's diagonal.

The quantum Tanner code is a CSS code, such that  $C_X$  is defined to be the classical Tanner code  $\mathcal{T}(G^+, (C_A^\perp \otimes C_B^\perp)^\perp)$  and  $C_Z$  is defined as  $\mathcal{T}(G^-, (C_A \otimes C_B)^\perp)$ . Note that in contrast to the classical Tanner code, in the quantum case it will be more convenient to think of codewords as assignments set on the squares and not on the edges.

Now we can state formally the theorem:

**Theorem 2.** *For  $\varepsilon \in (0, \frac{1}{2})$ ,  $\gamma \in (\frac{1}{2} + \varepsilon, 1)$ ,  $\delta_0 > 0$ , large enough  $\Delta$ , and small codes  $C_A, C_B$  with distance at least  $\delta_0 \Delta$  if the dual tensor code of  $C_A, C_B$  is  $w$ -robust and  $\Delta^\gamma$  resistance to puncturing, then there exists an infinite family of square complexes for which the Tanner code defined by the complexes and the dual tensor code such that any codeword with weigh less than  $\frac{\delta_0 n}{4\Delta^{3/2+\varepsilon}}$  is reducible definition 12.*

**Claim 19.** *Consider a codeword  $c$  such there exists a negative vertex  $v$  that adjoins only for normal vertices. Then  $c$  is reducible, and there exists  $y \in C_A^\perp \otimes C_B^\perp$  such that  $y$  supported only on  $v$  and  $|c_v + y| < |c_v|$ .*

*Proof.* □

**Claim 20.** *The distance of the dual tensor code is at least  $\delta \Delta$ .*

*Proof.* By the robustness property, any codeword of the dual tensor code with a weight less than  $\delta_0 \Delta$  is supported on at most one row. Let  $c$  be such a codeword and denote by  $i$  the number of the non-trivial row. Fix a  $c' \in C_A^\perp$  such that the  $i$ th coordinate of  $c'$  is non-zero and consider the multiplication of  $c$  with the codewords of  $C_A^\perp \otimes C_B^\perp$  of the following form:

$$J = \left\{ c' \otimes c_b : c_b \in C_B^\perp \right\}$$

So the  $i$ th row of any  $x \in J$  is a codeword of  $C_B^\perp$  and in total, collecting all the  $i$ th rows of codewords in  $J$  sums up to all the code words in  $C_B^\perp$ . On the other hand,  $c \cdot x = 0$  for all  $x \in J$ ; that is,  $c \cdot x = c_i \cdot c_b = 0$ . Thus we obtain that  $c_i \in C_B$  and therefore  $|c_i| \geq \delta_0 \Delta \Rightarrow |c| \geq \delta_0 \Delta$ , which is a contradiction. □

Denote by  $S$  and  $T$  the positive and negative vertices support the codeword  $x \in C_x$ .

**Claim 21.** *for any  $\varepsilon \in (0, 1)$  and large enough  $\Delta$  it holds that  $|S| \leq \Delta^\varepsilon |T|$*

*Proof.* Suppose not, namely that  $|S| > \Delta^\varepsilon |T|$ , then  $|x|/|T| > \Delta^\varepsilon |x|/|S| > \Delta^\varepsilon \cdot \delta_0 \Delta$  But:

$$\frac{|x|}{|T|} = \frac{\Theta(E(T, T))}{|T|} \leq \Theta(\Delta^2) \frac{|T|}{n} + \Theta(\Delta) \rightarrow_{n \rightarrow \infty} \Theta(\Delta)$$

□

**Claim 22.** *There is a negative vertex which adjoins to only normal vertices.*

*Proof.* Denote by  $S_e, S_n$  the exceptional and the normal vertices. Namely the weight of the local view for any vertex in  $S_e$  is grater than  $\Delta^{3/2+\varepsilon}$ , and the normal vertices is the complementary vertices set. Suppose through contradiction that any negative vertex  $v_- \in V_-$  has at least one sibling in  $S_e$ . Therefore  $|T| \leq \Delta |S_e|$  combining with claim 21 it follows that  $|S| \leq \Delta^{1+\varepsilon} |S_e|$ :

$$\begin{aligned} \Delta^{3/2+\varepsilon} &\leq \frac{E(S, S_e)}{|S_e|} = \Theta(\Delta^2) \frac{|S|}{n} + \Theta(\Delta) \sqrt{\frac{|S|}{|S_e|}} \\ &\leq \Theta(\Delta^2) \frac{|S|}{n} + \Theta(\Delta) \Theta\left(\Delta^{\frac{1+\varepsilon}{2}}\right) \end{aligned}$$

□

**Claim 23.** *Any  $x \in (C_A^\perp \otimes C_B^\perp)^\perp$  such that  $|x| \leq w$ , and denote by  $A', B'$  the rows and cols support  $x$ . Then  $x$  can decompose into a sum of  $x = t + s$  such that  $t \in C_A \otimes \mathbb{F}^{B'}$  and  $s \in \mathbb{F}^{A'} \otimes C_B$ .*

*Proof.* By induction on the number of rows supported  $x$ . Denote  $x$  by

$$\begin{aligned} x &= \sum_{i, c_a} c_a \otimes e_i + \sum_{i, c_b} e_i \otimes c_b \\ &= c_{a'} \otimes e_\tau + \sum_{i/\tau, c_a} c_a \otimes e_i + \sum_{i, c_b} e_i \otimes c_b \end{aligned}$$

Now, if  $c_{a'} \otimes e_\tau$  decrease the summation weight then it must holds that  $x$

$$= c_{a'} \otimes e_\tau + t' + s' = \overbrace{(c_{a'} \otimes e_\tau + t')}^{C_A \otimes \mathbb{F}^{B'}} + s'$$

□

**Claim 24.** *Let  $x$  be a codeword of  $(C_A^\perp \otimes C_B^\perp)^\perp$  such that  $d(x, \mathbb{F}^A \otimes C_B), d(x, C_A \otimes \mathbb{F}^B) \leq \xi < w$ . Then  $d(x, C_A \otimes C_B) < \xi$ .*

*Proof.*

□

**Claim 25.** *Suppose that  $v \in T$  is sarounded by normal vertices. Then:*

$$d(c|_v, c_A \otimes C_B) < \Theta(\Delta^{1+\varepsilon})$$

*Proof.* □

**Remark 2.**

$$\begin{aligned} \alpha|S_e| &\leq \theta^2 \frac{|S_e||S|}{|V|} - \lambda^2 \sqrt{|S_e||S|} & |S_e| &\leq \left(\alpha - \theta^2 \frac{|S|}{|V|}\right)^{-2} \lambda^4 |S| \\ \beta|T| &\leq 2\theta \frac{|T||S|}{2|V|} + 2\lambda \sqrt{|T||S|} & |T| &\leq \left(\beta - 2\theta \frac{|S|}{2|V|}\right)^{-2} 4\lambda^2 |S| \end{aligned}$$

$$\begin{aligned} \Delta^{1\frac{1}{2}+\varepsilon}|T| &\leq |x| \leq \Delta^{1\frac{1}{2}}|S| \\ |T|\Delta^\varepsilon &\leq |S| \end{aligned}$$

**Definition 16** (The Disagreement Code). *Given a Tanner code  $C = \mathcal{T}(G, C_0)$ , define the code  $C_\oplus$  to contain all the words equal to the formal summation  $\sum_{v \in V(G)} c_v$  when  $c_v$  is an assignment of a codeword  $c_v \in C_0$  on the edges of the vertex  $v \in V(G)$ . We call to such code the **disagreement code** of  $C$ , as edges are set to 1 only if their connected vertices contribute to the summation codewords that are different on the corresponding bit to that edge. In addition, we will call to any contribute  $c_v$ , the **suggestion** of  $v$ . And notice that by linearity, each vertex suggests, at most, a single suggestion.*

*Finally, given a bits assessment  $x \in \mathbb{F}_2^E$  over the edges of  $G$ , we will denote by  $x^\oplus \in C_\oplus$  the codeword which obtained by summing up suggestions set such each vertex suggests the closet codeword to his local view. Namely, for each  $v \in V$  define:*

$$\begin{aligned} c_v &\leftarrow \arg_{\tilde{c} \in C_0} \min d(x|_v, \tilde{c}) \quad \forall v \in V \\ x^\oplus &\leftarrow \sum_{v \in V} c_v \end{aligned}$$

*We will think about  $x^\oplus$  as the disagreement between the vertices over  $x$ .*

**Lemma 5** (Linearity Of The Disagreement). *Consider the code  $C = \mathcal{T}(G, C_0)$ . Let  $x \in \mathbb{F}_2^E$  then for any  $y \in C$  it holds that:*

$$(x + y)^\oplus = (x)^\oplus$$

*Proof.* Having that  $y \in C$  follows  $y|_v \in C_0$  and therefore  $\arg_{\tilde{c} \in C_0} \min d(z, \tilde{c}) = y|_v + \arg_{\tilde{c} \in C_0} \min d(z, \tilde{c} + y|_v)$ , Hence the suggestion made by vertex  $v$  is:

$$\begin{aligned} c_v &\leftarrow \arg_{\tilde{c} \in C_0} \min d((x + y)|_v, \tilde{c}) \\ &\leftarrow y|_v + \arg_{\tilde{c} \in C_0} \min d((x + y)|_v, \tilde{c} + y|_v) \\ &\leftarrow y|_v + \arg_{\tilde{c} \in C_0} \min d(x|_v, \tilde{c}) \end{aligned}$$

It follows that:

$$\begin{aligned} (x + y)^\oplus &= \sum_{v \in V} c_v = \sum_{v \in V} y|_v + \sum_{v \in V} \arg_{\tilde{c} \in C_0} \min d(x|_v, \tilde{c}) \\ &= y^\oplus + x^\oplus = x^\oplus \end{aligned}$$

When the last transition follows immediately by the fact that  $y \in C$  and therefore any pair of connected vertices contribute the same value for their associated edge □



## 4.2 Decoding and Testing

For completeness, we show exactly how Theorem 1 implies testability. The following section repeats Leiverar's and Zemor's proof [LZ22]. Consider a binary string  $x$  that is not a codeword. The main idea is the observation that the number of bits flipped by (any) decoder, while decoding  $x$ , bounds the distance  $d(x, C)$  from above. In addition, the number of positive checks in the first iteration is exactly the number of violated restrictions.

**Definition 17.** Let  $L = \{L_i\}_0^{2|E|}$  be a series of  $2|E|$ . Such that for each vertex  $v \in V$   $\sum_{e=\{u,v\}} L_{e_v} \in C_0$ . We will call  $L$  a Potential list and refer to the  $e_v$ 's as the element of  $L$  as a suggestion made by the vertex  $v \in V$  for the edge  $e \in E$ . Sometimes we will use the notation  $L_v$  to denote all the  $L$ 's coordinates of the form  $L_{e_v} \forall e \in \text{Support}(v)$ . Define the Force of  $L$  to be the following sum  $F(L) = \sum_{e=\{v,u\} \in E} (L_{e_v} + L_{e_u})$  and notice that  $F(L) \in C_\oplus$ . And define the state  $S(L) \subset \mathbb{F}_2^{|E|}$  of  $L$  as the vector obtained by choosing an arbitrary value from  $\{L_{e_v}, L_{e_u}\}$  for each edge  $e \in E$ .

**Claim 26.** Let  $L$  be the Potential list. If  $F(L) = 0$  then  $S(L) \in C$ .

*Proof.* Denote by  $\phi(e) \in \{L_{e_v}, L_{e_u}\}$  the value which was chosen to  $e = \{v, u\} \in E$ . By  $F(L) = 0$ , it follows that  $L_{e_v} + L_{e_u} = 0 \Rightarrow L_{e_v} = L_{e_u} = \phi(e)$  for any  $e \in E$ . Hence for every  $v \in V$  we have that  $S(L)|_v = \sum_{u \sim v} \phi(\{v, u\}) = \sum_{u \sim v} L_{e_v} \in C_0 \Rightarrow S(L) \in C$   $\square$

The decoding goes as follows. First, each vertex suggests the closest  $C_0$ 's codeword to his local view. Those suggestions define a Potential list, denote it by  $L$ , then if  $F(L) < \tau$ , by Theorem 1, one could find a suggestion of vertex  $v$  and a codeword  $c_v$  such that updating the value of  $L_v \leftarrow L_v + c_v$  yields a Potential list with lower force. Therefore repeating the process till the force vanishes, obtain a Potential list in which its state is a codeword.

**Definition 18.** Let  $\tau > 0, f : \mathbb{N} \rightarrow \mathbb{R}^+$ , and consider a Tanner Code  $C = \mathcal{T}(G, C_0)$ . Let us Define the following decoder and denote it by  $\mathcal{D}$ .

**Data:**  $x \in \mathbb{F}_2^n$   
**Result:**  $\arg \min \{y \in C : |y + x|\}$  if  $d(y, C) < \tau$  and False otherwise.

```

1  $L \leftarrow \text{Array}\{\}$ 
2 for  $v \in V$  do
3    $c'_v \leftarrow \arg \min \{y \in C_0 : |y + x|_v|\}$ 
4    $L_v \leftarrow c'_v$ 
5 end
6  $z \leftarrow \sum_{v \in V} c'_v$ 
7 if  $|z| < \tau \frac{n}{f(n)}$  then
8   while  $|z| > 0$  do
9     find  $v$  and  $c \in C_0$  such that  $|z + c_v| < |z|$ 
10     $z \leftarrow z + c_v$ 
11     $L_v \leftarrow L_v + c_v$ 
12  end
13 else
14   reject.
15 end
16 return  $S(L)$ 
```

**Algorithm 1:** Decoding

**Theorem 3.** Consider a Tanner Code  $C = [n, n\rho, n\delta]$  and the disagreement code  $C_\oplus$  defined by it. Suppose that for every codeword  $z \in C_\oplus$  in  $C_\oplus$  such that  $|z| < \tau'n/f(n)$ , there exists another codeword  $y \in C_\oplus$  such that  $|y| < |z|$ , set  $\tau \leftarrow \frac{\tau'}{6\Delta}\delta$  then,

1.  $\mathcal{D}$  corrects any error at a weight less than  $\tau n/f(n)$ .
2.  $C$  is  $f(n)$  testable code.

*Proof.* So it is clear from the claim claim 26 above that if the condition at line (6) is satisfied, then  $\mathcal{D}$  will converge into some codeword in  $C$ . Hence, to complete the first section, it left to show that  $\mathcal{D}$  returns the closest codeword. Denote by  $e$  the error, and by simple counting arguments; we have that  $\mathcal{D}$  flips at most:

$$d_{\mathcal{D}}(x, C) \leq 2|e|\Delta + \tau \frac{n}{f(n)} \Delta$$

bits. Hence, by the assumption,

$$d_{\mathcal{D}}(x, C) \leq 3\Delta\tau \frac{n}{f(n)} \leq 3\Delta\tau\delta n < \frac{1}{2}\delta n$$

Therefore the code word returned by  $\mathcal{D}$  must be the closet. Otherwise, it contradicts the fact that the relative distance of the code is  $\delta$ . To obtain the correctness of the second section, we will separate when the conditional at the line (5) holds and not. And prove that the testability inequality holds in both cases. Let  $x \in \mathbb{F}_2^n$  and consider the running of  $\mathcal{D}$  over  $x$ . Assume the first case, in which the conditional at line (5) is satisfied. In that case,  $\mathcal{D}$  decodes  $x$  into its closest codeword in  $C$ . Therefore:

$$\begin{aligned} d(x, C) &\leq d_{\mathcal{D}}(x, C) \leq m\xi(x)\Delta + |z|\Delta \\ &\leq m\xi(x)\Delta + m\xi(x)\Delta^2 \\ \frac{d(x, C)}{n} &\leq \kappa_1\xi(x) \end{aligned}$$

Now, consider the other case in which:  $|z| \geq \tau \frac{n}{f(n)}$ .

$$\begin{aligned} \frac{d(x, C)}{n} &\leq 1 \leq \frac{|z|}{\tau n} f(n) \leq \frac{m}{n} \frac{1}{\tau} \Delta \xi(x) f(n) \\ &\leq \kappa_2 \xi(x) f(n) \end{aligned}$$

Picking  $\kappa \leftarrow \max\{\kappa_1, \kappa_2\}$  proves  $f(n)$ -testability □

## Chapter 5

# First Attempt For Good qLTC.

### 5.1 The Polynomial-Code Is Not $w$ -Robust.

One idea for constructing is to use the polynomial code instead of  $C_0$ . This follows from the fact that if one picks a degree strictly greater than  $\Delta/2$ , then  $C_0^\perp \subset C_0$  and therefore one could choose  $C_z$  to be the same code defined on the negative vertices of the graph.

Here we prove that the dual-tensor code, in that case, is not  $w$ -robust, meaning that any such construction should be considered another way of proving the Reduction Lemma.

**Claim 27.** *Let  $C_0$  be the  $[\Delta, d, \Delta - d]$  polynomial code. Then any code word in  $(C_0^\perp \otimes C_0^\perp)^\perp$  is a polynomial in  $F[x, y]$  at degree at most  $\Delta + d$*

*Proof.* Consider base element  $C_0 \otimes \mathbb{F}$ , denote it by  $c = g_i \otimes e_j$ . And notice that  $c$  has representation in  $F[x, y]$  of  $\prod_{y' \neq j} (y - y')g_i(x)$ . By the fact that  $g_i(x) \in C_0$  we have that degree of  $c$  is at most  $\Delta + \delta$ . Hence any element in the subspace of  $C_0 \otimes \mathbb{F}$  is a polynomial at degree at most  $\Delta + d$ .  $\square$

**Claim 28.** *The dual-tensor polynomial code is not  $w$ -robust.*

*Proof.* Consider the following polynomial

$$P(x, y) = \prod_{i \neq \Delta-1} (x + iy) = \prod_{i \neq 1} (x - iy)$$

The degree of any monomial is at most  $\Delta - 1$ , Thus it clear that  $P \in (C_0^\perp \otimes C_0^\perp)^\perp$ . And by the fact that for any  $x \neq y$  there exists  $i \neq 1$  such that  $x = iy$  we have that  $P(x, y) = 0$ . Hence the weight of  $P$  is at most  $|\{(x, y) : x = y\}| = \Delta$ . Yet, for any  $x = y$  it follows:

$$P(x, x) = \prod_{i \neq \Delta-1} (x + ix) = x^{\Delta-1} \prod_{i \neq \Delta-1} (1 + i) = (\Delta - 1)! =_{\Delta} -1 \neq_{\Delta} 0$$

Put it differently, the diagonal of the matrix has only non-zero values, therefore the  $P$  is supported on the entire collection of rows and columns. Namely, we found a codeword in the dual tensor code at weight less than  $\Delta^{1/2}$  which is not restricted to at most  $\Delta^{1/2}/\delta_0\Delta$ . So, the dual-tensor polynomial code is not a  $w$ -robust code, for  $w = \Delta^{1/2}$ .  $\square$



## Chapter 6

# Local Majority $\neq$ Local Testability.

**Claim 29.** Suppose that  $G$  is an expander graph with a second eigenvalue  $\lambda$ , then For any layer  $U$  there exist a layer  $U'$  such that:

- (1)  $|U'| \geq |U|$
- (2)  $w_{E/E'}(x|_{U'}) \geq \Delta|U'| \left( \delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta} \right)$

*Proof.* Consider layer  $U$  and denote by  $U_{-1}$  and  $U_{+1}$  the preceding and the following layers to  $U$  in  $T$ . It follows from the expander mixing lemma that:

$$\begin{aligned} w_{E/E'}(x|_U) &\geq \delta_0 \Delta |U| - w \left( E(U_{-1} \cup U_{+1}, U) \right) \geq \\ &\delta_0 \Delta |U| - E(U_{-1} \cup U_{+1}, U) \\ &\delta_0 \Delta |U| - \Delta \frac{|U||U_{-1}|}{n} - \Delta \frac{|U||U_{+1}|}{n} \\ &\quad - \lambda \sqrt{|U||U_{-1}|} - \lambda \sqrt{|U||U_{+1}|} \end{aligned}$$

**Claim 30.** We can assume that  $|U| \geq |U_{-1}|, |U_{+1}|$ .

*Proof.* Suppose that  $|U_{+1}| > |U|$ , so we could choose  $U$  to be  $U_{+1}$ . Continuing stepping deeper till we have that  $|U| > |U_{+1}|, |U_{-1}|$ . Simiraly, if  $|U| > |U_{+1}|$  but  $|U_{-1}| > |U|$ , the we could take steps upward by replacing  $U_{-1}$  with  $U$ . At the end of the process, we will be left with  $U$  at a size greater than the initial layer and  $|U| > |U_{+1}|, |U_{-1}|$   $\square$

Using claim 30, we have that  $(|U_{+1}| + |U_{-1}|)/n < \frac{2}{3}$  and therefore:

$$w_{E/E'}(x|_U) \geq \left( \delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta} \right) \Delta |U|$$

$\square$

That immediately yields the following: let  $U_{\max} = \arg \max_{U \text{ layer in } T} |U|$  then:

$$|x| \geq w_{E/E'}(x|_{U_{\max}}) \geq \left( \delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta} \right) \Delta |U_{\max}|$$

**Claim 31.** Consider again the maximal layer  $U_{\max}$  then:

$$w_{E/E'}(x) \geq \left( \delta_0 - \frac{|U_{\max}|}{n} - \frac{\lambda}{\Delta} \right) \Delta |T|$$

*Proof.* Similarly to above, now we will bound the flux that all the nodes in  $T$  induce over  $E/E'$ . Denote by  $U_0, U_1 \dots U_m$  the layers of  $T$  ordered corresponded to their height, thus we obtain:

$$\begin{aligned}
w_{E/E'}(x) &\geq \delta_0 \Delta |T| - \sum_{i \in [m]} w(E(U_i, U_{i+1})) \\
&\geq \delta_0 \Delta |T| - \sum_{i \in [m]} E(U_i, U_{i+1}) \\
&\geq \delta_0 \Delta |T| - \sum_{i \in [m]} \frac{\Delta}{n} |U_i| |U_{i+1}| + \lambda \sqrt{|U_i| |U_{i+1}|} \\
&\geq \delta_0 \Delta |T| - \sum_{i \in [m]} \frac{\Delta}{n} |U_i| |U_{i+1}| + \lambda \frac{|U_i| + |U_{i+1}|}{2} \\
&\geq \delta_0 \Delta |T| - \frac{\Delta}{n} |T| |U_{\max}| - \lambda |T| \\
&\geq \left( \delta_0 - \frac{|U_{\max}|}{n} - \frac{\lambda}{\Delta} \right) \Delta |T|
\end{aligned}$$

□

**Claim 32.** Consider  $G = (V, E)$  a  $\Delta$ -ramunjan graph and let  $U$  be a subset of  $V$  such that  $|U| \geq \frac{1}{9}n$  then, there is must to be at least one vertex in  $U$  such the number of closed loops pass through it, is less than  $\sqrt{\Delta} \cdot n$ .

**Claim 33.** Alternate proof of flux inequality, which doesn't assume that there is no interference inside the layers.  $w(E(U, U)) > 0$ .

*Proof.* Separate into the following cases, First assume that  $|U_{\max}|/n > \frac{1}{3}$  then we have that the total interference with  $U_{\max}$  layers is at most:

$$\begin{aligned}
&\frac{\Delta |U_{\max}| (n - |U_{\max}|)}{n} + \lambda \sqrt{|U_{\max}| n} \leq \left( 1 - \frac{|U_{\max}|}{n} + \sqrt{3} \frac{\lambda}{\Delta} \right) \Delta |U_{\max}| \\
&\leq \left( \frac{2}{3} + \sqrt{3} \frac{\lambda}{\Delta} \right) \Delta |U_{\max}|
\end{aligned}$$

And therefore we have that the flux induced by  $U_{\max}$  is at least:

$$\left( \delta_0 \Delta - \frac{2}{3} + \sqrt{3} \frac{\lambda}{\Delta} \right) \Delta |U_{\max}|$$

So it lefts to consider the case in which for every layer it holds that  $|U_{\max}| \leq \frac{1}{3}n$ . At that case we count the flux induced by the whole three  $T$  which is what exactly we have prove in ?? minus the inner interference at the tree, That it we need only to subtract  $\sum \frac{\Delta |U_i|^2}{n} + \lambda |U_i| \leq \left( \frac{|U_{\max}|}{n} + \lambda/\Delta \right) |T|$  So we obtained that in that case:

$$w_{E/E'}(x) \geq \left( \delta_0 - 2 \frac{|U_{\max}|}{n} - 2\lambda/\Delta \right) \Delta |T| \geq \left( \delta_0 - \frac{2}{3} - 2 \frac{\lambda}{\Delta} \right) \Delta |T|$$

□

*Proof of Theorem 1.* Consider the size of the maxiaml layer  $|U_{\max}|$  and sepearate to the following two cases. First, consider the case that  $|U_{\max}| \geq \alpha n$  in that case it follows immediy by claim 29 that if  $\delta_0 > \frac{2}{3} - \frac{2\lambda}{\Delta}$  there exists  $\alpha' > 0$  such that:

$$|x| \geq \left( \delta_0 - \frac{2}{3} - \frac{2}{\lambda} \Delta \right) \Delta |U_{\max}| \geq \alpha' n$$

So, it is lefts to consider the second case in which  $|U_{\max}| < \alpha n$  in that case, we have from claim 31 inequality that:

$$\begin{aligned} |x| &\geq w_{E/E'}(x) \geq \left( \delta_0 - \frac{|U_{\max}|}{n} - \frac{\lambda}{\Delta} \right) \Delta |T| \\ &\geq \left( \delta_0 - \alpha - \frac{\lambda}{\Delta} \right) \Delta |T| \end{aligned}$$

Setting  $\alpha \geq \frac{2}{3}$  we complete the proof  $\square$

Unfortunately, Singelton bound doesn't allow both  $\delta_0 > \frac{2}{3}$  and  $\rho_0 \geq \frac{1}{2}$ , so in total, we prove the existence of code LDPC code which is good in terms of testability and distance yet has a zero rate. In the following subsection, we will prove that one can overcome this problem, by considering a variton of Tanner code, in which every vertex cheks only a  $\frac{2}{3}$  fraction of the edges in his support.

### 6.0.1 Overcoming The Vanishing Rate.

Consider the following code; instead of associating each edge with pair of checks, let's define the vertices to be the checks of small codes over  $q \in [0, 1]$  fraction of their edges. That is, now each vertex defines only  $(1 - \rho_0) q \Delta$  restrictions. Hence, the rate of the code is at least:

$$\begin{aligned} \rho \frac{1}{2} \Delta n &\geq \frac{1}{2} \Delta n - (1 - \rho_0) q \Delta n \\ \Rightarrow \rho &\geq \left( 2\rho_0 + \left( \frac{1}{q} - 2 \right) \right) q \\ \rho_0 &\geq 1 - \frac{1}{2q} \end{aligned}$$

for example, if  $q = 2/3$ , then for having constant rate, it is enough to ensure that  $\rho_0 \geq 1 - \frac{3}{4} = \frac{1}{4}$ .

**Intuition For Testability.** Before expand the construction let's us justify why one should even expects that removing constraints preserves testability. Assume that is guaranteed that the lower bound of the flux on the trivial vertices remains up to multiplication by the fraction factor  $q$ , or put it differently, one could just stick  $q$  in every inequality without lose correctness, Then:

$$\begin{aligned} w_{E/E'}(x|_U) &\geq \delta_0 q \Delta |U| - q w(E(U_{-1} \cup U_{+1}, U)) \\ \Rightarrow |x| &\geq \left( \delta_0 - \frac{2}{3} - \frac{2\lambda}{\Delta} \right) q \Delta |U_{\max}| \end{aligned}$$

As you can see, irreducible [12] words of the disagreement have a linear weight, despite that the original code has non-vanish rate.

Yet, We still require more to prove a linear distance. By repeating on the *Singleton Bound 2.3* proof it follows that the small code  $\tilde{C}_0$  obtained by ignoring arbitrary  $(q - \frac{1}{2}) \Delta$  coordinates yield a code with distance:

$$\left( \delta_0 q - \left( q - \frac{1}{2} \right) \right) \Delta$$

So assume that we could engineer an expander family such that the graphs obtained by removing  $\frac{1}{2}$  of the edges connected for each vertex result also expanders, and in addition, regarding  $\tilde{C}_0$  each edge is checked by both vertices on its support. Namely, a good Tanners Code could be defined on the restricted graphs; Then, any string that satisfies the original checks also has a linear weight. To achieve this property, we will restrict ourselves to a particular family of Cayley Graphs.

**Theorem** (Theorem 1+). *There exist a constant  $\alpha > 0$  and infinite family of codes which satisfies Theorem 1 and also good.*

**Definition 19** (Testability Tanner Code). *Let  $q > \frac{1}{2}$  and let  $J$  be a generator set for group  $\Gamma$  such that  $|J| = \Delta$ ,  $q|\Delta$ ,  $J$  closed for inverse, and there exist subset of  $J$ , denote it by,  $J'$  such that  $J'$  is a generator set of  $\Gamma$  and  $|J'| = \frac{1}{2}\Delta$ . Let  $C_0$  be a code with parameters  $C_0 = q\Delta [1, \rho_0, \delta_0]$ . For any vertex associate a subset  $\bar{J}_v \subset J/J'$  at size:*

$$|\bar{J}_v| = \left( q - \frac{1}{2} \right) \Delta \Rightarrow |\bar{J}_v \cup J'| = q\Delta$$

Define the code  $\mathcal{T}(J, q, C_0)$  to be the subspace such that any vertex's local view over the edges defined by  $\bar{J}_v \cup J'$  is a codeword of  $C_0$ . In addition, let's associate a code  $\tilde{C}_v$  obtained for any vertex by ignoring the bits supported on the  $\bar{J}_v$  coordinates. Notice that code defined by requiring that the local view of any vertex  $v$  of  $\text{Cayley}(\Gamma, J')$  is a codeword of  $\tilde{C}_v$  is a TannerCode. Denote it by  $\tilde{\mathcal{T}}(J, q, C_0)$ .

**Claim 34.** *Let  $J$  be defined as above such that both  $\text{Cayley}(\Gamma, J)$ ,  $\text{Cayley}(\Gamma, J')$  are expanders with algebraic expansion greater then  $\lambda$  and  $C_0$  with the parameters  $\rho_0 > 1 - \frac{1}{2q}$  and  $\delta_0 q - (q - \frac{1}{2}) > 2\lambda/\Delta$ . Then the code  $\mathcal{T}(J, q, C_0)$  is a good code.*

*Proof.* We already proved that the code has a positive rate, So it left to show a constant relative distance.

Consider a codeword  $x$  and denote by  $x'$  the restriction of  $x$  to  $\text{Cayley}(\Gamma, J')$  which is a codeword of  $\tilde{C} = \tilde{\mathcal{T}}(J, q, C_0)$ . But  $\tilde{C}$  is a Tanner Code such that any vertex sees at least  $\tilde{\delta}_0 \Delta := (\delta_0 - (q - \frac{1}{2})) \Delta$  nontrivial bits. Denote by  $S$  the vertices subset supports  $x'$ , and by  $E(S, S)$  the edges from  $S$  to itself, and by using the fact that  $\text{Cayley}(\Gamma, J')$  is an expander with second eigenvalue at most  $\delta$  we have that:

$$\frac{|x'|}{|S|} \geq \tilde{\delta}_0 \Delta \Rightarrow |S| \geq \left( \tilde{\delta} - \frac{2\lambda}{\Delta} \right) \Delta n$$

By the assumption that  $\tilde{\delta} > 2\lambda/\Delta$  we have that  $S$  must has liner size, and therefore  $|x'|$  also must to be linear in  $n$ . Finally as  $x' \subset x$  we obtain the correctness of the claim.  $\square$



**Claim 35** (Existence of such *Cayley's*). *Let  $S$  be a generator set such that  $\text{Cayley}(\Gamma, S)$  has a second largest eigenvalue greater than  $\lambda$ , And consider an arbitrary group element  $g \in \Gamma$  and denote by  $S_g$  the set  $gSg^{-1}$ . Then the second eigenvalue of the graph obtained by  $(\Gamma, S) \cup (\Gamma, S)$  is at most  $2\lambda$ .*

*Proof.* Denote by  $G, G'$  the *Cayley* graphs corresponding to  $S, S_g$ , for convenient we will use the notation of  $\sum_{v \sim_G u}$  to denote a summation over all the neighbors of  $v$  in the graph  $G$ . Let  $A_{G'}$  be the adjacency matrix of  $G'$ . Recall that  $G'$  is a  $\Delta$  regular graph, and therefore the uniform distribution  $\mathbf{1}$  is the eigenstate with the maximal eigenvalue, and the second eigenvalue is given by the min-max principle:

$$\begin{aligned}
\max_{f \perp \mathbf{1}} \frac{f^\top A_{G'} f}{f^\top f} &= \max_{f \perp \mathbf{1}} \sum_v \sum_{u \sim_{G'} v} \frac{f(u) f(v)}{f^\top f} \\
&= \max_{f \perp \mathbf{1}} \sum_v \sum_{\tau \in S} \frac{f(g\tau g^{-1}v) f(v)}{f^\top f} \\
&= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{\tau \in S} \frac{f(g\tau g^{-1}gv) f(gv)}{f^\top f} \\
&= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{\tau \in S} \frac{f(g\tau v) f(gv)}{f^\top f} \\
&= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{u \sim_G v} \frac{f(gu) f(gv)}{f^\top f}
\end{aligned}$$

As for any function  $f : V \rightarrow \mathbb{R}$  one could define a function  $f' : E \rightarrow \mathbb{R}$  such that  $f'(v) = f(v)$  and  $f'$  preserves the norm:

$$\begin{aligned}
f'^\top f' &= \sum_{v \in V} f'(v) f'(v) = \sum_{v \in V} f^\top(vg) f(vg) = f^\top f \\
\Rightarrow \max_{f \perp \mathbf{1}} \frac{f^\top A_{G'} f}{f^\top f} &= \max_{f \perp \mathbf{1}} \sum_{gv} \sum_{u \sim_G v} \frac{f(gu) f(vg)}{f^\top f}
\end{aligned}$$

By the Interlacing Theorem, [Hae95] the second eigenvalue of any subgraph of  $G'$  is less than the  $\lambda'$ , In particular, the eigenvalue of the graph obtained by taking the edges that are associated with elements of the  $S_g/S$ .

Denote that subgraph by  $G'_{/S}$ . Because  $S_g/S \cap S = \emptyset$ , we have that the edges sets of  $G, G'$  are disjointness sets. Hence the adjacency matrix of the graphs union equals the sum of their adjacency matrices. So in total, we obtain that:

$$\begin{aligned}
\lambda' &= \max_{f \perp \mathbf{1}} \frac{f^\top (A_G + A_{G'_{/S}}) f}{f^\top f} \\
&\leq \max_{f \perp \mathbf{1}} \frac{f^\top A_G f}{f^\top f} + \max_{f \perp \mathbf{1}} \frac{f^\top A_{G'_{/S}} f}{f^\top f} \\
&\leq \lambda + \lambda = 2\lambda
\end{aligned}$$

□

**Claim 36.** *If  $\Delta$  is a constant greater than two, and  $G$  is a  $\lambda$ -algebraic expander with girth at length  $\Omega(\log n)$ , then there exists a  $g \in \Gamma$  such that  $S_g \cap S = \emptyset$ .*

*Proof.* As  $\Delta > 2$  there must be at least two different elements  $s_1, s_2 \in S$  such that  $s_1 \neq s_2, s_2^{-1}$ . Pick  $g = s_1 s_2$ . Now assume through contradiction that there exists a pair  $s, r \in S$  such that  $g s g^{-1} = r \Rightarrow g s = r g$  and notice that the fact that  $s_1 \neq s_2^{-1}$  guarantees that both terms are a product of 3 element group. Therefore either that there is a 6-length cycle in the graph, Or that there is element-wise equivalence, namely  $s_1 = r, s_2 = s_1, s = s_2$ . The first case contradict the lower bound on the expander girth, which is at least  $\Omega(\log_\Delta(n))$ , while the other stand in contradiction to the fact that  $s_1 \neq s_2$   $\square$

**Remark. Regarding Quantum Codes.** Notice that any complex designed to hold CSS qLDPC codes must have constant length cycles. Otherwise, the distance of  $C_x$  will not be constant, and therefore the condition  $H_x H_z^\top = 0$  could be satisfied only if  $H_z$  is not a constant row-weight matrix, Put differently  $C_z$  is not an LDPC code. Consequently, any trial to generalize the construction for obtaining quantum codes must not rely on claim 36.

**Remark. Note On Random Construction.** One might wondering if using *Cayley* is necessary. We conjecture that there is a constant  $c > 0$  such that sampling pair of  $(1+c)\frac{1}{2}\Delta$  regular random graphs, and than take the anti-symmetry union of them might also obtain a good expander such that each of the residue part also has good expansion with heigh probability.

**Claim 37.** *Consider the graph  $G$  and the code  $C$  as defined in [19] and let  $S, T$  be a pair of disjointness vertices subsets. And let  $x_S$  and  $x_T$  codewords of the  $C_\oplus$  such that  $x_S$  suggested only by vertices in  $S$ , and in similar manner  $x_T$  suggested only by  $T$ 's vertices. Then the flux of  $S$  over  $T$  is at most:*

$$w_T(x_S) \leq E_{G'}(S, T) \leq \frac{1}{2}\Delta \frac{|S||T|}{n} + \lambda\sqrt{|S||T|}$$

*Proof.* The only edges that can interfere are the edge defined by  $J'$ , Namely the edges which belong to *Cayley*( $\Gamma, J'$ ). Therefore it's enough to use the mixing expander lemme on the  $\frac{1}{2}\Delta$ -regular graph.  $\square$

*Proof of Theorem 1+.* Notice that  $\frac{1}{2} < \frac{2}{3} = q$ , Thus repeating exactly over proof above obtains that:

$$\begin{aligned} w_{E/E'}(x|_U) &\geq \delta_0 q \Delta |U| - w\left(E(U_{-1} \cup U_{+1}, U)\right) \\ &\geq \delta_0 q \Delta |U| - \frac{\Delta}{2} \frac{|U| (|U_{-1}| + |U_{+1}|)}{n} - \lambda \sqrt{|U| (|U_{-1}| + |U_{+1}|)} \\ &\geq \left(\delta_0 - \frac{2}{3} - \frac{1}{q} \frac{\lambda}{\Delta}\right) q \Delta |U| \end{aligned}$$

When the last inequality follows form the fact that the proof of ?? doesn't relay on graph structure arguments. The same arguments leads also to analog inequality for ??.

Choosing  $J$  such that *Cayley*( $\Gamma, J$ ) is ramnujan provide that  $\frac{2\lambda}{\Delta q}$  scale as  $\Theta\left(\frac{1}{\sqrt{\Delta}}\right)$ . That close the case in which there is a linear size layer of nontrivial suggestions. In other case, in which any such layer is at size less than  $\alpha'n$  ( $\alpha' = (\delta_0 - (q - \frac{1}{2}))$  ? ) then we obtain the testability for free  $\square$

# Bibliography

- [Ham50] R. W. Hamming. “Error detecting and error correcting codes”. In: *The Bell System Technical Journal* 29.2 (1950), pp. 147–160. DOI: [10.1002/j.1538-7305.1950.tb00463.x](https://doi.org/10.1002/j.1538-7305.1950.tb00463.x).
- [RS60] Irving S. Reed and Gustave Solomon. “Polynomial Codes Over Certain Finite Fields”. In: *Journal of The Society for Industrial and Applied Mathematics* 8 (1960), pp. 300–304.
- [Tan81] R. Tanner. “A recursive approach to low complexity codes”. In: *IEEE Transactions on Information Theory* 27.5 (1981), pp. 533–547. DOI: [10.1109/TIT.1981.1056404](https://doi.org/10.1109/TIT.1981.1056404).
- [Hae95] Willem H. Haemers. “Interlacing eigenvalues and graphs”. In: *Linear Algebra and its Applications* 226-228 (1995). Honoring J.J.Seidel, pp. 593–616. ISSN: 0024-3795. DOI: [https://doi.org/10.1016/0024-3795\(95\)00199-2](https://doi.org/10.1016/0024-3795(95)00199-2). URL: <https://www.sciencedirect.com/science/article/pii/0024379595001992>.
- [Sho95] Peter W. Shor. “Scheme for reducing decoherence in quantum computer memory”. In: *Phys. Rev. A* 52 (4 Oct. 1995), R2493–R2496. DOI: [10.1103/PhysRevA.52.R2493](https://doi.org/10.1103/PhysRevA.52.R2493). URL: <https://link.aps.org/doi/10.1103/PhysRevA.52.R2493>.
- [CS96] A. R. Calderbank and Peter W. Shor. “Good quantum error-correcting codes exist”. In: *Physical Review A* 54.2 (Aug. 1996), pp. 1098–1105. DOI: [10.1103/physreva.54.1098](https://doi.org/10.1103/physreva.54.1098). URL: <https://doi.org/10.1103/physreva.54.1098>.
- [Gro96] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: [quant-ph/9605043](https://arxiv.org/abs/quant-ph/9605043) [quant-ph].
- [SS96] M. Sipser and D.A. Spielman. “Expander codes”. In: *IEEE Transactions on Information Theory* 42.6 (1996), pp. 1710–1722. DOI: [10.1109/18.556667](https://doi.org/10.1109/18.556667).
- [Sho97] Peter W. Shor. “Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer”. In: *SIAM Journal on Computing* 26.5 (Oct. 1997), pp. 1484–1509. DOI: [10.1137/s0097539795293172](https://doi.org/10.1137/s0097539795293172). URL: <https://doi.org/10.1137/s0097539795293172>.
- [AB99] Dorit Aharonov and Michael Ben-Or. *Fault-Tolerant Quantum Computation With Constant Error Rate*. 1999. arXiv: [quant-ph/9906129](https://arxiv.org/abs/quant-ph/9906129) [quant-ph].
- [AK99] Ashish Ahuja and Sanjiv Kapoor. *A Quantum Algorithm for finding the Maximum*. 1999. arXiv: [quant-ph/9911082](https://arxiv.org/abs/quant-ph/9911082) [quant-ph].
- [Den+02] Eric Dennis et al. “Topological quantum memory”. In: *Journal of Mathematical Physics* 43.9 (Sept. 2002), pp. 4452–4505. DOI: [10.1063/1.1499754](https://doi.org/10.1063/1.1499754). URL: <https://doi.org/10.1063/1.1499754>.

- [HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. “Expander graphs and their applications”. In: *Bulletin of the American Mathematical Society* 43.4 (2006), pp. 439–561.
- [Got14] Daniel Gottesman. *Fault-Tolerant Quantum Computation with Constant Overhead*. 2014. arXiv: [1310.2984 \[quant-ph\]](#).
- [TZ14] Jean-Pierre Tillich and Gilles Zemor. “Quantum LDPC Codes With Positive Rate and Minimum Distance Proportional to the Square Root of the Block-length”. In: *IEEE Transactions on Information Theory* 60.2 (Feb. 2014), pp. 1193–1202. DOI: [10.1109/tit.2013.2292061](#). URL: <https://doi.org/10.1109/2Ftit.2013.2292061>.
- [LTZ15] Anthony Leverrier, Jean-Pierre Tillich, and Gilles Zemor. “Quantum Expander Codes”. In: *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, Oct. 2015. DOI: [10.1109/focs.2015.55](#). URL: <https://doi.org/10.1109/2Ffocs.2015.55>.
- [BGW19] Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. “Completeness Theorems for Non-Cryptographic Fault-Tolerant Distributed Computation”. In: *Providing Sound Foundations for Cryptography: On the Work of Shafi Goldwasser and Silvio Micali*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 351–371. ISBN: 9781450372664. URL: <https://doi.org/10.1145/3335741.3335756>.
- [PK21] Pavel Panteleev and Gleb Kalachev. *Asymptotically Good Quantum and Locally Testable Classical LDPC Codes*. 2021. DOI: [10.48550/ARXIV.2111.03654](#). URL: <https://arxiv.org/abs/2111.03654>.
- [AF22] “Maximum distance separable (MDS) code”. In: *The Error Correction Zoo*. Ed. by Victor V. Albert and Philippe Faist. 2022. URL: <https://errorcorrectionzoo.org/c/mds>.
- [Din+22] Irit Dinur et al. *Good Locally Testable Codes*. 2022. DOI: [10.48550/ARXIV.2207.11929](#). URL: <https://arxiv.org/abs/2207.11929>.
- [LZ22] Anthony Leverrier and Gilles Zémor. *Quantum Tanner codes*. 2022. arXiv: [2202.13641 \[quant-ph\]](#).