Understanding Quantumness And Testability.

David Ponarovsky

April 5, 2023

Contents

1 Introduction **7**

2 Codes **9**
2.1 Introduction . . . 9
2.1.1 Notations, Definitions, And Our Contribution . . . 9
2.1.2 Singleton Bound . . . 10
2.1.3 Tanner Code . . . 11
2.1.4 Expander Codes . . . 11

3 Locally Testable Codes. **13**
3.0.1 Polynomial Code. . . . 13

4 Quantum Error Correction Codes. **15**
4.1 Introduction. . . . 15

5 Good qLDPC and LTC. **17**
5.1 Decoding and Testing . . . 17

6 First Attempt For Good qLTC. **19**
6.1 The Polynomial-Code Is Not w-Robust. . . . 19

7 Local Majority $\neq$ Local Testability. **21**
7.0.1 Overcoming The Vanishing Rate. . . . 22

List of Figures

Chapter 1

Introduction

Chapter 2

Codes

2.1 Introduction

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

Non-formally, code is good if its distance and rate are scaled linearly in the encoded message length. In practice, one is also interested in implementing those 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.

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

First, we state the notations, definitions, and formal theorem in section 2. 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 should consider skipping directly to section 5, in which we provide our proof.

2.1.1 Notations, Definitions, And Our Contribution

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 ρn. 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 δ and ρ correspondingly.

Definition 2. *A **family of codes** is an infinite series of codes. Additionally, suppose the rates and relative distances converge into constant values ρ, δ. 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.*

2.1.2 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 $\rho \to 1$. 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 exits). Let's review the polynomial code family [RS60], which is a code over none binary field that achieve the Singleton Bound.

Note On Quantum Polynomial Code.

Let's define the code C such that any state in C is a coset of the polynomials at degree at most d shifted by $x \in \mathbb{F}_p$. In other words the codeword associated with x is the state $|\underline{c}\rangle = \sum_{\substack{f \in \mathbb{F}_d[x] \\ f(0)=0}} |c + f\rangle$.

The inner product between any d-degree polynomial with zero free coefficient is:

$$\langle f|x^j\rangle = \sum_{i\leq d}\langle a_i x^i|x^j\rangle = \sum_{i\leq d} a_i \mathbf{E}\left[x^i x^j\right] = \sum_{i\leq d} a_i \mathbf{1}_{i+j=_n 0}$$

[COMMENT] Say some words about the classily testability of the polynomial code, and why for quantum it doesn't work. (The dual space of polynomials of low degree is the subspace of all the polynomials with heigh degree.)

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.

2.1.3 Tanner Code

The constructions require two main ingredients: a graph Γ, and for simplicity, we will restrict ourselves to a Δ regular graph. Secondly, 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 E 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 Γ, C_0. Notice that if C_0 is a binary linear code, So C is.*

It's also worth mentioning that the first construction of good classical codes, due to Sipser and Shpilman, are Tanner codes over expanders graphs [SS96].

Theorem. *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 □

Setting C_0 To Be The Polynomial Code.

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

2.1.4 Expander Codes

We saw how a graph could give us arbitrarily long codes with a positive rate. We will show, Sipser's result that if the graph is also an expander, we can guarantee a positive relative distance. We notice that the name expander codes is coined for a more general version than the one we will present.

Definition 5. *Denote by λ the second eigenvalue of the adjacency matrix of the Δ-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:*

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

This bound is known as the Expander Mixining Lemma. We refer the reader to [HLW06] for more deatilied survery.

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 Tunner 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 Mixining Lemma we have that:

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

Namely, for any such sublinear weight string, x, the average of nontrivial edges for the vertex is less than λ. So there must be at least one vertex $v \in S$ that, on his local view, sets a string at a weight less than λ. 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 $\square$

Chapter 3

Locally Testable Codes.

Apart from distance and rate here, we interest also that the checking process will be robust. In particular, we wish that against significant errors, forgetting to perform a single check will sabotage the computation only with a tiny probability.

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-testability*** *$f(n)$ If there exists $\kappa > 0$ such that*

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

3.0.1 Polynomial Code.

Consider the field $\mathbb{F}_m$ for an arbitrary prime power $m = q^l$ greater than n. The polynomial codes relay on the fact that any two different polynomials in the ring $\mathbb{F}_m[x]$ at degree at most d different by at least $m - d + 1$ points. For example consider a polynomials pair at degree 1, namely two linear straight lines. If they are not identical than they have at most single intersection point, and the disagree on each of the $n - 1$ remaining points.

So by define the code to be the subspace contains all the polynomials at degree at most d, in such way that any codeword is an image of such polynomail encoded by n numbers, one can garntee 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 $a_0, a_1, a_2, \ldots a_n$ distinct points of the field $\mathbb{F}_m = R$ and define the code $C \subset R$ as follows:*

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

Observe that C is a linear code at length n over the aleph-bet $\mathbb{F}_m$. The following Lemma states the realtion between the maximal degree of the polynomials and the properites of the code.

Lemma 1. *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 equals to the dimension of the polynomials space at degree at most d which is spanned by the monomial base $e_0, e_1, e_2 \ldots e_d = 1, x \ldots x^d$ and therefore is $d + 1$. In addition suppose that f, g are different polynomials i.e $f \neq g$.

Hence $h = f - g$ is a non-0 polynomial at degree at most d and therefore has at most d roots. Namely at most d points in which f equals g and at least $n - d$ in which they disagree. Put in another way the distance between any two different codewords of the code is at least $n - d$. □

Lemma 2 (Testability Of Polynomail Code.)**.** *Let G be an algortithm that given $d+1$ points return the polynomail that pass throu****[COMMENT]*** *here Testability*

Proof.

$$\begin{aligned}\text{agree} &\leq \mathbf{Pr}_{x_1,x_2\ldots x_d\sim_U, x_{d+1}\sim\mathbb{F}}\left[(Gf)(x) = f(x)\right] \\ &\leq \mathbf{Pr}\left[(Gf)(x) = f(x)\right] \leq \varepsilon \\ 1 - \text{agree} &\geq 1 - \varepsilon\end{aligned}$$

□

Chapter 4

Quantum Error Correction Codes.

4.1 Introduction.

It's wide believed that quantum machines have a significant advantage over the classical in range of computational tasks[Gro96], [AK99]. Simple algorithms, which could be interpreted as the quantum version of scanning all the options cut the running time by square root of the classical magnitude.

Nevertheless, Shore shown a polynomial depth quantum circuit that solve the hidden abelian subgroup [Sho97], what is considered as breakthrough, as it made the computer science community to believe that a quantum computer might offer an exponential advantage.

Yet, even those that there is a general consensus about the superiority of ideal quantum computation model,it is still unclear that it feasible to implement such machine in the presence of noise. Still just point about the existences of noise is not powerful enough to cancel feasibility of computation and evidence of this is the fact that classical computers are also suffer form a certain rate of faults. Thus, for getting a full understandings of the hardness, let us compare two main reasons that made realize an hard task. First is the magnitude of the error rate, the classical computers also have errors, and sometimes when that happen we are wetness for systems failures (blue screen for example). The error rate of modern computers is so low such that the probability for error to propagate stay negligible even if the length of the computation is polynomial in the scale of what considered as reasonable input size. It's worth to mention, that in the area of exascale computing, when super computers preform around 10^{18} operations per second, It is hard to miss the faults. In quantum we become aware to their existences much more before.

The second difference, which is a really tricky point, is that quantum sates are sensitive for additional type of error. Along with the chance for bit flip error, quantum state might change their phase. For example, consider the initial state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, and suppose that due to noise the state transformed into $\frac{1}{\sqrt{4}}\left(\sqrt{3}\,|0\rangle + |1\rangle\right)$. While classical circuits are blind to such faults, namely their run would stay identical as no error occurs, Quantum circuits, usually, would affect and might fail. Furthermore, when planing a decoder for quantum error correction codes, If one is willing to use a classical code to defend against phase flips he has to make sure that the decoding itself doesn't translate bit flip errors.

Definition 8 (Bit and pahse 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 applied on one of our state's qubits. The bit-flip event could be thought and be treated as exactly as the standard bit-flip error in the classical regime. In similar manner, phase-flip occurs when the Pauli Z applied on one of the qubits.*

Notice that together with the identity I the set $\{I, X \otimes e_i, Z \otimes e_j\}_{i,j\in[n]}$ span the matrices act on n qubits.

However, even though quantum noise is so violent, It was proven that any ideal circuit at polynomial depth can be transformed to a robust circuit at poly-logarithmic cost [AB99]. Or in other words, There is a threshold, If the physicists would provide qubits and a finite gate set that suffer from a rate of noise below that threshold, than BQP, the class of polynomial time ideal quantum computation is feasible and could computed on realistic machine.

The basic ingredient in [AB99] was to show the existence of quantum error correction code, such that one can perform all the logic operations in a way that restrict present errors from propagate on. That allow them separate any operation of the computation into stages, one of them is the operation itself, another one is an error correction stage. That process become with an additional cost, in both space and time, yet it might decrease the probability that the final state at the end would be faulted. The trade-off between the resource needed to pay and the decreasing rate define the threshold. And if the balance is positive then one can repeat in recursion manner, and after a log-log iterations the failure probability decay to zero while the circuit would scale at most poly-logarithmic wide and depth factors.

Let's return back to repeation code presented in chapter 2. We would like to think about an analog, a first and natural attempt might considering to duplicate copies of the state, Unfortunately copying a general state is not a linear operation and therefore can not be done in the circuit model (and any other believed to be feasible). In particular there is no circuit U which duplicate simultaneity the states $|0\rangle, |1\rangle, |+\rangle, |-\rangle$.

To overcome the issue Shor came up with the nine-quibt code [Sho95], which in first glance might seems a naive straightforward implantation of "duplication", but instead use a clever insight about quantum's in general. Any operation can been seen as linear (and even unitary) operation over a subspace embedded in large enough dimensions. By quadric the dimension of the repeation code,..
[COMMENT] Here

in which the encode state is a tree copies of the encoding of $|pm\rangle$ states encoded in the repeation code.

Chapter 5

Good qLDPC and LTC.

Definition 9 (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:

Lemma 3 (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:*

Definition 10. *Let $C = \mathcal{T}(G, C_0)$. We say that $x \in C_\oplus$ is* ***reducable*** *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 reducable codeword then we say that x is* ***ireducable*** *.*

5.1 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 filliped 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 11. *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'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 $L_{e_v} \forall e \in$ 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 1. *Let L be the Potential list. If $F(L) = 0$ then $S(L) \in C$.*

Proof. Denote by $\phi(e) \subset \{L_{e_v}, L_{e_u}\}$ 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 $S(L)|_v = \sum_{u \sim v} \phi(\{v,u\}) = \sum_{u \sim v} L_{e_v} \in C_0 \Rightarrow S(L) \in C$ □

The decoding goes as follows. First, each vertex suggests the closet 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 12. *Let $\tau > 0, f : \mathbb{N} \to \mathbb{R}^+$, 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 $L \leftarrow$ 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 1. *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 1 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:

Chapter 6

First Attempt For Good qLTC.

6.1 The Polynomial-Code Is Not w-Robust.

One idea for constructing is to use the polynomial code instead C_0, The follow form the fact that if one pick 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 consider other way for proving the reduction Lemma.

Claim 2. *Let C_0 be the $[\Delta, d, \Delta - d]$ polynomial code. Then any code word in $\left(C_0^{\perp} \otimes C_0^{\perp}\right)^{\perp}$ 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 $\Delta + d$. □

Claim 3. *The dual-tensor polynomial code is not w-robust.*

Proof. Consider the following polynomial

Chapter 7

Local Majority $\neq$ Local Testability.

Claim 4. *Suppose that G is an expander graph with a second eigenvalue λ, then For any layer U there exist a layer U' such that:*

Claim 5. *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 $|U| > |U_{+1}|, |U_{-1}|$ □

Using claim 5, we have that $\left(|U_{+1}| + |U_{-1}|\right)/n < \frac{2}{3}$ and therefore:

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

Claim 6. *Consider $G = (V, E)$ a Δ-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 7. *Alternate proof of fulx inequality, which doesn't assume that there is no interference inside the layers. $w\left(E\left(U, U\right)\right) > 0$.*

Proof. Separeate 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:

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 fulx 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:

Proof of Theorem 1. Consider the size of the maxiaml layer $|U_{\max}|$ and sepearte to the following two cases. First, consider the case that $|U_{\max}| \geq \alpha n$ in that case it follows immedily by claim 4 that if $\delta_0 > \frac{2}{3} - \frac{2\lambda}{\Delta}$ there exists $\alpha' > 0$ such that:

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.

7.0.1 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:

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:

Yet, We still require more to prove a linear distance. By repeating on the *Singleton Bound* 2.1.2 proof it follows that the small code $\tilde{C}_0$ obtained by ignoring arbitrary $\left(q - \frac{1}{2}\right)\Delta$ coordinates yield a code with distance:

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

Definition 13 (Testability Tanner Code)**.** *Let $q > \frac{1}{2}$ and let J be a generator set for group Γ 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 Γ 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:*

Consider a codeword x and denote by x' the restriction of x to Cayley (Γ, 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 $\tilde{\delta}_0\Delta := \left(\delta_0 - \left(q - \frac{1}{2}\right)\right)\Delta$ 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* Cayley(Γ, J') *is an expander with second eigenvalue at most δ we have that:*

Claim 8 (Existence of such *Cayley's*)**.** *Let S be a generator set such that* Cayley(Γ, S) *has a second largest eigenvalue greater then λ, 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 2λ.*

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 Δ 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:

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:

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 9. *Consider the graph G and the code C as defined in [13] 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:*

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

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 $\alpha' n$ ($\alpha' = \left(\delta_0 - \left(q - \frac{1}{2}\right)\right)$?) then we obtain the testability for free ☐

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.

[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.

[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. URL: https://link.aps.org/doi/10.1103/PhysRevA.52.R2493.

[Gro96] Lov K. Grover. *A fast quantum mechanical algorithm for database search*. 1996. arXiv: 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.

[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. URL: https://doi.org/10.1137%2Fs0097539795293172.

[AB99] Dorit Aharonov and Michael Ben-Or. *Fault-Tolerant Quantum Computation With Constant Error Rate*. 1999. arXiv: quant-ph/9906129 [quant-ph].

[AK99] Ashish Ahuja and Sanjiv Kapoor. *A Quantum Algorithm for finding the Maximum*. 1999. arXiv: quant-ph/9911082 [quant-ph].

[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.

[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.

[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.

[LZ22] Anthony Leverrier and Gilles Zémor. *Quantum Tanner codes*. 2022. arXiv: 2202.13641 [quant-ph].