Simple Almost LTC Good LDPC Codes

David Ponarovsky

November 15, 2022

Abstract

We propose a new simple construction based on Tanner Codes, which yields a good LDPC code with testability query complexity of $\Theta\left(n^{1-\varepsilon}\right)$ for any $\varepsilon > 0$.

1 Preambles

In this work, we propose a new construction for good LDPC codes, which also have a good testability parameter. In the sense that one can engineer the code such that doing $\Theta\left(n^{1-\varepsilon}\right)$ random checks, at most, would be enough to detect any error with probability proportional to its size. When ε is a small, as we wish, fixed positive number. In contrast to previews, constructions made by C and D, our construction does not involve any amplification and is almost identical to the expander code construction, which is considered by many simple and easy to implement. We remark that Dinur, in her previous work [Din+22], obtained a better testability code with only constant query complexity. Yet, her construction relies on the existence of a specific square complex. That complex, which is also an essential ingredient of the Quantum good LDPC proof, allows engineering a restrictions system with high degeneration. This complex is also necessary to provide a structure of commuting Pauli cheks which is needed to define a quantum CSS code [PK21], [LZ22]. That fact raises the question: Could the preview constructions also provide such degeneration? What is the exact relation between Quantemness and Testability? Could one create a testability code without having a quantum code? We partially answer those questions above by showing that a slight adaptation of the Tanner-Expander code provides effective, as we wish, testability. Our proof also indirectly answers the following question. Why most of the good LDPC codes are known to be bad in terms of detecting errors? In other words, It seems that for most of them, there exist strings that are very far from being in the code and, meanwhile, fail to satisfy only a small number of restrictions. While the previous LDPC constructions focused on ensuring that the yielded code would have a good rate and distance parameters, our construction enforces the restrictions collection to have a nontrivial fraction of degeneration. That is, removing a single restriction will not change the code, as any restriction is linearly dependent on the others.

2 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 could recover the original message. Also, We could think about the code as all the strings that satisfy a specified restrictions collection. Non-formally, we say that code is good if its distance and rate are scaled linearly in the encoded message length. In practice, one might also intrested to be able to implement 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. And finally, another characteristic of the code is its testability, which is the complexity of the number of random checks one should be done in order to negate that a given candidate is in the code. Besides the fact that good codes are 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 in order 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 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. 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. 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 etiher 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. We will say that a family of codes is a *good code* if its parameters converge into positive values.

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. 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 exsits $\kappa > 0$ such that

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

Nowadays, we are aware of a wide range of constructions yield good codes, including the expander codes of Sipser and Spilman [SS96] and the LTC codes of Dinur [Din+22], [PK21], [LZ22]. Thus if a decade ago, the main question was the existence of a good code and its construction, now, and particularly in this work, we concentrate on getting a deep understanding of what makes those constructions work. By utilizing those insights, we seccussed in achieving a significantly simpler both constructions and their correctness proof. Our results:

Theorem: For every $\varepsilon > 0$ there is a family of good LDPC codes with $\Theta\left(n^{1-\varepsilon}\right)$ testability.

2.2 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 $C_p \subset \mathbb{F}_2^{n+1}$, which it's 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. In the next section, we will review the Singleton bound, which states that for any code (not necessarily good), there must be a zero-sum game between the relative distance and the rate. Now, we are ready to formulate our contribution.

Besides of been 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. 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.

2.3 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. 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 □

2.4 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. 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:

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

This bound is known as the Expander Mixining Lemma.

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, 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 a trivial string. 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 $\square$

2.5 Tanner testability.

This subsection will explain why testability is so hard to achieve. Let C be a good Tanner expander code as defined above. And consider an arbitrary vertex $u \in V$ and arbitrary restriction of C_0, h. Now define $\tilde{C}$ as the code obtained by requiring all the restrictions of C except h on u. That it, u is satisfied if his local view satisfies all the C_0 restrictions apart from h. Also, for convenience, denote the small code u enforces on his local view by C_0^u. Let us assume that the distance of C_0^u is at least $\delta\Delta$. Then, by repeating almost exactly the steps above with caution, one could prove that $\tilde{C}$ also has a linear distance.

Assume that $\tilde{C} \neq C \Rightarrow$ there exists $x \in \tilde{C}/C$. By definition, for any $v \in V/\{u\}$ it holdes that $x|_v \in C_0$. Hence, the assumption that $x \notin C$ implies $x|_u \notin C_0$ So, clearly, x fails at a constant number of C's checks. On the other hand, the closest codeword $y \in C$ to x is also a codeword of $\tilde{C}$ as $y|_v \in C_0$ for every $v \in V$. Hence:

$$d(x, C) = d(x, y) \geq d\left(\tilde{C}\right) = \Theta(n)$$

That is, even that a linear number of bits are needed to be flipped to correct x, only a single check observes that x is indeed an error.

3 Construction

3.1 Almost LTC With Zero Rate

Definition. 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.

Theorem 1. For every $\varepsilon > 0$, there exist $\Delta_\varepsilon \in \mathbb{N}$ and $\alpha > 0$ such that if $\Delta \geq \Delta_\varepsilon$, then for every $x \in C_\oplus$ at Hamming weight at most $\alpha|E|^{1-\varepsilon}$, there exists a vertex $v \in V$ and a small codeword $c_v \in C_0$ such that adding the assignment of it over the v's edges to x define the codeword $x + c_v$ such that $|x + c_v| < |x|$.

Proof. By induction over the number of vertices $V' \subset V$, which suggest a nontrivial codeword to x. Base, assume that there is just a single vertex $v \in V$ that suggests a nontrivial codeword $c_v \in C_0$. Then it's clear that $x = c_v$. And therefore, we have that $|x + c_v| = 0 < |x|$.

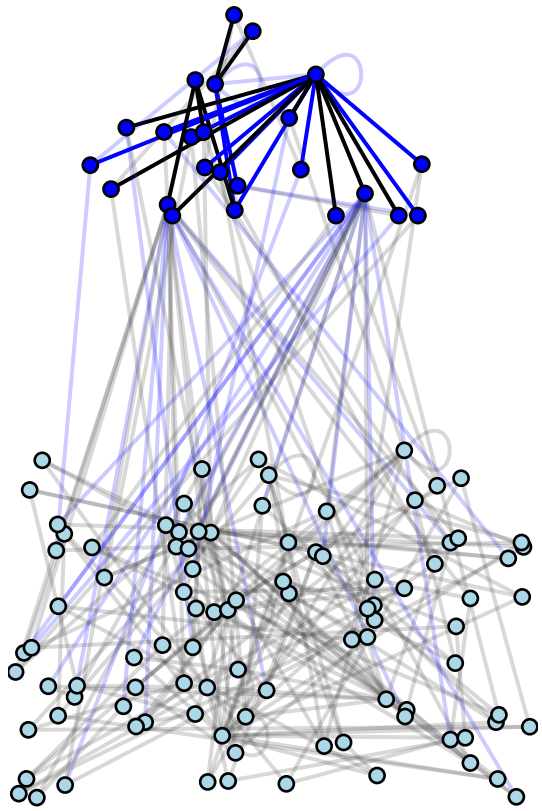
Assume the correctness of the argument for every codeword defined by at most m nontrivial suggestions made by $V' \subset V$. And consider the graph (V', E') induced by them. If the graph has more than a single connectivity component, then any of them is also a codeword of $C_\oplus$ but composed by at most $m - 1$ nontrivial suggestions. Therefore, by the assumption, we could find a vertex v and a proper small codeword $c_v \in C_0$, such that the addition of the suggestion will decrease the weight of the codeword defined on that component and therefore decrease the total weight of x.

So, we can assume that the vertices in V' compose a single connectivity component. Let be $x|_v \in \mathbb{F}_2^\Delta$ the bits of x on the indices corresponding to v's edges. If there is any v, with suggestion c_v, such that $\frac{1}{2}w(c_v) < w(x|_v)$, then we could pick to turn on c_v again and have that:

$$\begin{aligned}|x + c_v| = |x_{/v} + x_v + c_v| &= |x_{/v}| + w(c_v) + w(x|_v) \\ &< |x_{/v}| + \frac{1}{2}w(c_v) \leq |x_{/v}| + w(x|_v) = |x|\end{aligned}$$

Hence it is left to consider the case that for any $v \in V'$, it holds that $\frac{1}{2}w(c_v) > w(x|_v)$ (Notice that if they equal, then by turn on c_v, we back again to codeword made by $m - 1$ nontrivial suggestions). We will prove that this case is possible only for codewords with wight at least $\alpha|E|^{1-\varepsilon}$.

For any $S \subset E$, define $w_S(x)$ to be the weight that x induces over S. And notice that any edge of E connected only to a single vertex in V' equals the corresponding bit in the original suggestion made by c_v. Hence for every $v \in V'$, it holds that $w_{E/E'}(x|_v) = w_{E/E'}(c_v)$.



Claim. For any $v \in V'$ and corresponded suggestion c_v it holds that: $w_{E'}(c_v) \geq \frac{1}{2}\delta_0\Delta$.

Proof: By using the previews insight we get:

$$w_{E'}(c_v) = w(c_v) - w_{E/E'}(c_v) = w(c_v) - w_{E/E'}(x|_v)$$
$$\geq w(c_v) - w(x|_v) \geq \frac{1}{2}w(c_v) = \frac{1}{2}\delta_0\Delta$$

□

Consider an arbitrary vertex $r \in V'$, and consider the DAG obtained by the BFS walk over the subgraph (V', E') starting at r. Denote this directed tree by T.

Claim. . The size of T is at least:

$$|T| \geq \left(\frac{1}{4}\delta_0 - \frac{\lambda}{\Delta}\right)n$$

Proof: By the fact that for any $v \in T$ the degree of v is at least $\frac{1}{2}\delta_0\Delta$ we have that: $E(T,T) \geq \frac{1}{2}\cdot\frac{1}{2}\delta_0\Delta|T|$. Combine the Mixining Expander Lemma we obtain:

$$\frac{1}{4}\delta_0\Delta|T| \leq \frac{\Delta}{n}|T|^2 + \lambda|T|$$
$$\Rightarrow \left(\frac{\Delta}{n}|T| + \lambda - \frac{1}{4}\delta_0\Delta\right)|T| \geq 0$$
$$\Rightarrow |T| \geq \left(\frac{1}{4}\delta_0 - \frac{\lambda}{\Delta}\right)n$$

□ Denote by $x|_U$ the bits of $x \in C$ correspond to the edges that connected to at least one vertex in U. And denote by $w_{E/E'}(x|_U)$ the weight induced by the $x|_U$ over the edges in E/E'.

Claim. Suppose that G is an expander graph with a second eigenvalue λ, 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:

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

Claim. 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 size grater than the inital layer and $|U| > |U_{+1}|, |U_{-1}|$ □

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

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

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

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

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

Theorem 1+. For every $\varepsilon > 0$, there exist $\Delta_\varepsilon \in \mathbb{N}$ and $\alpha > 0$ such that if $\Delta \geq \Delta_\varepsilon$, then for every $x \in C_\oplus$ at Hamming weight at most $\alpha|E|^{1-\varepsilon}$, there exists a vertex $v \in V_\pm$ and a small codeword $c_v \in C_0^\pm$ such that adding the assignment of it over the v's edges to x define the codeword $x + c_v$ such that $|x + c_v| < |x|$.

Proof: Define the T precisely as is in the proof of Theorem 1, And assume, without loss of generality $\delta_- < \delta_+$. Hence it is clear that for any $v \in V'$ it holdes that $w_{E'}(c_v) > \frac{1}{2}\delta_-\Delta$, so we could repeat exactly the above steps to get that there exists a layer $U \subset T$ at height $\frac{3}{4}g$ such that $|U| = \Omega\left(n^{1-\varepsilon}\right)$ for large enough Δ. Furthermore:

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

Claim.

$$|U||U_{-1}| + |U||U_{+1}| + |U_{+1}||U_{+2}| \leq \left(|U| + |U_{+1}|\right)\frac{3}{4}n$$

Proof: Recall that $|U_{-1}| + |U_{+1}| = |U_{-1} \cup U_{+1}| \leq |V_\pm| = \frac{1}{2}n$. Then:

$$\begin{aligned} & |U||U_{-1}| + |U||U_{+1}| + |U_{+1}||U_{+2}| = \\ & \frac{1}{2}\left(|U|(|U_{-1}| + |U_{+1}|) + |U_{+1}||U_{+2}|\right) + \\ & \frac{1}{2}\left(|U||U_{-1}| + |U_{+1}|(|U| + |U_{+2}|)\right) \leq \\ & \frac{1}{2}\left(|U_{-1}||U| + |U_{+1}||U_{+2}|\right) + \frac{1}{2}n\left(|U| + |U_{+1}|\right) \\ & \leq \left(\frac{1}{4}n + \frac{1}{2}n\right)\left(|U| + |U_{+1}|\right)\square \end{aligned}$$

Also, we could again assume that $|U| \geq |U_{-1}|, |U_{+1}|, |U_{+2}|$ and then:

$$\begin{aligned} \sqrt{|U_i||U_j|} = \frac{\sqrt{|U_i||U_j|}}{|U| + |U_{+1}|} \cdot \left(|U| + |U_{+1}|\right) \leq \\ \frac{\sqrt{|U_i||U_j|}}{|U|} \cdot \left(|U| + |U_{+1}|\right) \leq \left(|U| + |U_{+1}|\right) \end{aligned}$$

So in total, we have that:

$$w_{E/E'}\left(x|_{U\cup U_{+1}}\right) \geq \left(\delta_- + \delta_+ - \frac{3}{4} - \frac{4\lambda}{\Delta}\right)\Delta\left(|U| + |U_{+1}|\right)$$

And the fact that $|x| \geq w_{E/E'}\left(x|_{U\cup U_{+1}}\right)$ proves Theorem 1+ $\square$

3.2 Tanner Code Testability For Small $\left(n^{\frac{2}{3}-\varepsilon}\right)$ Errors

Another side result obtained by the proof of Theorem 1 is that one cannot hide errors in a tree at a length less than half of the girth. Namely, any codeword of the disagreement code with a weight less than $\omega^{\frac{1}{2}g}$ could be reduced, regardless of the parameters of the small code.

Theorem 1-. For every $\delta_0, \varepsilon > 0$, there exist $\Delta_\varepsilon \in \mathbb{N}$ and $\alpha > 0$ such that if $\Delta \geq \Delta_\varepsilon$, then for every $x \in C_\oplus$ at Hamming weight at most $\alpha|E|^{\frac{g}{2}-\varepsilon}$, there exists a vertex $v \in V$ and a small codeword $c_v \in C_0$ such that adding the assignment of it over the v's edges to x define the codeword $x + c_v$ such that $|x + c_v| < |x|$.

Proof: Proof, Define T again to be the DAG constructed as in the proof of Theorem 1 and denote by U the last layer of T. As the height of the tree is less than $\frac{1}{2}g$, then it follows from the claim that any vertex v in U has at most one parent in T, and therefore:

$$w_{E/E'}\left(x|_v\right) \geq w(c_v) - 1 \geq \frac{1}{2}w(c_v)$$

Assuming Girth is greater than $\frac{4}{3}\log_{\Delta-1}(n)$ gives us a decoder against errors at weight at most $n^{\frac{3}{2}-\varepsilon}$ $\square$.

4 Decodeing 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. 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'th 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 $F(L) = \sum_{e=\{v,u\}\in E}\left(L_{e_v} + L_{e_u}\right)$ 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. 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 $S(L)|_v = \sum_{u\sim v}\phi(\{v, u\}) = \sum_{u\sim v} L_{e_v} \in C_0 \Rightarrow S(L) \in C\square$

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. 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 $\mathcal{D}$.

Algorithm 1: Decoding

Data: $x \in \mathbb{F}_2^n$

Result: $\arg\min\{y \in C : |y + x|\}$ if $d(y, C) < \tau$ and False otherwise.

$L \leftarrow \text{Array}\{\}$
for $v \in V$ **do**
  $c'_v \leftarrow \arg\min\{y \in C_0 : |y + x|_v|\}$
  $L_v \leftarrow c'_v$
$z \leftarrow \sum_{v \in V} c'_v$
if $|z| < \tau \frac{n}{f(n)}$ **then**
  while $|z| > 0$ **do**
    find v and $c \in C_0$ such that $|z + c_v| < |z|$
    $z \leftarrow z + c_v$
    $L_v \leftarrow L_v + c_v$
else
  reject.
return $S(L)$

Theorem. 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 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 δ. To obtain the correctness of the second section, we will separate when the conditional at 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 the line (5) is satisfied. In that case, $\mathcal{D}$ decods x into it's closest codeword in C. Therefore:

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

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 $\square$

References

[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.

[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.

[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.

[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].