Simple Almost LTC Good LDPC Codes

David Ponarovsky

October 28, 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

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.

In this work, we propose a new construction for good LDPC codes, which also have a good testability parameter. 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 last work [Din+22], obtained a better testability code with only constant query complexity. Yet, here construction requires using algebraic structures, which complicates the process. 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.

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 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.

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 how to construct a good code, then now it is replaced by what is the best code one could expect.

To get a feeling of the behavior of the distance-rate tradeof, 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.

Theorem 1 Let C be a family of good codes, with parameters $[n, \rho n, \delta n]$. Then:

$$\rho + \frac{5}{16}\delta \leq \frac{3}{4} + \frac{1}{16}$$

3 Singleton Bound

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.

4 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 □

5 Construction

5.1 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:

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

□

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. Let g be the girth of the graph, and consider a layer U in T at height $h(U)$ satisfies the inequality $h(U) < \frac{1}{2}g + l$ for some integer l.

Claim. . The number of vertices with more than a single parent in T is at most $\binom{\Delta^l}{2}$.

Proof: Assume, towards contradiction, that there are more than $\binom{\Delta^l}{2}$ in U that has at least two parents, And consider all the subtrees of T whose roots are placed in height l. There are, at most Δ^l, such subtrees. Hence, by the assumption, there must be a pair of vertices in U with parents in the same subtree. Because each subtree has a height at most half of g, there is a path connecting those parents at length at most g. Then, one can concatenate that path with the corresponding vertices in U and get a cycle shorter than

the girth. □

Lemma . Let U be a layer at height l and denote by $\omega = \frac{1}{2}\delta_0\Delta$. The number of vertices in U is at least:

$$\left(\frac{1}{2}\delta_0\Delta\right)^l - \frac{\left(\frac{\Delta^2}{\omega}\right)\Delta^{2l-g} - \omega^{l-g/2}}{\frac{\Delta^2}{\omega}-1}$$

Proof: We will think of T as the tree obtained by purning a tree, with degree greater than $\frac{1}{2}\delta_0\Delta$, at height l, namely any vertex at height $i \le l$ with two parents can be counted as subtree with height $l-i$ that has pruned. Then the problem is equivalence to bounding from above the number of pruned vertices at height l. Define X_i to be the number of pruned subtrees whose root is at height i. Then consider the next maximization problem:

$$X_i \le \Delta^{2\max\{0,l-g/2\}}$$

$$f \le \sum_{g/2}^{l} \omega^{l-i} X_i \le \omega^{l-g/2} \sum_{0}^{l-g/2} \omega^{-i} \Delta^{2i} =$$

$$\omega^{l-g/2}\frac{\left(\frac{\Delta^2}{\omega}\right)^{l-g/2+1}-1}{\frac{\Delta^2}{\omega}-1} = \frac{\left(\frac{\Delta^2}{\omega}\right)\Delta^{2l-g}+\omega^{l-g/2}}{\frac{\Delta^2}{\omega}-1}$$

By using the inequality $|U| \ge \omega^l - f$ we get the bound □

Claim. there exists a constant Δ' that depends only on δ_0, such that if $\Delta \ge \Delta'$, then the number number of vertices at height $\frac{3}{4}g$ is at least $\frac{1}{2}\omega^{\frac{3}{4}g}$.

Proof: The above lower bound yields:

$$|U| \ge \omega^{\frac{3}{4}g} - \frac{\left(\frac{\Delta^2}{\omega}\right)\Delta^{\frac{1}{2}g}}{\frac{\Delta^2}{\omega}-1} \ge \omega^{\frac{3}{4}g} - \frac{2\Delta^{\frac{1}{2}g}}{\frac{\Delta^2}{\omega}-1}$$

So it is enough to require that: $\left(\frac{1}{2}\delta_0\Delta\right)^{\frac{3}{4}g} \ge 4\Delta^{\frac{1}{2}g}$:

$$\Delta^{\frac{1}{4}g} \ge 4\cdot\left(\frac{2}{\delta_0}\right)^{\frac{3}{4}g} \Rightarrow \Delta \ge 4^{\frac{4}{g}}\left(\frac{2}{\delta_0}\right)^3$$

So any Δ satisfies $\Delta > \left(\frac{2}{\delta_0}\right)^3$ is also satisfying the above inequality, and therefore for such Δ-regular graphs, the size of U will be at least $\frac{1}{2}\omega^{\frac{3}{4}g}$ □

Claim. Assume that $g \ge \frac{4}{3}\log_{\Delta-1}(n)$, Then we have

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:

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

Claim.

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

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

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

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

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

So in total, we have that:

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

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

5.3 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