Improved Decoding of Tanner Codes

Zhaienhe Zhou
School of the Gifted Young & College of Computer Science
University of Science and Technology of China
Hefei 230026, China.
Email: zhaienhezhou@gmail.com

Zeyu Guo
Department of Computer Science and Engineering
The Ohio State University
Columbus, OH, USA.
Email: zguotcs@gmail.com

***Abstract*—In this paper, we present improved decoding algorithms for expander-based Tanner codes.**

We begin by developing a randomized linear-time decoding algorithm that, under the condition that $\delta d_0 > 2$, corrects up to αn errors for a Tanner code $T(G, C_0)$, where G is a (c, d, α, δ)-bipartite expander with n left vertices, and $C_0 \subseteq \mathbb{F}_2^d$ is a linear inner code with minimum distance d_0. This result improves upon the previous work of Cheng, Ouyang, Shangguan, and Shen (RANDOM 2024), which required $\delta d_0 > 3$.

We further derandomize the algorithm to obtain a deterministic linear-time decoding algorithm with the same decoding radius. Our algorithm improves upon the previous deterministic algorithm of Cheng et al. by achieving a decoding radius of αn, compared with the previous radius of $\frac{2\alpha}{d_0(1+0.5c\delta)}n$.

Additionally, we investigate the size-expansion trade-off introduced by the recent work of Chen, Cheng, Li, and Ouyang (IEEE TIT 2023), and use it to provide new bounds on the minimum distance of Tanner codes. Specifically, we prove that the minimum distance of a Tanner code $T(G, C_0)$ is approximately $f_\delta^{-1}\left(\frac{1}{d_0}\right)\alpha n$, where $f_\delta(\cdot)$ is the Size-Expansion Function. As another application, we improve the decoding radius of our decoding algorithms from αn to approximately $f_\delta^{-1}\left(\frac{2}{d_0}\right)\alpha n$.

I. Introduction

Tanner codes are constructed by assigning a linear inner code C_0 of length d and minimum distance d_0 to the vertices of a sparse bipartite graph. Specifically, bits are placed on the left side of the bipartite graph, and each vertex on the right side is assigned an inner code that imposes constraints on the connected bits. To analyze the decoding algorithms of LDPC and Tanner codes, Sipser and Spielman [1] introduced the concept of vertex expansion. Expander codes are a special class of Tanner codes constructed from (c, d, α, δ)-bipartite expanders, where c, d, α, and δ are constants. Specifically, the graph $G = (L \cup R, E)$ is left-regular of degree c and right-regular of degree d, and for any $S \subseteq L$ with $|S| \leq \alpha n$, the number of neighbors of S is at most $\delta c|S|$. Expander codes are known for their efficient decoding algorithms, which can correct $\Omega(n)$ errors in linear time. Research [1]–[8] has focused on optimizing the decoding radius and other parameters while keeping the decoding algorithm linear-time.

Consider the special case where the inner code is a parity-check code. In this case, the flip algorithm introduced by Sipser and Spielman [1] can decode up to $(2\delta - 1)\alpha n$ errors in linear time for any expander code with $\delta > \frac{3}{4}$. Later, Viderman [6] proposed a new decoding method, which corrects up to $\frac{3\delta-2}{2\delta-1}\alpha n$ errors in linear time when $\delta \geq \frac{2}{3}$.

More recently, Chen, Cheng, Li, and Ouyang [8] gave an improved decoding algorithm by combining previous approaches and introducing a method they term "expansion guessing." They also discovered a size-expansion trade-off, which enables the expansion of larger sets to be inferred from smaller sets. They showed that expander codes achieve a minimum distance of $\frac{1}{2(1-\delta)}\alpha n$, and their decoding algorithm achieves a decoding radius of $\frac{3}{16(1-\delta)}\alpha n$, which is nearly half of the code's distance. However, their algorithm still requires $\delta > \frac{3}{4}$ to use the flip algorithm. This raises an open question: What is the minimum δ required to decode a linear number of errors in linear time? It was shown in [6] that $\delta > \frac{1}{2}$ is necessary.

The above studies focus on the special case of expander codes where the inner code C_0 is a parity-check code. Progress has also been made on the general case [9]–[11]. Notably, Dowling and Gao [10] proved that the condition $d_0\delta^2 = \Omega(c)$ is sufficient for error correction using a flip-based decoding algorithm. More recently, Cheng, Ouyang, Shangguan, and Shen [11] improved this result by showing that $\delta d_0 > 3$ is sufficient for error correction, while $\delta d_0 > 1$ is necessary.

However, many questions remain open about the optimal parameters. In particular, there is still a gap between the sufficient and necessary conditions for δd_0 to enable a linear-time decoding. In this paper, we narrow this gap by proving that $\delta d_0 > 2$ is sufficient for expander-based Tanner codes.

A. Main Results

Let $T(G, C_0)$ be a Tanner code based on a bipartite expander G and an inner code C_0 (see Definition 2.2). Our first main result is a deterministic linear-time decoding algorithm for $T(G, C_0)$.

Theorem 1.1 (Informal version of Theorem 4.6): Suppose $\delta d_0 > 2$. There exists a deterministic $O(n)$-time algorithm that corrects up to αn errors for any Tanner code $T(G, C_0) \subseteq \mathbb{F}_2^n$, where G is a (c, d, α, δ)-bipartite expander and C_0 is an inner code with minimum distance d_0.

Previously, under the condition $\delta d_0 > 3$, Cheng et al. [11] gave a randomized $O(n)$-time decoding algorithm that corrects αn errors, as well as a deterministic $O(n)$-time algorithm with a smaller decoding radius $\frac{2\alpha}{d_0(1+0.5c\delta)}n$. Theorem 1.1 relaxes the condition to $\delta d_0 > 2$ and derandomizes the randomized decoding algorithm without reducing the decoding radius αn.

We also investigate the size-expansion trade-off introduced by [8]. Specifically, we define the Size-Expansion Function

$f_\delta(k)$ (Definition 5.1), which satisfies the following property: Any (c, d, α, δ)-bipartite expander is also a $(c, d, k\alpha, f_\delta(k))$-bipartite expander for $k > 1$. Consequently, our decoding algorithm achieves a decoding radius of approximately $f_\delta^{-1}\left(\frac{2}{d_0}\right)\alpha n$, which is strictly larger than αn.

Theorem 1.2 (Informal version of Theorem 5.5): Theorem 1.1 still holds with the decoding radius increased to approximately $f_\delta^{-1}\left(\frac{2}{d_0}\right)\alpha n$.

Finally, we establish the following tight bound on the minimum distance of $T(G, C_0)$:

Theorem 1.3 (Informal version of Theorems 5.4 and 5.6): Suppose $\delta d_0 > 1$. The minimum distance of $T(G, C_0)$ is at least approximately $f_\delta^{-1}\left(\frac{1}{d_0}\right)\alpha n$. This lower bound is tight in the sense that it is achieved by infinitely many examples.

II. Preliminaries

For $n \in \mathbb{N}$, denote by $[n]$ the set $\{1, 2, \ldots, n\}$.

a) Codes: All codes in this paper are assumed to be *Boolean linear codes*. The Hamming weight of $x \in \mathbb{F}_2^n$ is denoted $\mathrm{wt}(x)$. The Hamming distance between $x, y \in \mathbb{F}_2^n$ is $d_H(x, y) :- \mathrm{wt}(x - y)$. The *minimum distance* of a code C is $d_H(C) :- \min\{d_H(x, y) : x, y \in C, x \neq y\}$.

b) Bipartite graphs and expanders: A bipartite graph $G = (L \cup R, E)$ is called (c, d)-*regular* if $\deg(u) = c$ for all $u \in L$ and $\deg(v) = d$ for all $v \in R$.

For $S \subseteq L \cup R$, let $N(S)$ denote the set of all neighbors of S. Define $N_i(S)$ as the set of vertices adjacent to exactly i vertices in S. Additionally, define $N_{\geq i}(S) \coloneqq \bigcup_{j \geq i} N_j(S)$ and $N_{\leq i}(S) \coloneqq \bigcup_{j \leq i} N_j(S)$. Define $E(S, T)$ as the set of edges connecting the two vertex sets S and T.

Definition 2.1 (Bipartite expander): A (c, d, α, δ)-bipartite expander is a (c, d)-regular bipartite graph $G = (L \cup R, E)$ such that $|N(S)| \geq \delta c|S|$ for any $S \subseteq L$ with $|S| \leq \alpha|L|$. For $\emptyset \neq S \subseteq L$, call $\frac{N(S)}{c|S|}$ the *expansion factor* of S.

Definition 2.2 (Tanner code): Let C_0 be a code of length d. Let $G = (L \cup R, E)$ be a (c, d, α, δ)-bipartite expander, where $L = [n]$ for some $n \in \mathbb{N}^+$. For each $v \in R$, fix a total ordering on $N(v)$, and let $N(v, i)$ denote its i-th element for $i \in [d]$. For $x \in \mathbb{F}_2^n$ and $v \in R$, define

$$x_{N(v)} \coloneqq (x_{N(v,1)}, \ldots, x_{N(v,d)}) \in \mathbb{F}_2^d.$$

The Tanner code $T(G, C_0)$ is defined as

$$T(G, C_0) \coloneqq \{x \in \mathbb{F}_2^n : x_{N(v)} \in C_0 \text{ for all } v \in R\} \subseteq \mathbb{F}_2^n.$$

Throughout this paper, we fix positive integers c, d and real numbers $\alpha, \delta \in (0, 1]$ as constants. Also, let $G = (L \cup R, E)$ be a (c, d, α, δ)-bipartite expander with $L = [n]$, and let C_0 be a code of length d with minimum distance d_0. All lemmas and theorems are stated under the assumption that G and C_0 are given, without explicitly mentioning this.

For convenience, we introduce the following definition.

Definition 2.3 (Corrupt bits and unsatisfied checks): For $x, y \in \mathbb{F}_2^n$, define $F(x, y) = \{i \in [n] : x_i \neq y_i\}$. Define $F(x) = F(x, y)$, where y is the closest codeword to x in $T(G, C_0)$ with respect to the Hamming distance. (If there are multiple closest codewords, y is chosen to be the lexicographically smallest one.)

Let $U(x) \subseteq R$ denote the set of unsatisfied checks, defined as $U(x) = \{v \in R : x_{N(v)} \notin C_0\}$.

Finally, We present some useful auxiliary lemmas. The proofs are omitted and can be found in the full version [12].

Lemma 2.4: For any $S \subseteq L$ with $|S| \leq \alpha n$ and integer $t \geq 0$, it holds that $|N_{\leq t}(S)| \geq \frac{\delta(t+1)-1}{t} \cdot c|S|$.

Lemma 2.5: Let $x \in \mathbb{F}_q^n$ and $y \in T(G, C_0)$ such that $d_H(x, y) \leq \alpha n$. Let $F = F(x, y)$. Then $c|F| \geq |U(x)| \geq |N_{\leq d_0 - 1}(F)| \geq \frac{\delta d_0 - 1}{d_0 - 1} \cdot c|F|$.

Lemma 2.6: For any $S \subseteq L$, $|N_{\geq t}(S)| \leq \frac{c}{t}|S|$.

Lemma 2.7: Suppose G is a (c, d, α, δ)-bipartite expander and the inner code C_0 has distance d_0. If $\delta d_0 > 1$, then the distance of $T(G, C_0)$ is greater than αn.

III. Randomized Decoding

In this section, we present an improved randomized flipping algorithm and extend it to a randomized decoding algorithm. We follow the approach of [11] which uses the following idea: Let each unsatisfied check v cast a "vote" on which bits to flip. Then, each bit is flipped with a probability determined by the votes it receives. This process corrects a constant fraction of errors. By repeating it logarithmically many times, the received word can be corrected with high probability.

Our improvement is achieved by allowing each v to send a weighted vote based on $d_H(x_{N(v)}, y)$ when $d_H(x_{N(v)}, y) < d_0/2$, where $y \in C_0$ is the closest codeword to $x_{N(v)}$, rather than using an unweighted vote when $d_H(x_{N(v)}, y) < d_0/3$, as was done in [11]. (At a high level, this bears some similarity with the GMD decoding algorithms for concatenated codes [13], where a large $d_H(x_{N(v)}, y)$ suggests that y is likely incorrect.) This modification enables a tighter analysis.

A. Randomized Flipping

Let $\mathsf{Decode}(x)$ denote the function that, given $x \in \mathbb{F}_2^d$, returns $y \in C_0$ closest to x in Hamming distance, with ties broken by selecting the lexicographically smallest y.

We now present the randomized flipping algorithm.

Algorithm 1 $\mathsf{RandFlip}(x)$

Input: $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$, where $n = |L|$.

$t \leftarrow \frac{d_0}{2}$
$(p_1, \ldots, p_n) \leftarrow (0, \cdots, 0) \in \mathbb{R}^n$
for each $v \in R$ **do**
    $w_v \leftarrow \mathsf{Decode}(x_{N(v)})$
    if $1 \leq d_H(w_v, x_{N(v)}) < t$ **then**
        Choose the smallest $i \in N(v)$ where w_v and $x_{N(v)}$ differ
        $p_i \leftarrow p_i + \frac{t - d_H(w_v, x_{N(v)})}{ct}$
    end if
end for
for each $i \in [n]$ **do**
    Flip x_i with probability p_i
end for
return x

It is easy to see $p_i \in [0,1]$ at Line 11, ensuring the validity of this line. A proof can be found in the full version [12].

Theorem 3.1: Assume $d_0\delta > 2$. Let $\varepsilon_0 = \frac{d_0}{2} - \frac{1}{\delta} > 0$. Let $x \in \mathbb{F}_2^n$ and $y \in T(G, C_0)$ such that $d_H(x,y) \le \alpha n$. Let x' be the output of Algorithm 1 with x as input. Then $\mathbb{E}[d_H(x',y)|] \le (1 - \frac{\varepsilon_0\delta}{t}) d_H(x,y)$.

To prove Theorem 3.1, we need the following lemma. Recall that $F(x,y) = \{i \in [n] : x_i \neq y_i\}$.

Lemma 3.2: Let $x \in \mathbb{F}_2^n$, $y \in T(G, C_0)$, and $F = F(x,y)$. Let $v \in N_k(F)$ for some integer k. Let $w_v = \mathsf{Decode}(x_{N(v)})$ as in Algorithm 1. If $w_v = y_{N(v)}$, then $d_H(w_v, x_{N(v)}) = k$. On the other hand, if $w_v \neq y_{N(v)}$, then $d_0 - k \le d_H(w_v, x_{N(v)}) \le k$. The latter case occurs only if $k \ge \frac{d_0}{2} = t$, i.e., $v \in N_{\ge t}(F)$.

Proof: By the definition of F and the choice of v, we have $d_H(y_{N(v)}, x_{N(v)}) = k$. As $y \in T(G, C_0)$, we have $y_{N(v)} \in C_0$. As w_v is a vector in C_0 closest to $x_{N(v)}$, we have

$$d_H(w_v, x_{N(v)}) \le d_H(y_{N(v)}, x_{N(v)}) = k.$$

If $w_v = y_{N(x)}$, then $d_H(w_v, x_{N(v)}) = d_H(x_{N(v)}, y_{N(v)}) = k$. On the other hand, if $w_v \neq y_{N(x)}$, then the distance between these two codewords of C_0 is at least d_0, which implies $d_H(w_v, x_{N(v)}) \ge d_H(w_v, y_{N(v)}) - d_H(x_{N(v)}, y_{N(v)}) \ge d_0 - k$. This proves the lemma. ∎

Now we are ready to prove Theorem 3.1.

Proof of Theorem 3.1:

Let $F = \{i \in [n] : x_i \neq y_i\}$, whose size is $d_H(x,y) \le \alpha n$. By and linearity of expectation, we have

$$\mathbb{E}[d_H(x',y)] = |F| - \left(\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i \right). \tag{1}$$

Consider any $v \in R$. In the iteration of the first loop corresponding to v, some p_i may increase by $\frac{t - d_H(w_v, x_{N(v)})}{ct}$. We analyze how this affects the quantity $\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i$. Suppose v has k neighbors in F, i.e., $v \in N_k(F)$.

Case 1: $k = 0$. In this case, $d_H(w_v, x_{N(v)}) = 0$. Due to the condition $1 \le d_H(w_v, x_{N(v)}) < t$ at Line 5, the iteration corresponding to v does not affect $\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i$.

Case 2: $1 \le k < t$. In this case, we have $d_H(w_v, x_{N(v)}) = k \in [1, t)$ and $w_v = y_{N(v)}$ by Lemma 3.2. In the iteration corresponding to v, the index i chosen at Line 6 is in F since $w_v = y_{N(v)}$. Thus, this iteration contributes exactly $\frac{t - d_H(w_v, x_{N(v)})}{ct} = \frac{t-k}{ct}$ to $\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i$.

Case 3: $t \le k < d_0$. In this case, we have $d_H(w_v, x_{N(v)}) \ge d_0 - k = 2t - k$ by Lemma 3.2. Thus, the iteration corresponding to v contributes at least $-\frac{t - d_H(w_v, x_{N(v)})}{ct} \ge -\frac{t-(2t-k)}{ct} = \frac{t-k}{ct}$ to $\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i$.

Case 4: $k \ge d_0$. In this case, we have $d_H(w_v, x_{N(v)}) \ge 0$. Thus, the iteration corresponding to v contributes at least $-\frac{t - d_H(w_v, x_{N(v)})}{ct} \ge -\frac{t}{ct} \ge \frac{t-k}{ct}$ to $\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i$, where the last inequality uses the fact that $k \ge d_0 = 2t$.

By the above discussion, We have

$$\sum_{i \in F} p_i - \sum_{i \in [n] \setminus F} p_i \ge \sum_{k=1}^{d} \frac{t-k}{ct} |N_k(F)|. \tag{2}$$

By the definition of $N_k(\cdot)$ and the fact that G is left-regular of degree c, we have

$$\sum_{k=1}^{d} k|N_k(F)| = |E(F, N(F))| = c|F|. \tag{3}$$

As $|F| \le \alpha n$ and G is a (c, d, α, δ)-bipartite expander,

$$\sum_{k=1}^{d} |N_k(F)| = |N(F)| \ge \delta c |F|. \tag{4}$$

Multiplying both sides of (4) by $t = \frac{1}{\delta} + \varepsilon_0$ and subtracting both sides of (3), we obtain

$$\sum_{k=1}^{d} (t-k)|N_k(F)| \ge \varepsilon_0 \delta c |F|. \tag{5}$$

Combining (1), (2), and (5) shows $\mathbb{E}[d_H(x',y)] \le (1 - \frac{\varepsilon_0\delta}{t})|F| = (1 - \frac{\varepsilon_0\delta}{t}) d_H(x,y)$, as desired. ∎

Following analyses similar to those in [1], [10], [11], we have the following lemma, which bounds the time complexity of Algorithm 1. Its proof is provided in the full version [12].

Lemma 3.3: Algorithm 1 can be implemented to run in $O(|F(x,y)|)$ time, where y is any codeword of $T(G, C_0)$.

While Algorithm 1 is expected to reduce the number of corrupt bits by a constant factor, the number may increase. Nevertheless, the following lemma shows that any such increase will not be too large.

Lemma 3.4: Let $x \in \mathbb{F}_2^n$, $y \in T(G, C_0)$, and $F = \{i \in [n] : x_i \neq y_i\}$. The number of $i \in [n] \setminus F$ such that $p_i > 0$ at the end of $\mathsf{RandFlip}(x)$ is at most $\frac{c}{t}|F|$. In particular, for $x' = \mathsf{RandFlip}(x)$, we always have $d_H(x',y) \le (1 + \frac{c}{t}) d_H(x,y)$.

Proof: Consider $v \in U(x)$ such that the corresponding iteration increases p_i from zero to nonzero for some $i \in [n] \setminus F$. By the way i is chosen at Line 6, we know w_v and $x_{N(v)}$ differ at this bit. As $i \in [n] \setminus F$, we know $x_{N(v)}$ and $y_{N(v)}$ agree at this bit. So $w_v \neq y_{N(v)}$. By Lemma 3.2, this occurs only if $v \in N_{\ge t}(F)$. Finally, by Lemma 2.6, the number of $v \in N_{\ge t}(F)$ is at most $\frac{c}{t}|F|$. ∎

B. Flipping Iteratively

We present our randomized decoding algorithm:

Algorithm 2 $\mathsf{RandDecode}(x)$

Input: $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

```
while |U(x)| > 0 do
    x ← RandFlip(x)
end while
return x
```

Theorem 3.5: Suppose $d_0\delta > 2$. Let $\varepsilon_0 = \frac{d_0}{2} - \frac{1}{\delta} > 0$. Let $x \in \mathbb{F}_2^n$ and $y \in T(G, C_0)$ such that $d_H(x,y) \le \alpha n$. Given the input x, Algorithm 2 outputs y in $O(n)$ time with probability $1 - o(1)$.

We omit the proof of Theorem 3.5 and refer the reader to the full version [12]. We remark that Theorem 3.5 is not needed in the analysis of our deterministic decoding algorithm.

IV. Deterministic Decoding

We begin by applying the same derandomization technique as in [11] to develop a deterministic algorithm that corrects γn errors when $\delta d_0 > 2$, where $\gamma = \left(1+\frac{c}{t}\right)^{-1}\frac{\delta d_0 - 1}{d_0 - 1}\alpha$. Subsequently, we introduce an additional step before running this algorithm, extending the decoding radius to αn.

The main idea in [11] is as follows: The set $L = [n]$ is partitioned into $O(1)$ buckets based on their flipping probability p_i. It can be shown that at least one of these buckets contains a significantly higher proportion of corrupt bits than uncorrupt bits. By flipping the bits in this bucket, a small but constant fraction of the errors can be corrected.

However, the desired bucket is not known in advance. Therefore, we must recursively search through all possible choices until the number of errors is significantly reduced, as indicated by a substantial decrease in the size of $U(x)$. We prune branches where $|U(x)|$ does not decrease significantly. While this approach may appear to rely on brute force, careful analysis shows that the algorithm still runs in linear time.

The previous process requires the number of corrupt bits to be bounded by γn initially to guarantee that this number remains below αn during the search, allowing the expansion property to apply. Our key new idea is that, even if the initial number of corrupt bits exceeds γn (but remains bounded by αn), we can search through the first few steps to find a branch where the number of corrupt bits drops below γn. Although we cannot immediately verify which branch works, there is only a constant number of branches. So we can run the aforementioned decoding process on all these branches and check whether any of them produces a valid codeword.

A. Deterministic Flipping

We begin by modifying Algorithm 1 to obtain the following deterministic flipping algorithm.

Algorithm 3 $\mathsf{DeterFlip}(x, q)$

Input: $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$ and $q \in \mathbb{R}$

$t \leftarrow \frac{d_0}{2}$
$p = (p_1, \ldots, p_n) \leftarrow (0, \cdots, 0) \in \mathbb{R}^n$
for each $v \in R$ **do**
  $w_v \leftarrow \mathsf{Decode}(x_{N(v)})$
  if $1 \le d_H(w_v, x_{N(v)}) < t$ **then**
    Choose the smallest $i \in N(v)$ where w_v and $x_{N(v)}$ differ
    $p_i \leftarrow p_i + \frac{t - d_H(w_v, x_{N(v)})}{ct}$
  end if
end for
for each $i \in [n]$ **do**
  Flip x_i if $p_i = q$
end for
return x

Algorithm 3 is derived from Algorithm 1 with the following modifications: First, it takes an additional input $q \in \mathbb{R}$. Second, instead of flipping each x_i with probability p_i, it flips x_i when p_i equals q. In particular, Algorithm 3 is deterministic.

Define the finite set $W := \left\{\frac{i}{cd_0} : i \in \mathbb{Z}, 0 \le i \le cd_0\right\}$. Note that each $p_i \in [0,1]$ is an integral multiple of $\frac{1}{2ct} = \frac{1}{cd_0}$ and, therefore, lies within W. The following lemma shows that there exists $q \in W$ such that flipping all x_i with $p_i = q$ corrects a constant fraction of errors.

Lemma 4.1: Assume $d_0\delta > 2$ and let $\varepsilon_0 = \frac{d_0}{2} - \frac{1}{\delta} > 0$. Let $x \in \mathbb{F}_2^n$ and $y \in T(G, C_0)$ such that $d_H(x, y) \le \alpha n$. Let $F = F(x, y)$. For $q \in W$, let P_q be the set of $i \in [n]$ such that $p_i = q$ at the end of $\mathsf{DeterFlip}(x, q)$. Then there exists $q \in W \setminus \{0\}$ such that $|P_q \cap F| - |P_q \setminus F| \ge \frac{\varepsilon_0\delta}{2ct^2}|F|$.

Proof sketch: This follows from the proof of Theorem 3.1 and an averaging argument. See the full version [12]. ∎

As $\mathsf{DeterFlip}(x, q)$ only flips the bits x_i with $i \in P_q$, we immediately derive the following corollary:

Corollary 4.2: Under the notation and conditions in Lemma 4.1, there exists $q \in W \setminus \{0\}$ such that $|F(x', y)| \le \left(1 - \frac{\varepsilon_0\delta}{2ct^2}\right)|F(x, y)|$, where x' is the output of $\mathsf{DeterFlip}(x, q)$.

The proofs of Lemma 3.4 and Lemma 3.3 apply to Algorithm 3 as well and yield the following counterparts.

Lemma 4.3: Let $x \in \mathbb{F}_2^n$ and $y \in T(G, C_0)$. For all $q \in W \setminus \{0\}$ and $x' = \mathsf{DeterFlip}(x, q)$, it holds that $d_H(x', y) \le (1+\frac{c}{t})d_H(x, y)$, or equivalently, $|F(x', y)| \le (1+\frac{c}{t})|F(x, y)|$.

Lemma 4.4: For all $q \in W \setminus \{0\}$, Algorithm 3 can be implemented to run in $O(|F(x, y)|)$ time, where y is any codeword of $T(G, C_0)$.

B. Search for a Sequence of q

In the following, assume $\delta d_0 > 2$ and let $\varepsilon_0 = \frac{d_0}{2} - \frac{1}{\delta} > 0$.

Algorithm 4 $\mathsf{DeepFlip}(x)$

Input: $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

$s \leftarrow \left\lceil \frac{\log\left(\frac{\delta d_0 - 1}{2(d_0 - 1)}\right)}{\log(1-\varepsilon)} \right\rceil$, where $\varepsilon := \frac{\varepsilon_0\delta}{2ct^2}$ and $t = d_0/2$.
$k_{\min} \leftarrow |R| + 1$
$x_{\min} \leftarrow \perp$
for each $(q_1, \ldots, q_s) \in (W \setminus \{0\})^s$ **do**
  $x^{(0)} \leftarrow x$
  for $i \leftarrow 1$ to s **do**
    $x^{(i)} \leftarrow \mathsf{DeterFlip}(x^{(i-1)}, q_i)$
    if $|U(x^{(i)})| > c\gamma n$ **then**
      Exit the inner loop
    else if $i = s$ and $|U(x^{(s)})| < k_{\min}$ **then**
      $k_{\min} \leftarrow |U(x^{(s)})|$
      $x_{\min} \leftarrow x^{(s)}$
    end if
  end for
end for
return $x_{\min}$

Theorem 4.5: Let $x \in \mathbb{F}_2^n$ and $y \in T(G, C_0)$ such that $d_H(x, y) \le \gamma n$, where $\gamma = \left(1+\frac{c}{t}\right)^{-1}\frac{\delta d_0 - 1}{d_0 - 1}\alpha$. Then $\mathsf{DeepFlip}(x)$ outputs an element $x' \in \mathbb{F}_2^n$ in $O(|F(x, y)|)$ time such that $|F(x', y)| \le \frac{1}{2}|F(x, y)|$.

A similar statement was proved in [11]. We omit the proof and defer it to the full version [12].

C. The Deterministic Decoding Algorithm

We now present the deterministic decoding algorithm. Let $t = d_0/2$, $\gamma = \left(1+\frac{c}{t}\right)^{-1}\frac{\delta d_0 - 1}{d_0 - 1}\alpha$, $\varepsilon_0 = \frac{d_0}{2} - \frac{1}{\delta}$, and $\varepsilon = \frac{\varepsilon_0 \delta}{2ct^2}$.

Algorithm 5 MainDecode(x)

Input: $x = (x_1, \ldots, x_n) \in \mathbb{F}_2^n$

1: $r \leftarrow \left\lceil \frac{\log \gamma}{\log(1-\varepsilon)} \right\rceil$,
2: $r' \leftarrow \lceil \log_2(\gamma n) \rceil + 1$
3: **for** each $(q_1, \ldots, q_r) \in (W \setminus \{0\})^r$ **do**
4: $\quad \hat{x} \leftarrow x$
5: $\quad$ **for** $i \leftarrow 1$ to r **do**
6: $\quad\quad \hat{x} \leftarrow \mathsf{DeterFlip}(\hat{x}, q_i)$
7: $\quad$ **end for**
8: $\quad$ **for** $i \leftarrow 1$ to r' **do**
9: $\quad\quad \hat{x} \leftarrow \mathsf{DeepFlip}(\hat{x})$
10: $\quad\quad$ **if** $|U(\hat{x})| > c2^{-i}\gamma n$ **then**
11: $\quad\quad\quad \hat{x} \leftarrow \perp$
12: $\quad\quad\quad$ Exit the inner loop
13: $\quad\quad$ **end if**
14: $\quad$ **end for**
15: $\quad$ **return** $\hat{x}$ **if** $\hat{x} \neq \perp$ and $|U(\hat{x})| = 0$ and $d_H(x, \hat{x}) \leq \alpha n$
16: **end for**

Theorem 4.6: Suppose $\delta d_0 > 2$. Algorithm 5 can be implemented to correct αn errors in $O(n)$ time for $T(G, C_0)$.

Proof sketch: By Corollary 4.2, there exists $(q_1, \ldots, q_r) \in (W \setminus \{0\})^r$ such that in the corresponding iteration of the outer loop, the first inner loop reduces the number of corrupt bits to γn or below. By Theorem 4.5, the second loop further reduces the number of corrupt bits to fewer than one, which must be zero. While we do not know which $(q_1, \ldots, q_r)$ works, we enumerate all possibilities, which are constantly many. Line 15 verifies whether any of them produces the desired codeword. The running time can be shown to be $O(n)$ using Lemma 4.4 and Theorem 4.5. For details, see the full version [12]. ∎

V. Distance and Decoding Radius

In this section, we use the size-expansion trade-off introduced in [8] to bound the minimum distance of $T(G, C_0)$ and improve the decoding radius of our algorithms. This trade-off was originally used in [8] for the special case where C_0 is a parity-check code. Proofs and background are omitted, but a detailed treatment can be found in the full version [12].

Definition 5.1 (Size-Expansion Function): For $k > 1$, define $f_\delta(k)$ as the optimal value of the following LP.

$$\begin{aligned} \text{minimize} \quad & \frac{1}{k}\sum_{i=1}^{\infty} \beta_i \\ \text{subject to} \quad & \sum_{i=1}^{\infty} i \cdot \beta_i = k, \\ & \sum_{i=1}^{\infty} \left(1 - \left(1 - \frac{1}{k}\right)^i\right) \cdot \beta_i \geq \delta, \\ & \beta_i \geq 0, \quad \forall i. \end{aligned}$$

To prove a lower bound on $|N(S)|$ for a set of vertices $S \subseteq L$ of size $k\alpha n$, we look at the expected expansion of a random subset $S' \subseteq S$ of size αn. Let $\beta_i = \frac{|N_i(S)|}{c\alpha n}$. The first constraint comes from double counting the edges between S and $N(S)$. The second constraint relates to the expected number of neighbors of S'. The term $1-(1-\frac{1}{k})^i$ is approximately the probability that a vertex in $N_i(S)$ is a neighbor of S'. This constraint requires the expected expansion $\frac{\mathbb{E}[|N(S')|]}{c\alpha n}$ to be at least δ. Minimizing the objective $\frac{1}{k}\sum \beta_i = \frac{|N(S)|}{ck\alpha n}$ yields the minimum possible expansion factor $\frac{|N(S)|}{c|S|}$ for the set S, subject to the above constraints that are implied by the expansion property.

Lemma 5.2: f_δ satisfy the following properties:

1) f_δ is non-increasing.
2) $f_\delta(k) \geq \frac{\delta}{k}$.
3) $f_\delta(k)$ is continuous with $\lim_{k\to 1} f_\delta(k) = \delta$ and $\lim_{k\to\infty} f_\delta(k) = 0$.

Lemma 5.3 (Size-Expansion Trade-off [8]): For $k > 1$, a (c, d, α, δ)-bipartite expander with n left vertices is also a $(c, d, k\alpha, f_\delta(k) - O(\frac{1}{n}))$-bipartite expander.

Theorem 5.4: Suppose $\delta d_0 > 1$. The minimum distance of the Tanner code $T(G, C_0)$ is greater than $f_\delta^{-1}\left(\frac{1}{d_0} + \varepsilon\right)\alpha n$ for any constant $\varepsilon \in (0, \delta - \frac{1}{d_0})$ and all sufficiently large n.

Theorem 5.5: Suppose $\delta d_0 > 2$. Algorithm 2 and Algorithm 5 can decode up to $f_\delta^{-1}\left(\frac{2}{d_0} + \varepsilon\right)\alpha n$ errors in $O(n)$ time for any constant $\varepsilon \in (0, \delta - \frac{2}{d_0})$ and sufficiently large n.

Proof: Let $\delta' = \frac{2}{d_0} + \varepsilon$, $k = f_\delta^{-1}(\delta')$, and $\alpha' = k\alpha$. Since G is a (c, d, α, δ)-bipartite expander, it is also a (c, d, α', δ')-bipartite expander by Lemma 5.3. Since $\delta' d_0 = 2 + d_0\varepsilon > 2$, by Theorem 3.5 and Theorem 4.6, Algorithm 2 and Algorithm 5 can decode up to $\alpha' n$ errors in linear time. ∎

Finally, the bound in Theorem 5.4 is essentially tight:

Theorem 5.6: Given $\delta, d_0, \varepsilon > 0$ with $\delta d_0 > 1$, there exist constants c, d, and α such that for infinitely many n, a $(c, d, \alpha, \delta - \varepsilon)$-bipartite expander G with n left vertices exists. Moreover, for any $C_0 \subseteq \mathbb{F}_2^d$ of minimum distance d_0, there exists such a graph G such that, by fixing an appropriate total ordering on $N(v)$ for each $v \in R(G)$, the minimum distance of the resulting code $T(G, C_0)$ is at most $f_\delta^{-1}\left(\frac{1}{d_0}\right)\alpha n$.

Proof sketch: We build a (c, d_0)-regular graph G_0 with $k\alpha n$ left nodes. We also build an almost (c, d)-regular graph G_1 with $(1 - k\alpha)n$ left nodes and $\frac{c}{kd_0}n$ right nodes of degree $d - d_0$, while the other right nodes have degree d. Finally, we merge G_0 and G_1 by pairing degree-d_0 nodes with degree-$(d - d_0)$ nodes. When $1^{d_0}0^{d-d_0} \in C_0$, the indicator vector of G_0's left nodes forms a codeword with weight $k\alpha n$. It remains to prove that G is a $(c, d, \alpha, \delta - \varepsilon)$-bipartite expander with high probability. Details can be found in the full version [12]. ∎

Acknowledgments

The first author thanks Xue Chen for explaining their results [8]. The second author was supported by the NSF CAREER award CCF-2440926. He thanks Chong Shangguan and Yuanting Shen for helpful discussions and for explaining their results [11], and Zihan Zhang for additional discussions.

REFERENCES

[1] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, 1996.

[2] V. Skachek and R. Roth, "Generalized minimum distance iterative decoding of expander codes," in *Proceedings 2003 IEEE Information Theory Workshop (Cat. No.03EX674)*, 2003, pp. 245–248.

[3] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[4] J. Feldman, M. Wainwright, and D. Karger, "Using linear programming to decode binary linear codes," *IEEE Transactions on Information Theory*, vol. 51, no. 3, pp. 954–972, 2005.

[5] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Transactions on Information Theory*, vol. 53, no. 1, pp. 82–89, 2007.

[6] M. Viderman, "Linear-time decoding of regular expander codes," *ACM Trans. Comput. Theory*, vol. 5, no. 3, 2013. [Online]. Available: https://doi.org/10.1145/2493252.2493255

[7] ——, "LP decoding of codes with expansion parameter above 2/3," *Inf. Process. Lett.*, vol. 113, no. 7, p. 225–228, Apr. 2013. [Online]. Available: https://doi.org/10.1016/j.ipl.2013.01.012

[8] X. Chen, K. Cheng, X. Li, and M. Ouyang, "Improved decoding of expander codes," *IEEE Transactions on Information Theory*, vol. 69, no. 6, pp. 3574–3589, 2023.

[9] S. K. Chilappagari, D. V. Nguyen, B. Vasic, and M. W. Marcellin, "On trapping sets and guaranteed error correction capability of LDPC codes and GLDPC codes," *IEEE Transactions on Information Theory*, vol. 56, no. 4, pp. 1600–1611, 2010.

[10] M. Dowling and S. Gao, "Fast decoding of expander codes," *IEEE Transactions on Information Theory*, vol. 64, no. 2, pp. 972–978, 2018.

[11] K. Cheng, M. Ouyang, C. Shangguan, and Y. Shen, "When can an expander code correct $\Omega(n)$ errors in $O(n)$ time?" in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2024)*, 2024, pp. 61:1–61:23.

[12] Z. Zhou and Z. Guo, "Improved decoding of tanner codes," 2025. [Online]. Available: https://arxiv.org/abs/2501.12293

[13] G. Forney, "Generalized minimum distance decoding," *IEEE Transactions on Information Theory*, vol. 12, no. 2, pp. 125–131, Apr. 1966.