Improved Decoding of Expander Codes

Xue Chen, Kuan Cheng, Xin Li, and Minghui Ouyang

***Abstract*— We study the classical expander codes, introduced by Sipser and Spielman, (1996). Given any constants $0 < \alpha, \varepsilon < 1/2$, and an arbitrary bipartite graph with N vertices on the left, $M < N$ vertices on the right, and left degree D such that any left subset S of size at most αN has at least $(1-\varepsilon)|S|D$ neighbors, we show that the corresponding linear code given by parity checks on the right has distance at least roughly $\frac{\alpha N}{2\varepsilon}$. This is strictly better than the best known previous result of $2(1-\varepsilon)\alpha N$ Sudan, (2000), Viderman, (2013) whenever $\varepsilon < 1/2$, and improves the previous result significantly when ε is small. Furthermore, we show that this distance is tight in general, thus providing a complete characterization of the distance of general expander codes. Next, we provide several efficient decoding algorithms, which vastly improve previous results in terms of the fraction of errors corrected, whenever $\varepsilon < \frac{1}{4}$. Finally, we also give a bound on the list-decoding radius of general expander codes, which beats the classical Johnson bound in certain situations (e.g., when the graph is almost regular and the code has a high rate). Our techniques exploit novel combinatorial properties of bipartite expander graphs. In particular, we establish a new size-expansion tradeoff, which may be of independent interests.**

***Index Terms*— Expander codes, bipartite expanders, list decoding.**

I. Introduction

EXPANDER codes [1] are error-correcting codes derived from bipartite expander graphs that are notable for their ultra-efficient decoding algorithms. In particular, all known asymptotically good error-correcting codes which admit linear-time decoding algorithms for a constant fraction of adversarial errors are based on expander codes. At the same time, expander codes are closely related to low-density parity-check (LDPC) codes [4] — a random LDPC code is an expander code with high probability. Over the last twenty years, LDPC codes have received increased attention ([5], [6], [7], [8], [9] to name a few) because of their practical performance. Along this line of research, the study of decoding algorithms for expander codes, such as belief-propagation [1], [4], [10], message-passing [11], and linear programming [5], [6], [12], has laid theoretical foundations and sparked new lines of inquiry for LDPC codes.

In this work, we consider expander codes for adversarial errors. Briefly, given a bipartite graph G with N vertices of degree D on the left, M vertices on the right, we say it is an $(\alpha N, (1-\varepsilon)D)$ expander if and only if any left subset S with size at most αN has at least $(1-\varepsilon)D \cdot |S|$ distinct neighbors. The code $\mathcal{C}$ of an expander G assigns a bit to each vertex on the left and views each vertex on the right as a parity check over its neighbors. A codeword $C \in \mathcal{C}$ is a vector in $\{0,1\}^N$ that satisfies all parity checks on the right. Moreover, the distance of $\mathcal{C}$ is defined as the minimum Hamming distance between all pairs of codewords. We defer the formal definitions of expanders and expander codes to Section II. For typical applications, the parameters α, ε and D are assumed to be constants, and there exist explicit constructions (e.g., [13]) of such expander graphs with $M < N$.

For expander codes defined by $(\alpha N, (1-\varepsilon)D)$-expanders, the seminal work of Sipser and Spielman [1] gave the first efficient algorithm to correct a constant fraction (i.e., $(1-2\varepsilon)\cdot\alpha N$) of errors, when $\varepsilon < 1/4$. In fact, their algorithms are super efficient — they provide a linear time algorithm called belief-propagation and a logarithmic time parallel algorithm with a linear number of processors. Subsequently, Feldman et al. [6] and Viderman [3], [12] provided improved algorithms to correct roughly $\frac{1-3\varepsilon}{1-2\varepsilon}\cdot\alpha N$ errors, when $\varepsilon < 1/3$. This fraction of error is strictly larger than that of [1] whenever $\varepsilon < 1/4$. Viderman [3] also showed how to correct $N^{\Omega_{D,\varepsilon,\alpha}(1)}$ errors when $\varepsilon \in [1/3, 1/2)$, and that $\varepsilon < 1/2$ is necessary for correcting even 1 error. However, the following basic question about expander codes remains unclear.

Question: What is the best distance bound one can get from an expander code defined by arbitrary $(\alpha N, (1-\varepsilon)D)$-expanders?

This question is important since it is well known that for unique decoding, the code can and can only correct up to half the distance number of errors. In [1], Sipser and Spielman showed that the distance of such expander codes is at least αN, while a simple generalization improves this bound to $2(1-\varepsilon)\alpha N$ (see e.g., [2] and [3]). Perhaps somewhat surprisingly, this simple bound is the best known distance bound for an arbitrary expander code. In fact, Viderman [3] asserted that this is the best distance bound one can achieve based only on the expansion property of the graph, and hence when ε converges to 0, the number of errors corrected in [3], $\frac{1-3\varepsilon}{1-2\varepsilon}\cdot\alpha N$ converges to the half distance bound. Yet, no evidence was known to support this claim. Thus it is natural to ask whether

Manuscript received 9 January 2022; revised 22 December 2022; accepted 7 January 2023. Date of publication 23 January 2023; date of current version 19 May 2023. The work of Xin Li was supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575. *(Corresponding authors: Xue Chen; Kuan Cheng; Xin Li.)*

Xue Chen is with the CAS Key Laboratory of Wireless-Optical Communications and the College of Computer Science and Technology, University of Science and Technology of China (USTC), Hefei 230027, China (e-mail: xuechen1989@ustc.edu.cn).

Kuan Cheng is with the Center on Frontiers of Computing Studies and the Advanced Institute of Information Technology, Peking University, Beijing 100871, China (e-mail: ckkcdh@pku.edu.cn).

Xin Li is with the Department of Computer Science, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: lixints@cs.jhu.edu).

Minghui Ouyang is with the Department of Mathematical Science, Peking University, Beijing 100871, China (e-mail: ouyangminghui1998@gmail.com).

Communicated by V. Skachek, Associate Editor for Coding and Decoding.
Digital Object Identifier 10.1109/TIT.2023.3239163

any improvement is possible, and if so, can one design efficient algorithms to correct more errors?

A. Our Results

1) Distance of Expander Codes: In this work, we give affirmative answers to the above questions. Our first result shows that the best distance bound of expander codes defined by arbitrary $(\alpha N, (1-\varepsilon)D)$-expanders is roughly $\frac{\alpha N}{2\varepsilon}$.

Theorem 1: [Informal Versions of Theorem 9 and Theorem 10] Given any $(\alpha N, (1-\varepsilon)D)$-expander, let $\mathcal{C}$ be the expander code defined by it. The distance of $\mathcal{C}$ is at least $\frac{\alpha}{2\varepsilon} \cdot N - O_\varepsilon(1)$.

Moreover, for any constant $\eta > 0$ there exists an $(\alpha N, (1-\varepsilon)D)$-expander whose expander code has distance at most $(\frac{\alpha}{2\varepsilon} + \eta) \cdot N$.

We remark that the bound $\frac{\alpha}{2\varepsilon} \cdot N$ is always larger than the previous bound $2(1-\varepsilon)\alpha N$ since we always have $\varepsilon < 1/2$ in expander codes. For small ε, this improves upon the previous bound by a factor of $\frac{1}{4\epsilon}$ roughly, which can be quite significant.

2) Decoding Algorithms: Next we consider algorithms to correct more errors. Given the above bound on the distance of expander codes, the natural goal is to design efficient algorithms that can correct $\Theta(\alpha/\varepsilon) \cdot N$ errors. We achieve this goal for all $\varepsilon < 1/4$.

Theorem 2: [Informal version of Theorem 23] Given any constants $\alpha, \eta > 0$ and $0 < \varepsilon < 1/4$, there exists a linear time algorithm that for any expander code defined by an $(\alpha N, (1-\varepsilon)D)$-expander, corrects up to $(\frac{3\alpha}{16\varepsilon} - \eta) \cdot N$ adversarial errors.

The bound $\frac{3\alpha}{16\varepsilon} \cdot N$ is larger than all previous bounds for $\varepsilon < 1/4$ by at least a constant factor. For example, when ε is close to $1/4$, all previous works [1], [3], [6] can only correct roughly $\frac{\alpha}{2} \cdot N$ errors, while our algorithm can correct roughly $\frac{3}{4} \cdot \alpha N$ errors. When ε is smaller, the improvement is even more significant, as no previous work can correct more than αN errors. On the other hand, given Theorem 1, one can hope for correcting roughly $\frac{\alpha}{4\varepsilon} \cdot N$ errors, so Theorem 2 falls slightly short of achieving it.

Actually, we can correct more errors when ε is small. For example, when $\varepsilon < \frac{3-2\sqrt{2}}{2} \approx 0.0858$, our algorithm in Section VI can correct $\frac{\sqrt{2}-1}{2} \cdot \frac{\alpha N}{\varepsilon} > 0.207 \cdot \frac{\alpha N}{\varepsilon}$ errors. We summarize all our results informally in Table I, compared to the previous best results of [3], [6].

3) List-Decoding: Finally, we consider the list-decodability of expander codes. List-decoding, introduced by Elias [14] and Wozencraft [15] separately, is a relaxation of the classical notion of unique decoding. In this setting, the decoder is allowed to output a small list of candidate codewords that include all codewords within Hamming distance ρN of the received word. Thus, the list-decoding radius ρN could be significantly larger than half of the distance. For example, a very recent work by Mosheiff et al. [9] shows random LDPC codes have list-decoding radii close to their distance. In this setting, the classical Johnson bound shows that *any* binary code with distance d is list-decodable up to radius $r = \frac{N - \sqrt{N(N-2d)}}{2}$ with list size $N^{O(1)}$. If we set the Johnson bound r as the baseline, a natural question is whether expander codes can list-decode more than r errors given the distance $d = \frac{\alpha}{2\varepsilon} \cdot N$?

In Section VII, we consider expander codes defined by expanders that has a maximum degree $D_{\max} = O(1)$ on the right, like LDPC codes. Our main results provide an alternative bound on the list-decoding radius of such codes, and show that it is strictly better than the Johnson bound when α/ε is small and the right hand side is also almost regular, i.e., $D_{\max} \approx D_R$, where D_R is the *average* right degree.

Theorem 3: [Informal version of Theorem 29] Given any $(\alpha N, (1-\varepsilon)D)$-expander with regular degree D on the left and maximum degree $D_{\max}$ on the right, its expander code has a list-decoding radius at least $\rho N = (\frac{1}{2} + \Omega(1/D_{\max}))d$ and list size $N^{O(1)}$. Here d is the distance of the code.

Furthermore, if $D_{\max} \le 1.1\ D_R$, $\varepsilon \le 1/4$ and $\alpha/\varepsilon \le 0.1$, ρN is strictly larger than the Johnson bound r of binary codes with distance $d = \frac{\alpha}{2\varepsilon} \cdot N$.

We remark that the Johnson bound $r = d/2 + \Theta(d^2/N)$ for a small d (by the Taylor expansion on $r = \frac{N - \sqrt{N(N-2d)}}{2}$). While we did not attempt to optimize the constant hidden in the Ω notation of $\rho = (\frac{1}{2} + \Omega(1/D_{\max}))d$, we show that roughly $\frac{1}{D_R} \ge \frac{\alpha}{4\varepsilon}$ in Section III. When the expander is also almost regular on the right, e.g., $D_{\max} \le 1.1\ D_R$, this bound is better than the Johnson bound with $d = \frac{\alpha}{2\varepsilon} \cdot N$ and a small ratio α/ε. The second condition would follow from a large average right-degree D_R (equivalently, a small M/N or a large code-rate $1 - M/N$). In particular, this applies to the upper bound constructed for Theorem 1, which has distance arbitrarily close to $\frac{\alpha}{2\varepsilon} \cdot N$.

One intriguing question is to design efficient list-decoding algorithms for expander codes. Since these algorithms would also immediately improve all our results on unique decoding, we leave this as a future direction.

4) New Combinatorial Properties of Expander Graphs: Our distance bounds and decoding algorithms make extensive use of a new size-expansion tradeoff for bipartite expander graphs, which we establish in this paper. Specifically, we show that one can always trade the expansion for larger subsets in such a graph. In particular, given any $(\alpha N, (1-\varepsilon)D)$-expander, we prove in Section III that this graph is also roughly a $(k\alpha N, (1 - k\varepsilon)D)$-expander for any $k \ge 1$, provided that $k\alpha N \le N$. This size-expansion tradeoff is potentially of independent interest. For example, besides the applications in our distance bounds and decoding algorithms, we also use it to show a relation between the three basic parameters $(\alpha, \varepsilon, D_R)$ of bipartite expanders. Roughly, we always have $\frac{\alpha}{\varepsilon} \le \frac{4}{D_R}$ (see Fact 12 for a formal statement). On the other hand, using a random graph one can show the existence of $(\alpha N, (1-\varepsilon)D)$-expanders such that roughly $\frac{\alpha}{\varepsilon} \ge \frac{1}{eD_R}$ (see Proposition 33). Thus our upper bound is tight up to a constant factor.

B. Related Work

Sipser and Spielman's definition in [1] is actually more general, and is a variant of Tanner codes [16] based on expanders. Basically, the code requires all symbols in the neighbor set of a right vertex (in some fixed order) to be a codeword from an inner linear code $\mathcal{C}_0$. The expander code studied here is the most popular and well studied case, where the inner code consists of all strings with even weight. Instead

TABLE I
SUMMARY OF THE DISTANCE AND DECODING RADII FOR ε

	$\varepsilon \in (0, \frac{3-2\sqrt{2}}{2})$	$\varepsilon \in [\frac{3-2\sqrt{2}}{2}, 1/8)$	$\varepsilon \in [1/8, 1/4)$
Distance from Theorem I.1	$\frac{1}{2\varepsilon} \cdot \alpha N$	$\frac{1}{2\varepsilon} \cdot \alpha N$	$\frac{1}{2\varepsilon} \cdot \alpha N$
Decoding radius from [3, 6]	$\frac{1-3\varepsilon}{1-2\varepsilon} \cdot \alpha N$	$\frac{1-3\varepsilon}{1-2\varepsilon} \cdot \alpha N$	$\frac{1-3\varepsilon}{1-2\varepsilon} \cdot \alpha N$
Decoding radius from this work	$\frac{\sqrt{2}-1}{2\varepsilon} \cdot \alpha N$ from Theorem VI.1	$\frac{1-2\varepsilon}{4\varepsilon} \cdot \alpha N$ from Theorem VI.1	$\frac{3}{16\varepsilon} \cdot \alpha N$ from Theorem V.4

of vertex expansion, the expander based Tanner codes are analyzed based on edge expansion, a related concept which has also been well studied in both mathematics and computer science [17], [18]. We note that the distance of Tanner codes depends heavily on the inner code $\mathcal{C}_0$, and is thus generally incomparable to the distance of our code. To the best of our knowledge, the best bound on the distance of expander codes based on vertex expansion of bipartite expanders, as studied in this paper, was $2(1-\varepsilon) \cdot \alpha N$.

As mentioned before, expander codes are closely related to low-density parity-check (LDPC) codes introduced by Gallager [4], where the bipartite graph associated with the parity checks has bounded degree on the right but is not necessary an expander. There is a long line of research on random LDPC codes against *random errors* (see [7], [11], [19] and the references therein). While a random LDPC code is an expander code with high probability, our results are incomparable with those of random LDPC codes. This is because first, we consider expander codes defined by arbitrary expanders, while many results on random LDPC codes use more properties than the expansion, such as the girth of the underlying graph that can be deduced from random graphs. Second, we consider adversarial errors, while many results on random LDPC codes [7], [11] consider random errors or memoryless channels.

In the context of list-decoding, the work of RonZewi-Wootters-Zemor [20] studied the problem of erasure list-decoding of expander codes, based on algebraic expansion properties (i.e., eigenvalues of the corresponding adjacency matrix).

In the past few decades, a great amount of research has been devoted to expander graphs, leading to a plethora of new results. We refer the reader to the survey by Hoory, Linial, and Wigderson [21] for an overview. Specifically, giving explicit constructions of bipartite expander graphs for expander codes has been a challenge. In particular, Kahale [22] showed that general Ramanujan graphs [17] (with the minimum 2nd largest absolute eigenvalue among all D-regular graphs) cannot provide vertex expansion more than half of the degree, which is the threshold required to give expander codes. After decades of efforts, explicit constructions satisfying the requirements of expander codes have been provided in [13], [23] separately.

C. Technique Overview

Let $\mathcal{C}$ be an expander code defined by an $(\alpha N, (1-\varepsilon)D)$ expander. Our techniques for the improved distance bound and decoding algorithms are based on the combination of the following three ingredients, together with a new idea of guessing expansions:

1) A new size-expansion tradeoff for arbitrary bipartite expander graphs, which we establish in this paper.
2) A procedure of finding possible corruptions in [3], which we slightly adapt and establish new properties.
3) A procedure of flipping bits in the corrupted word to reduce the number of errors, introduced in [1].

We first briefly explain each ingredient.

1) The Size-Expansion Tradeoff: As mentioned before, we show that any $(\alpha N, (1-\varepsilon)D)$-expander is also roughly a $(k\alpha N, (1-k\varepsilon)D)$-expander for any $k \geq 1$. To prove this, assume for the sake of contradiction that there is a left subset S with size $k\alpha N$ that has smaller expansion. This then implies that there are many *collisions* (two different vertices on the left connected to the same vertex on the right) in the neighbor set of S, i.e., more than $k\varepsilon D \cdot k\alpha N = k^2\alpha\varepsilon ND$ collisions. Now we pick a *random* subset $T \subseteq S$ with size αN, then each previous collision will remain with probability roughly $1/k^2$. By linearity of expectation, more than $\alpha\varepsilon ND$ collisions are expected to remain in the neighbor set of T, thus implying the expansion of T is smaller than $(1-\varepsilon)D \cdot \alpha N$. This contradicts the expander property.

This convenient size-expansion tradeoff is used extensively in our bounds and algorithms. In fact, by using linear programming, we can get a better size-expansion tradeoff for $k \geq \frac{1}{2\varepsilon}$, which we use in our result on list-decoding expander codes.

2) The Procedure of Finding Possible Corruptions: Viderman [3] introduced the following procedure for finding possible corruptions. Maintain a set L of left vertices, a set R of right vertices and a fixed threshold h. Start with R being all the unsatisfied parity checks, then iteratively add left vertices with at least h neighbors in R to L, and their neighbors to R. Viderman showed that if the number of corruptions is not too large, then when this process ends, L will be a super set of all corruptions and the size of L is at most αN. Therefore, one can treat L as a set of erasures and decode from there.

In [3], Viderman used sophisticated inequalities to analyze this procedure. In this paper, we show that the process has the following property.

3) Property (*): If $h = (1-2\Delta)D$ such that any subset S of corrupted vertices has expansion at least $(1-\Delta)D|S|$, then all corruptions will be contained in L. Furthermore, we can assume without loss of generality that the set of corrupted vertices is added to L before any other vertex.

This allows us to simplify the analysis in [3] and combine with our size-expansion tradeoff.

4) The Procedure of Flipping Bits: Sipser and Spielman [1] introduced a procedure to flip bits in the corrupted word. Again, the idea is to set a threshold h, and flip every bit which has at least h wrong parity checks in its neighbors. Sipser

and Spielman showed that when $\varepsilon < 1/4$ and the number of corruptions is not too large, this procedure will reduce the number of errors by a constant factor each time. Thus one only needs to run it for $O(\log N)$ times to correct all errors.

5) Our Approaches: We now describe how to combine these ingredients to get our bounds and algorithms. For the distance lower bound, it suffices to choose k such that $1-k\varepsilon > 1/2$. Then a standard analysis as in [1] shows the distance of the code is at least $k\alpha N$. Thus, we can set $k \approx \frac{1}{2\varepsilon}$ so that the distance is roughly at least $\frac{\alpha}{2\varepsilon}N$. A subtle point here is that it is not a priori clear that we can choose $k \approx \frac{1}{2\varepsilon}$, since it may be that $k\alpha N = \frac{\alpha}{2\varepsilon}N > N$, and no left subset can have size larger than N. However, we again use the size-expansion tradeoff to show that this cannot happen. In particular, we show $\frac{\alpha}{\varepsilon} \leq \frac{4}{D_R}$ (recall D_R is the average degree on the right), and thus we can always set $k \approx \frac{1}{2\varepsilon}$. Section III-A gives a construction which shows this bound is almost tight.

Next we describe our decoding algorithms.

6) Unique Decoding for $\varepsilon < 1/4$: Our algorithm here is based on the following crucial observation. Let F denote the set of corrupted vertices any time during the execution of the algorithm, and assume $|\Gamma(F)| = (1-\gamma)D|F|$, where $\Gamma(F)$ denotes the neighbor set of F. If γ is large, or equivalently $|\Gamma(F)|$ is small, then the procedure of finding possible corruptions works well. This is because intuitively, the number of vertices added to L will be proportional to $|\Gamma(F)|$, and thus $|L|$ will be small. On the other hand, if γ is small, or equivalently $|\Gamma(F)|$ is large, then the procedure of flipping bits works well. This is because intuitively, the procedure of flipping bits works better when the expansion property is better.

Hence, we can combine both procedures and set a threshold for γ. If γ is larger than this threshold, we use the procedure of finding possible corruptions; otherwise we use the procedure of flipping bits. However, we don't know γ. Thus in our algorithm we guess γ, and for each possible value of γ we apply the corresponding strategy. This is a bit like list-decoding, where we get a small list of possible codewords, from which we can find the correct codeword by checking the Hamming distance to the corrupted word. Note that the procedure of finding possible corruptions always returns a possible codeword; while to get a codeword from the procedure of flipping bits, we need to apply it for a constant number of times, until the number of errors is small enough so that we can easily correct all errors using any known algorithm. Thus we also need to guess γ for a constant number of times.

Using these ideas, we show that Algorithm 2 can correct $(1-\varepsilon)\alpha N$ errors for any constant $\varepsilon < 1/4$. Now, we can improve this by combining with our size-expansion tradeoff. Specifically, for any constant $\varepsilon < 1/4$ we can choose any $k \geq 1$ such that $k\varepsilon < 1/4$. This implies that a modified algorithm can actually correct $(1-k\varepsilon)k\alpha N$ errors. Setting $k \approx \frac{1}{4\varepsilon}$ gives us an algorithm that can correct roughly $\frac{3\alpha}{16\varepsilon}N$ errors.

For the running time, each time we guess γ, we know $\gamma = 1 - \frac{|\Gamma(F)|}{D|F|}$ with $|\Gamma(F)| \in [M]$ and $|F| \in [N]$. Thus a naive enumeration will result in $O(MN) = O(N^2)$ possible values. Since we need to guess γ for a constant number of times, this will lead to a polynomial running time. However, instead we can enumerate γ from $\{0, \eta, 2\eta, \ldots, \lceil \frac{1}{\eta} \rceil \eta\}$ for a small enough constant $\eta > 0$. This reduces the running time to linear time, at the price of decreasing the relative decoding radius by an arbitrarily small constant. Finally, we remark that this algorithm can be executed in logarithmic time on a linear number of parallel processors, since its main ingredients from [1], [3] have parallel versions in logarithmic time.

7) Unique Decoding for Smaller ε: When ε is even smaller, e.g., $\varepsilon < 1/8$, our algorithm uses the procedure of finding possible corruptions, together with property (*) we established. Let F denote the set of corrupted vertices in the received word. To use property (*), we need to find a Δ such that for any $S \subseteq F$, S has expansion at least $(1-\Delta)D|S|$. Then we can set the threshold $h = (1-2\Delta)D$. In [3], one assumes $|F| \leq \alpha N$ and thus it is enough to set $\Delta = \varepsilon$. However, our goal here is to correct more than αN errors, thus this choice of Δ no longer works. Instead, we use our size-expansion tradeoff to show that if $|\Gamma(F)| = (1-\gamma)D|F|$, then when $S \subseteq F$ and $|S| \geq \alpha N$, we always roughly have $|\Gamma(S)| \geq \left(1 - \sqrt{\frac{\gamma|F|\varepsilon}{\alpha N}}\right) \cdot D|S|$, thus we can set $\Delta = \max\left\{\sqrt{\frac{\gamma|F|\varepsilon}{\alpha N}}, \varepsilon\right\}$.

However, again we don't know γ and $|F|$. Thus we apply the same trick as before, and guess both quantities. This leads to Algorithm 4. Since we have two possible cases ($\Delta = \sqrt{\frac{\gamma|F|\varepsilon}{\alpha N}}$ or $\Delta = \varepsilon$), we get two different decoding radii for different ranges of ε. The running time is polynomial if we use the naive enumeration of γ and $|F|$, but can be made linear by using a similar sparse enumeration as we discussed before.

8) List-Decoding Radius: Recall that our goal is to show that given any $y \in \mathbf{F}_2^N$, there is a list of at most $N^{O(1)}$ codewords within distance $\rho N = (\frac{1}{2} + \Omega(1/D_{\max}))d$ to y. Our analysis modifies the double counting argument that is used to show the Johnson bound. The modification is by using the special structure of expander codes.

In more details, suppose the list of L codewords within distance ρN to y, is $\{C_1, \ldots, C_L\}$. Let τ_i be the number of codewords in the list which have their i-bit different from y. We focus on counting the number T of "triples" (i, j_1, j_2), where the pair of codewords (C_{j_1}, C_{j_2}) are different in their i-th bit. Since the code has distance $d = \delta N$, we know $T \geq \binom{L}{2}\delta N$. We also know $T = \sum_{i\in[N]} \tau_i(L - \tau_i)$. The key observation in our analysis is that for expander codes, $\{\tau_i, i \in [N]\}$ have a large deviation. Specifically, we call τ_i *heavy* if $\tau_i \geq \frac{0.9}{D_{max}}L$, and show that the summation of heavy τ_i's is $\Theta(NL)$. By using this observation, we manage to get a better upper bound for T than that in the proof of the Johnson Bound in certain situations, which in turn yields a better list-decoding radius.

9) Organization: The rest of this paper is organized as follows. In Section II, we describe some basic notation, terms, definitions and useful theorems from previous work. In Section III, we show our improved distance bound for expander codes, and prove it is tight in general. In Section IV, we establish new properties of the algorithm which can find a super set of corruptions. In Section V, we provide our main unique decoding algorithm. In Section VI, we provide our improved unique decoding algorithm for smaller ε.

In Section VII, we show our list-decoding result. Finally, we conclude in Section VIII with some open questions. Appendix A contains some relatively standard materials omitted in the main body.

II. Preliminaries

We will use $1\{\mathcal{E}\} \in \{0,1\}$ to denote the indicator variable of an event $\mathcal{E}$. Moreover, we use C and c to denote different constants in various proofs of this paper.

A. Basic Definitions From Graph Theory

Given a graph G, we use $V(G)$ to denote its vertex set and $E(G)$ to denote its edge set. Given a bipartite graph G, we use $V_L(G)$ and $V_R(G)$ to denote the left hand side and right hand side of the bipartite graph separately. When G is clear, we simplify them as V_L and V_R. Moreover, we fix two notations $N := |V_L|$ and $M := |V_R|$.

For any subset $S \subseteq V_L \cup V_R$, we always use $\Gamma(S)$ to denote its neighbor set in G. If a vertex $v \in \Gamma(S)$ is connected to S by exactly one edge, we call v a unique neighbor of S and use $\Gamma^1(S)$ to denote the set of all unique neighbors of S.

In this work, we consider bipartite graphs that are regular on the left hand side. Thus we use D to denote the *regular* degree in V_L and D_R to denote the *average* degree in V_R. Since $N = |V_L|$ and $M = |V_R|$, we have $N \cdot D = M \cdot D_R$. Moreover, we will use $D_{\max}$ to denote the *maximum* degree in G, which would be the maximum degree in V_R given $M < N$.

A bipartite graph G is an $(\alpha N, (1-\varepsilon)D)$-expander if and only if for any left subset S of size at most αN, its neighbor set $\Gamma(S)$ has size $\geq (1-\varepsilon)D \cdot |S|$. For convenience, we call $\frac{|\Gamma(S)|}{|S|}$ the expansion of S and say G satisfies $(\alpha N, (1-\varepsilon)D)$ expansion if and only if it is an $(\alpha N, (1-\varepsilon)D)$-expander. Throughout this work, we assume that D and D_R are constants. Since we are interested in expanders with $\varepsilon < 1/2$ and $N > M$, we always assume $D \geq 3$ and $D_R > 3$.

B. Basic Definitions From Coding Theory

We recall several notations from coding theory and define expander codes formally.

Definition 4: An (N, k, d) binary error correcting code $\mathcal{C}$ is a set of codewords contained in $\mathbf{F}_2^N$, with $|\mathcal{C}| = 2^k$ such that $\forall C_1, C_2 \in \mathcal{C}$, the Hamming distance between C_1 and C_2 is at least d. Moreover we call k/N the rate of $\mathcal{C}$.

A linear code is a code whose codewords form a linear subspace of $\mathbf{F}_2^N$.

One fact about linear codes is that the distance of a linear code is equal to the minimum weight of a non-zero codeword in it. The decoding radius of a decoding algorithm of $\mathcal{C}$ refers to the largest number of errors that the algorithm can correct.

Definition 5 (Expander Codes [1]): Given an $(\alpha N, (1-\varepsilon)D)$ expander graph G with M right vertices, the expander code defined by G is $\mathcal{C} \subseteq \mathbf{F}_2^N$ such that

$$\mathcal{C} = \left\{ C \mid \forall i \in [M], \sum_{j \in \Gamma(i)} C_j = 0 \right\},$$

where the addition is over the field $\mathbf{F}_2$.

Given the definition of expander codes, we know its rate is at least $1 - M/N$ and its distance is the minimum weight of a non-zero codewords in $\mathcal{C}$.

Remark 6: The original definition of expander codes in [1] is more general, where each vertex on the right represents some linear constraints on the codeword bits corresponding to its neighbors. In this paper, we only consider the most popular and well studied case where each vertex on the right represents a parity check.

We use the following results of decoding for expander codes, from [3].

Theorem 7 [3]: Let G be an $(\alpha N, (\frac{1}{2}+\xi)D)$ expander with $\xi > 0$. For the expander code defined by G, there is a linear-time algorithm that can correct αN erasures.

Theorem 8 [3]: Let G be an $(\alpha N, (1-\varepsilon)D)$ expander for $\varepsilon < 1/3$. For the expander code defined by G, there is a linear-time algorithm that can correct $\frac{1-3\varepsilon}{1-2\varepsilon}\lfloor \alpha N \rfloor$ errors.

III. Improved Distance of Expander Codes

Let G be an $(\alpha N, (1-\epsilon)D)$ expander and $\mathcal{C}$ be the corresponding expander code. We show that when $\varepsilon < 1/2$, the distance of C is roughly $\frac{1}{2\varepsilon}\alpha N$.

Theorem 9: Let G be an $(\alpha N, (1-\varepsilon)D)$ bipartite expander. The distance of the expander code defined by G is at least $\frac{\alpha}{2\varepsilon} \cdot N - 1/\varepsilon$.

In Section III-A, we provide a construction of expander codes to show the above bound $\frac{\alpha}{2\varepsilon} \cdot N$ is almost tight in general.

Theorem 10: Given any constants $\varepsilon \in (0, 1/2)$ and $\eta > 0$, there exist constants D and $\alpha > 0$, such that for infinitely many N, there exist $(\alpha N, (1-\varepsilon-\eta)D)$-expanders with $M \in [N/2, 2N/3]$ where (1) the rate of the expander code is in $[1/3, 1/2]$; and (2) the distance of the expander code is at most $\frac{\alpha}{2\varepsilon} \cdot N$.

Remark: While the graphs we construct in Theorem 10 are not strictly regular on the right, they are "almost regular" in V_R, i.e., $D_{max} \leq 1.1\ D_R$, such that Theorem 3 indicates a larger list-decoding radius than the Johnson bound.

To prove Theorem 9, we start with the following lemma which gives a tradeoff between the two parameters α and ε. This is one of our main technical lemmas, and the proof is deferred to Section III-B.

Lemma 11: For any $k \in (1, 1/\alpha)$ and any left subset S of size $k\alpha N$, we have

- $|\Gamma(S)| \geq (1-k\varepsilon)D \cdot k\alpha N - 2\varepsilon k^2 \cdot D$.
- $|\Gamma(S)| \geq (1-\frac{2k\varepsilon-1}{3-2/k})D \cdot \frac{k}{2}\alpha N - O(k \cdot D)$ (which is better than the 1st bound for $k > 1/2\varepsilon$).

In particular, the first bound will be extensively used in our decoding algorithms, which shows an $(\alpha N, (1-\epsilon)D)$-expander is also roughly a $(k\alpha N, (1-k\epsilon)D)$-expander for any $k > 1$. While this bound is extremely useful for $k \leq 1/2\varepsilon$, we will use the second one for larger k to improve the list-decoding radius upon the standard Johnson bound.

Using the above lemma, we first prove the following facts in an expander graph.

Fact 12: Let G be an $(\alpha N, (1-\varepsilon)D)$-expander with left regular degree D and right average degree D_R. We always

have

1) $\varepsilon \geq 1/D$.
2) $\frac{\alpha}{4\varepsilon} \leq \frac{1}{D_R} \cdot (1 + \frac{2}{\alpha N})$.

Proof: To prove the first fact, let us consider the smallest non-trivial cycle C in the expander graph G. First of all, we observe that $|C| = O(\log |V|)$. To show this, we consider the argument to bound the girth of a graph. Let us fix a vertex v and consider the BFS tree with root v. The BFS procedure finds a non-trivial cycle when it finds a vertex for the 2nd time. Since G is D-regular in V_L, within depth $2\log_{D-1} M$, the BFS procedure will find a non-trivial cycle. Since each vertex in $C \cap V_R$ has two neighbors in $C \cap V_L$, $|\Gamma(C \cap V_L)| \leq D \cdot |C \cap V_L| - |C \cap V_R| = (D-1) \cdot |C \cap V_L|$.

For the second fact, consider a left subset S of size $2M/D$ in V_L. Since $M \leq N$ and $D \geq 3$, such an S always exists. Given $|\Gamma(S)| \leq M$, we plug $k = \frac{|S|}{\alpha N}$ into the 1st inequality of Lemma 11 to obtain

$$\left(1 - \varepsilon \cdot \frac{2M/D}{\alpha N}\right) \cdot D \cdot |S| - 2 \cdot \varepsilon D \cdot \left(\frac{2M/D}{\alpha N}\right)^2 \leq M.$$

We simplify it as follows:

$$\left(1 - \varepsilon \cdot \frac{2M/D}{\alpha N}\right) \cdot D \cdot 2M/D - \frac{8\varepsilon \cdot M^2}{\alpha^2 N^2 \cdot D} \leq M$$

$$\left(1 - \varepsilon \cdot \frac{2}{\alpha D_R}\right) \cdot 2M - \frac{8\varepsilon \cdot M}{\alpha^2 N \cdot D_R} \leq M$$

(recall $ND = MD_R$)

$$M - \frac{8\varepsilon \cdot M}{\alpha^2 N \cdot D_R} \leq \frac{4\varepsilon}{\alpha D_R} \cdot M.$$

So we have $\frac{1}{D_R} \geq \frac{\alpha}{4\varepsilon} - \frac{2}{\alpha N \cdot D_R}$, or equivalently, $\frac{\alpha}{4\varepsilon} \leq \frac{1}{D_R} \cdot (1 + \frac{2}{\alpha N})$. □

We can now prove Theorem 9.

Proof of Theorem 9: First of all, note that $\frac{\alpha}{2\varepsilon} \leq \frac{2}{D_R} \cdot (1 + \frac{1}{\alpha N})$ from Fact 12, and thus is strictly less than 1 (since $D_R > D \geq 3$). Now assume the claim is false, let us consider any non-zero codeword z with Hamming weight at most $\frac{\alpha}{2\varepsilon} \cdot N - 1/\varepsilon$.

Let $S \subset [N]$ denote the entries in z that are 1. By Lemma 11, $|\Gamma(S)| \geq (1 - \frac{|S|}{\alpha N} \cdot \varepsilon) D \cdot |S| - 2\varepsilon D \cdot (\frac{|S|}{\alpha N})^2$. Since $1 > \frac{2\varepsilon \cdot |S|}{\alpha N}$, we have

$$\left(\frac{1}{2} + \frac{1/\varepsilon}{\alpha N} \cdot \varepsilon\right) D \cdot |S| - 2\varepsilon D \cdot \left(\frac{|S|}{\alpha N}\right)^2$$

$$= \frac{D}{2} \cdot |S| + D \cdot \frac{|S|}{\alpha N} - D \cdot \frac{2\varepsilon \cdot |S|}{\alpha N} \cdot \frac{|S|}{\alpha N} > \frac{D}{2} \cdot |S|.$$

This implies the existence of unique neighbors in $\Gamma(S)$. Thus z is not a valid codeword, which contradicts our assumption. □

A. Distance Upper Bound of Expander Codes

In this section we prove Theorem 10. Given η and ε, let $D \geq \frac{1}{\varepsilon \cdot \eta^2}$ be a constant such that there exists a family of degree-D Ramanujan graphs [17] whose 2nd largest absolute value of eigenvalues of the adjacency matrix is $\lambda \leq 2\sqrt{D-1}$. In this proof, when the graph H is clear, we use $e(A, B)$ for $A, B \subset V(H)$ to denote the number of distinct edges between A and B. We state the following version of the expander mixing lemma for $e(A, A)$ [18], [24].

Lemma 13 (Theorem 8 in [24] for $A = B$): Let $H = (V, E)$ be an expander with degree D, where the second largest absolute value of eigenvalues of the adjacency matrix is λ. Then for any subset $A \subset V$ of size at most $|V|/2$, $e(A, A)$, the number of edges inside A, is bounded by

$$\left| e(A, A) - \frac{D}{2|V|} \cdot |A|^2 \right| \leq \frac{\lambda}{2} \cdot |A|.$$

We now construct an $(\alpha N, (1 - \varepsilon - \eta)D)$-expander graph with $N + M$ vertices by putting together two disjoint graphs G_0 and G_1. For G_0, we first choose a Ramanujan graph H with degree D and $N' = \frac{\alpha}{2\varepsilon} \cdot N$ vertices in the family for a sufficiently small α (compared to ε/D). To obtain $(1 - \varepsilon - \eta)D$ expansion, we modify H and construct a new graph G_0 based on the *vertex-edge incidence* graph of H. Thus the vertex expansion of G_0 is now the *vertex-edge expansion* of H, rather than the *vertex expansion* of H. More specifically, we set G_0 as follows: $V_L(G_0) = V(H)$ and $V_R(G_0) = E(H)$ such that $(v, e) \in E(G_0)$ if and only if $v \in V(H)$ is an endpoint in the edge $e \in E(H)$ of H. Notice that G_0 has left degree D.

Claim 14: The bipartite graph G_0 constructed above is an $(\alpha N, (1 - \varepsilon - \eta)D)$-expander.

Proof: For any $S \subseteq V_L(G_0)$, $|\Gamma(S)|$ is the number of distinct edges connected to $S \subset V(H)$ in H, i.e., $e(S, V(H))$ in the Ramanujan graph H. We rewrite $e(S, V(H)) = e(S, S) + e(S, \overline{S})$. Since $2e(S, S) + e(S, \overline{S}) = D \cdot |S|$, we upper bound $e(S, S)$ by the expander mixing lemma:

$$e(S, S) \leq \frac{D}{2|V(H)|} \cdot |S|^2 + \frac{\lambda}{2} \cdot |S| \leq D \cdot |S| \left(\frac{|S|}{2|V(H)|} + \frac{\lambda}{2D}\right).$$

Since $|S| \leq \alpha N$, $|V(H)| = \frac{\alpha}{2\varepsilon} \cdot N$ and $\lambda/D \leq \frac{2}{\sqrt{D}} \leq 2\eta\sqrt{\varepsilon}$, we have $e(S, S) \leq (\varepsilon + \eta) D \cdot |S|$ and $e(S, V(H)) = D \cdot |S| - e(S, S) \geq (1 - \varepsilon - \eta) \cdot D|S|$. □

While G_0 satisfies $(\alpha N, (1 - \varepsilon - \eta)D)$ expansion, its expander code has a small rate since its right hand side $|V_R(G_0)| = |E(H)| > |V(H)| = |V_L(G_0)|$. Then we construct G_1 such that the design rate of the expander code of $G = G_0 \cup G_1$ is $M/N \in [1/3, 2/3]$. For an even number N and some $M \in [N/2, 2N/3]$ chosen later, G_1 is defined as a random regular bipartite graph with $|V_L(G_1)| = N_1 = N - N'$, $|V_R(G_1)| = M_1 = M - DN'/2$, regular left degree D and regular right degree $D_1 = N_1 \cdot D/M_1$. Since we can choose $M \in [N/2, 2N/3]$ and α to be sufficiently small, such an integer D_1 exists. For example, we can set $M_1 = N_1/2$ and $D_1 = 2D$ give an even N_1 (based on even N' in [17] and even N in our choice).

Then, a random bipartite graph with such parameters satisfies $(\alpha N, (1-\varepsilon)D)$ expansion with high probability for a small α. For completeness we show this calculation in Appendix A and assume this property in the rest of this proof. Overall, because both G_0 and G_1 satisfy $(\alpha N, (1-\varepsilon-\eta)D)$ expansion, $G = G_0 \cup G_1$ is an $(\alpha N, (1 - \varepsilon - \eta)D)$ expander.

Next, consider a codeword that is all 1 in $V_L(G_0)$, and 0 everywhere else. It satisfies all parity checks since the right degree of $V_R(G_0)$ is 2. Moreover, its weight is $\frac{\alpha}{2\varepsilon} N$, and thus the distance of the corresponding expander code is at most $\frac{\alpha}{2\varepsilon} N$.

Finally, we remark that G is regular in $V_L(G)$ and is "almost regular" in $V_R(G)$. Its maximum degree $D_{\max} = D_1$ and average degree on right $D_R = \frac{(M-DN'/2)\cdot D_1 + DN'}{M}$. Since $N' = \frac{\alpha}{2\epsilon}N$ for a sufficiently small α (compared to ε/D) and $M \in [N/2, 2N/3]$, $D_R \geq 0.95\ D_1$ and $D_{\max} \leq 1.1\ D_R$. So our construction meets the requirement of Theorem 3 for a better list-decoding radius than the Johnson bound.

B. Proof of Lemma 11 and Its Generalization

We prove the first lower bound $|\Gamma(S)| \geq (1-k\varepsilon)D\cdot|S| - 2\varepsilon k^2\cdot D$ by a probabilistic argument, where we recall $|S| = k\alpha N$. Suppose $|\Gamma(S)|$ is small. Then we consider a random subset T of size αN in S and upper bound

$$\mathbb{E}\left[\left|\Gamma(T)\right|\right] \leq D\cdot|T| - \left(|S|\cdot D - \left|\Gamma(S)\right|\right)\cdot\frac{|T|\cdot(|T|-1)}{|S|\cdot(|S|-1)}.$$

As justification, consider any neighbor u of S, say u has $d_S(u)$ neighbors in S which are $v_1,\ldots,v_{d_S(u)}$. Observe that the following inequality holds for any $T \subseteq S$

$$1\{u\in\Gamma(T)\} \leq \sum_{i=1}^{d_S(u)} 1\{v_i\in T\} - \sum_{i=2}^{d_S(u)} 1\{v_1\in T\}\cdot 1\{v_i\in T\}.$$

So we take expectation (over T) on both sides:

$$\mathop{\mathbb{E}}_{T}[1\{u\in\Gamma(T)\}] \leq d_S(u)\cdot\frac{|T|}{|S|} - (d_S(u)-1)\cdot\frac{|T|\cdot(|T|-1)}{|S|\cdot(|S|-1)}. \tag{1}$$

At the same time, we know

$$\sum_{u\in\Gamma(S)} d_S(u) = D\cdot|S|$$
$$\text{and}\quad \sum_{u\in\Gamma(S)} (d_S(u)-1) = D\cdot|S| - |\Gamma(S)| \tag{2}$$

Then we consider the summations over $u\in\Gamma(S)$ on the two sides of (1): By linearity of expectation, it becomes

$$\begin{aligned}\mathop{\mathbb{E}}_{T}\left[\left|\Gamma(T)\right|\right] &\leq \sum_u d_S(u)\frac{|T|}{|S|} - \sum_u (d_S(u)-1)\cdot\frac{|T|(|T|-1)}{|S|(|S|-1)}\\ &= D\cdot|T| - \left(|S|\cdot D - \left|\Gamma(S)\right|\right)\cdot\frac{|T|\cdot(|T|-1)}{|S|\cdot(|S|-1)}\\ &\qquad\text{(plug the two summations of (2))}\\ &= |T|\cdot D\left(1 - \left(1-\frac{\left|\Gamma(S)\right|}{D\cdot|S|}\right)\cdot\frac{|T|-1}{|S|-1}\right).\end{aligned}$$

On the other hand, this is at least $|T|\cdot D(1-\varepsilon)$ by the expander property. So we have

$$\begin{aligned}1-\varepsilon &\leq 1 - \left(1-\frac{\left|\Gamma(S)\right|}{D\cdot|S|}\right)\cdot\frac{|T|-1}{|S|-1}\\ \Leftrightarrow\quad \varepsilon &/\left(\frac{|T|-1}{|S|-1}\right) \geq 1 - \frac{\left|\Gamma(S)\right|}{D\cdot|S|}.\end{aligned}$$

This gives

$$\frac{\left|\Gamma(S)\right|}{D\cdot|S|} \geq 1-\varepsilon\cdot\left(k+\frac{k-1}{\alpha N-1}\right).$$

We rewrite it to obtain

$$|\Gamma(S)| \geq (1-\varepsilon k)\cdot D|S| - \varepsilon\frac{(k-1)}{\alpha N-1}\cdot D|S| \geq (1-\varepsilon k)\cdot D|S| - 2\varepsilon Dk^2.$$

1) Generalization: Next we consider an alternative way to compute $\mathbb{E}\left[\left|\Gamma(T)\right|\right]$. The main motivation is to prove a better bound than the above one for $k > 1/2\varepsilon$.

Let us fix S of size $k\alpha N$ and consider $\Gamma(S)$. Since the total degree of S is $D\cdot k\alpha N$, let $\beta_j\cdot D\alpha N$ denote the number of vertices in $\Gamma(S)$ with exactly j neighbors in S. Since the largest degree is $D_{\max}$, by the definition,

$$|\Gamma(S)| = (\beta_1 + \cdots + \beta_{D_{\max}})\cdot D\alpha N.$$

Moreover, by summing up the degrees, we have

$$\beta_1 + 2\beta_2 + \cdots + D_{\max}\cdot\beta_{D_{\max}} = k.$$

Now we consider

$$\mathbb{E}\left[\left|\Gamma(T)\right|\right] = \sum_{i\in\Gamma(S)} \Pr_{T\sim\binom{S}{\alpha N}}[T\cap\Gamma(i)\neq\emptyset], \tag{3}$$

which is at least $(1-\varepsilon)D\alpha N$ from the property of expansion.

Since T is a uniformly random subset of size αN in S, $\Pr_T[T\cap\Gamma(i)\neq\emptyset]$ in (3) only depends on $|\Gamma(i)\cap S|$ — the number of neighbors in S. Hence, we use q_j to denote this probability $\Pr_T[T\cap\Gamma(i)\neq\emptyset]$ for vertices with exactly j neighbors in S. From the definition, q_j equals $\Pr_T[T\cap\Gamma(i)\neq\emptyset] = 1-\Pr_T[T\cap\Gamma(i)=\emptyset]$

$$= 1 - \frac{(|S|-|T|)\cdot(|S|-|T|-1)\cdots(|S|-|T|-j+1)}{|S|\cdot(|S|-1)\cdots(|S|-j+1)}. \tag{4}$$

Plugging this into Eq(3), we have the inequality

$$\sum_{j=1}^{D_{\max}} q_j\cdot\beta_j\cdot D\alpha N \geq (1-\varepsilon)\cdot D\alpha N.$$

To lower bound $|\Gamma(S)|$, we rewrite all constraints as a linear programming:

$$\begin{aligned}\min\quad & \beta_1+\cdots+\beta_{D_{\max}}\\ \text{subject to}\quad & \beta_1+2\cdot\beta_2+\cdots D_{\max}\cdot\beta_{D_{\max}} = k \qquad (5)\\ & \sum_{j=1}^{D_{\max}} q_j\cdot\beta_j \geq (1-\varepsilon) \qquad (6)\\ & \beta_j\geq 0,\quad \forall j.\end{aligned}$$

To prove a lower bound of the above linear program, consider the dual of the above linear program:

$$\begin{aligned}\max\quad & k\cdot x_1 + (1-\varepsilon)\cdot x_2\\ \text{subject to}\quad & j\cdot x_1 + q_j\cdot x_2 \leq 1 \quad \forall j=1,\ldots,D_{\max}, \qquad (7)\\ & x_2\geq 0.\end{aligned}$$

Now we prove the 2nd lower bound by presenting a feasible point (x_1,x_2) in the dual program. We consider the point where x_1 and x_2 are determined by (7) with $j=2$ and $j=3$:

$$2\ x_1 + q_2\cdot x_2 = 1, \qquad 3\ x_1 + q_3\cdot x_2 = 1.$$

We get $x_1 = \frac{q_2-q_3}{3q_2-2q_3}$ and $x_2 = \frac{1}{3q_2-2q_3}$. To simplify x_1 and x_2, we simplify q_2 and q_3 using the fact $k > \frac{1}{2\varepsilon} > 1$ as follows:

$$q_2 = 1 - \frac{(k-1)\alpha N\cdot[(k-1)\alpha N - \frac{k-1}{k} - \frac{1}{k}]}{k\alpha N\cdot(k\alpha N-1)}$$

$$= 1 - \left(\frac{k-1}{k}\right)^2 - \frac{(k-1)\alpha N \cdot (-\frac{1}{k})}{k\alpha N \cdot (k\alpha N - 1)}$$
$$= \frac{2}{k} - \frac{1}{k^2} + O\left(\frac{1}{k^2 \alpha N}\right)$$
$$q_3 = 1 - \frac{(k-1)\alpha N \cdot [(k-1)\alpha N - 1] \cdot [(k-1)\alpha N - 2]}{k\alpha N \cdot (k\alpha N - 1) \cdot (k\alpha N - 2)}$$
$$= 1 - \left(\frac{k-1}{k}\right)^3 + \left(\frac{1}{k} \cdot \frac{1}{k\alpha N - 1} + \frac{2}{k} \cdot \frac{1}{k\alpha N - 2}\right) \cdot \left(\frac{k-1}{k}\right)^2 - \frac{1}{k} \cdot \frac{1}{k\alpha N - 1} \cdot \frac{2}{k} \cdot \frac{1}{k\alpha N - 2} \cdot \frac{k-1}{k}$$
$$= \frac{3}{k} - \frac{3}{k^2} + \frac{1}{k^3} + O\left(\frac{1}{k^2 \cdot \alpha N}\right)$$

Given $k > 1$, we rewrite $x_1 = \frac{-1/k+2/k^2-1/k^3 \pm O\left(\frac{1}{k^2 \cdot \alpha N}\right)}{3/k^2-2/k^3 \pm O\left(\frac{1}{k^2 \cdot \alpha N}\right)} = \frac{2-1/k-k}{3-2/k} \pm O(\frac{1}{\alpha N})$ and $x_2 = \frac{k^2}{3-2/k} \pm O(\frac{1}{\alpha N})$ such that the objective value becomes

$$k \cdot x_1 + (1-\varepsilon) \cdot x_2 = k \cdot \frac{2-1/k-k+k(1-\varepsilon)}{3-2/k} \pm O\left(\frac{k}{\alpha N}\right) = \frac{k}{2} \cdot \frac{4-2/k-2k\varepsilon}{3-2/k} \pm O\left(\frac{k}{\alpha N}\right) = \frac{k}{2}\left(1 - \frac{2k\varepsilon - 1}{3-2/k}\right) \pm O\left(\frac{k}{\alpha N}\right).$$

We show this pair of (x_1, x_2) is feasible in the rest of this section. $k > 1$ implies $x_2 > 0$. Claim 15 below will show that (7) is true for any j. These conclude that $|\Gamma(S)|$ is at least $\left[\frac{k}{2}(1 - \frac{2k\varepsilon-1}{3-2/k}) - O(\frac{k}{\alpha N})\right] \cdot D\alpha N$. Finally we note that the first lower bound is obtained from the dual where x_1 and x_2 are determined by (7) with $j = 1$ and $j = 2$. In the rest of this section, we state Claim 15 and finish its proof.

Claim 15: $x_1 = \frac{q_2 - q_3}{3q_2 - 2q_3}$ and $x_2 = \frac{1}{3q_2 - 2q_3}$ satisfy (7) for any j.

Proof: Recall that x_2 and x_3 satisfy the equations of (7) with $j = 2$ and $j = 3$. For $j = 1$, (7) is true since $q_1 = 1/k$ and $x_1 + \frac{1}{k}x_2 = \frac{2-1/k}{3-2/k} \pm O(\frac{k}{\alpha N}) \leq 1$ given $k > 1$. Next, we show (7) is also true for $j \geq 4$ via the concavity of q_j (comparing to equations of $j = 2$ and $j = 3$).

Specifically, the key property is that these probabilities q_j defined in (4) constitute a strictly concave curve. Namely, for any $j > 1$,

$$q_j - q_{j-1} > q_{j+1} - q_j. \tag{8}$$

To verify $2q_j > q_{j-1} + q_{j+1}$, we prove

$$2 \cdot \frac{(|S|-|T|)\cdots(|S|-|T|-j+1)}{|S|\cdots(|S|-j+1)} < \frac{(|S|-|T|)\cdots(|S|-|T|-j+2)}{|S|\cdots(|S|-j+2)} + \frac{(|S|-|T|)\cdots(|S|-|T|-j)}{|S|\cdots(|S|-j)}.$$

Removing the common factor $\frac{(|S|-|T|)\cdots(|S|-|T|-j+2)}{|S|\cdots(|S|-j+2)}$, this is equivalent to showing

$$2 \cdot \frac{|S|-|T|-j+1}{|S|-j+1} < 1 + \frac{(|S|-|T|-j+1)\cdot(|S|-|T|-j)}{(|S|-j+1)\cdot(|S|-j)}$$

Let $a = |S|$ and $b = |S| - |T|$. This becomes

$$2(b-j+1)(a-j) < (a-j+1)(a-j) + (b-j+1)(b-j)$$

After some algebraic manipulation, it becomes proving

$$0 < (a^2 + b^2 + b) - (2ab + a).$$

The last inequality is equivalent to $0 < (a-b)^2 - (a-b)$, which is always true as long as $a - b = |T| > 1$.

Fix $\ell \geq 4$ and consider the linear combination of (7) with $j = 2$ and $j = \ell$ whose coefficients are $\frac{\ell-3}{\ell-2}$ and $\frac{1}{\ell-2}$. By the concavity of q_2, q_3, and q_ℓ, its L.H.S.

$$3 \cdot x_1 + \left(\frac{\ell-3}{\ell-2} \cdot q_2 + \frac{1}{\ell-2} \cdot q_\ell\right) \cdot x_2 < 3 \cdot x_1 + q_3 \cdot x_2.$$

Since $3 \cdot x_1 + q_3 \cdot x_2 = 1$ by the definition of (x_1, x_2), this implies the linear combination is strictly less than 1. Again, since $2 \cdot x_1 + q_2 \cdot x_2 = 1$, we have $\ell \cdot x_1 + q_\ell \cdot x_2 < 1$. □

Remark: In this remark, we justify our choice of (x_1, x_2) by showing that (1) the minimum value of the primal is achieved by β^* with at most two non-zero entries; (2) more importantly, if β^* has exactly two non-zero entries, they must be adjacent, i.e., $\beta_j > 0$ and $\beta_{j+1} > 0$ for some j. By complementary slackness, these imply that our choice of (x_1, x_2) is optimal for certain regime of parameters.

Specifically, if β^* is supported on three entries say $\ell_1 < \ell_2 < \ell_3$, we have $j \cdot x_1 + q_j \cdot x_2 = 1$ in the dual for $j = \ell_1, \ell_2, \ell_3$ by complementary slackness. Note that $x_2 > 0$ in order to satisfy any two equations. However, the two equations of (7) for $j = \ell_1$ and $j = \ell_3$ indicate $\ell_2 \cdot x_1 + q_{\ell_2} \cdot x_2 > 1$, as follows. Consider their linear combination with coefficients $\frac{\ell_3 - \ell_2}{\ell_3 - \ell_1}$ and $\frac{\ell_2 - \ell_1}{\ell_3 - \ell_1}$: it equals 1 on the RHS from these two equations; but the combination on the LHS is strictly less than $\ell_2 \cdot x_1 + q_{\ell_2} \cdot x_2$ by the concavity of q_j (and $x_2 > 0$). Similarly, if β^* is supported on two non-adjacent entries say ℓ_1 and ℓ_2 with $\ell_1 + 1 < \ell_2$, we have two equations for $j = \ell_1$ and $j = \ell_2$ separately. However, the solution (x_1, x_2) which satisfies these two equations violates other constraints in the dual — one can show $(\ell_1 + 1) \cdot x_1 + q_{\ell_1 + 1} \cdot x_2 > 1$ by the same argument again.

IV. Decoding From Erasures, and Finding Possible Corruptions

First, we show that by combining Lemma 11 and Theorem 7, we can also get a stronger result for decoding from erasures.

Theorem 16: For every $\varepsilon < 1/2$, consider an expander code defined by an $(\alpha N, (1-\varepsilon)D)$ expander G. For every $\xi > 0$, there is a linear-time algorithm that corrects $\frac{1-\xi}{2\varepsilon}\alpha N$ erasures.

Proof: By Lemma 11, for any $1 < k < \frac{1}{\alpha}$ the expander is also a $(k\alpha N, (1 - k\varepsilon)D - \frac{2\varepsilon Dk}{\alpha N}))$ expander. Thus if $1 - k\varepsilon - \frac{2\varepsilon k}{\alpha N} > 1/2 + \xi'$ for a $\xi' > 0$, then by Theorem 7, one can decode from $k\alpha N$ erasures, using the same algorithm. By Fact 12, $\frac{1-\xi}{2\varepsilon} \leq \frac{1}{\alpha}$. This means k can be as large as $\frac{1-\xi}{2\varepsilon}$ for any $\xi > 0$. Notice that if $\frac{1-\xi}{2\varepsilon} \leq 1$, then the theorem is implied by Theorem 7. □

Next, we provide a simple algorithm to find a super set of the corruptions, which is adapted from a similar algorithm

Algorithm 1 The Basic Algorithm Finding a Super Set of Corruptions

```
function FIND(y ∈ F_2^N and Δ ∈ R)
    L ← ∅
    R ← {unsatisfied parity checks of y}
    h ← (1 − 2Δ)D
    while ∃i ∈ V_L \ L s.t. |Γ(i) ∩ R| ≥ h do
        L ← L ∪ {i}
        R ← R ∪ Γ(i)
    end while
    return L
end function
```

in [3]. Let G be an $(\alpha N, (1-\varepsilon)D)$ expander with N left vertices, M right vertices, and left degree D. Let $\mathcal{C}$ be an expander code defined by G. The input y is a corrupted message of a codeword $C_0 \in \mathcal{C}$. Let F be the set of corruptions in y compared to C_0. We use Algorithm 1 to find a super set of F given certain parameters.

By a similar proof to that of proposition 4.3 in [3], we have the following properties.

Lemma 17: If $|\Gamma^1(S)| \geq (1-2\Delta)D|S|$ for any non-empty $S \subseteq F$, then F is contained in L after the while loop.

Proof: Suppose not, then let B be $F \setminus L$ after running the algorithm, $B \neq \emptyset$. Since $B \subseteq F$, we have $|\Gamma^1(B)| \geq (1-2\Delta)D|B|$. So there is a vertex $u \in B$ such that u has at least $(1-2\Delta)D$ neighbors in $\Gamma^1(B)$. We know that $|\Gamma(u) \cap R| < (1-2\Delta)D$, because otherwise u should be added to L then. Thus there has to be a neighbor v of u, such that v is not in R and is only connected to one vertex in B, which is u. As $F \setminus B \subseteq L$, we know $\Gamma(F \setminus B) \subseteq R$. So v connects to one vertex, i.e., u in F. This is not possible since then v has to be unsatisfied and thus it is already in R. □

Lemma 18: In every iteration, if there are multiple vertices that can be added to L and we choose one of them arbitrarily, then we always get the same L after all the iterations.

Proof: Consider two different procedures where they choose different vertices to add to L in their corresponding iterations. Suppose that they get two different L, say L_1 for the first procedure and L_2 for the second. Without loss of generality assume $L_1 \setminus L_2 \neq \emptyset$. Let u be the first vertex in $L_1 \setminus L_2$ that is added in procedure 1. Then all the vertices in L_1 added before u, denoted by the set A, is also contained in L_2. Since vertices can only be added to the set R, for procedure 2 we should always have $|\Gamma(u) \cap R| \geq h$ when $A \subseteq L_2$ and $u \notin L_2$. Thus u has to be added to L_2 in procedure 2. This is a contradiction. Therefore $L_1 = L_2$. □

Lemma 19: If $|\Gamma^1(S)| \geq (1-2\Delta)D|S|$ for any non-empty $S \subseteq F$, then there exists a sequence of choices of the algorithm such that all the elements of F can be added to L in the first $|F|$ iterations.

Proof: We use induction to show that in each of the first $|F|$ iterations, there exists an element in $F \setminus L$ which can be added to L.

In the first iteration, since $|\Gamma^1(F)| \geq (1-2\Delta)D|F|$, there exists $u \in F$ such that $|\Gamma(u) \cap \Gamma^1(F)| \geq h$ for $h = (1-2\Delta)D$. Observe that $\Gamma^1(F) \subseteq R$. So u can be added to L.

Assume in each of the first $i-1 < |F|$ iterations, the algorithm can find a distinct element in F to add to L. In the i-th iteration, let $F' = F \setminus L$. Notice that $|F'| = |F| - (i-1) \geq 1$. Hence $|\Gamma^1(F')| \geq (1-2\Delta)D|F'|$. Thus there exists $u \in F'$ such that $|\Gamma(u) \cap \Gamma^1(F')| \geq (1-2\Delta)D = h$. Observe that $\Gamma^1(F') \subseteq \Gamma^1(F) \subseteq R$. So u can be added to L. □

The above lemmas imply that as long as $|\Gamma^1(S)| \geq (1-2\Delta)D|S|$ for any non-empty $S \subseteq F$, when analyzing Algorithm 1, we can assume without loss of generality that the algorithm first adds all corrupted bits into the set L.

V. Unique Decoding by Guessing Expansion With Iterative Flipping

Let $\varepsilon \in (0, 1/4)$ be an arbitrary constant in this section. We first show an algorithm which has a decoding radius $(1-\varepsilon)\alpha N$. Then by using Lemma 11, we show that the algorithm achieves a decoding radius approximately $\frac{3\alpha}{16\varepsilon}N$ for any $\varepsilon < 1/4$.

The basic idea of the algorithm is to guess the expansion of the set of corrupted entries in the algorithm, say $(1-\gamma)D$. Assume we can correctly guess γ. For the case of $\gamma \geq \frac{2}{3}\varepsilon$, we use a procedure similar to [3] to find a super set of possible corruptions, and then decode from erasures. For the case of $\gamma < \frac{2}{3}\varepsilon$, we first consider the left subset which contains all vertices with at least $(1-3\gamma)D$ unsatisfied checks, and show that this set contains (a constant fraction) more corrupted bits than correct bits. Thus we can flip all bits in this set and reduce the number of errors by a constant fraction. The algorithm then repeats this step for a constant number of times, until the number of errors is small enough, where we can apply an existing algorithm to correct the remaining errors.

We describe our algorithm in Algorithm 2 and then state our main result of this section.

Theorem 20: For every small constant $\beta > 0$, and every $\varepsilon \leq 1/4 - \beta$, let $\mathcal{C}$ be an expander code defined by a $(\alpha N, (1-\varepsilon)D)$ expander graph. There is a linear time decoding algorithm for $\mathcal{C}$ with decoding radius $(1-\varepsilon) \cdot \alpha N$.

To prove the theorem, we focus on the i-th iteration of function DECODING, and show that we can make progress (either reducing the number of errors or decoding the original codeword) in this iteration. Let F_i denote the set of errors at the beginning of iteration i and $\gamma(F_i) \in [0, \varepsilon]$ be the parameter such that $|\Gamma(F_i)| = (1-\gamma(F_i)) \cdot D|F_i|$.

First we show function FIXEDFINDANDDECODE will recover the codeword directly whenever $\gamma_i \geq \frac{2\varepsilon}{3} + \eta$.

Claim 21: If $|F_i| \leq (1-\varepsilon) \cdot \alpha N$, $\gamma_i \geq \frac{2\varepsilon}{3} + \eta$, and $\gamma(F_i) \in [\gamma_i - \eta, \gamma_i)$, then function FIXEDFINDANDDECODE in DECODING will return a valid codeword directly.

Proof: First notice that when $\gamma_i \geq \frac{2\varepsilon}{3} + \eta$, this iteration of DECODING will go to function FIXEDFINDANDDECODE. Let $\gamma := \gamma(F_i)$. We prove that L after FIND has size at most αN. Suppose not. Since $|F_i| \leq (1-\varepsilon) \cdot \alpha N$, by the expander property, for every nonempty $F' \subseteq F_i, |\Gamma(F')| \geq (1-\varepsilon)D|F'|$, so by Lemma 17, after FIND, L covers all the errors. Consider the moment $|L| = \alpha N$. Without loss of generality, we assume $F_i \subseteq L$ (otherwise we can adjust the order of vertices added to L by Lemma 18).

Algorithm 2 Decoding Algorithm for $\varepsilon = 1/4 - \beta$

function MAIN($y \in \mathbf{F}_2^n, \alpha \in \mathbf{R}, \varepsilon \in \mathbf{R}$) //The main procedure.
  Let $\ell = \lceil \log_{1-\beta} \frac{1}{3} \rceil = O(1/\beta)$
  for every $i \in [\ell]$, every $\gamma_i \in \{\eta, 2\eta, \ldots, \lceil \frac{1}{\eta} \rceil \eta\}$, where $\eta := \beta/100$ **do**
    $C' \leftarrow$ DECODING($y, \gamma_1, \ldots, \gamma_\ell, \alpha, \varepsilon$)
    if C' is a valid codeword and the distance between C' and y is at most $(1-\varepsilon)\alpha N$ **then**
      return C'
    end if
  end for
end function

function DECODING($y \in \mathbf{F}_2^n$ and $(\gamma_1, \ldots, \gamma_\ell) \in \mathbf{R}^\ell, \alpha \in \mathbf{R}, \varepsilon \in \mathbf{R}$)
  $z \leftarrow y$
  for $i = 1, \ldots, \ell$ **do**
    if $\gamma_i \geq 2\varepsilon/3 + \eta$ **then**
      $z \leftarrow$ FIXEDFINDANDDECODE(z, α, ε)
      return z
    else
      Let L_0 denote all bits in z with at least $(1 - 3\gamma_i)D$ wrong parity checks
      Flip all the bits in L_0
    end if
  end for
  Apply the decoding of Theorem 8 on z and return the result
end function

function FIXEDFINDANDDECODE($y \in \mathbf{F}_2^N, \alpha \in \mathbf{R}, \varepsilon \in \mathbf{R}$)
  $L \leftarrow$ FIND(y, ε), where FIND is from Algorithm 1
  $y' \leftarrow$ Replace all symbols of y in L by the erasure symbol
  return codeword C' decoded by Theorem 16 on y'
end function

Then we have

$$(1-\varepsilon)D\alpha N \leq |\Gamma(L)| \leq (1-\gamma)D \cdot |F_i| + 2\varepsilon D(\alpha N - |F_i|),$$

because the expansion of F_i is $(1-\gamma)D \cdot |F_i|$ and when adding any vertex in $L \setminus F_i$ to L, the cardinality of R increases by at most $2\varepsilon D$. So

$$(1-\varepsilon)\alpha N \leq (1-\gamma) \cdot |F_i| + 2\varepsilon(\alpha N - |F_i|).$$

As $\gamma \leq \varepsilon$ and $\varepsilon \leq 1/4$, $1 - \gamma - 2\varepsilon > 0$. This implies $|F_i| \geq \frac{1-3\varepsilon}{1-\gamma-2\varepsilon} \cdot \alpha N$. Since $\gamma \geq \gamma_i - \eta \geq \frac{2\varepsilon}{3}$, we have $|F_i| \geq \frac{1-3\varepsilon}{1-8\varepsilon/3}\alpha N$. When $\varepsilon \leq 1/4 - \beta$, one can check that $\frac{1-3\varepsilon}{1-8\varepsilon/3} > 1 - \varepsilon$ always holds. It is contradicting the assumption that $|F_i| \leq (1-\varepsilon)\alpha N$.

As $L \supseteq F_i$ and is of size at most αN, the algorithm can correct all the errors using L and z, given $\varepsilon < 1/4 - \beta$, by Theorem 16. □

Next we discuss the case where $\gamma_i < 2\varepsilon/3 + \eta$, which will result in the function DECODING finding the set L_0 and flipping all the bits in L_0. We show that this will reduce the number of errors by a constant fraction.

Claim 22: If $|F_i| \leq (1-\varepsilon)\alpha N$, $\gamma_i < \frac{2\varepsilon}{3} + \eta$, and $\gamma(F_i) \in [\gamma_i - \eta, \gamma_i)$, then flipping L_0 will decrease the number of errors in z by at least a β fraction.

Proof: Let $\gamma := \gamma(F_i)$ and $N' := \frac{(1+3\eta)|F_i|}{(1-\varepsilon)\alpha}$. We show that $|F_i \cup L_0| < \alpha N'$. To prove it, assume $|F_i \cup L_0| = \alpha N'$, i.e., we only take $\alpha N' - |F_i|$ elements from $L_0 \setminus F_i$, consider these elements together with elements in F_i. Note that by definition of N', as $|F_i| \leq (1-\varepsilon)\alpha N$, $\alpha N' \leq (1+3\eta)\alpha N$. By Lemma 11, $(1 - (1+3\eta)\varepsilon)D\alpha N' - 2\varepsilon D(1+3\eta)^2 \leq |\Gamma(F_i \cup L_0)|$. Notice that $|\Gamma(F_i)| = (1-\gamma)D|F_i|$. Also notice that dding each element of $L_0 \setminus F_i$ to L_0 contributes at most $3\gamma_i D$ to $|\Gamma(F_i \cup L_0)|$, since each element in L_0 has at least $(1-3\gamma_i)D$ wrong parity checks and $\Gamma(F_i \cup L_0)$ contains all the wrong parity checks. So

$$\left(1 - (1+3\eta)\varepsilon - \frac{2\varepsilon(1+3\eta)^2}{\alpha N'}\right) D\alpha N' \leq |\Gamma(F_i \cup L_0)| \leq (1-\gamma)D|F_i| + 3\gamma_i D \cdot (\alpha N' - |F_i|).$$

This implies $|F_i| \geq \frac{1-(1+3\eta)\varepsilon - \frac{2\varepsilon(1+3\eta)^2}{\alpha N'} - 3\gamma_i}{1-\gamma-3\gamma_i} \cdot \alpha N'$. As $\gamma_i \leq \gamma + \eta$, this is $\geq \frac{1-(1+3\eta)\varepsilon - \frac{2\varepsilon(1+3\eta)^2}{\alpha N'} - 3\gamma - 3\eta}{1-4\gamma-3\eta} \cdot \alpha N'$. It is minimized when $\gamma = 0$, since this is $(1 + \frac{\gamma - (1+3\eta)\varepsilon - \frac{2\varepsilon(1+3\eta)^2}{\alpha N'}}{1-4\gamma-3\eta}) \cdot \alpha N'$, which has its derivative being non-negative when $\gamma \in [0, 1/4 - \beta]$. Thus $|F_i| \geq \frac{(1-(1+3\eta)\varepsilon - \frac{2\varepsilon(1+3\eta)^2}{\alpha N'} - 3\eta)}{1-3\eta}\alpha N' = (1 - \varepsilon - \frac{6\eta}{1-3\eta}\varepsilon - \frac{2\varepsilon(1+3\eta)^2}{(1-3\eta)\alpha N'})\alpha N'$. But we know that $|F_i| = \frac{1-\varepsilon}{1+3\eta}\alpha N' = (1 - \varepsilon - \frac{3\eta - 3\varepsilon\eta}{1+3\eta})\alpha N'$. This is a contradiction, since $\varepsilon = 1/4 - \beta$, $\eta = \beta/100$ where β is a small enough constant, which implies $\frac{6\eta}{1-3\eta}\varepsilon + \frac{2\varepsilon(1+3\eta)^2}{(1-3\eta)\alpha N'} - \frac{3\eta - 3\varepsilon\eta}{1+3\eta} = \frac{9\eta\varepsilon + 27\eta^2\varepsilon - 3\eta - 9\eta^2}{1-9\eta^2} + \frac{2\varepsilon(1+3\eta)^2}{(1-3\eta)\alpha N'} \leq \frac{-\eta}{2-18\eta^2}$ which is a negative constant.

Now consider $|F_i \cap L_0|$. The number of vertices in F_i having at least $(1-3\gamma_i)D$ unsatisfied neighbors has to be at least $|F_i|/3$, since otherwise there are $> 2|F_i|/3$ vertices in F_i having $< (1-3\gamma_i)D$ unsatisfied neighbors and this implies the number of unsatisfied neighbors of F_i is $< (1-2\gamma)D|F_i|$, a contradiction. So $|F_i \cap L_0| \geq |F_i|/3$.

Then consider $|L_0 \setminus F_i|$. Because $|F_i \cup L_0| < \alpha N' = \frac{1+3\eta}{1-\varepsilon}|F_i|$, it holds that $|L_0 \setminus F_i| = |F_i \cup L_0| - |F_i| < \frac{\varepsilon+3\eta}{1-\varepsilon}|F_i|$. Because $\varepsilon = 1/4 - \beta, \eta = \beta/100$, this is $\frac{1/4-\beta+3\eta}{3/4+\beta}|F_i| < (1/3 - \beta)|F_i|$.

Hence when we flip all bits in L_0, the number of corruptions decreases by at least $|F_i \cap L_0| - |L_0 \setminus F_i| \geq \beta|F_i|$. □

Proof of Theorem 20: The decoding algorithm is Algorithm 2. The key point is that in the enumerations of the γ_i's, one sequence $(\gamma_i)_{i \in [\ell]}$ provides a good approximation of the actual expansion parameters, i.e. $\forall i \in [\ell]$ in the i-th iteration, $\gamma(F_i) \in [\gamma_i - \eta, \gamma_i)$. Now for every $i \in [\ell]$, we consider i-th iteration. If $\gamma_i \geq 2\varepsilon/3 + \eta$, then by Claim 21, the algorithm returns the correct codeword. If $\gamma_i < 2\varepsilon/3 + \eta$, then by Claim 22, the number of errors can be reduced by β fraction. So in the worst case, when $\ell \geq \log_{1-\beta} \frac{1}{3}$, the number of errors can be reduced to at most $\alpha N/3$ in a

Algorithm 3 Decoding Algorithm for $\varepsilon < 1/4$ With Larger Decoding Radius

function FINAL DECODING FOR LARGE RADIUS($y \in \mathbf{F}_2^N, \alpha \in \mathbf{R}, \varepsilon \in \mathbf{R}$)
  Let $k = \frac{1/4-\eta'}{(1+\frac{2}{\alpha N})\varepsilon}$, with $\eta' = \eta/100$
  Let $z \leftarrow$ MAIN($y, k\alpha, 1/4-\eta'$) from Algorithm 2
  return z
end function

constant number of iterations. Finally the algorithm applies the decoding algorithm from Theorem 8, which corrects the remaining errors.

The running time of Algorithm 2 is linear, since $\ell = O(1)$ and there are constant number of choices for each γ_i takes constant time. The procedures FIXEDFINDANDDECODE and the decoding from Theorem 8 both run in linear time as well. □

By using Theorem 20 and Lemma 11 we can get the following result.

Theorem 23: For all constants $\varepsilon \in (0, \frac{1}{4}), \eta > 0$, if $\mathcal{C}$ is an expander code defined by an $(\alpha N, (1-\varepsilon)D)$ expander, then there is a linear time decoding algorithm for $\mathcal{C}$ with decoding radius $(\frac{3\alpha}{16\varepsilon} - \eta)N$.

Proof: Consider Algorithm 3. By Lemma 11, the expander graph is also a $(k\alpha N, (1-k\varepsilon)D - \frac{2\varepsilon Dk}{\alpha N})$ expander for $k \geq 1$. If k satisfies $k\varepsilon + \frac{2\varepsilon k}{\alpha N} \leq 1/4 - \eta'$ for a small constant η', then by Theorem 20, there is a decoding algorithm with radius $(1 - k\varepsilon - \frac{2\varepsilon k}{\alpha N})k\alpha N$. When $k = \frac{1/4-\eta'}{(1+\frac{2}{\alpha N})\varepsilon}$, this is maximized to be $\frac{\frac{3}{16} - \frac{\eta'}{2} - \eta'^2}{(1+\frac{2}{\alpha N})\varepsilon}\alpha N$. We take η' to be $\eta/100$ such that $k \geq 1$ and the decoding radius becomes $(\frac{3\alpha}{16\varepsilon} - \eta)N$. The running time is linear by Theorem 20. □

VI. IMPROVED UNIQUE DECODING FOR $\varepsilon \leq 1/8$

In this section we provide Algorithm 4 with a better decoding radius for $\varepsilon \leq 1/8$. We state the main result in Theorem 24.

Theorem 24: For all constants $\varepsilon \in (0, 1/8], \eta > 0$, if $\mathcal{C}$ is an expander code defined by an $(\alpha N, (1-\varepsilon)D)$ expander, then there is a linear time decoding algorithm for $\mathcal{C}$ with decoding radius $(\frac{\sqrt{2}-1}{2\epsilon}\alpha - \eta)N$ for $\varepsilon < \frac{3-2\sqrt{2}}{2}$ and decoding radius $(\frac{1-2\varepsilon}{4\varepsilon}\alpha - \eta)N$ for $\varepsilon \geq \frac{3-2\sqrt{2}}{2}$.

Algorithm 4 is again by guessing the correct expansion of the set of corrupted entries. To guarantee that the running time is linear in n, it guesses the expansion with a fine net $\eta' = \varepsilon \cdot \eta/2$. One remark is that one could extend Algorithm 4 to a polynomial time algorithm, which enumerates all possible expansions and replaces the $-\eta \cdot N$ term in the decoding radius by a constant.

In the rest of this section, we prove the correctness of Algorithm 4. Again F denotes the set of corrupted entries. And we assume $|F| = x\alpha N$ and $|\Gamma(F)| = (1-\gamma)D|F|$. Since we enumerate $\tilde{\gamma}\tilde{x}$ from a sequence with gap η', one of them satisfies $\gamma x \in [\tilde{\gamma}\tilde{x}, \tilde{\gamma}\tilde{x} + \eta']$. Now we only consider this pair of $\tilde{\gamma}$ and $\tilde{x}$ in the following analysis.

Next we can bound the expansion of all subsets in F.

Algorithm 4 Decoding Algorithm for $\varepsilon \leq 1/8$

function DECODING($y \in \mathbf{F}_2^N, \varepsilon \in \mathbf{R}, \alpha \in \mathbf{R}, \eta \in \mathbf{R}$)
  for every $\tilde{\gamma}\tilde{x}$ from $\{\eta', 2\eta', \ldots, \lceil \frac{1}{\eta'} \rceil \eta'\}$, where $\eta' = \varepsilon\eta/2$ **do**
    if $\tilde{\gamma}\tilde{x} \geq \varepsilon$ **then**
      $\Delta \leftarrow \sqrt{\tilde{\gamma}\tilde{x}\varepsilon} + \eta'$.
    else
      $\Delta \leftarrow \varepsilon + 2\eta'$.
    end if
    $L \leftarrow$ FIND($y \in \mathbf{F}_2^n$, Δ)
    $y' \leftarrow$ Replace all symbols of y in L by the erasure symbol
    $C' \leftarrow$ Apply the decoding from Theorem 16 on y'.
    return C' if the distance between C' and y is $\leq \frac{1-2\varepsilon}{4\varepsilon}\alpha N$
  end for
end function

Claim 25: Our choice of Δ always satisfies that

$$\forall F' \subseteq F, |\Gamma(F')| \geq (1-\Delta) \cdot D|F'|.$$

Proof: Let $F' \subseteq F$ be an arbitrary non-empty set, and $|F'| = x' \cdot \alpha N$.

If $x' > 1$, then assume $|\Gamma(F')| = (1-\beta)Dx'\alpha N$. We consider the collisions in $\Gamma(F)$ and $\Gamma(F')$. Recall that by collision we mean that given an arbitrary order of the edges, if one edge in this order has its right endpoint the same as any other edge prior to it, then this is called a collision. Note that the total number of collisions for edges with left endpoints in F' is at most the total number of collisions for edges with left endpoints in F, because a collision in $\Gamma(F')$ is also a collision in $\Gamma(F)$. Thus

$$\beta x' \leq \gamma x.$$

Also, since F' has size $x' \cdot \alpha N$, by Lemma 11 we have $|\Gamma(F')| \geq (1-x'\varepsilon)Dx'\alpha N - 2\varepsilon x'^2 D$. So $\beta \leq x'\varepsilon + \frac{2\varepsilon x'}{\alpha N}$. Hence $\beta(\beta - \frac{2\varepsilon x'}{\alpha N})/\varepsilon \leq \gamma x$. Thus $\beta \leq \sqrt{\gamma x \varepsilon} + \frac{2\varepsilon x'}{\alpha N}$ and $|\Gamma(F')| = (1-\beta)D|F'| \geq (1 - \sqrt{\gamma x \varepsilon})D|F'| - 2\varepsilon x'^2 D$. When $\tilde{\gamma}\tilde{x} \geq \varepsilon$, the algorithm sets $\Delta = \sqrt{\tilde{\gamma}\tilde{x}\varepsilon} + \eta'$. So $|\Gamma(F')| \geq (1-\Delta)D|F'|$. When $\tilde{\gamma}\tilde{x} < \varepsilon$, the algorithm sets $\Delta = \varepsilon + 2\eta'$. Notice that $\sqrt{\gamma x \varepsilon} \leq \sqrt{(\tilde{\gamma}\tilde{x} + \eta')\varepsilon} \leq \varepsilon + \eta'$. Hence again $|\Gamma(F')| \geq (1-\Delta)D|F'|$.

If $x' \leq 1$, then again we have two cases. When $\tilde{\gamma}\tilde{x} \geq \varepsilon$, we know $\Delta \geq \varepsilon + \eta'$. So by expansion, $|\Gamma(F')| \geq (1-\varepsilon)D|F'| \geq (1-\Delta)D|F'|$. When $\tilde{\gamma}\tilde{x} < \varepsilon$, the algorithm sets $\Delta = \varepsilon + 2\eta'$. So $|\Gamma(F')| \geq (1-\varepsilon)D|F'| \geq (1-\Delta)D|F'|$. □

Given the guarantee in Claim 25, one can show that L contains all the errors.

Claim 26: After step 9 in Algorithm 4, we have $F \subseteq L$.

Proof: By Claim 25, $\forall F' \subseteq F, |\Gamma(F')| \geq (1-\Delta) \cdot D|F'|$. So $\forall F' \subseteq F, |\Gamma^1(F')| \geq (1-2\Delta) \cdot D|F'|$, since $(1-2\Delta) > 0$ in our setting. By Lemma 17, we know $F \subseteq L$ after FIND. □

Then we calculate the decoding radius and the size of L.

Claim 27: For the branch $\Delta = \sqrt{\tilde{\gamma}\tilde{x}\varepsilon} + \eta'$, if $x \leq \frac{\sqrt{2}-1}{2\varepsilon} - \eta'/\varepsilon$, then $|L| < \frac{1-2\varepsilon}{2\varepsilon}\alpha N$.

Proof: We will use the fact $\Delta \le \sqrt{\gamma x \epsilon} + \eta'$ (since $\gamma x \in [\tilde{\gamma}\tilde{x}, \tilde{\gamma}\tilde{x} + \eta']$ in the correct guessing) extensively in this proof. Now suppose after the iterations, $|L| \ge \frac{1-2\varepsilon}{2\varepsilon}\alpha N$. By Claim 25 and Lemma 19, we denote L' as a set constituted by first adding F and then adding another $\frac{1-2(\sqrt{\gamma x\varepsilon}+\eta')}{2\varepsilon}\alpha N - x\alpha N$ elements. Let $\delta = \frac{|L|-|F|}{\alpha N} = \frac{1-2(\sqrt{\gamma x\varepsilon}+\eta')}{2\varepsilon} - x$. Notice that $|L'| = \frac{1-2(\sqrt{\gamma x\varepsilon}+\eta')}{2\varepsilon} \le \frac{1-2\varepsilon}{2\varepsilon}$. We show that even having this L' leads to a contradiction.

We show that $\delta \ge 0$ and $x+\delta \ge 1$. The reason is as follows. First consider the case $x \ge 1$. Notice that $\gamma \le x\varepsilon + \frac{2\varepsilon x}{\alpha N}$ by Lemma 11 when $x \ge 1$. So $\delta = \frac{1-2(\sqrt{\gamma x\varepsilon}+\eta')}{2\varepsilon} - x \ge \frac{1}{2\varepsilon} - (1 + \sqrt{1+\frac{2}{\alpha N}})x - \eta'/\varepsilon$. When $x \le \frac{\sqrt{2}-1}{2\varepsilon} - \eta'/\varepsilon$ and $\varepsilon \le 1/8$, this is at least 0. $x + \delta \ge 1$ immediately follows. Second if $x < 1$, then $\tilde{\gamma}\tilde{x} \le \gamma x < \varepsilon$, since $\gamma x \in [\tilde{\gamma}\tilde{x}, \tilde{\gamma}\tilde{x} + \eta']$ and $\gamma \le \varepsilon$ by definition of γ. Thus the algorithm should not go to this branch.

Next notice that all the unsatisfied checks are in $\Gamma(F)$ where $|\Gamma(F)| = (1-\gamma)D|F|$, and every element in $L' \setminus F$ contributes at most $2\Delta D$ vertices to R. Hence $|\Gamma(L')| \le |\Gamma(F)| + 2\Delta D \cdot \delta\alpha N$. On the other hand, Lemma 11 implies $|\Gamma(L')| \ge (1 - (x+\delta)\varepsilon)D \cdot (x+\delta)\alpha N - 2\varepsilon(x+\delta)^2 D$. Thus we have

$$\begin{aligned}&(1-(x+\delta)\varepsilon)\cdot(x+\delta)\alpha N - 2\varepsilon(x+\delta)^2\\ \le&(1-\gamma)x\alpha N + 2\Delta\cdot\delta\alpha N \le (1-\gamma)x\alpha N + 2(\sqrt{\gamma x\varepsilon}+\eta')\cdot\delta\alpha N.\end{aligned}$$

In the rest of this proof, we show that our choice of δ yields

$$(1-(x+\delta)\varepsilon)\cdot(x+\delta) - \frac{2\varepsilon(x+\delta)}{\alpha N} > (1-\gamma)x + 2(\sqrt{\gamma x\varepsilon}+\eta')\cdot\delta, \tag{9}$$

which gives a contradiction. Towards that, we rewrite inequality (9) as

$$0 > \varepsilon\delta^2 + (2\varepsilon x - 1 + 2(\sqrt{\gamma x\varepsilon}+\eta'))\,\delta + \varepsilon x^2 - \gamma x + \frac{2\varepsilon(x+\delta)}{\alpha N}.$$

When $(2\varepsilon x - 1 + 2(\sqrt{\gamma x\varepsilon}+\eta'))^2 - 4\varepsilon\left(\varepsilon x^2 - \gamma x + \frac{2\varepsilon(x+\delta)}{\alpha N}\right) > 0$, the quadratic polynomial will be negative at $\delta = \frac{1-2\varepsilon x - 2(\sqrt{\gamma x\varepsilon}+\eta')}{2\varepsilon}$. To verify this, we set $z = \varepsilon x$ and only need to guarantee that

$$(2z - 1 + 2\sqrt{\gamma z})^2 - 4\,z^2 + 4\gamma z + 2(2z - 1 + 2\sqrt{\gamma z})\eta' > 0.$$

This is equivalent to

$$\begin{aligned}&8\gamma z + (8z-4)\sqrt{\gamma z} + 1 - 4z - 2\eta' > 0\\ \Rightarrow\ &8\left(\sqrt{\gamma z} + \frac{2z-1}{4}\right)^2 - 2\eta' + 1 - 4z - 8\left(\frac{2z-1}{4}\right)^2 > 0.\end{aligned}$$

When $z = \varepsilon x \le \frac{\sqrt{2}-1}{2} - \eta'$ (namely $x \le \frac{\sqrt{2}-1}{2\varepsilon} - \eta'/\epsilon$), the residue $1 - 4z - 8(\frac{2z-1}{4})^2 - 2\eta' = \frac{1}{2} - 2z - 2z^2 - 2\eta' > 0$. So the inequality holds. □

Claim 28: For the branch $\Delta = \varepsilon + 2\eta'$, if $x \le \frac{1-2\varepsilon}{4\varepsilon} - 2\eta'/\epsilon$, then $|L| < \frac{1-2\varepsilon}{2\varepsilon}\alpha N$.

Proof: Suppose $|L| \ge \frac{1-2\varepsilon}{2\varepsilon}\alpha N$. Consider $L' \subseteq L$ with $|L'| = \frac{1-2\varepsilon}{2\varepsilon}\alpha N$. Let $\delta = \frac{1-2\varepsilon}{2\varepsilon} - x$. Notice that $\delta \ge 0$ because $x \le \frac{1-2\varepsilon}{4\varepsilon} - 2\eta'/\epsilon, \varepsilon \le 1/8$. Also $x + \delta \ge 1$ since $\varepsilon \le 1/8$.

By Lemma 11, $|\Gamma(L')| \ge (1-(x+\delta)\varepsilon)D|L'| - 2\varepsilon(x+\delta)^2 D$. By Lemma 19 we can consider L' as being constituted by first adding all elements in F and then add another $\delta\alpha N$ elements by the algorithm. Notice that all the unsatisfied checks are in $\Gamma(F)$, $|\Gamma(F)| \le D|F|$, and every element in $L' \setminus F$ contributes at most $2\Delta D$ vertices to R. Hence $|\Gamma(L')| \le D|F| + 2\Delta D\delta\alpha N$. So we have

$$(1-(x+\delta)\varepsilon)D|L'| - 2\varepsilon(x+\delta)^2 D \le |\Gamma(L')| \le D|F| + 2\Delta D\delta\alpha N.$$

Thus

$$(1-(x+\delta)\varepsilon)\cdot(x+\delta) - \frac{2\varepsilon(x+\delta)}{\alpha N} \le x + 2\Delta\delta = x + 2(\varepsilon + 2\eta')\delta.$$

So this is equivalent to

$$(1 - 2\varepsilon - \varepsilon(x+\delta))(x+\delta) - 4\delta\eta' - \frac{2\varepsilon(x+\delta)}{\alpha N} \le (1-2\varepsilon)x.$$

Recall that $\delta + x = \frac{1-2\varepsilon}{2\varepsilon}$. To get a contradiction, we only need

$$(1-2\varepsilon)x < (1-2\varepsilon)^2/4\varepsilon - 4\delta\eta' - \frac{2\varepsilon(x+\delta)}{\alpha N}.$$

This is satisfied by $x \le \frac{1-2\varepsilon}{4\varepsilon} - 2\eta'/\epsilon$. □

Proof of Theorem 24: In Algorithm 4, one of our enumerations has $\tilde{\gamma}\tilde{x}$ such that $\gamma x \in [\tilde{\gamma}\tilde{x}, \tilde{\gamma}\tilde{x} + \eta']$. Now consider this specific enumeration. After the function Find, all the errors are in L by Claim 26.

Now we bound $|L|$. We can pick the smaller bound of x from Claim 27 and Claim 28. If $\varepsilon < \frac{3-2\sqrt{2}}{2}$, then $\frac{\sqrt{2}-1}{2\varepsilon} < \frac{1-2\varepsilon}{4\varepsilon}$. So by Claim 27 and Claim 28 when $x \le \frac{\sqrt{2}-1}{2\varepsilon} - \eta'/\varepsilon$ we have $|L| < \frac{1-2\varepsilon}{2\varepsilon}\alpha N$. If $\varepsilon \in [\frac{3-2\sqrt{2}}{2}, 1/8]$, then $\frac{\sqrt{2}-1}{2\varepsilon} \ge \frac{1-2\varepsilon}{4\varepsilon}$. So by Claim 27 and Claim 28, when $x \le \frac{1-2\varepsilon}{4\varepsilon} - 2\eta'/\varepsilon$, we have $|L| < \frac{1-2\varepsilon}{2\varepsilon}\alpha N$. Since the expander is an $(\alpha N, (1-\varepsilon)D)$ expander, by Theorem 16, one can correct all the errors efficiently using L (as the set of erasures) and the corrupted codeword.

The decoding algorithm runs in linear time because we only have a constant number of enumerations, and each enumeration takes linear time. □

VII. List-Decoding Radius

In this section, we consider expander graphs with bounded maximum degree $D_{\max} = O(1)$. Our main result of this section is the following theorem about the list-decoding radius of almost-regular expander codes. For convenience, we only consider relative distance and relative radii. Throughout this section, $\delta = \alpha/2\varepsilon$ denotes the relative distance, r denotes the relative decoding radius from the Johnson bound, and ρ denotes the relative decoding radius that we will prove.

Theorem 29: Given any $(\alpha N, (1-\varepsilon)D)$-expander G with a regular degree D in V_L and a maximum degree $D_{\max}$ in V_R, its expander code has a relative list-decoding radius at least $\rho = (\frac{1}{2} + \Omega(1/D_{\max})) \cdot \delta$ and list size $O(1)$.

In particular, when $\varepsilon \le 1/4$, $\alpha/\varepsilon \le 0.1$, and $D_{\max} \le 1.1\ D_R$ for the average right degree D_R, the relative list-decoding radius ρ is strictly larger than the Johnson bound r of binary codes with relative distance $\delta = \frac{\alpha}{2\varepsilon}$.

We remark that $D_{\max} \le 1.1\ D_R$ is a relaxation for D_R-regular graphs, which are a standard instantiation of LDPC codes. One immediate open question is to design efficient list-decoding algorithms for expander codes. Since these algorithms provide efficient algorithms for unique decoding with a radius up to

half of the distance, they would improve our decoding results immediately. Hence, we leave this as a future direction.

We finish the proof of Theorem 29 in the rest of this section. For $y \in \mathbf{F}_2^N$, let $|y|$ denote its Hamming weights. To prove Theorem 29, recall that the Johnson bound r of binary codes with relative distance δ is $\frac{1-\sqrt{1-2\delta}}{2}$ [2], which is the limit of the inequality

$$\delta/2 + r^2 - r > 0.$$

Our basic idea is to use locality (which we will define more precisely in the proof) of expander codes to improve the average case in the argument of the Johnson bound. In particular, for L codewords $C_1, \ldots, C_L$ within distance ρN to some string y, we will show that the 1s in $C_1+y, \ldots, C_L+y$ are concentrated on a constant fraction of positions. More precisely, we pick a threshold $\theta := 0.9/D_{\max}$ to show the concentration of 1s. We use the following fact about θ and r in the proof.

Claim 30: When $\varepsilon \le 1/4$ and $\alpha/\varepsilon \le 0.1$, the relative list-decoding radius r of the Johnson bound of relative distance $\delta := \alpha/2\varepsilon$ of binary codes is less than 0.53δ. Furthermore, when $D_{\max} \le 1.1\ D_R$, our choice $\theta := 0.9/D_{\max}$ is at least 0.544δ, which is greater than r.

We defer the proof of Claim 30 to Section VII-B.1 and finish the proof of Theorem 29 here.

Proof of Theorem 29: We fix the threshold $\theta := 0.9/D_{\max}$ as in Claim 30. For convenience, we assume that $\rho < 0.54\delta$ in this proof — otherwise $\rho \ge 0.54\delta$ satisfies $\rho = \frac{\delta}{2}(1 + \Omega(1/D_{\max}))$ and $\rho \ge 0.54\delta$ is strictly larger than $r < 0.53\delta$ (from the above claim) for the second case.

We fix an arbitrary string $y \in \mathbf{F}_2^N$ and consider codewords within relative distance ρ to it, say, there are L codewords $C_1, \ldots,$ and C_L. Let $\Gamma_{odd}(S)$ denote the neighbors of S with an odd number of edges to S. Given $z \in \mathbf{F}_2^N$, let S_z denote the set of 1-entries and $\Gamma_{odd}(z) := \Gamma_{odd}(S_z)$. Back to the codewords $C_1, \ldots, C_L$, since $(y + C_i) + (y + C_j) = C_i + C_j$ is a codeword, $\Gamma_{odd}(y + C_1) = \cdots = \Gamma_{odd}(y + C_L)$ from the definition of the expander code — all codewords satisfy those parity checks. Hence we use Γ_{odd} to denote this neighbor set $\Gamma_{odd}(y + C_1) = \cdots = \Gamma_{odd}(y + C_L)$.

First of all, we lower bound $|\Gamma_{odd}|$. We pick C_j such that $|y + C_j| \in [0.5\delta \cdot N, \rho \cdot N]$. Note that such a C_j exists as long as $L \ge 2$. Then $|\Gamma_{odd}(y+C_j)| \ge (1 - 2\varepsilon \cdot \frac{|y+C_j|}{\alpha N})D \cdot |y + C_j| - 16\varepsilon D$ from Lemma 11. This is at least $0.46\rho D \cdot N - 16\varepsilon D$ given the range of $|y + C_j| \in [0.5\delta \cdot N, 0.54\delta \cdot N]$ (recall $\rho < 0.54\delta$ in this proof). For ease of exposition, we use a simplified lower bound $|\Gamma_{odd}| \ge 0.45\rho \cdot DN$ in the rest of this proof.

Let τ_i denote how many codewords of C_j have ith bit different from the corresponding bit in y, i.e., $\sum_{j=1}^{L} 1\{i \in \mathsf{supp}(y + C_j)\}$. Since $|y + C_j| \le \rho N$, we have $\sum_i \tau_i \le \rho N \cdot L$ — in another word, $\mathbb{E}_i[\tau_i] \le \rho L$. The key difference between our calculation and the Johnson bound is that we will prove $\tau_1, \ldots, \tau_n$ have a large deviation. We call $i \in V_L$ heavy if and only if $\tau_i \ge \theta \cdot L$ for $\theta = 0.9/D_{\max}$ and show that their sum is $\Theta(NL)$:

$$S_h := \sum_{\text{heavy } i} \tau_i \ge 0.45\rho N \cdot (L - D_{\max} \cdot \theta L) = 0.045\rho NL. \tag{10}$$

In particular, for certain choices of parameters, $\theta \ge 0.544\delta$ (from Claim 30) would be strictly larger than $\rho < 0.54\delta$. This implies that $\tau_1, \ldots, \tau_N$ have a large deviation.

To prove Eq (10), the starting observation is that for each $v \in \Gamma_{odd} \subseteq V_R$, $\sum_{i \in \Gamma(v)} \tau_i \ge L$ by the definition of Γ_{odd}. Since v has $\le D_{\max}$ neighbors,

$$\sum_{\text{heavy } i \in \Gamma(v)} \tau_i \ge L - D_{\max} \cdot \theta L.$$

By the double counting argument,

$$\sum_{v \in \Gamma_{odd}} \sum_{\text{heavy } i \in \Gamma(v)} \tau_i \ge (L - D_{\max} \cdot \theta L) \cdot |\Gamma_{odd}| \ge (L - D_{\max} \cdot \theta L) \cdot 0.45\rho D \cdot N.$$

So

$$\sum_{\text{heavy } i} \tau_i \ge \frac{\sum_{v \in \Gamma_{odd}} \sum_{\text{heavy } i \in N(v)} \tau_i}{D} \ge 0.45\rho N \cdot (L - D_{\max} \cdot \theta L).$$

Moreover, let N_h denote the number of heavy elements. We have $\theta L \cdot N_h \le S_h$, which upper bounds N_h by $S_h/(\theta L)$.

Similar to the argument of the Johnson bound, let T denote all triples of the form (i, j_1, j_2) where $i \in [N]$, $j_1, j_2 \in [L]$ and $C_{j_1}(i) \ne C_{j_2}(i)$. Since the distance between C_{j_1} and C_{j_2} is at least δN for any $j_1 \ne j_2$, the number of triples is at least $\binom{L}{2} \cdot \delta N$.

On the other hand, T is equal to $\sum_{i \in [n]} \tau_i(L - \tau_i)$. Then we provide an upper bound on $\sum_{i \in [n]} \tau_i(L - \tau_i)$ under the two constraints $\sum_i \tau_i \le \rho N \cdot L$ and $\sum_{\text{heavy } i} \tau_i \ge 0.45\rho N \cdot (L - D_{\max} \cdot \theta L)$.

Claim 31: Given $\sum_i \tau_i \le \rho N \cdot L$, the threshold $\theta > \rho$, and $\sum_{\text{heavy } i: \tau_i \ge \theta L} \tau_i \ge 0.45\rho N \cdot (L - D_{\max} \cdot \theta L)$, we have

$$\sum_{i \in [n]} \tau_i(L - \tau_i) \le N_h^* \cdot \theta L(L - \theta L) + (N - N_h^*) \cdot \eta L(L - \eta L),$$

where N_h^* is equal to the upper bound $S_h^*/(\theta L)$ for $S_h^* = 0.45\rho N \cdot (L - D_{\max} \cdot \theta L)$ and $\eta = \frac{\rho LN - S_h^*}{L(N - N_h^*)}$.

In another word, the lower bound is obtained when (1) all heavy τ_is are equal to θ with a sum S_h^* equal to the lower bound $0.45\rho N \cdot (L - D_{\max} \cdot \theta L)$; and (2) the light ones have the same value η, which is $< \rho$, such that the total sum $N_h^* \cdot \theta + (N - N_h^*) \cdot \eta = \rho N$ where $N_h^* = S_h^*/\theta$.

We defer the proof of Claim 31 to Section VII-B.2 and combine the two bounds of T to get

$$\binom{L}{2}\delta N \le T \le N_h^* \cdot \theta L(L - \theta L) + (N - N_h^*) \cdot \eta L(L - \eta L)$$

where the right hand side is obtained at $N_h^* = S_h^*/\theta L$ for $S_h^* = 0.45\rho N \cdot (L - D_{\max} \cdot \theta L) \ge 0.045 \cdot \rho NL$ and $\eta = \frac{\rho LN - S_h^*}{L(N - N_h^*)}$. This implies

$$\left(\delta/2 + \frac{N_h^*}{N} \cdot \theta^2 + \frac{N - N_h^*}{N} \cdot \eta^2 - \rho\right) L \le \delta/2.$$

So $L = O(1)$ when the decoding radius ρ satisfies $\delta/2 + \frac{N_h^*}{N}\theta^2 + \frac{N - N_h^*}{N}\eta^2 - \rho = \Omega(1)$. For convenience, let ρ^* be the

limit of ρ satisfying the above inequality such that

$$\delta/2 + \frac{N_h^*}{N}\theta^2 + \frac{N-N_h^*}{N}\eta^2 - \rho^* = 0. \tag{11}$$

Next, we provide explicit bounds on ρ based on (11) and equation $\frac{N_h^*}{N}\theta + \frac{N-N_h^*}{N}\eta = \rho$. Recall that the Johnson bound r is obtained from (11) with $\theta = \eta = r$:

$$\delta/2 + r^2 - r = 0. \tag{12}$$

This implies $r = \frac{1-\sqrt{1-2\delta}}{2}$, which is $\frac{\delta}{2} + \Theta(\delta^2)$ for small δ.

A. Showing $\rho^* = (\frac{1}{2} + \Omega(1/D_{\max}))\delta$

When $1/D_{\max} < 2.5\delta$, the list-decoding radius $r = \frac{\delta}{2} + \Omega(\delta^2)$ from the Johnson bound is $\big(\frac{1}{2} + \Omega(1/D_{\max})\big)\delta$. Hence we only consider $1/D_{\max} \geq 2.5\delta$ to prove $\rho^* = \big(\frac{1}{2} + \Omega(1/D_{\max})\big)\delta$. Observe that $\theta = 0.9/D_{\max} > 2\delta$ is larger than ρ here. We simplify ρ^* in (11) to

$$\begin{aligned}\rho^* > \delta/2 + \frac{N_h^*}{N}\theta^2 &= \delta/2 + \frac{S_h^*}{NL}\cdot\theta \\ &\text{Recall} N_h^* = S_h^*/(\theta L) \text{from Claim 31} \\ &> \delta/2 + 0.045\rho^*\cdot 0.9/D_{\max} \\ &S_h^* \geq 0.045\cdot\rho^* NL \text{from our choice of} \theta.\end{aligned}$$

This implies $\rho^* > \frac{\delta/2}{1-0.04/D_{\max}} = \delta/2\cdot(1+\Omega(1/D_{\max}))$.[1]

B. Showing $\rho^* > r$

In this case, we show $\rho^* = r + \Omega(\delta^3)$ given $\varepsilon \leq 1/4$, $\alpha/\varepsilon \leq 0.1$ and $D_{\max} \leq 1.1\ D_R$, which implies the list-decoding radius of such an expander code is larger than the Johnson bound. To simplify ρ^* in (11), the key is to apply $\frac{N_h^*}{N}\theta + \frac{N-N_h^*}{N}\eta = \rho^*$ to rewrite the two middle terms as

$$\begin{aligned}\frac{N_h^*}{N}\theta^2 + \frac{N-N_h^*}{N}\eta^2 &= (\rho^*)^2 + \frac{N_h^*}{N}(\theta-\rho^*)^2 + \frac{N-N_h^*}{N}(\eta-\rho^*)^2 \\ &= (\rho^*)^2 + \frac{N_h^*}{N}\cdot\frac{N-N_h^*}{N}\cdot(\theta-\eta)^2.\end{aligned}$$

Comparing to (12), the extra term $\frac{N_h^*}{N}\cdot\frac{N-N_h^*}{N}(\theta-\eta)^2$ would always increase the range of ρ^*. Specifically, (11) minus (12) implies

$$\begin{aligned}&(\rho^*)^2 - r^2 + \frac{N_h^*}{N}\cdot\frac{N-N_h^*}{N}\cdot(\theta-\eta)^2 - \rho^* + r = 0 \\ \Leftrightarrow\ &(\rho^*-r)\cdot(1-\rho^*-r) = \frac{N_h^*}{N}\cdot\frac{N-N_h^*}{N}\cdot(\theta-\eta)^2 \\ \Leftrightarrow\ &\rho^* - r = \frac{\frac{N_h^*}{N}\cdot\frac{N-N_h^*}{N}(\theta-\eta)^2}{1-\rho^*-r}.\end{aligned}$$

Since $\theta > 0.544\delta$ and $\eta < \rho \in [0.5\delta, 0.54\delta]$ (from Claim 31), we have $\theta - \eta = \Omega(\delta)$. Moreover, $N_h^*/N = \frac{S_h^*}{\theta L\cdot N}$ from Claim 31 is $\Omega(\rho/\theta)$ which is $\Omega(\delta\cdot D_{\max})$ given $\theta = 0.9/D_{\max}$ and $\rho \geq \delta/2$; then both r and ρ^* are less than 0.05 because the distance $\delta = \frac{\alpha}{2\varepsilon} \leq 0.05$ from the condition $\alpha/\varepsilon \leq 0.1$ of this case. From all discussion above, we have $\rho^* - r = \Omega(D_{\max}\cdot\delta^3)$. □

[1]While a better constant in $\Omega(1/D_{\max})$ is 0.1125 obtained via $\theta = \frac{1}{2D_{\max}}$, we did not intend to optimize the constants in this work.

1) Proof of Claim 30: When $\alpha/\varepsilon \leq 0.1$, the Johnson bound $r = \frac{1}{2}(1-\sqrt{1-2\delta})$ has a Taylor expansion $\frac{\delta}{2} + \frac{2^{-2}}{2\cdot 2!}\cdot(2\delta)^2 + \cdots$ for $\delta = \alpha/2\varepsilon$. This is at most $1.06\cdot\frac{\delta}{2} = \frac{0.265\alpha}{\varepsilon}$.

Then, we show $\frac{1}{D_R} \geq \frac{0.33\alpha}{\varepsilon} - O(kD/M)$. We plan to apply the 2nd lower bound in Lemma 11 for $k := 0.95/\varepsilon$. A subset of size $k\alpha N$ exists because $0.95\alpha/\varepsilon \leq \frac{3.8}{D_R}\cdot(1+\frac{2}{\alpha N})$ from Fact 12. Since $D_R \geq D \geq 4$ for $\varepsilon < 1/4$, $0.95\alpha/\varepsilon$ is less than 1 such that one could find a subset S of size $k\alpha N$ in V_L. Next we apply Lemma 11 to $\Gamma(S)$ and obtain

$$\frac{k}{2}\left(1 - \frac{2k\varepsilon - 1}{3 - 2/k}\right)\cdot D\alpha N - O(k\cdot D) \leq M.$$

For $k = 0.95/\varepsilon$, we use $DN = D_R M$ to simplify it to

$$\frac{0.95}{2\varepsilon}\cdot\left(1 - \frac{0.9}{3 - 2\varepsilon/0.95}\right)\cdot\alpha D_R M - O(kD) \leq M.$$

Since $\varepsilon \leq 1/4$, we have

$$\frac{0.95}{2\varepsilon}\cdot\left(1 - \frac{0.9}{3}\right)\cdot\alpha \leq 1/D_R + O(kD/M),$$

which shows $1/D_R \geq \frac{0.3325\alpha}{\varepsilon} - O(kD/M)$

Given $D_{\max} \leq 1.1\ D_R$, we have that $\theta := 0.9/D_{\max} \geq 0.9/(1.1\ D_R) \geq 0.272\alpha/\varepsilon$ is strictly larger than $r < 0.265\alpha/\varepsilon$.

2) Proof of Claim 31: Our goal is to provide an upper bound on

$$\sum_{i\in[n]} \tau_i(L - \tau_i) \tag{13}$$

given $\sum_i \tau_i \leq \rho N\cdot L$, threshold $\theta > \rho$, and $\sum_{\text{heavy } i:\tau_i \geq \theta L} \tau_i \geq 0.45\rho N\cdot(L - D_{\max}\cdot\theta L)$. We divide the argument into four steps. N_h denotes the number of heavy τ_i and S_h denotes their sum $\sum_{\text{heavy } i}\tau_i$ in this proof.

- When $\sum_i \tau_i$, S_h and N_h are fixed, $\sum_{i\in[n]}\tau_i(L - \tau_i) = \sum_i \tau_i L - \sum_{\text{heavy } i}\tau_i^2 - \sum_{\text{non-heavy } i}\tau_i^2$ is maximized at $\tau_i = S_h/N_h$ for all heavy elements and $\tau_i = (\sum_i \tau_i - S_h)/(N - N_h)$ for non-heavy elements. So we assume heavy elements and non-heavy elements have the same values of τ_i separately. So (13) becomes $N_h\cdot\frac{S_h}{N_h}(L - \frac{S_h}{N_h}) + (N - N_h)\cdot\eta L(L - \eta L)$ for $\eta = \frac{\sum_i \tau_i - S_h}{L(N-N_h)}$.
- Then we fix S_h and N_h and focus on $\sum_i \tau_i$. From the 1st step, $\tau_i = \eta L$ for all non-heavy elements where $\eta = \frac{\sum_i \tau_i - S_h}{L(N-N_h)}$. This is less than $1/2$ since $\eta < \rho < 1/2$ for binary codes. Increasing $\sum_i \tau_i$ will increase $\eta(L-\eta)$ and $N_h\cdot\frac{S_h}{N_h}(L - \frac{S_h}{N_h}) + (N - N_h)\cdot\eta L(L - \eta L)$. So we fix $\sum_i \tau_i = \rho LN$ to be the largest for an upper bound.
- Next, when S_h is fixed, consider the upper bound with N_h:

$$N_h\frac{S_h}{N_h}\left(L - \frac{S_h}{N_h}\right) + (N - N_h)\frac{\rho LN - S_h}{N - N_h}\left(L - \frac{\rho LN - S_h}{N - N_h}\right) \tag{14}$$

$$= \rho L^2 N - \frac{S_h^2}{N_h} - \frac{(\rho LN - S_h)^2}{N - N_h}. \tag{15}$$

Its derivative $(\frac{S_h}{N_h})^2 - (\frac{\rho LN - S_h}{N - N_h})^2$ on N_h is positive, because $\frac{S_h}{N_h}$ is the τ-value for heavy elements and $\frac{\rho LN - S_h}{N - N_h}$ is the value for non-heavy ones. To estimate an upper bound, we fix $N_h^* = S_h/\theta L$ to be the largest possible value.

- Finally, since $\sum_i \tau_i = \rho LN$ is fixed and $\sum_i \tau_i^2$ are convex, the upper bound in (15) is maximized when $\frac{S_h}{N_h^*} - \frac{\rho LN - S_h}{N - N_h^*}$ is minimized. This is achieved at the smallest possible $S_h^* = 0.45\rho N(L - D_{\max} \cdot \theta L)$.

So we obtain an upper bound where for the smallest possible $S_h^* = 0.45\rho N(L - D_{\max} \cdot \theta L)$, $N_h^* = S_h^*/\theta L$ heavy elements have $\tau_i = \theta L$ and the rest of the elements have $\tau_i = \frac{\rho LN - S_h^*}{N - N_h^*}$.

VIII. Open Questions

Our work leaves many intriguing open questions, and we list some of them here.

1) Our distance in Theorem 1 is only shown to be tight by a graph that is not strictly regular on the right. For bipartite expander graphs that are regular on both sides, is it possible to get an improved distance bound, or is the bound in Theorem 1 still tight?
2) Can one design efficient algorithms to correct more errors? In particular, much less is know about $\varepsilon \geq 1/4$ — so far all our improvements over previous results are only for the case of $\varepsilon < 1/4$. Can one get any improvements for the case of $\varepsilon \geq 1/4$?
3) Alternatively, is there any hardness result that prevents us from decoding close to the half distance bound?
4) Can one get a better list-decoding radius for general expander codes? Can one design efficient list-decoding algorithms? As mentioned before, any efficient list-decoding algorithm would also immediately improve our results on unique decoding, and in particular imply unique decoding up to half distance. If there is any hardness result for unique decoding close to half distance, this would also rule out the possibility of list-decoding for general expander codes.

Appendix A
Supplemental Proofs

We finish the calculation omitted in Section III-A here, by showing that random bipartite graphs with certain parameters are good expanders with high probability. We provide one calculation for graphs that is not necessarily regular on the right and another calculation for regular graphs.

Proposition 32: If parameters $\alpha, \epsilon, M, N, D$ satisfies $\left(\frac{e}{\alpha}\right) \cdot \left(\frac{e\alpha ND}{\epsilon M}\right)^{\epsilon D} < 1$, then the probability of a random bipartite graph, where each vertex in V_L has D random neighbors, is $(\alpha N, (1-\epsilon)D)$–expander is $\geq 1 - \left(\left(\frac{e}{\alpha}\right) \cdot \left(\frac{e\alpha ND}{\epsilon M}\right)^{\epsilon D}\right)^{\alpha N}$.

Proof: Suppose the left part of the bipartite graph is $[N]$. Fix a subset X of $[N]$ with size αN, and let $y_i^1, \cdots, y_i^D$ be the neighbours of the i–th vertex in X. Then the expansion of X is less than $(1-\epsilon)D$ is equivalent to $\#\left\{y_i^j\right\} < (1-\epsilon)D\alpha N$, where $i \in X$ and $j \in [D]$.

Arrange y_i^j in the lexicographic order of (i, j). The probability of the value of y_i^j has been taken before it does not exceed $\frac{\#\left\{y_{i'}^{j'} \;\middle|\; (i',j') \prec (i,j)\right\}}{M} < \frac{\alpha ND}{M}$.

So the probability that the expansion of X is less than $(1-\epsilon)D$, is less than $\binom{\alpha ND}{\epsilon\alpha ND} \cdot \left(\frac{\alpha ND}{M}\right)^{\epsilon\alpha ND}$.

Hence, the probability of the random graph is not $(\alpha N, (1-\epsilon)D)$–expander is less than

$$\binom{N}{\alpha N} \cdot \binom{\alpha ND}{\epsilon\alpha ND} \cdot \left(\frac{\alpha ND}{M}\right)^{\epsilon\alpha ND} \tag{16}$$

By the approximation of binomial coefficient: $\binom{A}{B} < \left(\frac{eA}{B}\right)^B$, (16) is less than

$$\left(\frac{eN}{\alpha N}\right)^{\alpha N} \cdot \left(\frac{e\alpha ND}{\epsilon\alpha ND}\right)^{\epsilon\alpha ND} \cdot \left(\frac{\alpha ND}{M}\right)^{\epsilon\alpha ND} = \left(\left(\frac{e}{\alpha}\right) \cdot \left(\frac{e\alpha ND}{\epsilon M}\right)^{\epsilon D}\right)^{\alpha N}$$

□

Given any constant $\varepsilon \in (0, 1)$, by choosing a large enough constant D and let $D_R = \frac{DN}{M}$ be the average degree on the right, Proposition 32 immediately implies the following proposition.

Proposition 33: For any constants $\varepsilon, \eta \in (0, 1)$, there exist constants D, α and $(\alpha N, (1-\varepsilon)D)$-expanders such that $\frac{\alpha}{\varepsilon} \geq \frac{1/e - \eta}{D_R}$.

One can also obtain a regular expander by choosing an integer $D_R = \frac{DN}{M}$ and generating D_R permutations. That such a random graph is an expander has been proved in [1]. We provide an argument for completeness.

Here is a technical lemma summarized from [1].

Proposition 34: Let B be a random (D, D_R)–regular bipartite graph with left size N and right size $\frac{D \cdot N}{D_R}$.[2] Then for all $0 < \alpha < 1$, with probability $\geq 1 - \left(\frac{e}{\alpha}\right)^{-\alpha N}$, all sets of αn vertices in the left part have at least

$$N\left(\frac{D}{D_R}\left(1 - (1-\alpha)^{D_R}\right) - 2\alpha \cdot \sqrt{D \ln e/\alpha}\right)$$

neighbours.

Before we prove this proposition, we show how to choose the parameters to make the expansion at least $(1-\varepsilon)D$. Recall that in the proof of Theorem 10 in Section III-A, we are looking at a random bipartite graph with $N_1 = N - N' \geq N/2$ left vertices, $M_1 = M - DN'/2$ right vertices, regular left degree D and regular right degree $D_R = N_1 \cdot D/M_1$. Since $M_1 \geq M/2 \geq N/4$ and $N_1 \leq N$, we have $D_R \leq 4D$. Next we choose $\alpha = 10^{-3} \cdot (\varepsilon/D)^2$ such that for any $\alpha' \leq 2\alpha$, $(1-\alpha')^{D_R} \in [1 - \alpha' D_R, 1 - (1-\varepsilon/2)\alpha' D_R]$ and $1 - (1-\alpha')^{D_R} \in \left[(1-\varepsilon/2)\alpha' D_R, \alpha' D_R\right]$. Note that any subset of size αN has size $\alpha' N_1$ with $\alpha \leq \alpha' \leq 2\alpha$. Thus we simplify the bound in the above proposition to get the desired expansion

$$N_1\left(\frac{D}{D_R} \cdot (1 - \varepsilon/2)\alpha' D_R - 2\alpha' \cdot \sqrt{D \ln(e/\alpha')}\right)$$

[2]One can think of the random graph as being generated following the Gallager's distribution, i.e. there are D rounds. In each round, randomly generate N/D_R new right vertices by randomly partition the left vertices evenly into N/D_R groups and connect vertices in the i-th group to the i-th right vertex.

$$=N_1 D\alpha' \cdot \left(1-\varepsilon/2-2\sqrt{\frac{\ln(e/\alpha')}{D}}\right)$$
$$\geq N_1 D\alpha' \cdot (1-\varepsilon) = \alpha N D \cdot (1-\varepsilon),$$

for a sufficiently large constant $D = D(\varepsilon)$.

Proof: [Proof of Proposition 34] First, we fix a set of αN vertices in the left part, V, and estimate the probability that $\Gamma(V)$ is small. The probability of a certain vertex in the right part is contained in $\Gamma(V)$ is at least $1-(1-\alpha)^{D_R}$. Thus the expected number of neighbours of V is at least $M \cdot (1-(1-\alpha)^{D_R}) = \frac{nD\left(1-(1-\alpha)^{D_R}\right)}{D_R}$. We will use Azuma inequality to derive that $|\Gamma(V)|$ has a small deviation property, and hence the probability that $|\Gamma(V)|$ less than the expectation minus some deviation is exponentially small.

Actually, we number the edges outgoing from V by 1 through $D\alpha N$. Let X_i be the random variable of the expected size of $|\Gamma(V)|$ given the choice of the first i edges leaving from V. Clearly, $X_1, \cdots, X_{D\alpha N}$ form a martingale and $|X_{i+1} - X_i| \leqslant 1$.

By Azuma's inequality, we have:

$$\mathbb{P}\left(\mathbb{E}\left(X_{D\alpha n}\right) - X_{D\alpha N} > \lambda\sqrt{D\alpha N}\right) < \exp\left(-\lambda^2/2\right)$$

Since there are $\binom{N}{\alpha N}$ choices for the set V, it suffices to choose λ such that

$$\binom{N}{\alpha N} e^{-\lambda^2/2} \text{ is exponentially small.}$$

Since $\binom{N}{\alpha N} \leq (e/\alpha)^{\alpha N}$, we choose $\lambda = 2 \cdot \sqrt{\alpha N \cdot \ln(e/\alpha)}$ to make it exponentially small. Then the deviation becomes

$$\sqrt{D\alpha N} \cdot 2\sqrt{\alpha N \cdot \ln(e/\alpha)} = 2\alpha N \cdot \sqrt{D \ln(e/\alpha)}$$

□

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for many useful suggestions regarding the presentation of this article.

REFERENCES

[1] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Trans. Inf. Theory*, vol. 42, no. 6, pp. 1710–1722, Nov. 1996.
[2] M. Sudan. (2000). *A Crash Course on Coding Theory*. [Online]. Available: http://people.seas.harvard.edu/madhusudan/MIT/coding/ibm/
[3] M. Viderman, "Linear-time decoding of regular expander codes," *ACM Trans. Comput. Theory*, vol. 5, no. 3, pp. 1–25, Aug. 2013.
[4] R. G. Gallager, *Low-Density Parity-Check Codes*. Cambridge, MA, USA: MIT Press, Sep. 1963, doi: 10.7551/mitpress/4347.001.0001.
[5] J. Feldman, M. J. Wainwright, and D. R. Karger, "Using linear programming to decode binary linear codes," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 954–972, Mar. 2005.
[6] J. Feldman, T. Malkin, R. A. Servedio, C. Stein, and M. J. Wainwright, "LP decoding corrects a constant fraction of errors," *IEEE Trans. Inf. Theory*, vol. 53, no. 1, pp. 82–89, Jan. 2007.
[7] S. Arora, C. Daskalakis, and D. Steurer, "Message-passing algorithms and improved LP decoding," *IEEE Trans. Inf. Theory*, vol. 58, no. 12, pp. 7260–7271, Dec. 2012.
[8] A. G. Dimakis, R. Smarandache, and P. O. Vontobel, "LDPC codes for compressed sensing," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3093–3114, May 2012.
[9] J. Mosheiff, N. Resch, N. Ron-Zewi, S. Silas, and M. Wootters, "LDPC codes achieve list decoding capacity," in *Proc. IEEE 61st Annu. Symp. Found. Comput. Sci. (FOCS)*, Nov. 2020, pp. 458–469.
[10] M. G. Luby, M. A. Shokrolloahi, M. Mizenmacher, and D. A. Spielman, "Improved low-density parity-check codes using irregular graphs and belief propagation," in *Proc. IEEE Int. Symp. Inf. Theory*, Jan. 1998, p. 117.
[11] T. J. Richardson and R. L. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 599–618, Feb. 2001.
[12] M. Viderman, "LP decoding of codes with expansion parameter above 2/3," *Inf. Process. Lett.*, vol. 113, no. 7, pp. 225–228, Apr. 2013.
[13] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson, "Randomness conductors and constant-degree lossless expanders," in *Proc. 34th Annu. ACM (STOC)*, 2002, pp. 659–668.
[14] P. Elias, "List decoding for noisy channels," Res. Lab. Electron., Massachusetts Inst. Technol., Cambridge, MA, USA, Tech. Rep. 335, 1957.
[15] J. M. Wozencraft, "List decoding," Res. Lab. Electron., Massachusetts Inst. Technol., Cambridge, MA, USA, Quart. Prog. Rep., Jan. 1958, vol. 48, pp. 90–95.
[16] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Trans. Inf. Theory*, vol. IT-27, no. 5, pp. 533–547, Sep. 1981.
[17] A. Lubotzky, R. Phillips, and P. Sarnak, "Ramanujan graphs," *Combinatorica*, vol. 8, no. 3, pp. 261–277, 1988.
[18] N. Alon and F. R. K. Chung, "Explicit construction of linear sized tolerant networks," *Discrete Math.*, vol. 72, no. 1, pp. 15–19, Dec. 1988. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0012365X88901896
[19] A. Shokrollahi, "LDPC codes: An introduction," in *Coding, Cryptography and Combinatorics*, K. Feng, H. Niederreiter, and C. Xing, Eds. Basel, Switzerland: Birkhauser Basel, 2004, pp. 85–110.
[20] N. Ron-Zewi, M. Wootters, and G. Zemor, "Linear-time erasure list-decoding of expander codes," *IEEE Trans. Inf. Theory*, vol. 67, no. 9, pp. 5827–5839, Sep. 2021.
[21] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bull. Amer. Math. Soc.*, vol. 43, pp. 439–561, Aug. 2006.
[22] N. Kahale, "Eigenvalues and expansion of regular graphs," *J. ACM*, vol. 42, no. 5, pp. 1091–1106, Sep. 1995.
[23] N. Alon and M. Capalbo, "Explicit unique-neighbor expanders," in *Proc. 43rd Annu. IEEE Symp. Found. Comput. Sci.*, Nov. 2002, p. 73.
[24] S. Kopparty. (2018). *Expander Graphs, Mixing Lemma and Applications to Randomness*. [Online]. Available: https://sites.math.rutgers.edu/sk1233/courses/topics-S18/lec2.pdf

Xue Chen received the bachelor's degree from Tsinghua University through Yao Class and the Ph.D. degree from The University of Texas at Austin. He was a Post-Doctoral Researcher with the Theory Group, Northwestern University, USA, and an Assistant Professor with George Mason University, USA. He is currently a Faculty Member of the School of Computer Science and Technology, USTC, China. His research interests include theoretical computer science.

Kuan Cheng received the B.S.E. degree from Shandong University in 2011, the M.S.E. degree from Tsinghua University in 2014, and the Ph.D. degree from Johns Hopkins University in 2019. He was a Post-Doctoral Researcher at The University of Texas at Austin. He is currently an Assistant Professor with the Center on Frontiers of Computing Studies, Peking University. His research interests include randomness in computation, coding theory, and machine learning.

Xin Li received the B.S. and M.S. degrees from Tsinghua University, China, and the Ph.D. degree from The University of Texas at Austin in 2011. He is currently an Associate Professor with the Computer Science Department, Johns Hopkins University. His research interests include theoretical computer science, the use of randomness in computation, complexity theory, coding theory, cryptography, and algorithms.

Minghui Ouyang received the B.S. degree from Peking University in 2021, where he is currently pursuing the Ph.D. degree with the School of Mathematical Science.