Electronic Colloquium on Computational Complexity, Report No. 104 (2014)

It'll probably work out: improved list-decoding through random operations*

Atri Rudra[†] Mary Wootters[‡]

August 8, 2014

[†] Department of Computer Science and Engineering,
University at Buffalo, SUNY
atri@buffalo.edu

[‡] Department of Mathematics,
University of Michigan
wootters@umich.edu

Abstract

In this work, we introduce a framework to study the effect of random operations on the combinatorial list decodability of a code. The operations we consider correspond to row and column operations on the matrix obtained from the code by stacking the codewords together as columns. This captures many natural transformations on codes, such as puncturing, folding, and taking subcodes; we show that many such operations can improve the list-decoding properties of a code. There are two main points to this. First, our goal is to advance our (combinatorial) understanding of list-decodability, by understanding what structure (or lack thereof) is necessary to obtain it. Second, we use our more general results to obtain a few interesting corollaries for list decoding:

1. We show the existence of binary codes that are combinatorially list-decodable from $1/2 - \varepsilon$ fraction of errors with optimal rate $\Omega(\varepsilon^2)$ that can be encoded in *linear* time.
2. We show that any code with $\Omega(1)$ relative distance, when randomly folded, is combinatorially list-decodable $1 - \varepsilon$ fraction of errors with high probability. This formalizes the intuition for why the folding operation has been successful in obtaining codes with optimal list decoding parameters; previously, all arguments used algebraic methods and worked only with specific codes.
3. We show that any code which is list-decodable with suboptimal list sizes has many subcodes which have near-optimal list sizes, while retaining the error correcting capabilities of the original code. This generalizes recent results where subspace evasive sets have been used to reduce list sizes of codes that achieve list decoding capacity.

The first two results follow from the techniques of Wootters (STOC 2013) and Rudra and Wootters (STOC 2014); one of the main technical contributions of this paper is to demonstrate the generality of the techniques in those earlier works. The last result follows from a simple direct argument.

*AR's research supported in part by NSF CAREER grant CCF-0844796 and NSF grant CCF-1161196. MW's research supported in part by a Rackham predoctoral fellowship.

ISSN 1433-8092

1 Introduction

The goal of *error correcting codes* is to enable communication between a sender and receiver over a noisy channel. For this work, we will think of a code $\mathcal{C}$ of block length n and size N over an alphabet Σ as an $n \times N$ matrix over Σ, where each column in the matrix $\mathcal{C}$ is called a *codeword.* The sender and receiver can use $\mathcal{C}$ for communication as follows. Given one of N messages—which we think of as indexing the columns of $\mathcal{C}$—the sender transmits the corresponding codeword over a noisy channel. The receiver gets a corrupted version of the transmitted codeword and aims to recover the originally transmitted codeword (and hence the original message). Two primary quantities of interest are the fraction ρ of errors that the receiver can correct (the *error rate*); and the redundancy of the communication, as measured by the *rate* $R := \frac{\log_{|\Sigma|} N}{n}$ of the code. The central goal is to design codes $\mathcal{C}$ so that both R and ρ are large.

A common approach to this goal is to first design a code matrix $\mathcal{C}_0$ that is "somewhat good," and to modify it to obtain a better code $\mathcal{C}$. Many of these modifications correspond to row or column operations on the matrix $\mathcal{C}_0$: for example, dropping of rows or columns, taking linear combinations of rows or columns, and combining rows or columns into "mega" rows or columns. In this work, we study the effects of such row- and column-operations on the *list decodability* of the code $\mathcal{C}_0$.

List decoding. In the list decoding problem [Eli57, Woz58], the receiver is allowed to output a small list of codewords that includes the transmitted codeword, instead of having to pin down the transmitted codeword exactly. The remarkable fact about list decoding is that the receiver may correct twice as many adversarial errors as is possible in the unique decoding problem. Exploiting this fact has led to many applications of list decoding in complexity theory and in particular, pseudorandomness.[1]

Perhaps the ultimate goal of list decoding research is to solve the following problem.

Problem 1. *For $\rho \in (0, 1 - 1/q)$, construct codes with rate $1 - H_q(\rho)$ that can correct ρ fraction of errors with linear time encoding and linear time decoding.*[2] *Above, H_q denotes the q-ary entropy, and $1 - H_q(\rho)$ is known to be the optimal rate.*

Even though much progress has been made in algorithmic list decoding, we are far from answering the problem above in its full generality. If we are happy with polynomial time encoding and decoding (and large enough alphabet size), then the problem was solved by Guruswami and Rudra [GR08], and improved by several follow-up results [GW13, Kop12, GX12, GX13, DL12, GK13]. However, even with all of this impressive work on algorithmic list decoding, the landscape of list-decoding remains largely unexplored. First, while the above results offer concrete approaches to Problem 1, we do not have a good characterization of which codes are even *combinatorially* list-decodable at near-optimal rate. Second, while we have polynomial-time encoding and decoding, linear-time remains an open problem. In this work, we make some progress in both of these directions.

New codes from old: random operations. In this paper, we develop a framework to study the effect of random operations on the list-decodability of a code. Specific instantiations of these operations are a common approach to Problem 1. For example,

1. In the Folded Reed-Solomon codes mentioned above, one starts with a Reed-Solomon code and modifies it by applying a *folding* operation to each codeword. In the matrix terminology, we bunch up rows to construct "mega" rows.

2. In another example mentioned above [GX13], one starts with a Reed-Solomon code and picks certain positions in the codeword, and also throws away many codewords—that is, one applies a *puncturing* operation the codewords, and then considers a *subcode.* In matrix terminology, we drop rows and columns.

[1] See the survey by Sudan [Sud00] and Guruswami's thesis [Gur04] for more on these applications.

[2] One needs to be careful about the machine model when one wants to claim linear runtime. In this paper we consider the RAM model. For the purposes of this paper, it is fine to consider linear time to mean linear number of $\mathbb{F}_q$ operations and the alphabet size to be be small, say polynomial in $1/\varepsilon$.

3. In [Tre03, IJKW10], the *direct product* operation and the *XOR* operation are used to enhance the list-decodability of codes. In matrix terminology, the direct product corresponds to bunching rows and the XOR operation corresponds to taking inner products of rows.

4. In [GI01, GI03, GI05], the *aggregation* operation is used to construct efficiently list-decodable codes out of list-recoverable codes. In matrix terminology, this aggregation again corresponds to bunching rows.

However, in all of these cases, the operations used are very structured; in the final two, the rate of the code also takes a hit.[3] It is natural to ask how generally these operations can be applied. In particular, if we considered random versions of the operations above, can we achieve the optimal rate/error rate/list size trade-offs? If so, this provides more insight about why the structured versions work.

Recently the authors showed in [RW14] that the answer is "yes" for puncturing of the rows of the code matrix: if one starts with *any* code with large enough distance and randomly punctures the code, then with high probability the resulting code is nearly optimally combinatorially list-decodable. In this work, we extend those results to other operations.

1.1 Our contributions and applications

The contributions of this paper are two-fold. First, the goal of this work is to improve our understanding of (combinatorial) list-decoding. What is it about these structured operations that succeed? How could we generalize? Of course, this first point may seem a bit philosophical without some actual deliverables. To that end, we show how to use our framework to address some open problems in list decoding. We outline some applications of our results below.

In order to state our main results, we pause briefly to set the quantitative stage. There are two main parameter regimes for list-decoding, and we will focus on both in this paper. In the first regime, corresponding the the traditional communication scenario, the error rate ρ is some constant $0 < \rho < 1 - 1/q$. In the second regime, motivated by applications in complexity theory, the error rate ρ is very large. For q-ary codes, these applications require correction from a $\rho = 1 - 1/q - \varepsilon$ fraction of errors, for small $\varepsilon > 0$. In both settings, the best possible rate is given by

$$R^* = 1 - H_q(\rho),$$

where H_q denotes the q-ary entropy. In the second, large-q, regime, we may expand $H_q(1-1/q-\varepsilon)$ to obtain an expression

$$R^*(q,\varepsilon) := 1 - H_q(1 - 1/q - \varepsilon) = \min\left\{\varepsilon, \frac{q\varepsilon^2}{2\log(q)} + O_q(\varepsilon^3)\right\}.$$

For complexity applications it is often enough to design a code with rate $\Omega(R^*(q,\varepsilon))$ with the same error correction capability.

1.1.1 Linear time encoding with near optimal rate.

We first consider the special case of Problem 1 that concentrates on the encoding complexity for binary codes in the high error regime:

Question 1. *Do there exist binary codes with rate $\Omega(\varepsilon^2)$ that can be encoded in linear time and are (combinatorially) list-decodable from a $1/2 - \varepsilon$ fraction of errors?*

Despite much progress on related questions, obtaining linear time encoding with (near-)optimal rate is still open. More precisely, for q-ary codes (for q sufficiently large, depending on ε), Guruswami and Indyk showed that linear time encoding and decoding with near-optimal rate is possible for *unique* decoding [GI05]. For list decoding, they prove a similar result for list decoding but the rate is exponentially small in $1/\varepsilon$ [GI03]. This result can be used with code concatenation to give a similar result for binary codes (see Appendix B

[3] It must be noted that in the work of [Tre03, IJKW10] the main objective was to obtain sub-linear time list decoding and the suboptimal rate is not crucial for their intended applications.

for more details) but also suffers from an exponentially small rate. If we allow for super-linear time encoding in Question 1, then it is known that the answer is yes. Indeed, random linear codes will do the trick [ZP82, CGV13, Woo13] and have quadratic encoding time; In fact, near-linear time encoding with optimal rate also follows from known results.[4]

Our results. We answer Question 1 in the affirmative. To do this, we consider the row-operation on codes given by taking random XORs of the rows of $\mathcal{C}_0$. We show that this operation yields codes with rate $\Omega(\varepsilon^2)$ that are combinatorially list-decodable from $1/2-\varepsilon$-fraction of errors, provided the original code has constant distance and rate. Instantiating this by taking $\mathcal{C}_0$ to be Spielman's code [Spi96], we obtain a linear-time encodable binary code which is nearly-optimally list-decodable.

1.1.2 The folding operation, and random t-wise direct product.

The result of Guruswami and Rudra [GR08] showed that when the *folding* operation is applied to Reed-Solomon codes, then the resulting codes (called folded Reed-Solomon codes) can be list decoded in polynomial time with optimal rate. The folding operation is defined as follows. We start with a q-ary code $\mathcal{C}_0$ of length n_0, and a partition of $[n_0]$ into n_0/t sets of size t, and we will end up with a q^t-ary code $\mathcal{C}$ of length $n = n_0/t$. Given a codeword $c_0 \in \mathcal{C}_0$, we form a new codeword $c \in \mathcal{C}$ by "bunching" together the symbols in each partition set and treating them as a single symbol. A formal definition is given in Section 2. For large enough t, this results in codes that can list decode from $1-\varepsilon$ fraction of errors with optimal rate [GR08, GX12, GX14] when one starts with Reed-Solomon or more generally certain algebraic-geometric codes. In these cases, the partition for folding is very simple: just consider t consecutive symbols to form the n/t partition sets.

Folding is a special case of t-wise aggregation of symbols. Given a code $\mathcal{C}_0$ of length n_0, we may form a new code $\mathcal{C}_0$ of length n by choosing n subsets $S_1, \ldots, S_n \subset [n_0]$ and aggregating symbols according to these sets. This operation has also been used to good effect in the list-decoding literature: in [GI01, GI03, GI05], the sets S_i are defined using expander codes, and the original code $\mathcal{C}_0$ is chosen to be *list-recoverable.* This results in efficiently list-decodable codes, although not of optimal rate. We can also view this t-wise aggregation as a puncturing of a t-wise direct product (where $n = \binom{n_0}{t}$ and all sets of size t are included).

There is a natural intuition for the effectiveness of the folding operation in [GR08, GR09], and for the t-wise aggregation of symbols in [GI01, GI03, GI05]. In short, making the symbols larger increases the size of the "smallest corruptable unit," which in turn decreases the number of error patterns we have to worry about. (See Section 5.2 for more on this intuition). In some sense, this intuition is the reason that random codes over large alphabets can tolerate more error than random codes over small alphabets: indeed, an inspection of the proof that random codes obtain optimal list-decoding parameters shows that this is the crucial difference. Since a random code over a large alphabet is in fact a folding of a random code over a small alphabet, the story we told above is at work here.

Despite this nice-sounding intuition—which doesn't use anything specific about the code—the known results mentioned above do not use it, and rely crucially on specific properties of the original codes, and on algorithmic arguments. It is natural to wonder if the intuition above can be made rigorous, and to hold for *any* original code $\mathcal{C}_0$. In particular,

Question 2. *Can the above intuition be made rigorous? Precisely, are there constants $\delta_0, c_0 > 0$, so that for any $\varepsilon > 0$, any code with distance at least δ_0 and rate at most $c_0\varepsilon$ admits a t-wise folding (or other t-wise aggregation of symbols with $n = n_0/t$) for t depending only on ε, such that the resulting code is combinatorially list-decodable from a $1 - \varepsilon$ fraction of errors?*

The first question mimics the parameters of folded Reed-Solomon codes; the second part is for the parameter regime of [GI01, GI03, GI05]. Notice that both the requirements (distance $\Omega(1)$ and rate $O(\varepsilon)$) are necessary. Indeed, if the original code does not have distance bounded below by a constant, it is easy to

[4] For example, Guruswami and Rudra [GR10] showed that folded Reed-Solomon codes—which can be encoded in near-linear time—concatenated with random inner codes with at most logarithmic block length achieve the optimal rate and fraction of correctable errors tradeoff.

come up with codes where the answer to the above question is "no." The requirement of $O(\varepsilon)$ on the rate of the original code is needed because folding preserves the rate, and the list-decoding capacity theorem implies that any code that can be list decoded from $1-\varepsilon$ fraction of errors must have rate $O(\varepsilon)$.

Our results. We answer Question 2 in the affirmative by considering the operation of random t-wise aggregation. We show that if $n = n_0/t$ (the parameter regime for t-wise folding), the resulting code is list-decodable from a $1-\varepsilon$ fraction of errors, as long as $t = O(\log(1/\varepsilon))$. Our theory can also handle the case when $n \ll n_0$, and obtain near-optimal rate at the same time.

1.1.3 Taking sub-codes.

The result of Guruswami and Rudra [GR08], even though it achieves the optimal tradeoff between rate and fraction of correctable errors is quite far from achieving the best known combinatorial bounds on the worst-case list sizes. Starting with the work of Guruswami [Gur11], there has been a flurry of work on using *subspace evasive subsets* to drive down the list size needed to achieve the optimal list decodability [GW13, DL12, GX12, GX13, GK13]. The basic idea in these works is the following: we first show that some code $\mathcal{C}_0$ has optimal rate vs fraction of correctable tradeoff but with a large list size of L_0. In particular, this list lies in an affine subspace of roughly $\log L_0$ dimensions. A subspace evasive subset is a subset that has a small intersection with any low dimension subset. Thus, if we use such a subset to pick a subcode of $\mathcal{C}_0$, then the resulting subcode will retain the good list decodable properties but now with smaller worst-case lists size. Perhaps the most dramatic application of this idea was used by Guruswami and Xing [GX13] who show that certain Reed-Solomon codes have (non-trivial) exponential list size and choosing an appropriate subcode with a subspace evasive subset reduces the list size to a constant.

However, the intuition that using a subcode can reduce the worst-case list size is not specifically tied to the algebraic properties of the code (i.e, to Reed-Solomon codes and subspace evasive sets). As above, it is natural to ask if this intuition holds more broadly.

Question 3. *Given a code, does there always exist a subcode that has the same list decoding properties as the original code but with a smaller list size? In particular, is this true for random sub-codes?*

Our results. We answer Question 3 by showing that for any code, a random subcode with the rate smaller only by an additive factor of ε can correct the same fraction of errors as the original code but with a list size of $O(1/\varepsilon)$ as long as the original list size is at most N^{ε}. Guruswami and Xing [GX13] showed that Reed-Solomon codes defined over (large enough) extension fields with evaluation points coming from a (small enough) subfield has non-trivial list size of N^{ε}. Thus, our result then implies the random sub-codes of such Reed-Solomon codes are optimally list decodable.[5] We also complement this result by showing that the tradeoff between the loss in rate and the final list size is the best one can hope for in general. We also use the positive result to show another result: given that $\mathcal{C}_0$ is optimally list decodable up to rate ρ_0, its random subcodes (with the appropriate rate) with high probability are also optimally list decodable for any error rate $\rho > \rho_0$.

1.1.4 Techniques

Broadly speaking, the operations we consider fall into two categories: row-operations and column-operations on the matrix $\mathcal{C}$. We use different approaches for the different types of operations.

For row operations (and Questions 1 and 2) we use the machinery of [Woo13, RW14] in a more general context. In those works, the main motivations were specific families of codes (random linear codes and Reed-Solomon codes). In this work, we use the technical framework (implicit in) those earlier papers to answer new questions. Indeed, one of the contributions of the current work is to point out that in fact these

[5] Guruswami and Xing also prove a similar result (since a random subset can be shown to be subspace evasive) so ours gives an arguably simpler alternate proof.

previous arguments apply very generally. For column operations, our results follow from a few simple direct arguments (although the construction for the lower bound requires a bit of care).

Remark 4. *We will specifically handle all row operations on the code matrix mentioned at the beginning of the introduction. For column operations, we handle only column puncturing (taking random subcodes). For many operations, this is not actually an omission: some of the column-analogues of the row-operations we consider are redundant. For example, taking random linear combinations of columns of a* linear code *has the same distribution as a random column puncturing. We do not handle bunching up of columns into mega columns, which would correspond to designing interleaved codes—see Section 2 for a formal definition—and we leave the solution of this problem as an open question.*

1.2 Organization

In Section 2, we set up our formal framework and present an overview of our techniques in Section 3. In Section 4, we state and prove our results about the list-decodability of codes under a few useful random operations; these serve to give examples for our framework. They also lay the groundwork for Section 5, where we return to the three applications we listed above, and resolve Questions 1, 2, and 3. Finally, we conclude with some open questions.

2 Set-up

In this section, we set notation and definitions, and formalize our notion of row and column operations on codes. Throughout, we will be interested in codes $\mathcal{C}$ of length n and size N over an alphabet Σ. Traditionally, $\mathcal{C} \subset \Sigma^n$ is a set of codewords. As mentioned above, we will treat $\mathcal{C}$ as a matrix in $\Sigma^{n\times N}$, with the codewords as columns. We will abuse notation slightly by using $\mathcal{C}$ to denote both the matrix and the set; which object we mean will be clear from context. For a prime power q, we will use $\mathbb{F}_q$ to denote the finite field with q elements.

For $x, y \in \Sigma^n$, we will use $d(x, y)$ to denote the Hamming distance between x and y, and we will use $\mathrm{agr}(x, y) := n - d(x, y)$ to denote the agreement between x and y. We study the list-decodability of $\mathcal{C}$: we say that $\mathcal{C}$ is (ρ, L)-*list-decodable* if for all $z \in \Sigma^n$, $|\{c \in \mathcal{C} : d(c, z) \le \rho\}| < L$. In this work, we will also be interested in the slightly stronger notion of *average-radius list-decodability.*

Definition 1. *A code $\mathcal{C} \subset \Sigma^n$ is (ρ, L)-average-radius list-decodable if for all sets $\Lambda \subset \mathcal{C}$ with $|\Lambda| = L$,*

$$\max_z \sum_{c\in\Lambda} \mathrm{agr}(c, z) \le (1 - \rho)nL.$$

Average-radius list-decodability implies list-decodability [GN13, RW14]. Indeed, the mandate of average-radius list decodability is that, for any L codewords in $\mathcal{C}$, they do not agree too much on average with their center, z. On the other hand, standard list decodability requires that for any L codewords in $\mathcal{C}$, at least one does not agree too much with z. As the average is always smaller than the maximum, standard list-decodability follows from average-radius list-decodability.

We will create new codes $\mathcal{C} \in \Sigma^{n\times N}$ from original codes $\mathcal{C}_0 \in \Sigma_0^{n_0\times N_0}$; notice that we allow the alphabet to change, as well as the size and block length of the code. We will consider code operations $f : \Sigma_0^{n_0\times N_0} \to \Sigma^{n\times N}$ which act on rows and columns of the matrix $\mathcal{C}_0$.

We say that a *basic row operation* takes a code $\mathcal{C}_0$ and produces a row of a new matrix $\mathcal{C}$: that is, it is a function

$$r : \Sigma_0^{n_0\times N_0} \to \Sigma^{N_0}.$$

Two examples of basic row operations that we will consider in this paper are taking linear combinations of rows or aggregating rows. That is:

(a) When $\Sigma = \Sigma_0 = \mathbb{F}_q$, and for a vector $v \in \mathbb{F}_q^{n_0}$, the row operation corresponding to linear combinations of rows is $r_v^{(\mathrm{ip})} : \mathbb{F}_q^{n_0 \times N} \to \mathbb{F}_q^N$, given by

$$r_v^{(\mathrm{ip})}(\mathcal{C}_0) = v^T \mathcal{C}_0.$$

(b) Let $S \subset [n_0]$ be a set of size t, and let $\Sigma = \Sigma_0^t$. Then the row operation corresponding to aggregating rows is $r_S^{(\mathrm{agg})} : \Sigma_0^{n_0 \times N} \to (\Sigma_0^t)^N$, given by

$$r_S^{(\mathrm{agg})}(M) = \left((M_{i,1})_{i \in S}, (M_{i,2})_{i \in S}, \ldots, (M_{i,N})_{i \in S} \right).$$

(Above, we have replaced $\mathcal{C}_0$ with M to ease the number of subscripts).

We will similarly consider *basic column operations*

$$c : \Sigma_0^{n_0 \times N_0} \to \Sigma^{n_0},$$

which take a code $\mathcal{C}_0$ and produce a new column of a matrix $\mathcal{C}$. Analogous to the row operations, we have the following two examples.

(a) When $\Sigma = \Sigma_0 = \mathbb{F}_q$, and for a vector $w \in \mathbb{F}_q^{N_0}$, we can consider

$$c_w^{(\mathrm{ip})}(\mathcal{C}_0) = \mathcal{C}_0 w.$$

(b) Let $T \subset [N_0]$ be a set of size t, and let $\Sigma = \Sigma_0^t$. Then

$$c_T^{(\mathrm{agg})}(M) = \left((M_{1,j})_{j \in T}, (M_{2,j})_{j \in T}, \ldots, (M_{n,j})_{j \in T} \right).$$

The code operations that we will consider in this paper are distributions over a collection of random basic row operations or collection of random basic column operations:

Definition 2. *A random row operation is a distribution $\mathcal{D}$ over n-tuples of basic row operations. We treat a draw $f = (r_1, \ldots, r_n)$ from $\mathcal{D}$ as a code operation mapping $\mathcal{C}_0$ to $\mathcal{C}$ by defining the i^{th} row of $\mathcal{C} = f(\mathcal{C}_0)$ to be $r_i(\mathcal{C}_0)$. Similarly, a random column operation is a distribution $\mathcal{D}$ over N-tuples of basic column operations.*

We say a random row (column) operation $\mathcal{D}$ has independent symbols *(*independent codewords *resp.) if the coordinates are independent. We say a random row operation $\mathcal{D}$ has symbols drawn* independently without replacement *if $(r_1, \ldots, r_n)$ are drawn uniformly at random without replacement from some set R of basic row operations.*

Finally, for a random row operation $\mathcal{D}$ and a sample f from $\mathcal{D}$ note that the columns of $f(\mathcal{C})$ are in one-to-one correspondence with the columns of $\mathcal{C}$. Thus, we will overload notation and denote $f(c)$ for $c \in \mathcal{C}$ to denote the column in $f(\mathcal{C})$ corresponding to the codeword $c \in \mathcal{C}$.

Below, we list several specific random row operations that fit into our framework.

1. *Random Sampling:* Let $\Sigma = \Sigma_0$ be any alphabet, and let $\mathcal{D} = (\mathcal{U}_r)^n$, where $\mathcal{U}_r$ is the uniform distribution on the n_0 basic row operations $r_{e_j}^{(\mathrm{ip})}$ for $j \in [n_0]$, where e_j is the j^{th} standard basis vector. Thus, each row of $\mathcal{C}$ is a row of $\mathcal{C}_0$, chosen independently uniformly with replacement.

2. *Random Puncturing:* Same as above except $r_1, \ldots, r_n$ are chosen *without* replacement.

3. *Random t-wise XOR:* Let $\Sigma_0 = \Sigma = \mathbb{F}_2$ and $\mathcal{D} = (\mathcal{U}_{\oplus,t})^n$. $\mathcal{U}_{\oplus,t}$ is the uniform distribution over the $\binom{n_0}{t}$ basic row operations

$$\left\{ r_v^{(\mathrm{ip})} \, : \, v \in \mathbb{F}_2^{n_0} \text{ has weight } t \right\}.$$

That is, to create a new row of $\mathcal{C}$, we choose t positions from $\mathcal{C}_0$ and XOR them together.

4. *Random t-wise aggregation:* Let $\Sigma = \Sigma_0^t$, for any alphabet Σ_0, and let $\mathcal{D} = (\mathcal{U}_{t,\mathrm{dp}})^n$, where $\mathcal{U}_{t,\mathrm{dp}}$ is the uniform distribution over the $\binom{n_0}{t}$ basic row operations
$$\left\{ r_S^{(\mathrm{agg})} : S \subset [n_0], |S| = t \right\}.$$

5. *Random t-wise folding:* Let $\Sigma = \Sigma_0^t$, for any alphabet Σ_0. For each partition $\pi = (S_1, \ldots, S_{n_0/t})$ of $[n_0]$ into sets of size t, consider the row operation $f_\pi = (r_1, \ldots, r_n)$ where
$$r_j = r_{S_j}^{(\mathrm{agg})}.$$
Let $\mathcal{D}$ be the uniform distribution over f_π for all partitions π.

The following column operations also fit into this framework; in this paper, we consider only the first. We mention the second operation (random interleaving) in order to parallel the situation with columns. We leave it as an open problem to study the effect of interleaving.

1. *Random sub-code:* Let $\Sigma = \Sigma_0$ be any alphabet, and let $\mathcal{D} = (\mathcal{U}_c)^N$, where $\mathcal{U}_c$ is the uniform distribution on the N_0 basic column operations
$$\left\{ c_w^{(\mathrm{ip})} \, : \, w = e_i, i \in [N_0] \right\}.$$
That is, $\mathcal{C}$ is formed from $\mathcal{C}_0$ by choosing codewords independently, uniformly, with replacement from $\mathcal{C}_0$.

   Notice that if $\mathcal{C}_0$ is a linear code over $\mathbb{F}_q$, then this operation is the same if we replace $\{w = e_i \, : \, i \in [N_0]\}$ with all of $\mathbb{F}_q^n$, or with all vectors of a fixed weight, etc. Thus, we do not separately consider random XOR (or inner products), as we do with columns.

2. *Random t-wise interleaving:* In this case $\mathcal{D} = \left(\mathcal{U}_{t,\mathrm{dp}}^c\right)^n$. $\mathcal{U}_{t,\mathrm{dp}}^c$ is the uniform distribution over the $\binom{N_0}{t}$ basic column operations
$$\left\{ c_T^{(\mathrm{agg})} : T \subset [N_0], |T| = t \right\}.$$

3 Overview of Our Techniques

Random Row Operations. In addition to answering Questions 1 and 2, one of the contributions of this work is to exhibit the generality of the techniques developed in [RW14]. As such, our proofs follow their framework. In that work, there were two steps: the first step was to bound the list-decodability in expectation (this will be defined more precisely below), and the second step was to bound the deviation from the expectation. In this work, we use the deviation bounds as a black box, and it remains for us to bound the expectation. We would also like to mention that we could have answered Questions 1 and 2 by applying the random puncturing results from [Woo13, RW14] as a black box to the XOR and direct product of the original code. We chose to unpack the proof to illustrate the generality of the proof technique developed in [Woo13, RW14] (and they also seem necessary to prove the generalization to the operation of taking random linear combinations of the rows of the code matrix).

The results on random row operations in this paper build on the approaches of [Woo13, RW14]. While those works are aimed at specific questions (the list-decodability of random linear codes and of Reed-Solomon codes with random evaluation points), the approach applies more generally. In this paper, we interpret the lessons of [Woo13, RW14] as follows:

> If you take a code over Σ_0 that is list-decodable (enough) up to $\rho_0 = 1 - 1/|\Sigma_0| - \varepsilon$, and do some random (enough) stuff to the symbols, you will obtain a new code (possibly over a different alphabet Σ) which is list-decodable up to $\rho = 1 - 1/|\Sigma| - O(\varepsilon)$. If the random stuff that you have done happens to, say, increase the rate, then you have made progress.

First, our notion of a random row operation $\mathcal{D}$ being random enough is the same as $\mathcal{D}$ having independent symbols (or independent symbols without replacement). Now, we will quantify what it means to be "list-decodable enough" in the setup described above. We introduce a parameter $\mathcal{E} = \mathcal{E}(\mathcal{C}_0, \mathcal{D})$, defined as follows:

$$\mathcal{E}(\mathcal{C}_0, \mathcal{D}) := \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \mathbb{E}_{f \sim \mathcal{D}} \max_{z \in \Sigma^n} \sum_{c \in \mathcal{C}_0} \operatorname{agr}(f(c), z). \tag{1}$$

The quantity $\mathcal{E}$ captures how list-decodable $\mathcal{C}$ is in expectation. Indeed, $\max_z \sum_{c \in \mathcal{C}_0} \operatorname{agr}(f(c), z)$ is the quantity controlled by average-radius list-decodability (Definition 1). To make a statement about the actual average-radius list-decodability of $\mathcal{C}$ (as opposed to in expectation), we will need to understand $\mathcal{E}$ when the expectation and the maximum are reversed:

$$\mathbb{E}_{f \sim \mathcal{D}} \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \max_{z \in \Sigma^n} \sum_{c \in \mathcal{C}_0} \operatorname{agr}(f(c), z).$$

The work of [Woo13, RW14] shows the following theorem.

Theorem 2. *Let $\mathcal{C}_0, \mathcal{D}$ and $\mathcal{C}$ be as above, and suppose that $\mathcal{D}$ has independent symbols. Fix $\varepsilon > 0$. Then*

$$\mathbb{E}_f \max_{z \in \Sigma^n} \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \sum_{c \in \Lambda} \operatorname{agr}(f(c), z) \leq \mathcal{E} + Y + \sqrt{\mathcal{E} Y},$$

where

$$Y = CL \log(N) \log^5(L)$$

for an absolute constant C. For $|\Sigma| = 2$, we have

$$\mathbb{E}_f \max_{x \in \Sigma^n} \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \sum_{c \in \Lambda} \operatorname{agr}(f(c), z) \leq \mathcal{E} + CL\sqrt{n \ln(N)}.$$

Theorem 2 makes the intuition above more precise: Any "random enough" operation (that is, an operation with independent symbols) of a code with good "average-radius list-decodability" (that is, good $\mathcal{E}(\mathcal{C}_0, \mathcal{D})$) will result in a code which is also list-decodable. In Appendix C, we show that Theorem 2 in fact implies the same result when "random enough" is taken to be mean that $\mathcal{D}$ has symbols drawn independently at random instead:

Corollary 1. *Theorem 2 holds when "independent symbols" is replaced by "symbols drawn independently without replacement".*

In this work, we answer Questions 1 and 2 by coming up with useful distributions $\mathcal{D}$ on functions f and computing the parameter $\mathcal{E}$. To do this, we will make use of some average-radius Johnson bounds; we record these in Appendix A.

Random Column Operations. Our result on random subcodes follows from a simple probabilistic method. The argument for showing that the parameters in this positive result cannot be improved, we construct a specific code $\mathcal{C}_0$. The code $\mathcal{C}_0$ consists of various "clusters", where each cluster is the set of all vectors that are close to some vector in another code C^*. The code C^* has the property that it is list decodable from a large fraction of errors and that for smaller error rate its list size is suitably smaller– the existence of such a code with exponentially many vectors follows from the standard random coding argument. This allows the original code $\mathcal{C}_0$ to even have good average-radius list decodability. The fact that the cluster vectors are very close to some codeword in C^* (as well as the fact that C^* has large enough distance) basically then shows that the union bound used to prove the positive result is tight.

4 General Results

In this section, we state our results about the effects of some particular random operations—XOR, aggregation, and subcodes—on list-decodability. In Section 5, we will revisit these operations and resolve Questions 1, 2 and 3.

4.1 Random t-wise XOR

In this section, we consider the row-operation of t-wise XOR. We prove the following theorem.

Theorem 3. *Let $\mathcal{C}_0 \in \mathbb{F}_2^{n_0 \times N}$ be a code with distance $0 < \delta_0 < 1/2$. Let $\mathcal{D} = (\mathcal{U}_{\oplus,t})^n$, as defined in Section 2, and consider the code operation $f \sim \mathcal{D}$. Suppose that $t = 4\ln(1/\varepsilon)\delta_0^{-1}$. Then for sufficiently small $\varepsilon > 0$ and large enough n, with probability $1 - o(1)$, $\mathcal{C} = f(\mathcal{C}_0)$ is $(1/2(1 - O(\varepsilon)), \varepsilon^{-2})$-average-radius list decodable and has rate $\Omega(\varepsilon^2)$.*

With the goal of using Theorem 2, we begin by computing the quantity $\mathcal{E}(\mathcal{C}_0, \mathcal{D})$.

Lemma 1. *Let $\mathcal{C}_0 \in \mathbb{F}_2^{n_0}$ be a code with distance δ_0, and suppose $t \geq \frac{4\ln(1/\varepsilon)}{\delta_0}$. Then*

$$\mathcal{E}(\mathcal{C}_0, \mathcal{D}) \leq \frac{n}{2}\left(L(1+\varepsilon) + \sqrt{L}\right).$$

The proof of Lemma 1 follows from an application of an average-radius Johnson bound (see Appendix A for more on these bounds). The proof is given in Appendix D.1. Given Lemma 1, Theorem 2 implies that with constant probability,

$$\begin{aligned}\max_{z \in \mathbb{F}_2^n} \max_{\Lambda \subset \mathcal{C}, |\Lambda| = L} \frac{1}{L}\sum_{c \in \Lambda} \mathrm{agr}(c, z) &\leq \frac{\mathcal{E}}{L} + C\sqrt{n\ln(N)} \\ &\leq \frac{n}{2}\left(1 + \varepsilon + \frac{1}{\sqrt{L}}\right) + C\sqrt{n\ln N}.\end{aligned}$$

In particular, if $C\sqrt{n\ln N} \leq \varepsilon n$, then in the favorable case $\mathcal{C}$ is $(\rho, L-1)$-average-radius list-decodable, for $L = \varepsilon^{-2}$ and $\rho = \frac{1}{2}\cdot(1 - C'\varepsilon)$ for some constant C'.

It remains to verify the rate R of $\mathcal{C}$. Notice that if $|\mathcal{C}| = N$, then we are done, because then the requirement $C\sqrt{n\ln(N)} \leq \varepsilon n$ reads

$$R = \frac{\log_2(N)}{n} \leq \frac{\varepsilon^2}{C\ln(2)}.$$

Thus, to complete the proof we will argue that f is injective with high probability, and so in the favorable case $|\mathcal{C}| = N$. Fix $c \neq c' \in \mathcal{C}_0$. Then, by the same computations as in the proof of Lemma 1,

$$\mathbb{P}\left\{f(c) = f(c')\right\} = \left(\frac{1}{2}\left(1 + (1-\delta_0)^t\right)\right)^n \leq \left(\frac{1+\varepsilon^2}{2}\right)^n.$$

Using the fact that we will choose $n \geq C\ln(N)/\varepsilon^2$, the right hand side is

$$\left(\frac{1+\varepsilon^2}{2}\right)^{C\ln(N)/\varepsilon^2} = N^{-\ln\left(\frac{2}{1+\varepsilon^2}\right)C/\varepsilon^2} \leq N^{-3}$$

for sufficiently small ε. Thus, by the union bound on the $\binom{N}{2} \leq N^2$ choices for the pairs of distinct codewords (c, c'), we see that $\mathbb{P}\left\{|\mathcal{C}| < N\right\} \leq 1/N$, which is $o(1)$ as desired. This completes the proof of Theorem 3.

Remark 5 (Random inner products for $q > 2$)**.** *For our application (Question 1), $q = 2$ is the interesting case. However, the argument above goes through for $q > 2$. In this case, we may use the first statement of Theorem 2, and statements 2 or 3 of Theorem 8 for the average-radius Johnson bound.*

4.2 Random t-wise aggregation

Theorem 4 below analyzes t-wise aggregation in two parameter regimes. In the first parameter regime, we address Question 2, and we consider t-wise direct product where $n_0 = nt$. In this case, final code $\mathcal{C}$ will have the same rate as the original code $\mathcal{C}_0$, and so in order for $\mathcal{C}$ to be list-decodable up to radius $1 - \varepsilon$, the rate R_0 of $\mathcal{C}_0$ must be $O(\varepsilon)$. Item 1 shows that if this necessary condition is met (with some logarithmic slack), then $\mathcal{C}$ is indeed list-decodable up to $1 - \varepsilon$. In the second parameter regime, we consider what can happen when the rate R_0 of $\mathcal{C}_0$ is significantly larger. In this case, we cannot hope to take n as small as n_0/t and hope for list-decodability up to $1 - \varepsilon$. The second part of Theorem 4 shows that we may take n nearly as small as the list-decoding capacity theorem allows.

Theorem 4. *There are constants C_i, $i = 0, \ldots, 5$, so that the following holds. Suppose $q > 1/\varepsilon^2$. Let $\mathcal{C}_0 \subset \mathbb{F}_q^{n_0}$ be a code with distance $\delta_0 \geq C_2 > 0$.*

1. *Suppose $t \geq C_0 \log(1/\varepsilon) \geq 4\ln(1/\varepsilon)/\delta_0$. Suppose that $\mathcal{C}_0$ has rate*

$$R_0 \leq \frac{C_1 \varepsilon}{\log(q) t \log^5(1/\varepsilon)}.$$

Let $n = n_0/t$, and let $\mathcal{D}_{=} (\mathcal{U}_{t,\mathrm{dp}})^n$ be the t-wise aggregation operation of Section 2. Draw $f \sim \mathcal{D}$, and let $\mathcal{C} = f(\mathcal{C}_0)$. Then with high probability, $\mathcal{C}$ is $(1 - C_3\varepsilon, 1/\varepsilon)$-average-radius list-decodable, and further the rate R of $\mathcal{C}$ satisfies $R = R_0$.

2. *Suppose that $t \geq 4\ln(1/\varepsilon)/\delta_0$, and suppose that $\mathcal{C}_0$ has rate R_0 so that*

$$R_0 \leq \left(\frac{nt}{n_0}\right)\left(\frac{\log(1/\varepsilon)}{\log(q)}\right).$$

Choose n so that

$$n \geq \frac{\log(N)\log(1/\varepsilon)}{\varepsilon}.$$

Let $\mathcal{D}_{=} (\mathcal{U}_{t,\mathrm{dp}})^n$ be the t-wise aggregation operation of Section 2. Draw $f \sim \mathcal{D}$, and let $\mathcal{C} = f(\mathcal{C}_0)$. Then with high probability, $\mathcal{C}$ is $(1 - C_4\varepsilon, 1/\varepsilon)$-average-radius list-decodable, and the rate R of $\mathcal{C}$ is at least

$$R \geq \frac{C_5 \varepsilon}{t \log(q) \log^5(1/\varepsilon)}.$$

The rest of this section is devoted to the proof of Theorem 4. As before, it suffices to control $\mathcal{E}(\mathcal{C}_0, \mathcal{D})$.

Lemma 2. *With the set-up above, we have*

$$\mathcal{E}(\mathcal{C}_0, \mathcal{D}) \leq Cn.$$

Again, the proof of Lemma 2 follows from an average-radius Johnson bound. The proof is given in Appendix D.1. Then by Theorem 2, recalling that

$$Y = CL\log(N)\log^5(L),$$

and $N = |\mathcal{C}_0|$, we have with high probability that

$$\begin{aligned}\mathbb{E}_f \max_{z \in \Sigma^n} \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \sum_{c \in \Lambda} \mathrm{agr}(f(c), z) &\leq \mathcal{E}(\mathcal{C}_0, \mathcal{D}) + Y + \sqrt{\mathcal{E}(\mathcal{C}_0, \mathcal{D}) Y} \\ &\leq O\left(L \log(N) \log^5(L) + n\right).\end{aligned}$$

In the favorable case,

$$\mathbb{E}_f \max_{z \in \Sigma^n} \max_{\Lambda \subset \mathcal{C}, |\Lambda| = L} \frac{1}{L} \sum_{c \in \Lambda} \mathrm{agr}(c, z) \leq O\left(\log(N)\log^5(L) + n/L\right) = O\left(\log(N)\log^5(1/\varepsilon) + n\varepsilon\right). \tag{2}$$

As before, $\mathcal{C}$ is $(1 - C\varepsilon, L - 1)$ average-radius list-decodable, for some constant C, as long as the right hand side is no more than $O(n\varepsilon)$. This holds as long as

$$\log(N)\log^5(1/\varepsilon) \leq n\varepsilon. \tag{3}$$

Equation (3) holds for any choice of n. First, we prove item 1 and we focus on the case that $n_0 = nt$; this mimics the parameter regime the definition of folding (which addresses Question 2). Given $n_0 = nt$, we can translate (3) into a condition on R_0, the rate of $\mathcal{C}_0$. We have

$$R_0 = \frac{\log_q(N)}{n_0} = \frac{\log_q(N)}{nt},$$

and so translating (3) into a requirement on $R(\mathcal{C}_0)$, we see that as long as

$$R_0 \lesssim \frac{\varepsilon}{\log(q) t \log^5(1/\varepsilon)} \lesssim \frac{\varepsilon}{\log(q)\log^6(1/\varepsilon)},$$

then with high probability $\mathcal{C}$ is $(1 - C\varepsilon, L)$-list-decodable. Choose n so that this holds. It remains to verify that the rate R of $\mathcal{C}$ is the same as the rate R_0 of $\mathcal{C}_0$. The (straightforward) proof is deferred to Appendix D.2.

Claim 5. *With $\mathcal{C}_0$ as above and with $n_0 = nt$, $|\mathcal{C}| = N$ with probability at least $1 - o(1)$.*

By a union bound, with high probability both the favorable event (2) occurs, and Claim 5 holds. In this case, $\mathcal{C}$ is $(1 - C\varepsilon, L)$-list-decodable, and the rate R of $\mathcal{C}$ is

$$R = R_0.$$

Next, we consider Item 2, where we may choose $n < n_0/t$, thus increasing the rate. It remains true that as long as (3) holds, then $\mathcal{C}$ is $(1 - C\varepsilon, L)$-list-decodable. Again translating the condition (3) into a condition on $\log_{q^t}(N)/n$, we see that as long as

$$\frac{\log_{q^t}(N)}{n} \leq \frac{\varepsilon}{t\log(q)\log^5(1/\varepsilon)}, \tag{4}$$

then $\mathcal{C}$ is $(1 - C\varepsilon, L)$-list-decodable. Now we must verify that the left-hand-side of (4) is indeed the rate R of $\mathcal{C}$, that is, that $|\mathcal{C}| = N$. As before, the proof is straightforward and is deferred to Appendix D.3.

Claim 6. *With $\mathcal{C}_0$ as above and with n arbitrary, $|\mathcal{C}| = N$ with probability at least $1 - o(1)$.*

Now, recalling our choice of n in (4), with high probability both (2) occurs and Claim 6 holds. In the favorable case, $\mathcal{C}$ is $(1 - C\varepsilon, L)$-list-decodable, as long as the rate R satisfies

$$R = \frac{\log_{q^t}(|\mathcal{C}|)}{n} = \frac{\log_{q^t}(N)}{n} \leq \frac{C\varepsilon}{t\log^5(1/\varepsilon)\log(q)}.$$

This completes the proof of Theorem 4.

4.3 Random sub-codes

In this section we address the case of random sub-codes. Unlike the previous sections, the machinery of [RW14, Woo13] does not apply, and so we prove the results in this section directly. We have the following proposition.

Proposition 1. *Let $\mathcal{C}_0$ be any (ρ, L_0)-list decodable q-ary code. Let $\mathcal{C}$ be a random sub-code of $\mathcal{C}_0$ with $N = pN_0$ (as in the definition in Section 2), where*

$$p = \frac{1}{q^{\varepsilon n} \cdot L_0}.$$

With probability $1 - o(1)$, the random subcode $\mathcal{C}$ is $\left(\rho, \frac{3}{\varepsilon}\right)$-list decodable. Further, the number of distinct columns n $\mathcal{C}$ is at least $pN_0/2$.

The proof of Proposition 1 follows straightforwardly from some Chernoff bounds. We defer the proof to Appendix E.2.

Remark 6. *In Proposition 1, the choice of $3/\varepsilon$ for the final list size was arbitrary in the sense that the 3 can be made arbitrarily close to 1 (assuming ε is small enough).*

Proposition 1 only works for the usual notion of list decodability. It is natural to wonder if a similar result holds for average-radius list decodability. We show that such a result indeed holds (though with slightly weaker parameters) in Appendix E.

It is also natural to wonder if one can pick a larger value of p—closer to $1/L_0$ than to $1/(q^{\varepsilon n} L_0)$—in the statement of Proposition 1. In particular, if L_0 is polynomial in n, could we pick $p = q^{-o(\varepsilon n)}$? In Appendix E, we show that this is not in general possible. More precisely, we show the following theorem.

Theorem 7. *For every $\rho > 0$, and for every $0 < \alpha < \frac{1-\rho}{12}$, and for every n sufficiently large, there exists a code $\mathcal{C}_0$ with block length n that is (ρ, n)-average-radius list decodable such that the following holds. Let $\mathcal{C}$ be obtained by picking a random sub-code of $\mathcal{C}_0$ of size $N = pN_0$ where $p = q^{-\alpha n}/n$. Then with high probability if $\mathcal{C}$ is (ρ', L)-list decodable for any $\rho' \geq 1/n$, then $L \geq \Omega(1/\alpha)$.*

5 Applications

Finally, we use the results of Section 4 to resolve Questions 1, 2, and 3.

5.1 Linear time near optimal list decodable codes

First, we answer Question 1, and give linear-time encodable binary codes with the optimal trade-off between rate and list-decoding radius. Our codes will work as follows. We begin with a linear-time encodable code with constant rate and constant distance; we will use Spielman's variant on expander codes [Spi96, Theorem 19]. These codes have rate 1/4, and distance $\delta_0 \geq 0$ (a small positive constant). Notice that a random puncturing of $\mathcal{C}_0$ (as in [Woo13, RW14]) will not work, as $\mathcal{C}_0$ does not have good enough distance—however, a random XOR, as in Section 4.1 will do the trick.

Corollary 2. *There is a randomized construction of binary codes $\mathcal{C} \in \mathbb{F}_2^n$ so that the following hold with probability $1 - o(1)$, for any sufficiently small ε and any sufficiently large n.*

1. *$\mathcal{C}$ is encodable in time $O(n \ln(1/\varepsilon))$.*
2. *$\mathcal{C}$ is (ρ, L)-average-radius list-decodable with $\rho = \frac{1}{2}(1 - C\varepsilon)$ and $L = \varepsilon^{-2}$, where C is an absolute constant.*
3. *$\mathcal{C}$ has rate $\Omega(\varepsilon^2)$.*

Indeed, let $\mathcal{C}_0$ be as above. Let $t = 4\ln(1/\varepsilon)\delta_0^{-1}$, and choose $f \sim (\mathcal{U}_{\oplus,t})^n$, as in Theorem 3. Let $\mathcal{C} = f(\mathcal{C}_0)$. Items 2. and 3. follow immediately from Theorem 3, so it remains to verify Item 1 of Theorem 2, that $\mathcal{C}$ is linear-time encodable. Indeed, we have

$$\mathcal{C}(x) = A\mathcal{C}_0(x),$$

where $A \in \mathbb{F}_2^{n\times n_0}$ is a matrix whose rows are binary vectors with at most t nonzeros each. In particular, the time to multiply by A is $nt = O(n\ln(1/\varepsilon))$, as claimed.

5.2 Random Folding

Next, we further discuss Question 2, which asked for a rigorous version of the intuition behind results for folded Reed-Solomon codes and expander-based symbol aggregation. The intuition is that increasing the alphabet size effectively reduces the number of error patterns a decoder has to handle, thus making it easier to list-decode. To make this intuition more clear, consider the following example when $q = 2$. Consider an error pattern that corrupts a $1 - 2\varepsilon$ fraction of the *odd* positions (the rest do not have errors). This error pattern must be handled by any decoder which can list decode from $1/2 - \varepsilon$ fraction of errors. On the other hand, consider a 2-folding (with partition as above) of the code; now the alphabet size has increased, so we hope to correct $1 - 1/2^2 - \varepsilon = 3/4 - \varepsilon$ fraction of errors. However, the earlier error pattern affects a $1 - 2\varepsilon$ of the new, folded symbols. Thus, in the folded scenario, an optimal decoder need not handle this error pattern, since $1 - 2\varepsilon > 3/4 - \varepsilon$ (for small enough ε).

In Theorem 4, Item 1, we have shown that if $\mathcal{C}_0$ is any code with distance bounded away from 0 and with rate sufficiently small (slightly sublinear in ε), has abundant random t-wise aggregation of symbols which are list-decodable up to a $1 - \varepsilon$ fraction of errors, when $n = n_0/t$ and t is large enough (depending only on ε and q). This is the same parameter regime as folded Reed-Solomon codes (up to logarithmic factors in the rate), and thus the Theorem answers Question 2 insofar as it lends a rigorous way to interpret t-wise aggregation in this parameter regime.

Remark 7. *While the intuition above applies equally well to folding and more general t-wise symbol aggregation, We note that a random folding and a random symbol aggregation are not the same thing. In the latter, the symbols of the new code may overlap, while in the former they may not. However, allowing overlap makes our computations simple; since the goal was to better understand the intuition above, we have done our analysis for the simpler case of t-wise symbol aggregation. It is an interesting open question to find a (clean) argument for the folding operation, perhaps along the lines of the argument of Corollary 1 for puncturing vs. sampling.*

5.3 Applications of random sub-codes

Finally, we observe that Proposition 1 immediately answers Question 3 in the affirmative. Indeed, suppose that $\mathcal{C}_0$ is (ρ_0, L_0)-list-decodable with rate R_0. Then Proposition 1 implies that with high probability, for any sufficiently small ε, a random subcode of rate

$$R_0 - O\left(\varepsilon\log(q) + \frac{\log(L_0)}{n}\right)$$

is $(\rho_0, 3/\varepsilon)$-list-decodable. In particular, if we start out with a binary code with constant rate and large but subexponential list size, the resulting subcode will also have constant rate, and constant list size. Further, Guruswami and Xing [GX13] showed that for every real R, $0 < \varepsilon < 1 - R$ and prime power q, there is an integer $m > 1$ such that Reed-Solomon codes defined over $\mathbb{F}_{q^m}$ with the evaluation points being $\mathbb{F}_q$ of rate R can be list decoded from the optimal $1 - R - \varepsilon$ fraction of errors with list size N^{ε}. Thus, Proposition 1 that randoms sub-codes are optimally list decodable (in all the parameters). We remark that this result also follows from the work of Guruswami and Xing [GX13]: our proof is arguable simpler (though we have not algorithmic guarantee unlike the results of [GX13]).

Given this, it is natural to ask about the list-decodability of the subcode $\mathcal{C}$ when the error radius ρ may be different than ρ_0. It turns out that this also follows from Proposition 1: below, we will use Proposition 1 to argue that if a code $\mathcal{C}_0$ is optimally list decodable for some fixed $\rho_0 > 0$ fraction of errors, then its random subcodes with high probability are optimally list decodable from ρ fraction of errors for any $\rho_0 \le \rho < 1-1/q$. Towards that end, we will make the following simple observation:

Lemma 3. *Let $\mathcal{C}$ be (ρ, L)-list decodable q-ary code. Then for every $\rho \le \rho' < 1 - 1/q$, $\mathcal{C}$ is also (ρ', L')-list decodable, where*

$$L' \le L \cdot q^{n(H_q(\rho') - H_q(\rho) + o(1))} \cdot 2^n.$$

Proof. Consider a received word $y \in [q]^n$ such that $|\mathcal{C} \cap B_q(y, \rho' n)| = L'$. Now we claim that there exists a $z \in B_q(y, \rho' n)$ such that

$$\begin{aligned} |B_q(z, \rho n) \cap \mathcal{C}| &\ge L' \cdot \frac{(q-1)^{\rho n}}{|B_q(y, \rho' n)|} && (5) \\ &\ge L' \cdot \frac{q^{H_q(\rho)n - o(n)}}{2^n} \cdot \frac{1}{q^{H_q(\rho' n)}}. && (6) \end{aligned}$$

In the above the second inequality follows from the following facts: volume of q-ary Hamming balls of radius γn are bounded from above by $q^{H_q(\gamma)n}$ and from below by $q^{H_q(\gamma)n - o(n)}$ (and that $\binom{n}{\rho n}(q-1)^{\rho n} \ge q^{H_q(\rho)n - o(n)}$). (6) along with the fact that $\mathcal{C}$ is (ρ, L)-list decodable proves the claimed bound on L'.

To complete the proof we argue (5): we show the existence of z by the probabilistic method:[6] pick $z \in B_q(y, \rho' n)$ uniformly at random. Fix a $c \in \mathcal{C} \cap B_q(y, \rho' n)$. Then

$$\mathbb{P}\{c \in B_q(z, \rho n)\} = \frac{|B_q(c, \rho n) \cap B_q(y, \rho' n)|}{B_q(y, \rho' n)}.$$

Next we argue that

$$|B_q(c, \rho n) \cap B_q(y, \rho' n)| \ge (q-1)^{\rho n}. \tag{7}$$

Note that the above implies that

$$\mathbb{E}\left[|B_q(z, \rho n) \cap \mathcal{C}|\right] \ge L' \cdot \frac{(q-1)^{\rho n}}{|B_q(y, \rho' n)|},$$

which would prove (5). To see why (7) is true, consider any ρn positions where c and y agree on. Note that if we change all of those values (to any of the $(q-1)^{\rho n}$ possibilities) to obtain c', then we have $d(c', y) \le \rho' n$ and $d(c', c) = \rho n$, which proves (7). □

Lemma 3 along with Proposition 1 implies the following.

Corollary 3. *Let $q \ge 2^{1/\varepsilon}$. Let $\mathcal{C}_0$ be a (ρ, L)-list decodable q-ary code with optimal rate $1 - H_q(\rho) - \varepsilon$. Then for any $\rho' \ge \rho$, with probability at least $1 - o(1)$, a random subcode $\mathcal{C}$ of $\mathcal{C}_0$ of rate $1 - H_q(\rho') - O(\varepsilon)$ is $(\rho', O(1/\varepsilon))$-list decodable.*

Remark 8. *The bound in Lemma 3 is tight up to the $q^{o(n)} \cdot 2^n$ factor. In particular, one cannot have a bound of $L \cdot q^{\gamma n}$ for any $\gamma < H_q(\rho') - H_q(\rho)$ since that would contradict the list decoding capacity bounds.*

6 Open Questions

In this work we have made some (modest) progress on understanding on how random row and column operations change the list decodability of codes. We believe that our work highlights many interesting open questions. We list some of our favorites below:

[6] This part of the proof is similar to the argument used to prove the Elias-Bassalygo bound [GRS14].

1. Theorem 4 is proved for random t-wise direct product codes. It would be nice to prove the analog of item 1 in Theorem 4 for random t-wise folding so that we can formally answer Question 2 in the affirmative.

2. We did not present any results for random t-wise interleaving. Gopalan, Guruswami and Raghavendra have shown that for any code $\mathcal{C}_0$ its t-wise interleaved code $\mathcal{C}$ (that is the code that deterministically applies all possible basic column operations that bunch together the $\binom{N_0}{t}$ subsets of columns of size t) the list decodability does not change by much [GGR11]. In particular, they show that if $\mathcal{C}_0$ is (ρ, L)-list decodable then $\mathcal{C}$ is $(\rho, L^{O(1)})$-list decodable. However, for *random* t-wise interleaving the list decoding radius might actually improve.[7] We leave open the question of resolving this possibility.

3. Following the result of Guruswami and Xing [GX13], Corollary 3 implies that random sub-codes of Reed-Solomon codes over $\mathbb{F}_{q^m}$ (for large enough m) with evaluation points from the sub-field $\mathbb{F}_q$ have optimal list decodable properties. We believe that we should be able to derive such a result even if we start from any Reed-Solomon codes or at the very least if one starts off with a randomly punctured Reed-Solomon codes. Note that even though the results of [RW14] give near optimal list decodability results of Reed-Solomon codes, their results are logarithmic factors off from the optimal rate bounds. Can we prove non-trivial exponential bound on the list size for list decoding rate R Reed-Solomon codes from $1 - R - \varepsilon$ fraction of errors? A very special case of this is proved in [GX13], but the general question is open. Such a statement, along with Proposition 1, would imply that random sub-codes of Reed-Solomon codes with random evaluation points achieve list decoding capacity.

4. All of our results so far only use either just random row operation or just random column operations. An open question is to find applications where random row and column operations could be use together to obtain better results than either on their own. The above point would be such an example, if resolved.

Acknowledgments

We thank Swastik Kopparty and Shubhangi Saraf for initial discussions on Questions 1 and 2 (and for indeed suggesting the random XOR as an operation to consider) and Dagstuhl for providing the venue for these initial discussions. We thank Venkat Guruswami for pointing out the argument in Appendix B. Finally, we thank Parikshit Gopalan for pointing the connection of our results to existing results on XOR and direct product codes. MW also thanks the theory group at IBM Almaden for their hospitality during part of this work.

References

[Bli86] Volodia M. Blinovsky. Bounds for codes in the case of list decoding of finite volume. *Problems of Information Transmission*, 22(1):7–19, 1986.

[Bli05] V. M. Blinovsky. Code bounds for multiple packings over a nonbinary finite alphabet. *Probl. Inf. Transm.*, 41(1):23–32, 2005.

[Bli08] V. M. Blinovsky. On the convexity of one coding-theory function. *Probl. Inf. Transm.*, 44(1):34–39, 2008.

[CGV13] Mahdi Cheraghchi, Venkatesan Guruswami, and Ameya Velingker. Restricted isometry of fourier matrices and list decodability of random linear codes. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 432–442, 2013.

[DL12] Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC)*, pages 351–358, 2012.

[7] If this were to be the case then this could formalize the reason why the Parvaresh-Vardy codes [PV05], which are sub-codes of interleaving of Reed-Solomon codes, have good list decodability properties.

[Eli57] Peter Elias. List decoding for noisy channels. *Technical Report 335, Research Laboratory of Electronics, MIT*, 1957.

[GGR11] Parikshit Gopalan, Venkatesan Guruswami, and Prasad Raghavendra. List decoding tensor products and interleaved codes. *SIAM J. Comput.*, 40(5):1432–1462, 2011.

[GI01] Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *Proceedings of the 42nd Annual IEEE Symposium on the Foundations of Computer Science (FOCS)*, pages 658–667. IEEE, 2001.

[GI03] Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC)*, pages 126–135, 2003.

[GI05] Venkatesan Guruswami and Piotr Indyk. Linear-time encodable/decodable codes with near-optimal rate. *IEEE Transactions on Information Theory*, 51(10):3393–3400, 2005.

[GK13] Venkatesan Guruswami and Swastik Kopparty. Explicit subspace designs. In *FOCS*, 2013. To appear.

[GN13] Venkatesan Guruswami and Srivatsan Narayanan. Combinatorial limitations of average-radius list decoding. *RANDOM*, 2013.

[GR08] Venkatesan Guruswami and Atri Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, 2008.

[GR09] Venkatesan Guruswami and Atri Rudra. Error correction up to the information-theoretic limit. *Commun. ACM*, 52(3):87–95, 2009.

[GR10] Venkatesan Guruswami and Atri Rudra. The existence of concatenated codes list-decodable up to the hamming bound. *IEEE Transactions on Information Theory*, 56(10):5195–5206, 2010.

[GRS14] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory, 2014. Draft available at http://www.cse.buffalo.edu/ atri/courses/coding-theory/book/index.html.

[Gur04] Venkatesan Guruswami. *List Decoding of Error-Correcting Codes (Winning Thesis of the 2002 ACM Doctoral Dissertation Competition)*, volume 3282 of *Lecture Notes in Computer Science*. Springer, 2004.

[Gur11] Venkatesan Guruswami. Linear-algebraic list decoding of folded reed-solomon codes. In *IEEE Conference on Computational Complexity*, pages 77–85, 2011.

[GV10] Venkatesan Guruswami and Salil Vadhan. A lower bound on list size for list decoding. *Information Theory, IEEE Transactions on*, 56(11):5681–5688, 2010.

[GW13] Venkatesan Guruswami and Carol Wang. Linear-algebraic list decoding for variants of reed-solomon codes. *IEEE Transactions on Information Theory*, 59(6):3257–3268, 2013.

[GX12] Venkatesan Guruswami and Chaoping Xing. Folded codes from function field towers and improved optimal rate list decoding. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC)*, pages 339–350, 2012.

[GX13] Venkatesan Guruswami and Chaoping Xing. List decoding reed-solomon, algebraic-geometric, and gabidulin subcodes up to the singleton bound. In *Proceedings of the 45th ACM Symposium on the Theory of Computing (STOC)*, pages 843–852, 2013.

[GX14] Venkatesan Guruswami and Chaoping Xing. Optimal rate list decoding of folded algebraic-geometric codes over constant-sized alphabets. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1858–1866, 2014.

[IJKW10] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: Simplified, optimized, and derandomized. *SIAM J. Comput.*, 39(4):1637–1665, 2010.

[Kop12] Swastik Kopparty. List-decoding multiplicity codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:44, 2012.

[PV05] Farzad Parvaresh and Alexander Vardy. Correcting errors beyond the guruswami-sudan radius in polynomial time. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 285–294, 2005.

[Rud11] Atri Rudra. Limits to list decoding of random codes. *IEEE Transactions on Information Theory*, 57(3):1398–1408, 2011.

[RW14] Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Proceedings of the 46th annual ACM Symposium on the Theory of Computing (STOC)*, 2014. To appear.

[Spi96] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Transactions on Information Theory*, 42(6):1723–1731, 1996.

[Sud00] Madhu Sudan. List decoding: algorithms and applications. *SIGACT News*, 31(1):16–27, 2000.

[Tre03] Luca Trevisan. List-decoding using the xor lemma. In *Proceedings of the 44th Symposium on Foundations of Computer Science (FOCS)*, pages 126–135, 2003.

[Woo13] Mary Wootters. On the list decodability of random linear codes with large error rates. In *Proceedings of the 45th annual ACM Symposium on the Theory of Computing (STOC)*, pages 853–860. ACM, 2013.

[Woz58] John M. Wozencraft. List Decoding. *Quarterly Progress Report, Research Laboratory of Electronics, MIT*, 48:90–95, 1958.

[ZP82] Victor V. Zyablov and Mark S. Pinsker. List cascade decoding. *Problems of Information Transmission*, 17(4):29–34, 1981 (in Russian); pp. 236-240 (in English), 1982.

A Average case, average radius Johnson bounds

The *Johnson bound* states that any code with good enough distance is list-decodable with polynomial list sizes, up to a radius that depends on the distance. For this work, we will need some slight variants on the Johnson bound. We will be interested in average-radius list decoding, rather than the standard definition. We state three versions of an average-radius Johnson bound below, for different list sizes.

Theorem 8 (Average-radius Johnson bounds). *Let $\mathcal{C} : \mathbb{F}_q^k \to \mathbb{F}_q^n$ be any code. Then for all $\Lambda \subset \mathbb{F}_q^k$ of size L and for all $z \in \mathbb{F}_q^n$:*

- *If $q = 2$,*

$$\sum_{x \in \Lambda} \mathrm{agr}(\mathcal{C}(x), z) \leq \frac{n}{2} \left(L + \sqrt{L^2 - 2 \sum_{x \neq y \in \Lambda} d(\mathcal{C}(x), \mathcal{C}(y))} \right).$$

- *For all* $\varepsilon \in (0,1)$*,*

$$\sum_{x\in\Lambda} \mathrm{agr}(\mathcal{C}(x), z) \le \frac{nL}{q} + \frac{nL}{2\varepsilon}\left(1+\varepsilon^2\right)\left(1-\frac{1}{q}\right) - \frac{n}{2L\varepsilon}\sum_{x\neq y\in\Lambda} d(\mathcal{C}(x), \mathcal{C}(y)).$$

-

$$\sum_{x\in\Lambda} \mathrm{agr}(\mathcal{C}(x), z) \le \frac{1}{2}\left(n + \sqrt{n^2 + 4n^2L(L-1) - 4n^2\sum_{x\neq y\in\Lambda} d(\mathcal{C}(x), \mathcal{C}(y))}\right).$$

Proof. The proof of the second two statements (for general q) can be found in [RW14]. The statement for $q = 2$ follows by the computation below (implicit in [Woo13, CGV13]). Let $\Phi \in (\pm 1)^{n\times 2^k}$ be the matrix whose columns are indexed by $x \in \mathbb{F}_2^k$, so that $\Phi_{j,x} = (-1)^{\mathcal{C}(x)_j}$. Let φ_j denote the j-th column of Φ. Then

$$\begin{aligned}
\max_z \sum_{x\in\Lambda} \mathrm{agr}(\mathcal{C}(x), z) &= \sum_{j=1}^n \max_{b\in\{0,1\}} \sum_{x\in\Lambda} \mathbf{1}_{\mathcal{C}(x)_j=\alpha} \\
&= \sum_{j=1}^n \max_{\alpha\in\{0,1\}} \sum_{x\in\Lambda} \frac{(-1)^\alpha(-1)^{\mathcal{C}(x)_j}+1}{2} \\
&= \sum_{j=1}^n \left(nL + \sum_{j=1}^n |\langle\varphi_j, \mathbf{1}_\Lambda\rangle|\right) \\
&= \frac{1}{2}\left(nL + \|\Phi\mathbf{1}_\Lambda\|_1\right) \\
&\le \frac{1}{2}\left(nL + \sqrt{n}\,\|\Phi\mathbf{1}_\Lambda\|_2\right),
\end{aligned}$$

using Cauchy-Schwarz in the final line. The claim then follows from the definition of Φ and the fact that the (x,y)-entry of $\Phi^T\Phi$ is given by $n(1-2d(\mathcal{C}(x),\mathcal{C}(y)))$. Indeed, from this, we have

$$\|\Phi\mathbf{1}_\Lambda\|_2^2 = \mathbf{1}_\Lambda^T\Phi^T\Phi\mathbf{1}_\Lambda = n\sum_{x\in\Lambda}\sum_{y\in\Lambda}\left(1-2d(\mathcal{C}(x),\mathcal{C}(y))\right),$$

and plugging this in above gives the statement. □

B Linear time encodable and decodable binary list decodable codes

We will argue the following in this section:

Theorem 9. *For every* $\varepsilon > 0$*, there exists a binary code that can be encoded and list decoded in linear time from* $1/2-\varepsilon$ *fraction of errors with rate* $2^{-2^{O(\varepsilon^{-9})}}$.

In the rest of the section, we argue why the statement above is true. (We thank Venkat Guruswami for pointing out the following argument to us.)

We will crucially use the following result that follows from the work of Guruswami and Indyk:

Theorem 10 ([GI03])**.** *For every* $\gamma > 0$*, there exists a* q*-ary code that can be encoded and decoded in linear time from* $1-\gamma$ *fraction of errors for* $q = 1/\gamma$ *and with rate* $2^{-2^{O(-\gamma^3)}}$.

Ultimately we will use the above theorem with $\gamma = \varepsilon^3/8$ to get out outer code. Our inner code will be the binary Hadamard code with $q = 8/\varepsilon^3$ codewords in it. Since the binary Hadamard code has relative distance $1/2$, Johnson bound implies that it is $(1/2-\varepsilon/2, 8/\varepsilon^2)$-list decodable. Our final code will be the code concatenation of the outer and inner code.

Note that the rate of the concatenated code is at least $1/q \cdot 2^{-2^{O(\varepsilon^{-9})}}$, which is within the claimed bound on the rate. The claim on the encoding runtime follows from the fact that the outer code can be encoded in linear time and the inner code has constant size.

Finally, we look at the list decoding algorithm. The algorithm is simple:

1. Let $y = (y_1, \ldots, y_N)$ be the received word where each y_i is a valid received word for the inner code.

2. For each $i \in [N]$, compute the list of every message whose corresponding Hadamard codeword is within a relative Hamming distance of $1/2 - \varepsilon/2$ from y_i. Set y'_i be a random element from this list of messages.

3. Run the list decoding algorithm for the outer code on the intermediate received word $(y'_1, \ldots, y'_N)$.

It is easy to check that the above algorithm runs in linear time since the list decoder for the outer code runs in linear time and inner code has constant size.

Finally, we argue why the above algorithm works. Consider any codeword that is within $1/2 - \varepsilon$ fraction of the received word. Then by an averaging argument, one can show that for at least ε fraction of the positions $i \in [N]$, the corresponding value in the outer codeword belong to the list calculated in Step 2 above. Since the list has size $8/\varepsilon^2$, then in expectation the codeword agrees with the intermediate received word from Step 3 in $\varepsilon^3/8$ fraction of positions. This implies that the list decoder from Theorem 10 can recover the algorithm.[8]

C With replacement vs. without replacement

In this appendix, we show how to apply Theorem 2 to operations like puncturing and folding, where the symbols do not quite have full independence. Our first lemma justifies the extension of Theorem 2 to symbols which are sampled without replacement.

Lemma 4. *Suppose that $f \sim \mathcal{D}$ has symbols drawn independently without replacement from $\mathcal{S}_{\mathcal{D}}$, as in Definition 2. Let $\mathcal{D}'$ be the corresponding distribution with replacement: that is, each f_j is drawn i.i.d. uniformly at random from $\mathcal{S}_{\mathcal{D}}$. Then*

$$\mathbb{E}_{f \sim \mathcal{D}} \max_{z \in \Sigma^n} \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \sum_{c \in \Lambda} \mathrm{agr}(f(c), z) \leq \mathbb{E}_{f \sim \mathcal{D}'} \max_{z \in \Sigma^n} \max_{\Lambda \subset \mathcal{C}_0, |\Lambda| = L} \sum_{c \in \Lambda} \mathrm{agr}(f(c), z)$$

For example, suppose $f = (f_1, \ldots, f_n) \sim \mathcal{D}$ is random puncturing, so $f_j(c) = c_{i_j}$ for a random subset $\{i_1, \ldots, i_n\} \subset [N]$ chosen uniformly without replacement. Then $\mathcal{D}'$ would be the *random sampling* operation of [RW14]. That is, $f_j(c) = c_{i_j}$ chosen i.i.d. from $[N]$. Thus, Lemma 4 implies that the results of [RW14] for random sampling imply to random puncturing as well.

To prove Lemma 4, we will need to unpack the results of [RW14] a bit. We introduce the following definition.

Definition 3. *For a set $\Lambda \subset \mathcal{C}_0$, and an index $j \in [n]$, we define the* plurality *of the j'th symbol of $\mathcal{C}_0$ in Λ to be*

$$\mathbf{pl}_j(\Lambda) = \max_{\alpha \in \Sigma} |\{c \in \Lambda \,:\, f(c)_j = \alpha\}|\,.$$

Thus, $\mathbf{pl}_j(\Lambda)$ is a random variable, over the choice of $f \sim \mathcal{D}$. Further, we have

$$\max_{z \in \Sigma^n} \sum_{c \in \Lambda} \mathrm{agr}(f(c), z) = \max_{c \in \Lambda} \sum_{j=1}^{n} \mathbf{pl}_j(\Lambda).$$

[8] To be fully correct, we need to adjust the constants so that in expectation one has agreement in $\varepsilon^3/4$ fraction of location since then with high probability one would indeed have agreement of at least $\varepsilon^3/8$ for *all* codewords that need to be output. The latter is fine since it is known that the code from Theorem 10 is actually $(1-\gamma, O(\gamma^{-3}))$-list decodable– so a union bound would suffice.

Thus, when $f \sim \mathcal{D}'$ has independent symbols, the random variables $\mathbf{pl}_j(\Lambda)$ are independent for different j. When $f \sim \mathcal{D}$ is independent with replacement, then we have a sum of independent random variables with replacement. Thus, the following simple lemma will imply Lemma 4.

Lemma 5. *Suppose that $X_1, \ldots, X_n$ are drawn without replacement from a finite set $\mathcal{S} \subset \mathbb{R}^d$ of size N. Suppose that $Y_1, \ldots, Y_n$ are drawn independently and uniformly at random from $\mathcal{S}$. Then*

$$\mathbb{E}_X \left\| \sum_{i=1}^n X_i \right\|_\infty \leq \mathbb{E}_Y \left\| \sum_{i=1}^n Y_i \right\|_\infty .$$

Proof. Consider the following distribution. Draw $z_1, \ldots, z_N$ from a multinomial distribution with n trials and event probabilities $p_i = 1/N$ for $i = 1, \ldots, N$. Let $\hat{z}_i'$ denote the z_i sorted in decreasing order: notice that $\hat{z}_i = 0$ for all $i > n$. Draw a random permutation $\pi \sim S_n$ and definte $\hat{z}_i = \hat{z}'_{\pi(i)}$. Now we have $\sum_i \hat{z}_i = n$, and by symmetry, $\mathbb{E}\hat{z}_i = 1$. Now draw $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_n$ from $\mathcal{S}$, as in the lemma statement. Observe that the distribution of

$$\sum_{i=1}^n \hat{z}_i X_i$$

is the same as the distribution of

$$\sum_{i=1}^n Y_i.$$

In particular, we have

$$\mathbb{E}_{X,\hat{z}} \left\| \sum_{i=1}^n \hat{z}_i X_i \right\|_\infty = \mathbb{E}_Y \left\| \sum_{i=1}^n Y_i \right\|_\infty . \tag{8}$$

On the other hand, we have

$$\mathbb{E}_{X,\hat{z}} \left\| \sum_{i=1}^n \hat{z}_i X_i \right\|_\infty \geq \mathbb{E}_X \left\| \mathbb{E}_{\hat{z}} \sum_{i=1}^n \hat{z}_i X_i \right\|_\infty = \mathbb{E}_X \left\| \sum_{i=1}^n X_i \right\|_\infty , \tag{9}$$

using the fact that $\mathbb{E}_{\hat{z}} \hat{z}_i = 1$ for all $i = 1, \ldots, n$. Together, (8) and (9) imply that

$$\mathbb{E}_X \left\| \sum_{i=1}^n X_i \right\|_\infty \leq \mathbb{E}_Y \left\| \sum_{i=1}^n Y_i \right\|_\infty ,$$

as desired. □

Now Lemma 5 implies Lemma 4. Indeed, in Lemma 5, we may take the vectors $Y_i \in \mathbb{R}^d$ for $d = \binom{N}{L}$ to be given by

$$(Y_j)_\Lambda = \mathbf{pl}_j(\Lambda).$$

D Missing Proofs from Section 4

D.1 Controlling the parameter $\mathcal{E}$

In this section, we show how to control the parameter $\mathcal{E}$ for random t-wise XOR and for random t-wise aggregation, using the average-radius Johnson bound, Theorem 8.

Proof of Lemma 1. We will use the average-radius Johnson bound, Theorem 8. Thus, we start by computing the expected distance between two symbols of the code $\mathcal{C} \in \mathbb{F}_2^n$ obtained from $\mathcal{C}_0$ and $\mathcal{D}$. Let c, c' denote two distinct codewords in $\mathcal{C}_0$. Recall that $\mathcal{U}_{\oplus,t}$ is the uniform distribution over

$$\left\{ r_v^{(\mathrm{ip})} : v \in \mathbb{F}_2^N \text{ has weight } t \right\},$$

and write $f = (r_1, \ldots, r_n)$. Let $v_i \in \mathbb{F}_2^N$ denote the vector picked by the row operation r_i; thus, $v_i \in \mathbb{F}_2^N$ are chosen i.i.d. uniformly at random (with replacement). Then

$$\begin{aligned}
\mathbb{E}\delta(f(c), f(c')) &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\left\{f_i(c) \neq f_i(c')\right\} \\
&= \mathbb{P}\left\{\langle v_i, c\rangle \neq \langle v_i, c'\rangle\right\} \\
&= \frac{1}{2}\mathbb{P}\left\{(c-c')_{\mathrm{Supp}v_i} \neq 0\right\} \\
&= \frac{1}{2}\left(1 - (1-\delta_0)^t\right) \\
&\leq \frac{1}{2}\left(1 - e^{-\delta_0 t/2}\right).
\end{aligned}$$

In particular, if $t = \frac{4\ln(1/\varepsilon)}{\delta_0}$, then this is $\frac{1}{2}(1-\varepsilon^2)$. Then Theorem 8 implies that

$$\begin{aligned}
\mathcal{E}(\mathcal{C}_0, \mathcal{D}_{ip}(t)) &= \max_{\Lambda \subset \mathcal{C}_0} \mathbb{E}_{f\sim\mathcal{D}_{ip}(t)} \max_{z\in\mathbb{F}_2^n} \sum_{c\in\Lambda} \mathrm{agr}(f(c), z) \\
&\leq \max_{\Lambda} \mathbb{E}_f \max_{z\in\mathbb{F}_2^n} \frac{n}{2}\left(L + \sqrt{L^2 - 2\sum_{c\neq c'\in\Lambda}\delta(f(c), f(c'))}\right) \\
&\leq \max_{\Lambda} \frac{n}{2}\left(L + \sqrt{L^2 - 2\sum_{c\neq c'\in\Lambda}\mathbb{E}_f\delta(f(c), f(c'))}\right) \\
&\leq \frac{n}{2}\left(L + \sqrt{L^2 - 2\sum_{c\neq c'\in\Lambda}\frac{1}{2}(1-\varepsilon^2)}\right) \\
&= \frac{n}{2}\left(L + \sqrt{L^2\varepsilon^2 + L(1-\varepsilon^2)}\right) \\
&\leq \frac{n}{2}\left(L(1+\varepsilon) + \sqrt{L}\right).
\end{aligned}$$

□

Proof of Lemma 2. We wish to control $\mathcal{E}(\mathcal{C}_0, \mathcal{D})$, which we do via the average-radius Johnson bound (Theorem 8). Because we are interested in the parameter regime where $q \geq 1/\varepsilon^2$, we use the third statement in Theorem 8. Suppose $t \geq 4\ln(1/\varepsilon)/\delta_0$ and set $L = 1/\varepsilon$. For $c \neq c' \in \mathcal{C}_0$, we compute

$$\begin{aligned}
\mathbb{E}_{f\sim\mathcal{D}}\delta(f(c), f(c')) &= \frac{1}{n}\sum_{i=1}^{n} \mathbb{P}\left\{f_j(c) \neq f_j(c')\right\} \\
&= \mathbb{P}\left\{\exists j \in S_i \,:\, c_j \neq c'_j\right\} \\
&= 1 - (1-\delta_0)^t \\
&\leq 1 - \varepsilon^2,
\end{aligned}$$

using the choice of t in the final line. Thus, by Theorem 8, Item 3,

$$\begin{aligned}
\mathcal{E}(\mathcal{C}_0, \mathcal{D}) &= \max_{\Lambda \subset \mathcal{C}_0} \mathbb{E}_{f \sim \mathcal{D}_{dp}(t)} \max_{z \in \mathbb{F}_q^n} \sum_{c \in \Lambda} \mathrm{agr}(f(c), z) \\
&\leq \max_{\Lambda \subset \mathcal{C}_0} \mathbb{E}_{f \sim \mathcal{D}_{dp}(t)} \max_{z \in \mathbb{F}_q^n} \frac{1}{2} \left(n + \sqrt{n^2 + 4n^2 L(L-1) - 4n^2 \sum_{c \neq c' \in \Lambda} \delta(f(c), f(c'))} \right) \\
&= \max_{\Lambda \subset \mathcal{C}_0} \frac{1}{2} \left(n + \sqrt{n^2 + 4n^2 L(L-1) - 4n^2 \sum_{c \neq c' \in \Lambda} \mathbb{E}_f \delta(f(c), f(c'))} \right) \\
&\leq \frac{1}{2} \left(n + \sqrt{n^2 + 4n^2 L(L-1) - 4n^2 \sum_{c \neq c' \in \Lambda} (1 - \varepsilon^2)} \right) \\
&= \frac{n}{2} \left(1 + \sqrt{1 + 4L(L-1)\varepsilon^2} \right) \\
&\leq Cn,
\end{aligned}$$

using the choice of L and defining $C = (1 + \sqrt{5})/2$. □

D.2 Proof of Claim 5

Proof. The only way that $|\mathcal{C}| < N$ is if two codewords $c \neq c' \in \mathcal{C}_0$ collide, that is, if $f(c) = f(c')$. This is unlikely: we have

$$\mathbb{P}\{f(c) = f(c')\} = (1 - \delta_0)^{nt} \leq \varepsilon^{2nt}.$$

By a union bound over $\binom{N}{2} \leq N^2$ pairs $c \neq c'$, we conclude that the probability that $|\mathcal{C}| < N$ is at most

$$\mathbb{P}\{|\mathcal{C}| < N\} \leq N^2 \varepsilon^{2nt}. \tag{10}$$

If $nt = n_0$, we have

$$\mathbb{P}\{|\mathcal{C}| < N\} \leq q^{2n_0 R_0} \varepsilon^{2nt} = \left(q^{R_0} \varepsilon\right)^{2n_0}.$$

In particular, when $q^{R_0} < 1/\varepsilon$, this is $o(1)$. By our assumption, $R_0 < \varepsilon$, and so this is always true for sufficiently small ε. □

D.3 Proof Claim 6

Proof. As in (10), we have

$$\mathbb{P}\{|\mathcal{C}| < N\} \leq N^2 \varepsilon^{2nt}.$$

We may bound the right-hand-side by

$$N^2 \varepsilon^{2nt} = \left(q^{R_0 n_0 / n} \varepsilon^t\right)^{2n},$$

and for this to be $o(1)$, it is sufficient for

$$R_0 \leq \left(\frac{nt}{n_0}\right) \left(\frac{\log(1/\varepsilon)}{\log(q)}\right),$$

which was our assumption for part 2 of the theorem. □

E Missing details on random sub-codes

E.1 Preliminaries

We collect some known results that we will use. We begin with a form of Chernoff bound that will be useful for our purposes:

Theorem 11. *Let $X_1, \ldots, X_m$ are random independent binary random variables with bias p. Then*

$$\mathbb{P}\left\{\sum_i X_i > t\right\} \leq \left(\frac{pm}{t}\right)^{t-pm}.$$

Next, we state a conjecture concerning the tradeoff between list decodability and list size:

Conjecture 12. *Any (ρ, L)-list decodable q-ary code has rate at most $1 - H_q(\rho) - \Omega\left(\frac{1}{L}\right)$.*

There many reasons to believe that the conjecture above is true. Conjecture 12 is known to be true when ρ approaches 1 [GV10, Bli05, Bli08, Bli86]. Weaker versions of the conjecture are known to be true.

Theorem 13 ([GN13, Bli05, Bli08, Bli86])**.** *For constant ρ, any q-ary code that is (ρ, L)- list decodable must have rate at most $1 - H_q(\rho) - \Omega\left(\frac{1}{2^L}\right)$.*

Theorem 14 ([GN13])**.** *For constant ρ, any binary code that is (ρ, L)-average-radius list decodable must have rate at most $1 - H_2(\rho) - \Omega\left(\frac{1}{L^2}\right)$.*

Finally, the rate bound in Conjecture 12 is achieved by random codes and the bound in Conjecture 12 is known to be true for most codes [Rud11].

E.2 Proof of Proposition 1

We now give the proof of Proposition 1.

Proof of Proposition 1. Let Σ be the alphabet of size q. Consider any fixed $y \in \Sigma^n$, where n is the block length of $\mathcal{C}_0$ (and is assumed to be large enough). Then the list decodability of $\mathcal{C}_0$ implies that

$$|B_q(y, \rho n) \cap \mathcal{C}_0| \leq L_0, \tag{11}$$

where $B_q(y, r)$ is the q-ary Hamming ball of radius r centered at y. As in Section 2, write $\mathcal{C} = f(\mathcal{C}_0)$, where $f = (c_1, \ldots, c_N) \sim (\mathcal{U}_c)^N$. Now consider the random variable $|B_q(y, \rho n) \cap \mathcal{C}|$, where we are abusing our chosen notation slightly and treating $\mathcal{C}$ as a proper set, even though as a matrix, $\mathcal{C}$ may have repeated columns. This is bounded by the sum of N independent Bernoulli-$\left(\frac{|B_q(y,\rho n) \cap \mathcal{C}_0|}{N_0}\right)$ variables:

$$|B_q(y, \rho n) \cap \mathcal{C}| \leq \sum_{i=1}^{n} \mathbf{1}_{c_i(\mathcal{C}_0) \in B_q(y, \rho n)},$$

where again we have inequality rather than equality because of the possibility that $c_i(\mathcal{C}_0) = c_j(\mathcal{C}_0)$ for some $i \neq j$. We have

$$\mathbb{E}\left[\sum_{i=1}^{n} \mathbf{1}_{c_i(\mathcal{C}_0) \in B_q(y,\rho n)}\right] = N \cdot \frac{|B_q(y, \rho n) \cap \mathcal{C}_0|}{N_0} \leq q^{-\varepsilon n}, \tag{12}$$

where the last inequality follows from (11). Thus, by a Chernoff bound (Theorem 11) along with (12),

$$\mathbb{P}\left\{|B_q(y, \rho n) \cap \mathcal{C}| > \frac{3}{\varepsilon}\right\} \leq \mathbb{P}\left\{\sum_{i=1}^{n} \mathbf{1}_{c_i(\mathcal{C}_0) \in B_q(y,\rho n)} > \frac{3}{\varepsilon}\right\} \leq \left(\frac{\varepsilon}{3 \cdot q^{\varepsilon n}}\right)^{3/\varepsilon - q^{-\varepsilon n}} \leq \left(\frac{1}{q^{\varepsilon n}}\right)^{\frac{2}{\varepsilon}} = q^{-2n},$$

where the last inequality follows for large enough n. Taking the union bound over the q^n choices of y, we conclude that $\mathcal{C}$ is not $\left(\rho, \frac{3}{\varepsilon}\right)$-list decodable with probability at most q^{-n}, which completes the proof.

The claim on the size of $\mathcal{C}$ follows from the following simple argument. Note that $|\mathcal{C}| < pN_0/2$ implies that there exists a subset $S \subset [N_0]$ of size exactly $pN_0/2$ such that all codewords in $\mathcal{C}$ are contained in the columns of $\mathcal{C}_0$ indexed by S. Note that the probability of this happening for a fixed S is given by $(p/2)^{pN_0}$. Taking union bound over all choices of S, implies that the probability that $|\mathcal{C}| < pN_0/2$ is upper bounded by

$$\binom{N_0}{pN_0/2} \cdot \left(\frac{p}{2}\right)^{pN_0} \leq \left(\frac{2e}{p}\right)^{pN_0/2} \cdot \left(\frac{p}{2}\right)^{pN_0} = \left(\frac{ep}{2}\right)^{pN_0/2},$$

which is $o(1)$ by our choice of parameters. This completes the proof. □

E.3 Upper Bound

Proposition 1 only works for the usual notion of list decodability. It is natural to wonder if a similar result holds for average-radius list decodability. Next, we show that such a result indeed holds (though with slightly weaker parameters). Indeed the result follows from the following simple observation:

Proposition 2. *Let $\mathcal{C}$ be a (ρ, L)-list decodable code. Then for any $\gamma > 0$, $\mathcal{C}$ is also $\left(\rho - \gamma, \frac{L}{\gamma}\right)$-average-radius list decodable.*

Proof. Define $L' = L/\gamma$ and fix an arbitrary $\Lambda \subset \mathcal{C}$ such that $|\Lambda| = L'$. Define

$$\Lambda_- = \Lambda \cap B_q(y, \rho n) \text{ and } \Lambda_+ = \Lambda \setminus \Lambda_-.$$

Note that since $\mathcal{C}$ is (ρ, L)-list decodable, we have $|\Lambda_-| \leq L$. This implies that

$$\sum_{c \in \Lambda_-} \operatorname{agr}(c, y) \leq |\Lambda_-| \cdot n \leq nL \leq \gamma n L', \tag{13}$$

where the last inequality follows from the definition of L'. Further, by the definition of Λ_+, we have

$$\sum_{c \in \Lambda_+} \operatorname{agr}(c, y) < (1 - \rho)n \cdot |\Lambda_+| \leq (1 - \rho)nL'.$$

Combining the above with (13) implies that $\sum_{c \in \Lambda} \operatorname{agr}(c, y) < (1 - \rho + \gamma)nL'$, which completes the proof. □

Since a (ρ, L_0)-average-radius list decodable code is also (ρ, L)-list decodable, Propositions 1 and 2 implies the following:

Corollary 4. *Let $\mathcal{C}_0$ be an (ρ, L_0)-average-radius list decodable q-ary code. If we retain each codeword with probability $\frac{1}{q^{\varepsilon n} \cdot L_0}$, then the resulting code with high probability is $(\rho - \varepsilon, O(1/\varepsilon^2))$-average-radius list decodable.*

E.4 Lower Bound

It is natural to wonder if one can pick a larger value of p in Proposition 1, and whether the dependence $q^{\varepsilon n}$ is necessary. In particular, if L_0 is only polynomial in n, could we pick $p = q^{-o(\varepsilon n)}$? We will now argue that this is not possible.

First, we give a short argument, conditional on Conjecture 12. By standard random coding argument, there exists a q-ary code $\mathcal{C}_1$ with rate $1 - H_q(\rho) - \frac{1}{n}$ that is (ρ, n)-list decodable. Suppose Proposition 1 holds with $p = q^{-o(\varepsilon n)}$. If we applied this to the code $\mathcal{C}_1$, we would obtain a code that is $(\rho, O(1/\varepsilon))$-list decodable that has rate at least

$$1 - H_q(\rho) - o(\varepsilon),$$

assuming that ε is constant and n is growing. However, this contradicts Conjecture 12 for $L = O(1/\varepsilon)$.

Next, we argue an unconditional upper bound on p in Proposition 1. In fact, we will prove something stronger: we will show that one needs $p = 2^{-\Omega(\varepsilon n)}$ even if the the original code $\mathcal{C}_0$ has the stronger property of being (ρ, L_0)-average-radius list decodable (and the random subcode can have a weaker list decoding radius).

Theorem 15 (Theorem 7, repeated)**.** *For every $\rho > 0$, and for every $0 < \alpha < \frac{1-\rho}{12}$, and for every n sufficiently large, there exists a code $\mathcal{C}_0$ with block length n that is (ρ, n)-average-radius list decodable such that the following holds. Let $\mathcal{C}$ be obtained by picking a random sub-code of $\mathcal{C}_0$ of size $N = pN_0$ where $p = q^{-\alpha n}/n$. Then with high probability if $\mathcal{C}$ is (ρ', L)-list decodable for any $\rho' \geq 1/n$, then $L \geq \Omega(1/\alpha)$.*

In the rest of the subsection, we will prove Theorem 7.

E.4.1 Preliminaries

We will need the following technical result, which follows the standard random coding argument and its analysis to determine the list decodability of random codes.

Lemma 6. *Let $q \geq 2^{1/r}$ be an integer. Then there exists a code $\mathcal{C}^*$ with rate r and block length n such that for every $2r < \gamma \leq 1$, where γ is a power of $1/2$, $\mathcal{C}^*$ is $\left(1-\gamma, \left\lceil \frac{1}{\gamma - 2r} \right\rceil\right)$-list decodable. Further, $\mathcal{C}^*$ has relative distance $1 - O(r)$.*

Proof. Fix a γ with conditions as in the lemma statement. Let $\mathcal{C}^*$ be a random code of rate r; by standard arguments, this distance of this code is $1 - O(r)$ with high probability [GRS14]. Further, the standard random coding argument (see, for example, [GRS14]) implies that $\mathcal{C}^*$ is $(1-\gamma, L)$ list decodable except with probability at most

$$q^n \cdot q^{rn(L+1)} \cdot \left(\frac{q^{H_q(1-\gamma)n}}{q^n} \right)^{L+1}.$$

Rearranging, we can bound the expression above by

$$\begin{aligned} & q^{-n(L+1)\left(1-H_q(1-\gamma)-r-\frac{1}{L+1}\right)} \\ & \quad \leq q^{-n(L+1)\left(1-(1-\gamma+r)-r-\frac{1}{L+1}\right)} && (14) \\ & \quad = q^{-n(L+1)\left(\gamma-2r-\frac{1}{L+1}\right)} \\ & \quad \leq q^{-\Omega(n/L)} && (15) \end{aligned}$$

In the above, (14) follows from the following sequence of relations (that holds for any $0 \leq \rho \leq 1 - 1/q$):

$$H_q(\rho) = \rho \log_q(q-1) + \frac{H_2(\rho)}{\log q} \leq \rho + r,$$

where the inequality uses the fact that $q \geq 2^{1/r}$. (15) uses the fact that the choice of $L = \left\lceil \frac{1}{\gamma - 2r} \right\rceil$ implies that $\gamma - 2r - \frac{1}{L+1} > 0$.

Finally, since the bound in (15) holds for any fixed γ and that there are $O(\log(1/r))$ possible values of γ, the probability that the randomly chosen $\mathcal{C}^*$ does not have the required property is $o(1)$, which completes the proof. □

E.4.2 The Construction

We now present the code $\mathcal{C}_0$. Choose $\beta > 0$ to be the smallest number such that $1 - \rho - \beta$ is a power of $1/2$. We will construct $\mathcal{C}_0$ from $\mathcal{C}^*$ as given by Lemma 6 with rate $r = (1 - \rho - \beta)/6$. (Note that by our choice of β, this implies that $r \geq (1-\rho)/12$ and hence, $\alpha < r$.) The construction goes as follows. For every $c \in \mathcal{C}^*$, let $N(c)$ be any $\frac{\beta \cdot n}{8 \log(1/(1-\rho-\beta))} - 1$ distinct vectors with Hamming distance 1 from c. Then define

$$\mathcal{C}_0 = \cup_{c \in \mathcal{C}^*} N(c).$$

Having constructed $\mathcal{C}_0$, we argue next that it has good average-radius list-decodability.

Lemma 7. *$\mathcal{C}_0$ is (ρ, n)-average-radius list decodable.*

Proof. Recall that $1-\rho-\beta$ is a power of $1/2$. Fix an arbitrary z and $\Lambda \subset \mathcal{C}_0$ with $|\Lambda| = n$. We want to show that

$$\sum_{c\in\Lambda} \operatorname{agr}(z,c) < (1-\rho)n^2. \tag{16}$$

Define

$$\mathcal{B} = B_q(z, (\rho+\beta)n).$$

We will break up the left-hand-side of (16) into two parts, and handle $\Lambda \setminus \mathcal{B}$ and $\Lambda \cap \mathcal{B}$ separately. First, we have

$$\sum_{c\in\Lambda\setminus\mathcal{B}} \operatorname{agr}(z,c) < (1-\rho-\beta)n \cdot |\Lambda| = (1-\rho-\beta)n^2. \tag{17}$$

Next, we bound $\sum_{c\in\Lambda\cap\mathcal{B}} \operatorname{agr}(z,c)$. We break this sum up even further, and decompose $\mathcal{B}$ into the annuli

$$\mathcal{A}_i := B_q(z, (1-2^{-i-1})n) \setminus B_q(z, (1-2^{-i})n)$$

for $0 \le i < \log\left(\frac{1}{1-\rho-\beta}\right)$. Fix an $0 \le i < \log\left(\frac{1}{1-\rho-\beta}\right)$ and for notational convenience define $\gamma = 2^{-i-1}$. (This will agree with the use of γ in the statement of Lemma 6). Now consider $\Lambda \cap \mathcal{A}_i$, and consider the set

$$S := \{c \in \mathcal{C}^* \,:\, N(c) \cap (\Lambda \cap \mathcal{A}_i) \neq \emptyset\}$$

of "centers" in $\mathcal{C}^*$ whose "clusters" $N(c)$ appear in this set. We make the following two observations:

Claim 16. *$S \subset \mathcal{C}^* \cap B_q(z, (1-\gamma/2)n)$.*

Proof. Since all vectors in $N(c)$ are at Hamming distance 1 from a $c \in \mathcal{C}^*$ (and n is assumed to be large enough), we have that $c \in S$ implies that $c \in \mathcal{A}_{i-1} \cup \mathcal{A}_i \cup \mathcal{A}_{i+1}$. It is easy to see that the union of the three annuli is contained in $B_q(z, (1-\gamma/2)n)$, which completes the proof. □

The following follows from the construction:

Claim 17.

$$|\Lambda \cap \mathcal{A}_i| \le |S| \cdot \frac{\beta \cdot n}{8\log(1/(1-\rho-\beta))}.$$

Thus, using the list-decodability of $\mathcal{C}^*$ guaranteed by Lemma 6 and Claim 16, we have that $|S| \le \left\lceil \frac{1}{\gamma - 2r} \right\rceil$. (Note that we can apply Lemma 6 since by our choice of parameters we have $\gamma \ge 1-\rho-\beta$, which in turn implies that $\gamma/2 \ge (1-\rho-\beta)/2 = 3r > 2r$ as required.) Further, this with Claim 17 implies that

$$|\Lambda \cap \mathcal{A}_i| \le \left\lceil \frac{1}{\gamma-2r} \right\rceil \cdot \frac{\beta\cdot n}{8\log(1/(1-\rho-\beta))} \le \left(\frac{2}{\gamma-2r}\right) \cdot \frac{\beta \cdot n}{8\log(1-\rho-\beta)}. \tag{18}$$

Now, we may bound

$$\begin{aligned}
\sum_{c\in\Lambda\cap\mathcal{A}_i} \operatorname{agr}(z,c) &\le |\Lambda \cap \mathcal{A}_i| \cdot 2\gamma n && (19)\\
&\le \left(\frac{1}{\gamma-2r}\right) \cdot \frac{\beta\cdot n}{4\log(1/(1-\rho-\beta))} \cdot 2\gamma n && (20)\\
&= \left(\frac{\gamma}{\gamma-2r}\right) \cdot \frac{\beta \cdot n^2}{2\log(1/(1-\rho-\beta))} \\
&\le \frac{\beta n^2}{\log(1/(1-\rho-\beta))}. && (21)
\end{aligned}$$

In the above, (19) follows from the fact that $\mathcal{A}_i$ lies outside of $B_q(z,(1-2\gamma)n)$. (20) follows from (18) while (21) follows from the fact that $2r \leq (1-\rho-\beta)/2 \leq \gamma/2$. Finally, summing everything up and using (17) and (21), we bound

$$\sum_{c\in\Lambda\cap\mathcal{B}} \mathrm{agr}(c,z) \leq (1-\rho-\beta)n^2 + \sum_{i=1}^{\log(1-\rho-\beta)} \frac{\beta n^2}{-\log(1-\rho-\beta)} = (1-\rho)n^2.$$

This establishes (16). □

Random subcodes of $\mathcal{C}_0$ are typically not list-decodable. Fix $\alpha > 0$, and let $\mathcal{C}$ be a random subcode of $\mathcal{C}_0$ of size pN_0, for $p = q^{-\alpha n}/n$ as in the statement of the theorem. We finally argue that $\mathcal{C}_0$ has many sub-codes that have terrible list decodability, thus proving Theorem 7. For any $z \in \mathcal{C}^*$ and let $\Lambda(z)$ be an arbitrary subset of $N(z)$ such that $|\Lambda(z)| = D/\alpha$, where we will fix D later. Further, order the "centers" in $\mathcal{C}^*$ as $z_1, z_2, \ldots$. Then the following is the main technical lemma:

Claim 18. *For any $k \leq \frac{q^{rn}}{3}$, we have*

$$\mathbb{P}\left\{\Lambda(z_{k+1}) \subset \mathcal{C} | \mathcal{C} \cap \left(\cup_{i=1}^k N(z_i)\right)\right\} \geq \left(\frac{p}{2e}\right)^{\frac{D}{\alpha}} \geq q^{-2Dn}.$$

Once we establish Claim 18, we are done. Indeed, we have

$$\begin{aligned}\mathbb{P}\left\{\forall i \leq \frac{q^{rn}}{3}, \Lambda(z_i) \not\subset \mathcal{C}\right\} &= \prod_{k=0}^{q^{rn}/3} \mathbb{P}\left\{\Lambda(z_{k+1}) \not\subset \mathcal{C} \mid \mathcal{C} \cap \left(\cup_{i=1}^k \Lambda(z_i)\right)\right\} \\ &\leq \left(1 - q^{-2Dn}\right)^{q^{rn}/3}.\end{aligned}$$

Thus, the probability that there is some i with $\Lambda(z_i) \subset \mathcal{C}$ is at least

$$1 - (1-q^{-2Dn})^{q^{rn}/3} \geq 1 - e^{-q^{(r-2D)n}/3} \geq 1 - o(1), \tag{22}$$

where the last inequality follows if we pick $D = r/3$. In particular, $\mathcal{C}$ has list sizes at least $|\Lambda(z_i)|$, even at distance $\rho = 1/n$, which is the radius of $\Lambda(z_i)$.

We conclude by proving Claim 18. For notational convenience, define $\mathcal{C}_0^{(k)} = \cup_{i=1}^k N(z_i)$; thus, $\mathcal{C}_0^{(k)}$ is the code $\mathcal{C}_0$ after the first k clusters $N(z_i)$ have been added. Let $M_k = |\mathcal{C} \cap \mathcal{C}_0^{(k)}|$. Note that M_k is random variable. For $k > 0$, let $N_k = |\mathcal{C}_0^{(k)}| = k \cdot |N(z_1)| = \frac{\beta \cdot nk}{8\log(1/(1-\rho-\beta))}$. (The fact that for $k > 0$, $N_k = k|N(z_1)|$ follows because the sets $N(z_i)$ are all disjoint, which itself follows from the distance of the code and the fact that all the clusters are of the same size).

The main observation is that conditioned on $\mathcal{C} \cap \mathcal{C}_0^{(k)}$, the distribution on $\mathcal{C}$ is the same as the distribution where $pN_0 - M_k$ codewords are picked uniformly at random, with replacement, from $\mathcal{C}_0 \setminus \mathcal{C}_0^{(k)}$. Again, this follows because the clusters $N(z_i)$ are disjoint. Call this distribution μ. For all k, we have $M_k < M_{q^{rn}/3}$. A Chernoff bound implies that this latter is small:

$$\mathbb{P}\left\{M_{q^{rn}/3} \geq pN_0/2\right\} \leq \exp\left(-\Omega(pN_0)\right).$$

We will absorb this failure probability into the calculation in (22), and assume from now on that $M_k < pN_0/2$ for all k. Now, the probability we need to bound to prove Claim 18 is

$$\mathbb{P}\left\{\Lambda(z_{k+1}) \subset \mathcal{C} | \mathcal{C} \cap \mathcal{C}^{(k)}\right\} = \mathbb{P}_\mu\left\{\forall v \in \Lambda(z_{k+1}), v \in \mathcal{C}\right\} \geq \mathbb{P}_\mu\left\{\forall v \in \Lambda(z_{k+1}), v \in \mathcal{C} \text{ exactly once }\right\}.$$

Let $D' = D/\alpha$. Then the right hand side above, the probability that each of the vectors in $\Lambda(z_{k+1})$ is picked exactly once in $\mathcal{C}$ under μ, is given by

$$\binom{pN_0 - M_k}{D'}\left(1 - \frac{D'}{N_0 - N_k}\right)^{pN_0 - M_k - D'} \frac{(D')!}{(N_0 - N_k)^{D'}} \geq \binom{pN_0/2}{D'}\left(1 - \frac{2D'}{N_0}\right)^{pN_0 - D'} \frac{(D')!}{(N_0/2)^{D'}},$$

where the inequality follows from the fact that $pN_0 \geq pN_0 - M_k > pN_0/2$ and that for all k,

$$N_0 - N_k \geq N_0 - N_{q^{rn}/3} \geq 2N_0/3 > N_0/2,$$

where the second inequality follows from the fact that all the clusters $N(z_i)$ have the same size. Now it suffices to bound the last expression from below by $\left(\frac{p}{2e}\right)^{D'}$. And indeed, we have

$$\begin{aligned}
\binom{pN_0/2}{D'}\left(1-\frac{D'}{N_0/2}\right)^{pN_0-D'}\frac{(D')!}{(N_0/2)^{D'}} &\geq \left(\frac{pN_0}{2D'}\right)^{D'}\cdot\left(1-\frac{2D'}{N_0}\right)^{pN_0}\cdot\left(\frac{2D'}{eN_0}\right)^{D'}\\
&= \left(\frac{p}{e}\right)^{D'}\cdot\left(\left(1-\frac{2D'}{N_0}\right)^{N_0/(2D')}\right)^{2pD'}\\
&\geq \left(\frac{p}{4^{2p}e}\right)^{D'}\\
&\geq \left(\frac{p}{2e}\right)^{D'}.
\end{aligned}$$

In the above the second inequality follows for $N_0 \geq 4D'$ and the final inequality follows for $p \leq 1/4$ both of which are valid assumptions for our choices for p and N_0. Finally, we have

$$\left(\frac{p}{2e}\right)^{\frac{D}{\alpha}} \geq \left(\frac{1}{2enq^{\alpha n}}\right)^{\frac{D}{\alpha}} \geq \left(\frac{1}{q^{2\alpha n}}\right)^{\frac{D}{\alpha}} = q^{-2Dn},$$

for large enough n, which completes the claim.