Threshold Rates of Code Ensembles: Linear Is Best

Nicolas Resch
Cryptology Group, Centrum Wiskunde & Informatica, Amsterdam, The Netherlands

Chen Yuan
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, China

Abstract

In this work, we prove new results concerning the combinatorial properties of random linear codes. By applying the thresholds framework from Mosheiff et al. (FOCS 2020) we derive fine-grained results concerning the list-decodability and -recoverability of random linear codes.

Firstly, we prove a lower bound on the list-size required for random linear codes over $\mathbb{F}_q$ ε-close to capacity to list-recover with error radius ρ and input lists of size ℓ. We show that the list-size L must be at least $\frac{\log_q \binom{q}{\ell} - R}{\varepsilon}$, where R is the rate of the random linear code. This is analogous to a lower bound for list-decoding that was recently obtained by Guruswami et al. (IEEE TIT 2021B). As a comparison, we also pin down the list size of random codes which is $\frac{\log_q \binom{q}{\ell}}{\varepsilon}$. This result almost closes the $O(\frac{q \log L}{L})$ gap left by Guruswami et al. (IEEE TIT 2021A). This leaves open the possibility (that we consider likely) that random linear codes perform better than the random codes for list-recoverability, which is in contrast to a recent gap shown for the case of list-recovery from erasures (Guruswami et al., IEEE TIT 2021B).

Next, we consider list-decoding with constant list-sizes. Specifically, we obtain new lower bounds on the rate required for:

- List-of-3 decodability of random linear codes over $\mathbb{F}_2$;
- List-of-2 decodability of random linear codes over $\mathbb{F}_q$ (for any q).

This expands upon Guruswami et al. (IEEE TIT 2021A) which only studied list-of-2 decodability of random linear codes over $\mathbb{F}_2$. Further, in both cases we are able to show that the rate is larger than that which is possible for uniformly random codes.

A conclusion that we draw from our work is that, for many combinatorial properties of interest, random linear codes actually perform *better* than uniformly random codes, in contrast to the apparently standard intuition that uniformly random codes are best.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases Random Linear Codes, List-Decoding, List-Recovery, Threshold Rates

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.104

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://arxiv.org/abs/2205.01513`

Funding *Nicolas Resch*: Research supported in part by ERC H2020 grant No.74079 (ALGSTRONGCRYPTO).
Chen Yuan: Research supported in part by the National Natural Science Foundation of China under Grant 12101403, the National Natural Science Foundation of China under Grant 12031011 and National Key Research and Development Project 2021YFE0109900.

1 Introduction

Coding theory is concerned with developing efficient means to makes data robust to noise. The mathematical objects used for this purpose are *(error-correcting) codes*, which are just subsets $\mathcal{C} \subseteq \Sigma^n$, where Σ is a finite alphabet of size q. It is often convenient to set $\Sigma = \mathbb{F}_q$,

49th International Colloquium on Automata, Languages, and Programming (ICALP 2022).
Editors: Mikołaj Bojańczyk, Emanuela Merelli, and David P. Woodruff;
Article No. 104; pp. 104:1–104:19


Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

where $\mathbb{F}_q$ is the finite field of order q,[1] in which case we can insist that $\mathcal{C}$ be a subspace of $\mathbb{F}_q^n$. We call such a code *linear* and denote it $\mathcal{C} \leq \mathbb{F}_q^n$. As we are mostly concerned with linear codes in the sequel we will always set $\Sigma = \mathbb{F}_q$.[2]

In order for a code to be useful for information transmission in noisy environments, we require $\mathcal{C}$ to satisfy noise-resilience properties, which amounts to insisting that the codewords are "difficult to confuse." A basic way to do this is to define a distance metric on $\mathbb{F}_q^n$ and then insist that the codewords are not too clustered. The standard choice is the (relative) *Hamming distance* which is defined as $d(\vec{x}, \vec{y}) = \frac{1}{n}|\{i \in [n] : x_i \neq y_i\}|$ for $\vec{x}, \vec{y} \in \mathbb{F}_q^n$: in words, it is the fraction of coordinates on which the vectors $\vec{x}$ and $\vec{y}$ differ. The *minimum distance* of a code is then the minimum distance between two distinct codewords, i.e., $\delta := \min\{d(\vec{c}, \vec{d}) : \vec{c}, \vec{d} \in \mathcal{C}, \vec{c} \neq \vec{d}\}$.

Beyond the minimum distance, there are other proxies for a code's noise-resilience that are widely studied. First and foremost, a popular relaxed notion of noise-resilience is provided by *list-decodability*, which informally asks that the code not be "too" clustered around any one point. More precisely, a code is said to be (ρ, L)-list-decodable if there are never L or more codewords that are all within distance ρ of some vector $z \in \mathbb{F}_q^n$, i.e.,

$$\forall \vec{z} \in \mathbb{F}_q^n,\ |\{\vec{x} \in \mathcal{C} : d(\vec{x}, \vec{z}) \leq \rho\}| < L\ .$$

The integer L is called the *list-size*. This notion, originally introduced by Elias and Wozencraft [6, 35], finds uses within coding theory and beyond in, e.g., complexity theory [27, 2, 33], cryptography [25], and learning theory [8].

We will also investigate another relaxation of list-decoding: *list-recovery*. Here, we are given a collection of input lists $S_1, \ldots, S_n \subseteq \mathbb{F}_q$ of bounded size, and the requirement is that there are not too many codewords that agree too much with these input lists. More precisely, for an integer $\ell \leq q$ we require that

$$\forall \vec{S} = (S_1, \ldots, S_n) \in \binom{\mathbb{F}_q}{\ell}^n,\ |\{\vec{x} \in \mathcal{C} : d(\vec{x}, \vec{S}) \leq \rho\}| < L\ .$$

In the above, we are denoting by $\binom{\mathbb{F}_q}{\ell}$ the family of all ℓ-element subsets of $\mathbb{F}_q$, and we are extending the Hamming distance notation $d(\cdot, \cdot)$ via

$$d(\vec{x}, \vec{S}) = \frac{1}{n}|\{i \in [n] : x_i \notin S_i\}|\ .$$

Note that $(\rho, 1, L)$-list-recovery is equivalent to list-decoding, demonstrating that list-recoverability is indeed a generalization of list-decodability. While list-recovery was initially introduced as a stepping stone towards list-decoding [11, 12, 13, 14] it has since found many new uses in theoretical computer science more broadly [20, 24, 29, 7, 22, 23].

In order for a code to provide for efficient information transmission, we would like for the code's *rate* to be as large as possible, which is a measure of the amount of information transmitted per symbol of a codeword. More precisely, the code's rate R is defined as $\frac{\log_q |\mathcal{C}|}{n}$; when the code is linear, this is simply $\frac{\dim(\mathcal{C})}{n}$.

At its core, coding theory is concerned with determining the achievable tradeoffs between a code's rate and its noise-resilience for various noise models. In this work, we focus upon the list-decodability and list-recoverability of codes. An important question we ask is how

[1] In this case, we will of course insist that q be a prime power.

[2] For nonlinear codes this does potentially lose some generality, as the alphabet size in that case could be any integer. We do remark that our results concerning arbitrary codes hold for all integer q, but emphasizing this point is not relevant to our purposes.

large the list-size L must be for these tasks. This is useful in practice, as the main constraint on the run time of most list-decoding/recovery algorithms is due to the need to process the list. Further, in applications of list-recoverable codes to constructions of expanders [20] the quality of the expansion is directly governed by the list-size.

Random Ensembles of Codes. As a stepping-stone towards a thorough understanding of the achievable tradeoffs (which is believed to be a very challenging problem), we take cues from much of the literature and study the behaviour of "typical" codes. That is, we sample codes of a prescribed rate according to natural distributions and investigate their list-decodability/-recoverability. In particular, we consider random *linear* codes, which are uniformaly sampled subspaces of $\mathbb{F}_q^n$ of the prescribed dimension. We also study uniformly random subsets of $\mathbb{F}_q^n$ of the prescribed size, which we call *random codes*.

In our work, we endeavour to provide a more fine-grained understanding of the combinatorial properties of these code ensembles. In this way, we help to uncover the landscape of achievable parameters for various code properties of interest. Beyond its theoretical interest, many code constructions [14, 19, 22, 23] use (small) linear codes as a component, and better list-decodability/recoverability of these inner codes improves these constructions.

In our results, we highlight a (perhaps surprising) fact: for list-decoding/recovery, random linear codes seem to perform *better* than uniformly random codes. On the one hand, even for the basic property of minimum distance it has already been observed that random linear codes (which achieve the Gilbert-Varshamov bound) outperform uniformly random codes. On the other hand, for problems such as list-decoding and list-recovery much of the literature appears to be focused on showing that random linear codes are "not too much worse" than uniformly random codes. We hope our work encourages the coding theory community to change perspective and endeavour to prove that random linear codes are in fact better.

1.1 Our Results

List-Recoverability of Random Linear Codes. As a first result, we provide a new lower bound on the list-size of random linear codes for list-recoverability. For context, we recall the list-recoverability capacity theorem, which gives us some coarse-grained information regarding achievable tradeoffs. For an integer $1 \leq \ell < q$, error-radius $\rho \in (1 - \ell/q)$ and $\varepsilon > 0$ it states the following:

- If $R \leq 1 - h_{q,\ell}(\rho) - \varepsilon$, there exist (ρ, ℓ, L)-list-recoverable codes with $L = O(\ell/\varepsilon)$.
- If $R \geq 1 - h_{q,\ell}(\rho) + \varepsilon$, there *do not* exist (ρ, ℓ, L)-list-recoverable codes with $L = o(q^{\varepsilon n})$.

In the above, the function $h_{q,\ell}(\cdot)$ is the (q, ℓ)-entropy function; its precise definition is not important at the moment so we defer it to Section 2. Informally, when studying codes of rate ε below the capacity for a small $\varepsilon > 0$ we refer to them as *capacity-approaching* and call ε as the *gap-to-capacity*.

This already tells us that the capacity for (ρ, ℓ, L)-list-recovery is $1 - h_{q,\ell}(\rho)$ if we insist that L be subexponential in n. However, we can ask for more fine-grained information: in particular, exactly how large must the list-size L be as a function of ε and the other parameters?

For random linear codes, we prove the following lower bound.

▶ **Theorem 1** (List-Recoverability Lower Bound for Random Linear Codes). *Let $1 \leq \ell \leq q$ be integers with q a prime power and fix $\rho \in (0, 1 - \ell/q)$. Fix $\delta > 0$. For sufficiently small $\varepsilon > 0$, a random linear code in $\mathbb{F}_q^n$ of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ is whp not $\left(\rho, \ell, \lfloor \frac{\log_q \binom{q}{\ell} - (1 - h_{q,\ell}(\rho))}{\varepsilon} - \delta \rfloor\right)$-list-recoverable.*

Table 1 This table summarizes much of the work on the list-recoverability of random linear codes (RLC) and random codes (RC). The lower bound of [15] only applies when $q = p^{\Omega(1/\varepsilon)}$ for a prime p, and in [32] $\eta > 0$ is viewed as a small constant. [15] also offers a similar lower bound for the case of list-recovery from erasures.

Source	Model	Radius	Rate	List-size bound
Folklore	RC	$\rho > 0$	$1 - h_{q,\ell}(\rho) - \varepsilon$	$\leq O(\ell/\varepsilon)$
[37]	RLC	$\rho > 0$	$1 - h_{q,\ell}(\rho) - \varepsilon$	$\leq q^{O(\ell/\varepsilon)}$
[32]	RLC	$\rho = 1 - \frac{\ell}{q} - \eta$	$0.99(1 - h_{q/\ell}(\alpha) - \log_q(\ell))$	$\leq q^{O(\ln^2(\ell/\eta))}$
[15]	RLC	$\rho = 0$	$1 - \log_q(\ell) - \varepsilon$	$\geq \ell^{\Omega(1/\varepsilon)}$
Theorem 1	RLC	$\rho > 0$	$1 - h_{q,\ell}(\rho) - \varepsilon$	$> \frac{\log_q \binom{q}{\ell} - (1 - h_{q,\ell}(\rho))}{\varepsilon}$
Theorem 2	RC	$\rho > 0$	$1 - h_{q,\ell}(\rho) - \varepsilon$	$\approx \frac{\log_q \binom{q}{\ell}}{\varepsilon}$

For context, we consider the case of uniformly random codes. In this case, we obtain a tight result.

▶ **Theorem 2** (List-Recoverability for Random Codes). *Let $1 \leq \ell \leq q$ be integers with q a prime power and fix $\rho \in (0, 1 - \ell/q)$. Fix $\delta > 0$. For sufficiently small $\varepsilon > 0$, a random code in $\mathbb{F}_q^n$ of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ is whp* not *$\left(\rho, \ell, \lfloor \frac{\log_q \binom{q}{\ell}}{\varepsilon} - \delta \rfloor\right)$-list-recoverable.*

On the other hand, for any $\varepsilon > 0$ and n sufficiently large, a random code in $\mathbb{F}_q^n$ of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ is whp $\left(\rho, \ell, \lceil \frac{\log_q \binom{q}{\ell}}{\varepsilon} \rceil + 1\right)$-list-recoverable.

In this way, we pin down the list-recoverability for random codes to one of two or three possible values: $\lfloor \frac{\log_q \binom{q}{\ell}}{\varepsilon} + 0.99 \rfloor$, $\lceil \frac{\log_q \binom{q}{\ell}}{\varepsilon} \rceil$ (if it's different) or $\lceil \frac{\log_q \binom{q}{\ell}}{\varepsilon} \rceil + 1$.

Comparing Theorems 1 and 2 we see that our lower bound on random linear codes is less than the precise bound we have on random codes. One could potentially draw the conclusion that Theorem 1 should be improved. However, we believe that it is in fact tight. For the case of list-decoding binary codes it has already been shown that random linear performs better than uniformly random, and the bounds we obtain are the natural generalizations of the (tight) results for that case. We therefore conjecture that Theorem 1 is indeed tight. This stands in stark contrast to *erasure* list-recovery:[3] for this model, it is known that random linear codes can require lists of size $\ell^{\Omega(1/\varepsilon)}$ [15] (at least, if the field has large characteristic), whereas the lists for random codes can be just $O(\ell/\varepsilon)$. A summary of the state-of-the-art for list-recovery of RLCs and RCs is provided in Table 1.

▶ Remark 3. It might appear that our conjecture that random linear codes outperform random codes for list-recovery is contradicted by the result of [15]. However, we emphasize that the capacity for erasure list-recovery is larger, so if a code is ε-close to capacity for list-recovery from erasures for small $\varepsilon > 0$ it is above capacity for list-recovery from errors, the model we study. Hence, this lower bound does not contradict our conjecture. One can also consider the model where ρ approaches the limit $1 - \ell/q$ as is done in [32]; in this case we still suspect that random linear codes outperform uniformly random codes, but this is just speculation and further investigation is required.

[3] Here, the requirement is that for all subsets $S_1, \ldots, S_n \subseteq \mathbb{F}_q$ where at least $(1 - \rho)n$ of the S_i's satisfy $|S_i| \leq \ell$ (and the others may be all of $\mathbb{F}_q$), the number of codewords in $S_1 \times \cdots \times S_n$ is less than L.

List-decoding with small lists. Next, we turn our attention to the challenge of list-decoding when the output list-size L is a (small) constant. Thus, we are no longer in the regime where we can expect to approach the list-decoding capacity, and we are interested to know by how much we are required to back off if, say, $L = 3, 4$.

First, we consider the case where $L = 4$ for the binary field, which we also refer to as *list-of-*3 decoding. Here and throughout, we also use the following notation (which is slightly abusive): for $q \geq 2$ and nonnegative reals $x_1, \ldots, x_t$ with $x_1 + \cdots + x_t \leq 1$, $H_q(x_1, \ldots, x_t) = \sum_{i=1}^{t} x_i \log_q \frac{1}{x_i} + (1 - x_1 - \cdots - x_t) \log_q \frac{1}{1-x_1-\cdots-x_t}$.

We first prove the following possibility result for random linear codes. In the following,

$$\mathcal{B}_\rho = \left\{(x_1, x_2) \in \mathbb{R}^2 : x_1 + 2x_2 \leq 4\rho, x_1 + x_2 \leq 1, x_1, x_2 \geq 0\right\} .$$

▶ **Theorem 4** (List-of-3 decoding Random Linear Binary Codes). *Let $\rho \in (0, 5/16)$*[4] *and suppose*

$$R < 1 - \max_{(x_1,x_2)\in\mathcal{B}_\rho} \frac{H_2(x_1, x_2) + 2x_1 + x_2 \log_2 3}{3} .$$

Then a random linear code over $\mathbb{F}_q$ of rate R is whp $(\rho, 4)$-list-decodable.

For context, we also study the list-of-3 decodability of random codes over the binary alphabet. In this case, we can prove the following:

▶ **Theorem 5** (List-of-3 decoding Random Binary Codes). *Let $\rho \in (0, 5/16)$ and suppose*

$$R > 1 - \max_{(x_1,x_2)\in\mathcal{B}_\rho} \frac{1 + H_2(x_1, x_2) + 2x_1 + x_2 \log_2 3}{4} .$$

Then a random code over $\{0, 1\}$ of rate R is whp not *$(\rho, 4)$-list-decodable.*

On the other hand, if

$$R < 1 - \max_{(x_1,x_2)\in\mathcal{B}_\rho} \frac{1 + H_2(x_1, x_2) + 2x_1 + x_2 \log_2 3}{4} ,$$

then a random code over $\{0, 1\}$ is whp $(\rho, 4)$-list-decodable.

As $\frac{1+F}{4} \geq \frac{F}{3}$ whenever $F \leq 3$, we see that the bound in Theorem 4 is greater than the bound from Theorem 5. Using terminology that we later make precise, we see that the *threshold rate* for list-of-3 decoding binary random linear codes strictly exceeds that of binary random codes.

Next, we study list-of-2 decoding over alphabets of size $q > 2$. And again, our theorems demonstrate that random linear codes strictly outperform random codes. Define

$$\mathcal{D}_\rho := \{(x_1, x_2) \in \mathbb{R}^2 : x_1 + x_2 \leq 3\rho, x_1 + x_2 \leq 1, x_1, x_2 \geq 0\}.$$

▶ **Theorem 6** (List-of-2 decoding Random Linear q-ary Codes). *Let $\rho \in (0, 1/3)$ and suppose*

$$R < 1 - \max_{(x_1,x_2)\in\mathcal{D}_\rho} \frac{H_q(x_1, x_2) + x_1 \log_q 3(q-1) + x_2 \log_q (q-1)(q-2)}{2} .$$

Then a random linear code over $\mathbb{F}_q$ of rate R is whp $(\rho, 3)$-list-decodable.

[4] If $\rho \geq 5/16$ it is known that there are no $(\rho, 4)$-list-decodable codes with postive rate [1].

▶ **Theorem 7** (List-of-2 decoding Random q-ary Codes). *Let $\mathbb{F}_q$ be an alphabet of size q. Let $\rho \in (0, 1/3)$ and suppose*

$$R > 1 - \max_{(x_1,x_2)\in\mathcal{D}_\rho} \frac{1 + H_q(x_1, x_2) + x_1 \log_q 3(q-1) + x_2 \log_q (q-1)(q-2)}{3} .$$

Then a random code over $\mathbb{F}_q$ of rate R is whp not $(\rho, 3)$*-list-decodable.*

On the other hand, if

$$R < 1 - \max_{(x_1,x_2)\in\mathcal{D}_\rho} \frac{1 + H_q(x_1, x_2) + x_1 \log_q 3(q-1) + x_2 \log_q (q-1)(q-2)}{3} ,$$

then a random code over $\mathbb{F}_q$ is whp $(\rho, 3)$-list-decodable.

Again, we can see that the bound from Theorem 6 is greater than the bound from Theorem 7. We therefore conjecture that this phenomenon of random linear codes outperforming random codes extends to more values of L. To provide more evidence for this conjecture, we extend an argument for binary random linear codes of [10, 26] to larger values of L, and by comparing it to a computation of the threshold rate for random binary codes, show that for many parameter regimes of interest we do indeed have random linear codes outperforming random codes.

1.2 Techniques

In order to obtain our results, we rely on a recently developed toolkit for proving threshold rates for combinatorial properties of random (linear) codes. This toolkit was developed by Mosheiff et al. [28] on the way to proving that LDPC codes achieve list-decoding capacity; recent works [15, 16] have found further uses for the techniques in investing combinatorial properties of random linear codes. An analogous threshold toolkit for *random codes* was provided in [17].

Broadly speaking, the techniques of [28, 17] apply when considering a property of codes defined by forbidding a family of "bad" subsets, each of which have constant cardinality (independent of n). For example, the property of (ρ, L)-list-decodability is defined by forbidding all L-element subsets $B = \{x_1, \ldots, x_L\}$ of a Hamming ball $B(z, \rho) = \{x \in \mathbb{F}_q^n : d(x, z) \leq \rho\}$ from appearing in the code. In [28], it is proved that for any such local property there is a *threshold* rate R^* such that:

- If $R < R^*$, a random linear code satisfies the property with high probability;
- If $R > R^*$, a random linear code fails to satisfy the property with high probability.

The theorem furthermore characterizes the threshold rate R^* as the solution to a certain optimization problem. In this work, we endeavour to compute new bounds on the threshold rate R^* for various properties of interest.

In the remainder, we provide intuition for the characterization of the threshold rate from [28]. First, we identify subsets $B \subseteq \mathbb{F}_q^n$ of size L with the matrix in $\mathbb{F}_q^{n\times L}$ whose columns are given by B (the choice of ordering is immaterial), and we say that a matrix M is contained in a code $\mathcal{C}$ if $\mathcal{C}$ contains all of M's columns. For a collection of matrices $\mathcal{M} \subseteq \mathbb{F}_q^{n\times L}$, we would like to compute the threshold rate R^* for "$\mathcal{M}$-freeness," i.e., the code property of not containing a matrix in $\mathcal{M}$.

As we are interested in list-decoding/recovery, we define a set of matrices $\mathcal{M}$ such that if $\mathcal{C}$ contains a matrix from $\mathcal{M}$ then $\mathcal{C}$ is not list-decodable/recoverable. We say that the collection $\mathcal{M}$ is "bad" for list-decoding/recovery. As intuition, for list-decoding we can just take the set of matrices where each column lies in some ball $B(z, \rho)$. Next, we would like

to show that $\mathcal{M}$ is "abundant" in the sense that it is very likely that $\mathcal{C}$ contains a matrix $M \in \mathcal{M}$. In other words, if X_M denotes the indicator random variable for the event $M \subseteq \mathcal{C}$, then we should expect $X_{\mathcal{M}} := \sum_{M \in \mathcal{M}} X_M \geq 1$.

It is relatively easy to compute $\mathbb{E}[X_{\mathcal{M}}]$ and see when it exceeds 1; however, to conclude that $X_{\mathcal{M}}$ is likely to be large one needs a concentration bound. Such a bound is often provided by estimating the variance of $X_{\mathcal{M}}$. Broadly construed, [28] applies the second moment method to demonstrate that there is really only one reason that $X_{\mathcal{M}}$ would fail to be concentrated: it is because for some compressing matrix $A \in \mathbb{F}_q^{L \times L'}$ with $L' \leq L$ the set $\{MA : M \in \mathcal{M}\}$ is too small.

List-Recovery. First, we endeavour to prove a lower bound on the list-size for list-recovery. This means that we need to say that if the list-size is too small then the random linear code quite likely contains a matrix from a set $\mathcal{M}$ of bad matrices for list-recovery. In light of the above, to conclude our argument we need to show that for any compressing matrix A, the set $\{MA : M \in \mathcal{M}\}$ remains large.

To do this, we use information-theoretic techniques: we identify each of our bad matrices $M \in \mathcal{M}$ with an appropriate *type*, which is a distribution $\tau \sim \mathbb{F}_q^L$ defined as the empirical distribution of M's rows. A lower bound on $\{MA : M \in \mathcal{M}\}$ is then implied by a lower bound on the entropy of the random variable $A\vec{\boldsymbol{u}}$ for $\vec{\boldsymbol{u}} \sim \tau$. We are also free to choose the type τ which is "bad" for a certain property, in the sense that if a code contains a matrix of type τ then it fails to satisfy the property.

For the case of (ρ, ℓ, L)-list-recovery, the following type is bad: one samples uniformly $\boldsymbol{S} \in \binom{\mathbb{F}_q}{\ell}$ and then outputs $\vec{\boldsymbol{u}} = (\boldsymbol{u}_1, \ldots, \boldsymbol{u}_L) \in \mathbb{F}_q^L$, where each $\boldsymbol{u}_i$ is independently uniform over $\boldsymbol{S}$ with probability $1 - \rho$ and uniform over $\mathbb{F}_q \setminus \boldsymbol{S}$ otherwise. It thus follows that a lower bound on $\{AM : M \in \mathcal{M}\}$ is implied by a lower bound on the entropy of the random variable $A\vec{\boldsymbol{u}}$ for $\vec{\boldsymbol{u}} \sim \tau$.

Obtaining this lower bound requires a rather lengthy argument; we overview the main ideas now. We begin by partitioning the coordinates of $A\vec{\boldsymbol{u}}$ into subsets $J_1, \ldots, J_k \subseteq [L']$, where each J_i depends on at least 2 "fresh" coordinates from $\vec{\boldsymbol{u}}$, along with (perhaps) a set of leftover coordinates J_{k+1}. We then provide two arguments depending on the maximum size of a part. If, say, $|J_1|$ is large, then we can show that $(A\vec{\boldsymbol{u}})_{J_1}$ already experiences a large entropy increase. This is shown by demonstrating that these coordinates alone already allow us to nontrivially guess the subset $\boldsymbol{S}$. Otherwise, we argue that all the parts provide a nontrivial increase in the entropy, and since there must be a large number of parts in this case, by summing over all the parts we provide an adequate lower bound.

This result generalizes the list-decoding lower bound that was provided in [15, Theorem IV.1]. The argument in that paper exploited the fact that a sample from the bad type for list-decoding has a simpler structure: it looks like $\vec{\boldsymbol{v}} + \boldsymbol{\alpha}\vec{1}$, where $\vec{\boldsymbol{v}}$ is a q-ary Bernoulli random variable and $\boldsymbol{\alpha} \in \mathbb{F}_q$ is uniformly random. In our case, we do not have this nice linear structure,[5] making the analysis more intricate.

List-Decoding with Small Lists. For our results concerning list-decoding with small lists, we again use the thresholds framework. In this case, we need to consider *any* type that is bad for $(\rho, 3)$ or $(\rho, 4)$-list-decoding. For these small values of L, we are able to identify the linear map A which leads to the maximum relative entropy $\frac{H_q(A\tau)}{\dim(A\tau)}$: in each case, it is given by the map sending $(x_1, \ldots, x_L) \mapsto (x_1 - x_L, \ldots, x_{L-1} - x_L)$.

[5] One might be tempted to look at $\vec{\boldsymbol{v}} + \vec{\boldsymbol{w}}$ where $\vec{\boldsymbol{v}}$ is q-ary Bernoulli and $\vec{\boldsymbol{w}}$ is uniform over $\boldsymbol{S}$, but note that for $\ell - 1$ choices for $\boldsymbol{v}_i \in \mathbb{F}_q^*$ the sum $\boldsymbol{v}_i + \boldsymbol{w}_i$ still lies in $\boldsymbol{S}$.

To provide the proof, we break up the vector spaces based on the number of distinct coordinates of the entries, and observe that a type which is bad for list-decodability can only put so much probability mass on each of these parts. To conclude, we rely on the concavity of the entropy function as well as some combinatorial reasoning concerning the subspaces of $\mathbb{F}_2^4$ and $\mathbb{F}_q^3$. Even for these small values of L we need to be quite careful to avoid a massive explosion in the number of cases to consider, as we must look at all compressing linear maps A.

Random Codes. For the case of random codes, we can compute the threshold rates for all the properties of interest in a relatively straightforward way, as the characterization from [17] does not require us to consider any sort of compressing mapping on the types. Quite notably, in all cases we see that random linear codes appear to perform better than random codes. This is perhaps in contrast to commonly held beliefs: in this sense, a main goal of our work is to disseminate this counterintuitive phenomenon.

1.3 Related Work

In Section 1.2 we outlined the works [28, 15, 17] which developed and studied the thresholds toolkit that we apply. In this section, we provide more context for the study of random linear codes and their list-decodability/-recoverability. In what follows, q always denotes the alphabet size and ε the "gap-to-capacity" for a capacity-approaching code.

List Size Lower Bounds for Random (Linear) Codes. As we provide lower bounds for list-recovery of random linear codes, we briefly survey the known lower bounds for list-decoding. First, Guruswami and Narayanan [21] showed that capacity-approaching random (linear) codes require lists of size $\Omega_{\rho,q}(1/\varepsilon)$: by inspecting the proof one can note that the implied constant tends to 0 as $\rho \to 1-1/q$, or if $q \to \infty$. While on the surface their approach appears very different to ours, their use of a second-moment method is akin to the proofs underlying the thresholds framework from [28], so the approaches are in fact somewhat similar. Later, Li and Wootters [26] gave a $\sim 1/\varepsilon$ list-size lower bound for capacity-approaching random codes. Again, the argument relies on the second-moment method.

In [15], a lower bound for the list-decodability of capacity-approaching random linear codes is given, showing that lists of size $\sim \frac{h_q(\rho)}{\varepsilon}$ are required: our list-recovery list-size lower bound is a generalization of this result. Lastly, in [17] the threshold rate for $(\rho, 2)$-list-decodability is computed, providing a lower bound and an upper bound: this segues us nicely into a discussion of the work on computing upper bounds on list-sizes.

List Size Upper Bounds for Random Linear Codes. There has been a long line of work [37, 10, 9, 5, 34, 31, 32, 26, 17] studying the list-decodability of capacity-approaching random linear codes, and we now highlight some relevant results. First, Zyablov and Pinkser [37] demonstrated that capacity-approaching RLCs are indeed (ρ, L)-list-decodable, albeit with $L = q^{\Omega(1/\varepsilon)}$. Subsequent work has endeavoured to prove list-decodability with $L = O(1/\varepsilon)$. The existence of such linear codes over $\mathbb{F}_2$ was first demonstrated by [10]; later, [26] showed that this holds with high probability for randomly sampled linear codes, and subsequently [15] showed this is true for *average-radius*[6] list-decoding.

[6] In this model, it is required that the code does not contain L points whose average distance from a centre is less than ρ. Thus, it is a stricter requirement than standard list-decoding.

As for larger alphabets, [9] showed that lists of size $O_{\rho,q}(1/\varepsilon)$ do indeed suffice for random linear codes. We further remark that their argument uses a certain Ramsey-theoretic concept called a 2-increasing sequence to choose the order in which to reveal coordinates, which is vaguely reminiscent of the "fresh" coordinates that we have defined by the J_i's in our list-recovery lower bound argument. A drawback of this work is that the implied constant in the $O_{\rho,q}(\cdot)$ notation degrades as $\rho \to 1-1/q$ or if q grows too large. In light of this restriction, a line of works [5, 34, 31] has studied the "high noise regime," where $\rho = 1-1/q-\eta$ and one endeavours to show that lists of size $O(1/\eta^2)$ suffice for codes of rate $\Omega(\eta^2)$. These results are still not quite optimal in the sense that the implied constants (even for the rate) lag behind the parameters achievable by random codes. Lastly, for list-recoverability with input list-size ℓ it appears that the best upper bound on the list-size is due to [32], where it is shown that lists of size $(q\ell)^{O(\log(\ell)/\varepsilon)}$ suffice.

Lower Bounds for List Sizes of Arbitrary Codes. While we exclusively study random (linear) codes, we view these as a proxy for determining the actual achievable tradeoffs. As lists of size $\Theta(1/\varepsilon)$ are required for random codes, it is natural to wonder if all capacity-approaching (ρ, L)-list-decodable codes require lists of size $\Omega(1/\varepsilon)$. Blinovsky [4, 3] has shown a lower bound of $\Omega_\rho(\log(1/\varepsilon))$. In the high noise regime, viz., $\rho = 1-1/q-\eta$, Guruswami and Vadhan [21] provided a $\Omega_q(1/\eta^2)$ lower bound on the list size. Lastly, for *average-radius* list-decoding Guruswami and Narayanan [18] proved a $\Omega_\rho(1/\sqrt{\varepsilon})$ lower bound.

1.4 Open Problems

In this work, we have progressed our understanding of combinatorial properties of random (linear) codes. A main conclusion of our work is that for list-decoding/recovery, random linear codes perform better.[7]

There are many open problems which remain to be studied and we list some below.

- Provide the corresponding upper bounds on the threshold rate for $(\rho, 4)$-list-decoding binary random linear codes, and the threshold rate for $(\rho, 3)$-list-decoding q-ary random linear codes.
- Provide the corresponding lower bound on the threshold rate for (ρ, ℓ, L)-list-recovery in the capacity-approaching regime. In fact, for $q > 2$, the threshold rate for (ρ, L)-list-decoding is still open. This is quite likely a very challenging problem; the only tight argument we have is due to [10, 26] (see also [15]) which only applies to list-decoding over the binary field, and this argument appears too "rigid" to apply in more generality.
- Get a better understanding for *worst-case* codes. In particular, to the best of our knowledge the *Plotkin points* for (ρ, L)-list-decoding for $q > 2$ are not known. That is, compute the minimum value ρ^* such that for all $\rho > \rho^*$, there are no q-ary (ρ, L)-list-decodable code families with positive rate. (Recent work [36] expresses the Plotkin point as a solution to a certain optimization problem, but we do not see how to extract a simple expression from this.)

1.5 Organization

In the subequent section, we introduce the necessary notations and definitions that we will use in this work, along with the tools from [28, 17] that we apply. In Section 3, we provide our lower bound on the list-size for the list-recoverability of random linear codes which

[7] For list-recovery, we admittedly only provide some evidence in this direction.

approach capacity. In Section 4, we lower bound the threshold rate for list-of-2 decoding (for general q) and list-of-3 decoding (in the binary case). We also compare random linear codes to random codes over the binary alphabet for more values of L. For space reasons, most of the technical proofs are deferred to the full version.

2 Prelimaries

Miscellaneous Notations. For an integer $n \geq 1$, we denote $[n] := \{1, 2, \ldots, n\}$. For a set X we denote by $\binom{X}{\ell}$ the family of all subsets of X with ℓ elements, and similarly $\binom{X}{\leq \ell}$ denotes the family of all subsets of X with $\leq \ell$ elements. Throughout, $\mathbb{F}_q$ denotes the finite field with q elements, for q a prime power.

For clarity, vectors are typically denoted with an arrow overtop. Given a vector $\vec{x} \in \mathbb{F}_q^n$ and a subset $I \subseteq [n]$ we denote by $\vec{x}_I$ the length $|I|$ vector $(x_i : i \in I) \in \mathbb{F}_q^{|I|}$. We reserve $\vec{1}$ for the all-1's vector; if we wish to emphasize its length we subscript it, i.e., $\vec{1}_D$ is the all-1's vector of length D. Random variables are typically written in boldface, e.g., $\boldsymbol{x}, \boldsymbol{y}$, etc. In particular, random vectors are denoted, e.g., $\vec{\boldsymbol{u}}$.

Coding Theory Terminology. A *code* $\mathcal{C}$ is a subset of $\mathbb{F}_q^n$ for $\mathbb{F}_q$ the finite field of order q, a prime power. Elements $\vec{c} \in \mathcal{C}$ are called *codewords*, the integer n is the *block-length*, and the integer q is the *alphabet size*; such a code is also called *q-ary*. When $q = 2$ the code is deemed *binary*. We are typically interested in *linear* codes, which are $\mathcal{C} \leq \mathbb{F}_q^n$, i.e., they are subspaces. The *rate* of a code $\mathcal{C}$ is $R = R(\mathcal{C}) := \frac{\log_q |\mathcal{C}|}{n}$ and its minimum distance is $\delta = \delta(\mathcal{C}) := \min\{d(\vec{c}, \vec{d}) : \vec{c} \neq \vec{d}, \vec{c}, \vec{d} \in \mathcal{C}\}$, where $d(\vec{x}, \vec{y}) = \frac{1}{n}|\{i \in [n] : x_i \neq y_i\}|$ is the (relative) Hamming distance from $\vec{x}$ to $\vec{y}$. We also slightly extend this notation as follows: for a vector $\vec{x} \in \mathbb{F}_q^n$ and a tuple of subsets $\vec{S} = (S_1, \ldots, S_n)$, $S_i \subseteq \mathbb{F}_q$, we define $d(\vec{x}, \vec{S}) := \frac{1}{n}|\{i \in [n] : x_i \notin S_i\}|$, i.e., the fraction of coordinates i for which $\vec{x}$ "disagrees" with the corresponding subset of $\vec{S}$.

A *random linear code* of rate R is a uniformly random subspace of $\mathbb{F}_q^n$ of dimension Rn.[8] As this concept will arise regularly in this work, we occasionally use the abbreviation *RLC*. A *random code* of rate R is a random subset of $\mathbb{F}_q^n$ obtained by including each element independently with probability $q^{(R-1)n}$.[9] For this concept, we use the abbreviation *RC*.

2.1 List-decodability and List-recoverability

In this work, we study combinatorial properties of linear codes. Of primary interest to us are list-decodability and list-recoverability, which we now define.

▶ **Definition 8** (List-decodability). *Let $\rho \in (0, 1 - 1/q)$ and $L \geq 1$. A code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is called* (ρ, L)-list-decodable *if for all $\vec{z} \in \mathbb{F}_q^n$,*

$$|\{\vec{c} \in \mathcal{C} : d(\vec{c}, \vec{z}) \leq \rho\}| < L \ .$$

[8] In fact, there are different ways to sample linear codes. For concreteness, we typically implicitly use the model where a random parity check matrix $\boldsymbol{H} \in \mathbb{F}_q^{(1-R)n \times n}$ is sampled and we output $\boldsymbol{\mathcal{C}} = \ker(\boldsymbol{H})$. Of course, there is a small chance $\boldsymbol{\mathcal{C}}$ has rate larger than R, but as this probability is exponentially small in n it is immaterial to our conclusions. We also briefly use the model where a random $\boldsymbol{G} \in \mathbb{F}_q^{Rn \times n}$ is sampled and we output $\boldsymbol{\mathcal{C}} = \mathrm{im}(\boldsymbol{G})$.

[9] By Chernoff bounds, such a code as rate $R \pm o(1)$ with high probability.

We also use the terminology "list-of-L-decoding" for $(\rho, L+1)$-list-decoding, e.g., list-of-2-decoding corresponds to $(\rho, 3)$-list-decoding.

The list-decoding *capacity* is the value $R^*(\rho)$ such that for any $R < R^*(\rho)$ there exists $L > 1$ such that infinite families of (ρ, L)-list-decodable codes of rate at least R exist, but for any $R > R^*(\rho)$ such an infinite family does not exist. It is known that

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

where

$$h_q(\rho) = \rho \log_q \frac{q-1}{\rho} + \log_q \frac{1}{1-\rho}$$

is the q-ary entropy function.

▶ **Definition 9** (List-recoverability). *Let $\rho \in (0, 1-1/q)$, $1 \le \ell \le q$ and $L \ge 1$. A code $\mathcal{C} \subseteq \mathbb{F}_q^n$ is called* (ρ, ℓ, L)-list-recoverable *if for all tuples of subsets $\vec{S} = (S_1, \ldots, S_n) \in \binom{\mathbb{F}_q}{\le \ell}^n$,*

$$|\{\vec{c} \in \mathcal{C} : d(\vec{c}, \vec{S}) \le \rho\}| < L \ .$$

In analogy to the list-decoding capacity, the *list-recovery capacity* is the value $R^*(\rho, \ell)$ such that for any $R < R^*(\rho, \ell)$ there exists $L > 1$ such that infinite families of (ρ, ℓ, L)-list-recoverable codes of rate at least R exist, but for any $R > R^*(\rho, \ell)$ such an infinite family does not exist. It is known that

$$R^*(\rho, \ell) = 1 - h_{q,\ell}(\rho) \ ,$$

where

$$h_{q,\ell}(\rho) = \rho \log_q \frac{q-\ell}{\rho} + (1-\rho) \log_q \frac{\ell}{1-\rho}$$

is the (q, ℓ)-entropy function.

2.2 Information-Theoretic Concepts

For a random variable $\boldsymbol{x}$ over a domain $\mathcal{X}$ we denote its entropy by

$$H(\boldsymbol{x}) = \sum_{x \in \mathcal{X}} \Pr[\boldsymbol{x} = x] \log \frac{1}{\Pr[\boldsymbol{x} = x]} \ ,$$

where we use the convention $0 \log \frac{1}{0} = 0$. If τ is a distribution then we define $H(\tau)$ to be the entropy of a random variable distributed according to τ.

Given another random variable $\boldsymbol{y}$ supported on a set $\mathcal{Y}$, the *conditional entropy* of $\boldsymbol{x}$ given $\boldsymbol{y}$ is

$$H(\boldsymbol{x}|\boldsymbol{y}) = \mathop{\mathbb{E}}_{y \sim \boldsymbol{y}} [H(\boldsymbol{x}|\boldsymbol{y} = y)] = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \Pr[\boldsymbol{x} = x, \boldsymbol{y} = y] \log \frac{\Pr[\boldsymbol{x} = x]}{\Pr[\boldsymbol{x} = x, \boldsymbol{y} = y]} \ .$$

Intuitively, this is the expected amount of entropy remaining in $\boldsymbol{x}$ after revealing $\boldsymbol{y}$. Conditional entropy satisfies the *chain rule* $H(\boldsymbol{x}, \boldsymbol{y}) = H(\boldsymbol{x}|\boldsymbol{y}) + H(\boldsymbol{y})$, which can be extended by induction to larger collections of random variables.

We also use the notion of *mutual information*, which is a measure of the amount of information one random variable gives about another and is defined as follows:

$$I(\boldsymbol{x}; \boldsymbol{y}) = H(\boldsymbol{x}) - H(\boldsymbol{x}|\boldsymbol{y}) = H(\boldsymbol{y}) - H(\boldsymbol{y}|\boldsymbol{x}) = H(\boldsymbol{x}, \boldsymbol{y}) - H(\boldsymbol{x}) - H(\boldsymbol{y}) \ .$$

(The equalities are justified by the chain rule.) We also consider the *conditional mutual information*, defined as follows:

$$I(\boldsymbol{x};\boldsymbol{y}|\boldsymbol{z}) = H(\boldsymbol{x}|\boldsymbol{z}) - H(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{y}) = H(\boldsymbol{y}|\boldsymbol{z}) - H(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{z}) = H(\boldsymbol{x},\boldsymbol{y}|\boldsymbol{z}) - H(\boldsymbol{x}|\boldsymbol{z}) - H(\boldsymbol{y}|\boldsymbol{z}) \ ,$$

where $\boldsymbol{z}$ is another random variable.

Conditional entropy, mutual information and conditional mutual information all satisfy the *data-processing inequality*: for any function f supported on $\mathcal{Y}$ (the domain of Y), we have

$$H(\boldsymbol{x}|f(\boldsymbol{y})) \geq H(\boldsymbol{x}|\boldsymbol{y}) \ , I(\boldsymbol{x};\boldsymbol{y}) \geq I(\boldsymbol{x};f(\boldsymbol{y})) \ , I(\boldsymbol{x};\boldsymbol{y}|\boldsymbol{z}) \geq I(\boldsymbol{x};f(\boldsymbol{y})|\boldsymbol{z}) \ .$$

We will also use *Fano's inequality*.

▶ **Theorem 10** (Fano's Inequality.)**.** *Let $\boldsymbol{x}$ be a random variable supported on $\mathcal{X}$, $\boldsymbol{y}$ a random variable supported on $\mathcal{Y}$ and $f : \mathcal{Y} \to \mathcal{X}$. Define $p_{\mathrm{err}} := \Pr[f(\boldsymbol{y}) \neq \boldsymbol{x}]$. Then,*

$$H(\boldsymbol{x}|\boldsymbol{y}) \leq h(p_{\mathrm{err}}) + p_{\mathrm{err}} \cdot \log(|\mathcal{X}| - 1) \ .$$

When we wish to change the base of the logarithm with which the entropy or mutual information, the desired base is subscripted. That is,

$$H_q(\boldsymbol{x}) := \frac{H(\boldsymbol{x})}{\log q} \ , \qquad I_q(\boldsymbol{x};\boldsymbol{y}) := \frac{I(\boldsymbol{x};\boldsymbol{y})}{\log q} \ ,$$

and similarly for the conditional versions of these quantities. Finally, as a slight abuse of notation, we also write

$$H_q(x_1,\ldots,x_t) = \sum_{i=1}^{t} x_i \log_q \frac{1}{x_i} + (1 - x_1 - \cdots - x_t) \log_q \frac{1}{1 - x_1 - \cdots - x_t}$$

if $x_1,\ldots,x_t$ are positive numbers satisfying $\sum_{i=1}^{t} x_i \leq 1$. (We caution that for $q > 2$, $H_q(x) \neq h_q(x)$.)

2.3 Thresholds

We now introduce the specialized notations and tools that we will need in order to apply the machinery of [28]. First, for a distribution $\tau \sim \mathbb{F}_q^b$ and a linear map $A : \mathbb{F}_q^b \to \mathbb{F}_q^c$, we let $A\tau$ denote the distribution of the random vector $A\vec{\boldsymbol{u}}$ for $\vec{\boldsymbol{u}} \sim \tau$. In more detail, $A\tau$ has the following probability mass function:

$$\Pr_{\vec{\boldsymbol{v}} \sim A\tau}[\vec{\boldsymbol{v}} = \vec{y}] = \sum_{\vec{x} \in A^{-1}(\vec{y})} \Pr_{\vec{\boldsymbol{u}} \sim \tau}[\vec{\boldsymbol{u}} = \vec{x}] \ .$$

While we are generally concerned with understanding the probability that certain "bad sets" lie in our code, it is in fact more convenient to work with matrices. For a matrix $M \in \mathbb{F}_q^{n \times b}$ and a code $\mathcal{C} \subseteq \mathbb{F}_q^n$ we say that $\mathcal{C}$ contains M if the columns of M are contained in $\mathcal{C}$.

Every matrix is assigned a *type*, and the type of a matrix is determined by the matrix's empirical row distribution as follows:

▶ **Definition 11** ($\tau_M, \dim(\tau), \mathcal{M}_{n,\tau}$)**.** *For a matrix $M \in \mathbb{F}_q^{n \times b}$, we define its* type τ_M *to be the distribution given by the empirical distribution of M's rows. That is, for all $\vec{v} \in \mathbb{F}_q^b$ we have*

$$\tau_M(\vec{v}) := \frac{|\{i \in [n] : i\mathit{th\ row\ of\ } M \ \mathit{equals}\ \vec{v}\}|}{n} \ .$$

For a distribution τ on $\mathbb{F}_q^b$, $\dim(\tau)$ denotes the dimension of the span of τ's support, i.e.,

$$\dim(\tau) := \dim(\mathrm{span}(\mathrm{supp}(\tau))).$$

*We denote by $\mathcal{M}_{n,\tau}$ the set of all matrices in $\mathbb{F}_q^{b\times n}$ with empirical row distribution τ. We call a type τ b-*local *if $\tau \sim \mathbb{F}_q^b$; note that a b-local type has $\dim(\tau) \le b$.*

▶ Remark 12. Technically, for a distribution $\tau \sim \mathbb{F}_q^b$ it could be the case that $\mathcal{M}_{n,\tau}$ is empty just because, for some $\vec{v} \in \mathbb{F}_q^b$, $\tau(\vec{v}) \notin \{0, 1/n, 2/n, \ldots, (n-1)/n, 1\}$. For such τ, we can define $\mathcal{M}_{n,\tau}$ to consist of those matrices which contain either $\lfloor n \cdot \tau(\vec{v})\rfloor$ or $\lceil n \cdot \tau(\vec{v})\rceil$ copies of $\vec{v}$. As we are always dealing with the setting where n is assumed to be sufficiently large compared to all other parameters, this does not affect the analysis. Hence, we may safely ignore this technicality, which we do for the clarity of exposition.

Our target is an understanding of the threshold rate for a combinatorial property of random linear codes. The combinatorial properties that we will study are those that are defined by excluding a set of types, as follows.

▶ **Definition 13** (τ-freeness, local properties)**.** *Given a code $\mathcal{C}$ and a type τ, we say that $\mathcal{C}$ is* τ-free *if $\mathcal{C}$ does not contain any matrix $M \in \mathcal{M}_{n,\tau}$, i.e., no matrix M of type τ.*

For a set $\mathcal{T}$ of types, where each $\tau \sim \mathbb{F}_q^b$ for some $b \in \mathbb{N}$, we say that $\mathcal{C}$ is $\mathcal{T}$-free *if it is τ-free for all $\tau \in \mathcal{T}$. We refer to $\mathcal{T}$-freeness as a* b-local property *of codes.*

For a more in-depth discussion of the definition, we refer the reader to, [28, Section 2] or [30, Chapter 3]. To provide some intuition, we demonstrate how (ρ, ℓ, L)-list-recoverability may be described as an L-local property. We define $\mathcal{T}$ to be the set of all types $\tau \sim \mathbb{F}_q^L$ such that for some (correlated) distribution $\nu \sim \binom{\mathbb{F}_q}{\ell}$,

$$\forall i \in [L], \quad \Pr_{(\vec{\boldsymbol{u}},\boldsymbol{S})\sim(\tau,\nu)}[\boldsymbol{u}_i \notin \boldsymbol{S}] \le \rho \tag{1}$$

and furthemore we require

$$\forall 1 \le i < j \le L, \quad \Pr_{\vec{\boldsymbol{u}}\sim\tau}[\boldsymbol{u}_i \neq \boldsymbol{u}_j] > 0 \ .$$

(This second condition amounts to requiring that any matrix of type τ has distinct columns.) We refer to the collection of all these types as $\mathcal{T}_{\rho,\ell,L}$.

We now characterize (up to $o(1)$ terms) the threshold rate of a property.

▶ **Theorem 14** ([30], Theorem 3.3.9: Thresholds for Random Linear Codes)**.** *Fix $b \in \mathbb{N}$ and let $\mathcal{T}$ be a set of b-local types. Then the threshold rate for $\mathcal{T}$-freeness is*

$$1 - \max_{\tau\in\mathcal{T}} \min_{A} \left\{ \frac{H_q(A\tau)}{\dim(A\tau)} \right\} \pm o_{n\to\infty}(1) \ , \tag{2}$$

where the minimum is taken over all surjective linear maps $A : \mathbb{F}_q^b \to \mathbb{F}_q^c$ with $c \le b$.

Let us specialize to the case of τ-freeness for a single type τ. Suppose that $R > 1 - \min_A \left\{ \frac{H_q(A\tau)}{\dim(A\tau)} \right\}$. Theorem 14 tells us that it is unlikely that a RLC of rate R is τ-free. Stated differently, we can expect that there is at least one matrix of type τ contained in such an RLC. In fact, while we do not prove this, it is in fact likely that there will be *many* such matrices. For this reason, we use the following terminology for types τ satisfying $R > 1 - \min_A \left\{ \frac{H_q(A\tau)}{\dim(A\tau)} \right\}$: we call them *abundant*.

In proving an upper bound R_{upper} on the threshold rate for a property of interest (e.g., (ρ, ℓ, L)-list-recovery), we will follow the following steps. First, we define an appropriate set type τ and prove that a code satisfies the property of interest only if it is τ-free. Informally, we refer to this as a proof that τ is *bad* for the property of interest. Next, we show that for RLCs of rate R_{upper}, the type τ is abundant. This is the more challenging part of the theorem, as the minimization over the set of all linear maps A is quite challenging to control. Nonetheless, we are able to carry out this program for (ρ, ℓ, L)-list-recovery, as advertised.

In proving a lower bound on R_{low} on the threshold rate for a property of interest (e.g., $(\rho, 3)$-list-decoding), we need to consider *any* type that is bad for list-decoding, and then show that it is *implicitly rare*: that is, for some matrix A, there are relatively few matrices of type $A\tau$, and hence it is likely no matrix of that type lies in the RLC. That is, we must upper bound the ratio of the entropy of $A\tau$ with the dimension of $A\tau$. Here, we have the freedom to choose A, but the argument must apply to all types τ. This is especially tricky when given a type τ whose support is contained in a strict subspace, as then the bound on the entropy must be commensurately smaller. It is for this reason that we only consider small values of L, as one suffers from a combinatorial explosion in the number of possible support spaces for the types.

Thresholds for Random Codes. For thresholds of random codes, the characterization theorem is simpler in the sense that we do not have to minimize over compressive mappings, at least if the property satisfies certain technical conditions. Fortunately, the characterization applies to list-recoverability, and hence also list-decodability.

▶ **Theorem 15** ([17], Theorem 2: Thresholds for Random Codes)**.** *Let $b \in \mathbb{N}$ and let $\mathcal{T}$ be a set of b-local types. Let T be a convex approximation for $\mathcal{T}$. Then the threshold rate for $\mathcal{T}$-freeness is*

$$1 - \frac{\max_{\tau \in T} H_q(\tau)}{b} \ .$$

▶ **Proposition 16** ([17], Lemma 1)**.** *$\mathcal{T}_{\rho,\ell,L}$ is a convex approximation for the property of (ρ, ℓ, L)-list-recoverability.*

3 Lower Bound on List-Size for List-Recovery

Througout this section, the following notations are fixed:

- $q \in \mathbb{N}$ is a (fixed) prime power;[10]
- $\ell \in \mathbb{N}$ satisfies $1 \leq \ell < q$;
- $\rho \in \mathbb{R}$ satisfies $0 < \rho < 1 - \frac{\ell}{q}$; and
- $\delta > 0$ is a small constant.

All these parameters are constants, independent of the growing parameter n. Our main result in this section is the following theorem.

▶ **Theorem 17.** *There exists $\varepsilon_{q,\ell,\rho,\delta} > 0$ such that for all $0 < \varepsilon < \varepsilon_{q,\ell,\rho,\delta}$ and n sufficiently large, a random linear code in $\mathbb{F}_q^n$ of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ is not $\left(\rho, \ell, \lfloor \frac{\log_q \binom{q}{\ell} - (1 - h_{q,\ell}(\rho))}{\varepsilon} - \delta \rfloor\right)$-list-recoverable with probability $1 - o(1)$.*

[10] When we discuss random codes, q may be any positive integer.

The proof of this theorem follows the same outline as has been used in, e.g., [15]. Namely, we begin by defining a L-local type which we show is *bad* for (ρ, ℓ, L)-list-recovery. Later, we prove that the type is indeed *abundant*, which is the more challenging part of the theorem.

The bad L-local type is defined as follows.

▶ **Definition 18** (The bad type for (ρ, ℓ, L)-list-recoverability)**.** *Fix $L \in \mathbb{N}$. Define the distribution $\tau \sim \mathbb{F}_q^L$ via the following procedure for sampling a random vector $\vec{\boldsymbol{u}} = (\boldsymbol{u}_1, \dots, \boldsymbol{u}_L)$:*

- *First, $\boldsymbol{S} \sim \binom{\mathbb{F}_q}{\ell}$ is sampled uniformly at random;*
- *Second, for $i = 1, \dots, L$, we sample $\boldsymbol{u}_i \sim \mathbb{F}_q$ as*

$$\Pr[\boldsymbol{u}_i = x | \boldsymbol{S} = S] = \begin{cases} \frac{1-\rho}{\ell} & \text{if } x \in S \\ \frac{\rho}{q-\ell} & \text{if } x \notin S \end{cases},$$

and conditioned on $\boldsymbol{S} = S$, the coordinates $\boldsymbol{u}_1, \dots, \boldsymbol{u}_L$ are independent*.*

Note that such a type does indeed lie in the set $\mathcal{T}_{\rho,\ell,L}$. Indeed, if $\nu \sim \boldsymbol{S}$ we clearly have

$$\forall i \in [L], \quad \Pr_{(\vec{\boldsymbol{u}}, \boldsymbol{S}) \sim (\tau, \nu)}[\boldsymbol{u}_i \notin \boldsymbol{S}] = \rho$$

and we also readily have $\Pr_{\vec{\boldsymbol{u}} \sim \tau}[\boldsymbol{u}_i \neq \boldsymbol{u}_j] > 0$. From [17], we conclude that τ is bad for (ρ, ℓ, L)-list-recovery.

We now claim that the type τ is indeed abundant, i.e., that it has sufficiently large (relative) entropy. This is the more technical part of the proof, and its proof is deferred to the full version.

▶ **Lemma 19.** *There exists an integer $L_{\rho,q,\ell,\delta}$ such that for all integers $L \geq L_{\rho,q,\ell,\delta}$, the following holds. Let $\vec{\boldsymbol{u}} \sim \tau$, and let $A \in \mathbb{F}_q^{L' \times L}$ with $L' \leq L$ and $\operatorname{rank}(A) = L'$. Then*

$$H_q(A\vec{\boldsymbol{u}}) \geq L' \cdot h_{q,\ell}(\rho) + \log_q \binom{q}{\ell} - 1 + h_{q,\ell}(\rho) - \delta \geq L' \cdot \left(h_{q,\ell}(\rho) + \frac{\log_q \binom{q}{\ell} - 1 + h_{q,\ell}(\rho) - \delta}{L} \right).$$

Assuming Lemma 19, we now show that this does indeed yield our target Theorem 17.

Proof of Theorem 17. Let $L_{\rho,q,\ell,\delta/2}$ be the promised constant from Lemma 19, and choose $\varepsilon_{q,\ell,\rho,\delta} := \frac{\log_q \binom{q}{\ell} - 1 + h_{q,\ell}(\rho)}{L_{\rho,q,\ell,\delta/2}+1}$. Let $\varepsilon < \varepsilon_{q,\ell,\rho,\delta}$. Let $L = \left\lfloor \frac{\log_q \binom{q}{\ell} - 1 + h_{q,\ell}(\rho)}{\varepsilon} - \delta \right\rfloor$, and define τ as in Definition 18 with this choice of L.

By Lemma 19, as $L \geq L_{\rho,q,\ell,\delta/2}$ we have that for all surjective linear maps $A : \mathbb{F}_q^L \to \mathbb{F}_q^{L'}$

$$\frac{H_q(A\tau)}{L'} \geq h_{q,\ell}(\rho) + \frac{\log_q \binom{q}{\ell} - 1 + h_{q,\ell}(\rho) - \delta/2}{L}.$$

We note further that as τ has full support the same is true for $A\tau$, i.e., $\dim(A\tau) = L'$. Thus, by Theorem 14 we have that the threshold rate for τ-freeness is at most

$$1 - h_{q,\ell}(\rho) - \frac{\log_q \binom{q}{\ell} - 1 + h_{q,\ell}(\rho) - \delta/2}{L} - o_{n\to\infty} < 1 - h_{q,\ell}(\rho) - \varepsilon,$$

where the last inequality holds for large enough n. In other words, a random linear code of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ contains a matrix $M \in \mathcal{M}_{n,\tau}$ with probability $1 - o(1)$. As we know that a code $\mathcal{C}$ which contains a matrix of type τ is not (ρ, ℓ, L)-list-recoverable, our theorem is proved. ◀

3.1 List-recoverability lower bound for random codes

For context, we provide nearly matching upper and lower bounds for list-recovery for *uniformly* random codes. There is a similar result for list-recovery provided in [17], but it is not optimized for the case of capacity-approaching codes.

▶ **Theorem 20.** *There exists $\varepsilon_{q,\ell,\rho,\delta}$ such that for all $0 < \varepsilon < \varepsilon_{q,\ell,\rho,\delta}$ and n sufficiently large, a random code in $\mathbb{F}_q^n$ of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ is not $\left(\rho, \ell, \lfloor \frac{\log_q \binom{q}{\ell}}{\varepsilon} - \delta \rfloor\right)$-list-recoverable.*

On the other hand, for any $\varepsilon > 0$ and n sufficiently large, a random code in $\mathbb{F}_q^n$ of rate $1 - h_{q,\ell}(\rho) - \varepsilon$ is $\left(\rho, \ell, \lceil \frac{\log_q \binom{q}{\ell}}{\varepsilon} + 1 \rceil\right)$-list-recoverable.

In this way, we can essentially pin-down the list size of a rate $1 - h_{q,\ell}(\rho) - \varepsilon$ random code to one of three possible values. This is similar to the result on the list-decodability of binary random linear codes from [15]. Again, the proof is deferred to the full version.

4 List-Decoding with Small Lists

In this section, we investigate the list-decodability of random codes and random linear codes with constant list size. Specifically, for list-of-3 decoding over the binary field, we can show that the threshold rate for list-decoding of random linear codes is strictly better than that for list-decoding uniformly random codes. Further, for larger field sizes we are able to show that the threshold rate for list-of-2 decoding over $\mathbb{F}_q$ is strictly better for random linear codes than for uniformly random codes. This extends the result of [17] which only applies to list-of-2 decoding for binary codes.

For our lower bound on the threshold rates for RLCs, we follow the following procedure. First, we consider any type that is bad for, e.g., $(\rho, 3)$-list-decoding, i.e., a type from $\mathcal{T}_{\rho,1,3}$. For any such type τ, we upper bound $\frac{H_q(A\tau)}{\dim(A\tau)}$ for the linear map A sending $(x_1, x_2, x_3) \mapsto (x_1 - x_3, x_2 - x_3)$. This is straightforward when the $\dim(A\tau)$ is full (requiring essentially only the concavity of the entropy function); when it is smaller, more careful reasoning is required. For space reasons, all the proofs of this section are deferred to the full version.

As a final contribution, we recall that in [15] it is shown that over the binary field the threshold rate for random linear codes is strictly better than random codes in the capacity-approaching regime. We observe that their techniques can be extended to show that such a trend holds for any constant list size L (assuming the decoding radius ρ is not too large). To do this, we first prove a lower bound on the threshold rate of binary random linear codes by applying the argument in [26] and an upper bound on the threshold rate of binary random codes following the argument in [15]. Although our proof resorts to known techniques, such results were not stated before and greatly strengthen our belief that random linear codes perform better than random codes. In light of the available evidence, a reasonable conjecture would be that the for all alphabet sizes, the threshold rate of random linear codes is strictly better than that of random codes.

4.1 List-of-3 Decoding for Binary Alphabet

In this section, we study the threshold rate for list-of-3 decoding binary codes. We recall that the Plotkin point for list-of-3 decoding binary codes, i.e., the maximum value of ρ for which $(\rho, 4)$-list-decoding with positive rate is possible, is $5/16$ [1]. Our main theorem is the following:

▶ **Theorem 21.** *Let $\rho \in (0, 5/16)$. The threshold rate for $(\rho, 4)$-list-decoding a random linear code over $\mathbb{F}_2$ is at least*

$$1 - \max\left\{ \frac{H_2(x_1, x_2) + 2x_1 + x_2 \log_2 3}{3} : x_1 + 2x_2 \leq 4\rho, x_1 + x_2 \leq 1, x_1, x_2 \geq 0 \right\}.$$

Next, for context, we consider the threshold rate for $(\rho, 4)$-list-decoding uniformly random codes.

▶ **Theorem 22.** *Let $\rho \in (0, 5/16)$. The threshold rate for $(\rho, 4)$-list decoding random code over $\{0, 1\}$ is*

$$1 - \max\left\{ \frac{1 + H_2(x_1, x_2) + 2x_1 + x_2 \log_2 3}{4} : x_1 + 2x_2 \leq 4\rho, x_1 + x_2 \leq 1, x_1, x_2 \geq 0 \right\}.$$

As $\frac{1+F}{4} \geq \frac{F}{3}$ for all $F \leq 3$, the lower bound on the threshold rate provided by Theorem 21 is greater than the exact value from Theorem 22. This demonstrates that random linear codes do indeed perform better.

4.2 List-of-2 Decoding for Arbitrary Alphabets

We now study list-of-2 decoding over $\mathbb{F}_q$ for $q \geq 3$. Here, the Plotkin point is to the best of our knowledge unknown, and we just prove our result for $\rho < 1/3$.

▶ **Theorem 23.** *Let $\rho \in (0, 1/3)$. The threshold rate for $(\rho, 3)$-list decoding random linear code over $\mathbb{F}_q$ with $q \geq 3$ is at least*

$$1 - \max\left\{ \frac{H_q(x_1, x_2) + x_1 \log_q 3(q-1) + x_2 \log_q (q-1)(q-2)}{2} : x_1 + 2x_2 \leq 3\rho, x_1 + x_2 \leq 1, x_1, x_2 \geq 0 \right\}.$$

For context, we again consider random codes.

▶ **Theorem 24.** *Let $\rho \in (0, 1/3)$. The threshold rate for $(\rho, 3)$-list decoding random code over $\mathbb{F}_q$ is*

$$1 - \max\left\{ \frac{1 + H_q(x_1, x_2) + x_1 \log_q 3(q-1) + x_2 \log_q (q-1)(q-2)}{3} : x_1 + 2x_2 \leq 3\rho, x_1 + x_2 \leq 1, x_1, x_2 \geq 0 \right\}.$$

Again, by noting $\frac{1+F}{3} \geq \frac{F}{2}$ for all $F \leq 2$, we conclude that random linear codes do indeed perform better: the lower bound on the threshold rate furnished by Theorem 23 is strictly greater than the exact threshold rate of Theorem 24.

4.3 List Decoding for Binary Alphabets with Larger Lists

In this subsection, we observe that the list-decodability of random linear codes is better than random codes over the binary field for any list size L.

We begin by stating our possibility result for random linear codes. The proof is an adaptation of the argument from [10, 26] which we omit due to the space limit.

▶ **Theorem 25.** *For any fixed list size L and $\delta > 0$, a random linear code over the binary field of rate $1 - h_2(\rho) - \frac{h_2(\rho)}{L-1-2\delta} - \delta$ is (ρ, L)-list decodable with probability $1 - 2^{-\Omega_{\delta,L}(n)}$.*

Next, we provide an upper bound on the list size of a random code. The proof, which appears in the full version, uses the threshold framework.

▶ **Theorem 26.** *Let L be a fixed constant list size and δ be any positive constant. With high probability, a random code with rate $\frac{L-1}{L}(1-h_2(\rho)) - \frac{h_2(2\rho-2\rho^2)-h_2(\rho)}{L} + \delta$ is not (ρ, L)-list decodable.*

From these two theorems, we note the following. If we let δ tend to 0, the upper bound provided by Theorem 26 is smaller than that provided by Theorem 25 as $(3+\frac{1}{L-1})h_2(\rho) - h_2(2\rho-2\rho^2) < 1$, assuming ρ is not too large.

References

1 Noga Alon, Boris Bukh, and Yury Polyanskiy. List-decodable zero-rate codes. *IEEE Transactions on Information Theory*, 65(3):1657–1667, 2018.

2 Laszlo Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. Bpp has weak subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3(4):307–318, 1990.

3 Vladimir M Blinovsky. Code bounds for multiple packings over a nonbinary finite alphabet. *Problems of Information Transmission*, 41(1):23–32, 2005.

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

5 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 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 432–442, 2013. doi:10.1137/1.9781611973105.31.

6 Peter Elias. List decoding for noisy channels. *Wescon Convention Record, Part 2*, pages 94–104, 1957.

7 Anna C Gilbert, Hung Q Ngo, Ely Porat, Atri Rudra, and Martin J Strauss. l2/l2-foreach sparse recovery with low risk. In *International Colloquium on Automata, Languages, and Programming*, pages 461–472. Springer, 2013.

8 Oded Goldreich and Leonid A Levin. A hard-core predicate for all one-way functions. In *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)*, pages 25–32. ACM, 1989.

9 Venkatesan Guruswami, Johan Håstad, and Swastik Kopparty. On the list-decodability of random linear codes. *IEEE Trans. Information Theory*, 57(2):718–725, 2011. doi:10.1109/TIT.2010.2095170.

10 Venkatesan Guruswami, Johan Håstad, Madhu Sudan, and David Zuckerman. Combinatorial bounds for list decoding. *IEEE Trans. Information Theory*, 48(5):1021–1034, 2002. doi:10.1109/18.995539.

11 Venkatesan Guruswami and Piotr Indyk. Expander-based constructions of efficiently decodable codes. In *42nd Annual Symposium on Foundations of Computer Science, FOCS 2001, 14-17 October 2001, Las Vegas, Nevada, USA*, pages 658–667, 2001. doi:10.1109/SFCS.2001.959942.

12 Venkatesan Guruswami and Piotr Indyk. Near-optimal linear-time codes for unique decoding and new list-decodable codes over smaller alphabets. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 812–821, 2002.

13 Venkatesan Guruswami and Piotr Indyk. Linear time encodable and list decodable codes. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 126–135, 2003.

14 Venkatesan Guruswami and Piotr Indyk. Efficiently decodable codes meeting gilbert-varshamov bound for low rates. In *SODA*, volume 4, pages 756–757. Citeseer, 2004.

15 Venkatesan Guruswami, Ray Li, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters. Bounds for list-decoding and list-recovery of random linear codes. *IEEE Transactions on Information Theory*, 2021.

16 Venkatesan Guruswami and Jonathan Mosheiff. Punctured large distance codes, and many reed-solomon codes, achieve list-decoding capacity. *arXiv preprint*, 2021. arXiv:2109.11725.

17 Venkatesan Guruswami, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters. Threshold rates for properties of random codes. *IEEE Transactions on Information Theory*, 2021.

18 Venkatesan Guruswami and Srivatsan Narayanan. Combinatorial limitations of average-radius list-decoding. *IEEE Transactions on Information Theory*, 60(10):5827–5842, 2014.

19 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.

20 Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009.

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

22 Brett Hemenway, Noga Ron-Zewi, and Mary Wootters. Local list recovery of high-rate tensor codes & applications. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 204–215. IEEE, 2017.

23 Brett Hemenway and Mary Wootters. Linear-time list recovery of high-rate expander codes. *Information and Computation*, 261:202–218, 2018.

24 Piotr Indyk, Hung Q Ngo, and Atri Rudra. Efficiently decodable non-adaptive group testing. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 1126–1142. SIAM, 2010.

25 Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the Fourier spectrum. *SIAM Journal on Computing*, 22(6):1331–1348, 1993.

26 Ray Li and Mary Wootters. Improved list-decodability of random linear binary codes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

27 Richard J Lipton. Efficient checking of computations. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 207–215. Springer, 1990.

28 Jonathan Mosheiff, Nicolas Resch, Noga Ron-Zewi, Shashwat Silas, and Mary Wootters. Ldpc codes achieve list decoding capacity. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 458–469. IEEE, 2020.

29 Hung Q Ngo, Ely Porat, and Atri Rudra. Efficiently decodable error-correcting list disjunct matrices and applications. In *International Colloquium on Automata, Languages, and Programming*, pages 557–568. Springer, 2011.

30 Nicolas Resch. List-decodable codes: (randomized) constructions and applications. *School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep., CMU-CS-20-113*, 2020.

31 Atri Rudra and Mary Wootters. Every list-decodable code for high noise has abundant near-optimal rate puncturings. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 764–773. ACM, 2014.

32 Atri Rudra and Mary Wootters. Average-radius list-recovery of random linear codes. In *Proceedings of the 2018 ACM-SIAM Symposium on Discrete Algorithms, SODA*, 2018.

33 Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

34 Mary Wootters. On the list decodability of random linear codes with large error rates. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 853–860, 2013. doi:10.1145/2488608.2488716.

35 Jack Wozencraft. List decoding. *Quarter Progress Report*, 48:90–95, 1958.

36 Yihan Zhang, Amitalok J Budkuley, and Sidharth Jaggi. Generalized list decoding. In *2020 Information Theory and Applications Workshop (ITA)*, pages 51–1. IEEE, 2020.

37 Victor Vasilievich Zyablov and Mark Semenovich Pinsker. List concatenated decoding. *Problemy Peredachi Informatsii*, 17(4):29–33, 1981.