Zero-Rate Thresholds and New Capacity Bounds for List-Decoding and List-Recovery

Nicolas Resch
Informatics' Institute, University of Amsterdam, The Netherlands

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

Yihan Zhang
Institute of Science and Technology Austria, Klosterneuburg, Austria

Abstract

In this work we consider the list-decodability and list-recoverability of arbitrary q-ary codes, for all integer values of $q \geqslant 2$. A code is called $(p, L)_q$-list-decodable if every radius pn Hamming ball contains less than L codewords; $(p, \ell, L)_q$-list-recoverability is a generalization where we place radius pn Hamming balls on every point of a combinatorial rectangle with side length ℓ and again stipulate that there be less than L codewords.

Our main contribution is to precisely calculate the maximum value of p for which there exist infinite families of positive rate $(p, \ell, L)_q$-list-recoverable codes, the quantity we call the *zero-rate threshold*. Denoting this value by p_*, we in fact show that codes correcting a $p_* + \varepsilon$ fraction of errors must have size $O_\varepsilon(1)$, i.e., independent of n. Such a result is typically referred to as a "Plotkin bound." To complement this, a standard random code with expurgation construction shows that there exist positive rate codes correcting a $p_* - \varepsilon$ fraction of errors. We also follow a classical proof template (typically attributed to Elias and Bassalygo) to derive from the zero-rate threshold other tradeoffs between rate and decoding radius for list-decoding and list-recovery.

Technically, proving the Plotkin bound boils down to demonstrating the Schur convexity of a certain function defined on the q-simplex as well as the convexity of a univariate function derived from it. We remark that an earlier argument claimed similar results for q-ary list-decoding; however, we point out that this earlier proof is flawed.

2012 ACM Subject Classification Mathematics of computing → Coding theory

Keywords and phrases Coding theory, List-decoding, List-recovery, Zero-rate thresholds

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.99

Category Track A: Algorithms, Complexity and Games

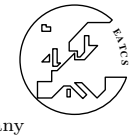
Related Version *Full Version*: `https://arxiv.org/abs/2210.07754` [46]

Funding *Nicolas Resch*: Research supported in part by ERC H2020 grant No.74079 (ALGSTRONGCRYPTO).
Chen Yuan: Research supported in part by the National Key Research and Development Projects under Grant 2022YFA1004900 and Grant 2021YFE0109900, the National Natural Science Foundation of China under Grant 12101403 and Grant 12031011.

Acknowledgements YZ is grateful to Shashank Vatedka, Diyuan Wu and Fengxing Zhu for inspiring discussions.

1 Introduction

Given a code $\mathcal{C} \subset [q]^n$, a fundamental problem of coding-theory is to determine how "well-spread" $\mathcal{C}$ can be if we also insist that $\mathcal{C}$ have large rate $R = \frac{\log_q |\mathcal{C}|}{n}$. The most basic way of quantifying "well-spread" is by insisting that all pairs of codewords are far apart. That is,



we hope that the minimum distance $d := \min\{d_{\mathrm{H}}(\boldsymbol{c}, \boldsymbol{c}') : \boldsymbol{c} \neq \boldsymbol{c}' \in \mathcal{C}\}$ is large, where $d_{\mathrm{H}}(\cdot, \cdot)$ denotes Hamming distance, i.e., the number of coordinates on which the two strings differ. Equivalently, given any word $\boldsymbol{y} \in [q]^n$, we have that $|\mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, r) \cap \mathcal{C}| \leqslant 1$, where $r = \lfloor d/2 \rfloor$ and $\mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, r) = \{\boldsymbol{x} \in [q]^n : d_{\mathrm{H}}(\boldsymbol{x}, \boldsymbol{y}) \leqslant r\}$ denotes the Hamming ball of radius r centered at $\boldsymbol{y}$.

One can naturally relax this requirement to the notion of list-decodability: instead of upper-bounding $|\mathcal{B}_{\mathrm{H}}(\boldsymbol{y}, r) \cap \mathcal{C}|$ by 1, we upper bound it by a larger integer $L - 1$.[1] Equivalently, if we place Hamming balls of radius r on each codeword of $\mathcal{C}$, no vector in $[q]^n$ is covered by L or more balls. If $\mathcal{C}$ satisfies this property we call it $(p, L)_q$-list-decodable. Initially introduced by Elias and Wozencraft in the 1950's [16, 50, 17], this relaxed notion of decoding has been intensively studied in recent years, in part motivated by purely coding-theoretic concerns, but also due to its connections with theoretical computer science more broadly [20, 3, 38, 37, 33, 47].

A further generalization of list-decoding is provided by *list-recoverability*. In this case, one considers tuples of input lists $\boldsymbol{\mathcal{Y}} = (\mathcal{Y}_1, \ldots, \mathcal{Y}_n)$ where each $\mathcal{Y}_i \subset [q]$ is of size at most ℓ, and the requirement is that the number of codewords $\boldsymbol{c}$ satisfying $|\{i \in [n] : c_i \notin \mathcal{Y}_i\}| \leqslant pn$ is at most $L - 1$. Such a code is deemed $(p, \ell, L)_q$-list-recoverable. Note that $(p, 1, L)_q$-list-recoverability is the same as $(p, L)_q$-list-decoding, demonstrating that list-recoverability is a more general notion. While it was originally defined as an abstraction required for the task of uniquely-/list-decoding concatenated codes [21, 22, 23, 24], it has since found myriad further applications in computer science more broadly, e.g., in cryptography [30, 31], randomness extraction [29], hardness amplification [14], group testing [32, 41], streaming algorithms [15], and beyond.

When it comes to list-decoding and list-recovery, the optimal tradeoff between decoding-radius p and rate R is well-understood if one is satisfied with list-sizes $L = O(1)$.[2] That is, there exist $(p, \ell, O(\ell/\varepsilon))_q$-list-recoverable codes of rate $1 - H_{q,\ell}(p) - \varepsilon$ where[3]

$$H_{q,\ell}(p) \coloneqq p \log_q \left(\frac{q - \ell}{p} \right) + (1 - p) \log_q \left(\frac{\ell}{1 - p} \right);$$

conversely, if the rate is at least $1 - H_{q,\ell}(p) + \varepsilon$ then it will not be list-recoverable for any $L = o\left(q^{\varepsilon n}\right)$ [44, Theorem 2.4.12]. (Note that setting $\ell = 1$ recovers the more well-known list-decoding capacity theorem.) While this already provides some "coarse-grained" information concerning the list-decodability/-recoverability of codes, it leaves many questions unanswered.

For example, one can ask about the maximum rate of a $(p, 3)_q$-list-decodable code. That is, what is the maximum rate of a code that never contains more than 2 points from a Hamming ball of radius pn? However, this question as stated appears be quite difficult to solve: any improvement for the special case of $L = 2$ and $q = 2$ would require improving either on the Gilbert-Varshamov bound [19, 48] (on the "possibility" side) or the linear programming bounds [49, 39, 13] (on the "impossibility" side). Unfortunately, despite decades of interest in this basic question hardly any asymptotic improvements on these bounds have been provided in the past fifty years.

Zero-rate thresholds for list-decoding and -recovery. We therefore begin by targeting a more modest question: what is the maximum $p_* = p_*(q, \ell, L)$ such that for any $p < p_*$ there exist infinite families of q-ary $(p, \ell, L)_q$-list-recoverable codes of positive rate? That is,

[1] We find it most convenient to let L denote 1 more than the list-size, which is admittedly nonstandard, but will make our computations much cleaner.

[2] Or indeed, if we insist on L just being subexponential.

[3] For $\ell = 1$, $H_{q,1}$ reduces to the q-ary entropy function denoted by H_q.

imagining the curve describing the achievable tradeoffs with the rate R on the y-axis and decoding radius p on the x-axis, instead of asking to describe this entire curve, we simply seek to determine the point where this curve crosses the x-axis (clearly, this curve is monotonically decreasing).

Over the binary alphabet, setting $\ell = 1$ and $L = 2$ in this question we recover a famous result of Plotkin [42]: the maximum fraction of errors that can be uniquely-decoded by an infinite family of positive rate binary codes is $1/4$. Over general q-ary alphabets, this value is similarly known to be $\frac{q-1}{2q}$ (folklore; see, e.g., [28, Theorem 4.4.1]). The value of $p_*(2, 1, L)$ has been computed by Blinovsky [5] for all L, and is known to be

$$p_*(2, 1, L) = \frac{1}{2} - \frac{\binom{2k}{k}}{2^{2k+1}} \text{ if } L = 2k \text{ or } L = 2k + 1.$$

While this expression is quite impenetrable at first glance, here is a natural probabilistic interpretation: given $x_1, \ldots, x_L \in \{0, 1\}$, let $\mathsf{pl}(x_1, \ldots, x_L)$ denote the number of times the more popular bit appears.[4] We then have

$$p_*(2, 1, L) = 1 - \frac{1}{L} \mathop{\mathbb{E}}_{(X_1,\cdots,X_L)\sim \mathrm{Bern}(1/2)^{\otimes L}} [\mathsf{pl}(X_1, \cdots, X_L)],$$

where the notation $(X_1, \cdots, X_L) \sim \mathrm{Bern}(1/2)^{\otimes L}$ denotes that L independent unbiased bits are sampled.

It is then not difficult to conjecture the value for $p_*(q, \ell, L)$: if $\mathsf{pl}_\ell(x_1, \ldots, x_L)$ denotes the top-ℓ-plurality value of $x_1, \ldots, x_L \in [q]$, i.e., $\mathsf{pl}_\ell(x_1, \ldots, x_L) = \max_{\Sigma\subseteq[q]:|\Sigma|=\ell} |\{i \in [L] : x_i \in \Sigma\}|$, then it should be that

$$p_*(q, \ell, L) = 1 - \frac{1}{L} \mathop{\mathbb{E}}_{(X_1,\cdots,X_L)\sim \mathrm{Unif}([q])^{\otimes L}} [\mathsf{pl}_\ell(X_1, \cdots, X_L)]. \tag{1}$$

This quantity is fairly natural: one can interpret it as the minimum radius of a list-recovery ball (i.e., a set of the form $\{\boldsymbol{v} \in [q]^n : v_i \in \mathcal{Y}_i \text{ for at least } (1-p)n \ i \in [n]\}$) that will contain L codewords in the "typical" case. For the case of $\ell = 1$, i.e., q-ary list-decoding, a proof is claimed in [6, 7]; however, as we outline in Section 3 this proof is flawed. In this work we provide a rigorous derivation of Equation (1) for all values of ℓ, L and q with $1 \leqslant \ell \leqslant q$.

More precisely, we obtain the following results:

- A proof that $(p, \ell, L)_q$-list-recoverable q-ary codes with $p > p_*(q, \ell, L)$ have *constant-size*, i.e., independent of n. This should be interpreted as a generalization of the Plotkin bound [42], which states that binary codes uniquely-decodable from a $1/4 + \varepsilon$ fraction of errors have size at most $O(1/\varepsilon)$. For this reason we call our result a "Plotkin bound for list-recovery."
- Adapting the Elias-Bassalygo argument [4], we subsequently derive upper bounds on the rate of $(p, \ell, L)_q$-list-recoverable q-ary codes when $p < p_*(q, \ell, L)$.
- To complement this, we show that there exist infinite families of positive rate q-ary codes that are $(p, \ell, L)_q$-list-recoverable whenever $p < p_*(q, \ell, L)$. We are therefore justified in calling $p_*(q, \ell, L)$ the zero-rate threshold for list-recovery.

We now describe our techniques in more detail.

[4] We use pl to stand for "plurality". However, we caution that this function does not output a most popular symbol (as is perhaps more in line with the standard meaning of plurality), but the number of $i \in [L]$ for which x_i equals a most popular symbol.

1.1 Our techniques

Schur convexity of the function $f_{q,L,\ell}$. Following prior work [6],[5] our task requires us to answer the following question. Consider the function on distributions P over the alphabet $[q]$ defined as

$$f_{q,L,\ell}(P) \coloneqq \mathop{\mathbb{E}}_{(X_1,\cdots,X_L)\sim P^{\otimes L}} \left[\mathsf{pl}_\ell(X_1,\cdots,X_L)\right].$$

Analogously to before, the notation $(X_1,\cdots,X_L)\sim P^{\otimes L}$ means that L independent samples are taken from the distribution P. A crucial ingredient for deriving the Plotkin bound is a demonstration that this function is minimized by the uniform distribution.

There is a well-studied class of functions on finite distributions with the property that they are minimized by the uniform distribution: *Schur convex* functions. These are the functions that are monotonically-increasing with respect to the *majorization*-ordering, which compares vectors of real numbers by first sorting the vectors in descending order and then checking to see if all the prefix sums of one vector is greater than or equal to the prefix sums of the other. The important detail for us is that the uniform vector $(1/q,\ldots,1/q)\in\mathbb{R}^q$, corresponding to the uniform distribution, is majorized by *every* other vector corresponding to a distribution over $[q]$.

To demonstrate the Schur convexity of this function, we use the Schur-Ostrowski criterion, which states that Schur-convexity is equivalent to the non-negativity of a certain expression involving partial derivatives. Showing that this expression is non-negative boils down to a combinatorial accounting game, where we can show that the positive contributions arising from certain terms exceed the negative contributions arising from others.

Convexity of the univariate function $g_{q,L,\ell}$. Another important technical ingredient that we need for the proof of the Plotkin bound is the convexity of the univariate function

$$g_{q,L,\ell}(w) \coloneqq f_{q,L,\ell}(P_{q,\ell,w}),$$

where the distribution $P_{q,\ell,w}=(p_1,\ldots,p_q)$ is defined as

$$p_i = \begin{cases} \frac{w}{q-\ell} & \text{if } i \leqslant q-\ell \\ \frac{1-w}{\ell} & \text{if } i \geqslant q-\ell+1 \end{cases}.$$

In order to show the function is convex, we prove the second derivative is non-negative. In differentiating, we use the expression for $g_{q,L,\ell}$ in terms of $f_{q,L,\ell}$ and apply the chain rule. Showing the resulting expression is positive is again a sort of combinatorial accounting game: we can show the positive terms contribute more than the negative terms.

Quite interestingly, for $\ell=1$ (i.e., the case relevant for list-decoding) we only prove the convexity of the function $f_{q,1,L}$ on the interval $[0,(q-1)/q]$. Fortunately, as we can also easily show that $g_{q,1,L}$ decreases on the interval $[0,(q-1)/q]$ and then increases on the interval $[(q-1)/q,1]$,[6] convexity of $f_{q,1,L}$ on $[0,(q-1)/q]$ suffices for our purposes. And indeed, this is not an artifact of the proof: Blinovsky had already observed that convexity of $f_{q,1,L}$ does not hold on the entire interval $[0,1]$ [6, 7]. However, for $\ell\geqslant 2$ we obtain that convexity of $f_{q,\ell,L}$ does indeed hold on the entire interval $[0,1]$. We note that the second derivative does behave qualitatively differently, so this is perhaps not too surprising in hindsight; we comment on this further in [46, Remark 5].

---

[5] In fact, [6] only considers list-decoding, so a slight adaptation of this argument is required for list-recovery.
[6] This is in fact an easy corollary of the Schur convexity of $f_{q,1,L}$.

Plotkin bound. Armed with these (Schur-)convexity results, we aim to prove a Plotkin bound for list-decoding/-recovery. That is, if a q-ary code is $(p, \ell, L)_q$-list-recoverable with $p \geqslant p_*(q, \ell, L) + \varepsilon$, how large can the code be? Following the template of the standard argument (although certain subtleties arise when generalizing to list-recovery), we can show that such a code must be of constant size, i.e., independent of n.

Informally, the argument begins with a "preprocessing step" that prunes away some (but, crucially, not too many) codewords and yields a more structured subcode that we can subsequently analyze. The codewords of this subcode are very "balanced" in the sense that all patterns of symbols appear with roughly the same frequency. In particular, every pattern of length t should appear roughly a $1/q^t$ fraction of the time (or the code is very "biased," in which case a separate argument bounds its size).

To analyze this subcode $\mathcal{C}'$ we apply a double-counting argument to the average radius (see [46, Definition 10]) to cover L-subsets (where for list-recoverability, this radius is measured via the distance to a tuple of input lists). The lower bound on this quantity follows quite naturally from the list-decodability/-recoverability of the code, together with the "balancedness" of the subcode. For the upper bound, we compute the radius of an L-subset in terms of the empirical distribution of a coordinate $k \in [n]$, i.e., each $x \in [q]$ is assigned probability mass $P_k(x) = \frac{1}{M} \sum_{\boldsymbol{x} \in \mathcal{C}'} \mathbb{1}\{x_k = x\}$. By the Schur convexity of the function $f_{q,L,\ell}$ and the convexity of the univariate function $g_{q,L,\ell}$, we can bound this in terms of a distribution placing total mass $w \leqslant \frac{q-\ell}{q}$ on the last ℓ elements of $[q]$ and mass $\frac{1-w}{q-\ell}$ on each of the others. The result then follows.

We remark that, due to our use of Ramsey-theoretic arguments, the precise bound we obtain on the code size is quite poor. We have made no effort to optimize this constant. However, we do believe it would be interesting to improve this bound; we discuss this further in Section 4.

Elias-Bassalygo-style bound. After deriving this Plotkin bound, a well-known argument template (typically attributed to Elias and Bassalygo [4]) allows one to derive more general tradeoffs between the rate R and the noise-resilience parameters $(p, \ell, L)_q$. Informally, this proceeds by covering the space $[q]^n$ by a bounded number of list-recovery balls. The radius of these balls is carefully chosen to allow one to apply the Plotkin bound to the subcodes obtained by taking the intersection of the code with these balls. On the other hand, the number of list-recovery balls needed to cover $[q]^n$, known as the covering number, can be sharply estimated. From the above two bounds (the Plotkin bound and the covering number), a bound on the size of the whole code can be derived.

Possibility result: random code with expurgation. To complement the Plotkin bound, we show that if the decoding radius p is less than $p_*(q, \ell, L)$ then there exist infinite families of $(p, \ell, L)_q$-list-recoverable q-ary codes. This justifies our "zero-rate threshold" terminology for $p_*(q, \ell, L)$. The argument is completely standard, obtained by sampling a random code and subsequently expurgating codewords to destroy all size-L lists that can fit into Hamming balls of radius np. In fact, the lower bound on achievable rate is derived from the exact large deviation exponent of a certain quantity known as the average radius (cf. [46, Definition 12]) of a tuple of random vectors. Therefore the bound holds under a stronger notion called average-radius list-recovery: namely, for any subset of L codewords $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_j$ and any tuple of input lists $(\mathcal{Y}_1, \ldots, \mathcal{Y}_n)$, we have

$$\sum_{j=1}^{L} |\{i \in [n] : x_{j,i} \notin \mathcal{Y}_i\}| > Lpn \ .$$

1.2 Discussion on related work

Lower bounds for small q and/or L. For the case of $(p,3)_2$-list-decoding, it was shown in [26, Theorem 6.1] that the *threshold rate*[7] of random binary *linear* codes equals

$$\frac{1}{2}(2 - H_2(3p) - 3p\log_2(3)). \tag{2}$$

The term *threshold* refers to the critical rate below which a random binary linear code is $(p,3)_2$-list-decodable with high probability and above which it is not with high probability. This result was recently extended to the following two cases [45]. For $(p,4)_2$-list-decoding, the threshold rate of random binary linear code is lower bounded by [45, Theorem 1.3]

$$\frac{1}{3}\min_{\substack{x_1,x_2\geqslant 0\\ x_1+2x_2\leqslant 4p\\ x_1+x_2\leqslant 1}} 3 - \eta_2(x_1,x_2) - 2x_1 - x_2\log_2(3). \tag{3}$$

Here we use the notation

$$\eta_q(x_1,\ldots,x_t) \coloneqq \sum_{i=1}^{t} x_i \log_q \frac{1}{x_i} + \left(1 - \sum_{i=1}^{t} x_i\right)\log_q \frac{1}{1-\sum_{i=1}^{t} x_i}$$

for a partial probability vector $(x_1,\ldots,x_t) \in \mathbb{R}^t_{\geqslant 0}$ satisfying $t \leqslant q$ and $x_1 + \cdots + x_t \leqslant 1$. Note that $\eta_2(x) = H_2(x)$, however, this is no longer the case for $q > 2$. Moreover, for $(p,3)_q$-list-decoding, [45, Theorem 1.5] showed that the threshold rate of random linear code is at least

$$\frac{1}{2}\min_{\substack{x_1,x_2\geqslant 0\\ x_1+2x_2\leqslant 3p\\ x_1+x_2\leqslant 1}} 2 - \eta_q(x_1,x_2) - x_1\log_q(3(q-1)) - x_2\log_q(q-1)(q-2). \tag{4}$$

Our general lower bound (cf. Theorem 14) for list-recovery (numerically) matches Equations (2)–(4) upon particularizing the parameters q,ℓ,L suitably. See Figures 1a–1c. It is possible to analytically prove this observation, though we do not pursue it in the current paper. The rationale underlying this phenomenon is that the threshold rate of random linear codes for list-recovery is expected to match the rate achieved by random codes with expurgation (with the notable exception of zero-error list-recovery [25]). This conjecture, in its full generality, remains unproved, although it is partially justified in several recent works [40, 25, 26, 45].

Hash codes. One may note that for $\ell \geqslant 2$, our upper and lower bounds typically exhibit a large gap even at $p = 0$. See Figures 1e–1h. We provide evidence below indicating that closing this gap is in general a rather challenging task and necessarily requires significantly new ideas. Let us focus on the vertical axis $p = 0$, known as *zero-error list-recovery*. We observe that some configurations of q,ℓ,L in this regime encode several longstanding open questions in combinatorics. Indeed, consider $\ell = q-1, L = q$. The $(0,q-1,q)_q$-list recoverability condition can then be written as: for any $\mathcal{Y}_1,\cdots,\mathcal{Y}_n \in \binom{[q]}{q-1}$,

$$|\{\boldsymbol{x}\in\mathcal{C} : |\{j\in[n] : x_j\notin\mathcal{Y}_j\}| = 0\}| \leqslant L-1,$$

[7] We warn the reader not not to confuse this concept with that of the zero-rate threshold.

i.e.,

$$|\{\boldsymbol{x} \in \mathcal{C} : \forall j \in [n],\, x_j \in \mathcal{Y}_j\}| \leqslant L - 1.$$

Taking the contrapositive, we note that this condition is further equivalent to: for any $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_L\} \in \binom{\mathcal{C}}{L}$, there exists $j \in [n]$ such that $|\{x_{1,j}, \cdots, x_{L,j}\}| = q$. In words, for any q-tuple of codewords in a $(0, q-1, q)_q$-list-recoverable code, there must exist one coordinate such that the corresponding q-ary symbols in the tuple are all distinct. Such a code is also known as a *q-hashing* in combinatorics. It is well-known [18, 35] that a probabilistic construction yields such codes of rate[8] at least

$$C_{(0,q-1,q)_q} \geqslant \frac{1}{q-1} \log_q \frac{1}{1 - \frac{q!}{q^q}}. \tag{5}$$

In the same paper [18] also proved an upper bound

$$C_{(0,q-1,q)_q} \leqslant \frac{q!}{q^{q-1}} \log_q(2). \tag{6}$$

Another upper bound

$$C_{(0,q-1,q)_q} \leqslant \log_q \frac{q}{q-1} \tag{7}$$

can be proved using either a double-counting argument (a.k.a. first moment method), or (hyper)graph entropy [35, 36, 34]. Equation (6) is much better than Equation (7) for $q \geqslant 4$. However, the latter bound $\log_3 \frac{3}{2}$ remains the best known for $q = 3$ (called the *trifference problem* by Körner). For larger q, both lower [51] and upper bounds [2, 11, 27, 10, 12] can be improved. However, improving the bound for $q = 3$ is recognized as a formidable challenge. We will show in [46, Remark 9] that our lower bound for list-recovery (cf. Theorem 14) recovers Equation (5) for q-hashing upon setting $\ell = q - 1, L = q$. Furthermore, our upper bound Theorem 16 recovers Equation (7) for q-hashing (cf. [46, Remark 7]).

A generalization of q-hashing known as *(q, L)-hashing* ($q \geqslant L$) can also be cast as zero-error list-recoverable codes with more general values of ℓ, L. Indeed, taking $L = \ell + 1$ and $\ell \leqslant q - 1$, we can write $(0, \ell, \ell + 1)_q$-list-recoverability alternatively as: for any $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_{\ell+1}\} \in \binom{\mathcal{C}}{\ell+1}$, there exists $j \in [n]$ such that $|\{x_{1,j}, \cdots, x_{\ell+1,j}\}| = \ell + 1$. This is in turn the precise definition of $(q, \ell + 1)$-hashing. It can be immediately seen that (q, q)-hashing is nothing but q-hashing. The upper and lower bounds in [18] also extend to $(q, \ell + 1)$-hashing and read as follows:

$$\frac{1}{\ell} \log_q \frac{1}{1 - \frac{\binom{q}{\ell+1}(\ell+1)!}{q^{\ell+1}}} \leqslant C_{(0,\ell,\ell+1)_q} \leqslant \frac{\binom{q}{\ell}\ell!}{q^\ell} \log_q(q - \ell + 1). \tag{8}$$

Our lower bound for list-recovery in Theorem 14 also recovers the above lower bound for $(q, \ell + 1)$-hashing by [18] upon setting $L = \ell + 1$ (see [46, Remark 8]). The upper bound was later improved in [36] for $q > L$ using the notion of hypergraph entropy:

$$C_{(0,\ell,\ell+1)_q} \leqslant \min_{0 \leqslant j \leqslant \ell-1} \frac{\binom{q}{j+1}(j+1)!}{q^{j+1}} \log_q \frac{q-j}{\ell-j}, \tag{9}$$

though it coincides with Equation (6) when $\ell = q - 1$. Some improved upper bounds in [27, 12] apply to $(q, \ell + 1)$-hashing as well. To the best of our knowledge, no improvement on lower bounds is known for $\ell < q - 1$.

[8] The bounds in [36, 18] are slightly adjusted so that they are consistent with our definition of code rate which adopts a $\log_q$ normalization (cf. [46, Definition 6]).

Zero-rate thresholds for general adversarial channels. The problem of locating the zero-rate threshold has been addressed in a much more general context [52]. The results in [52] on *general adversarial channel* model can be specialized to the list-recovery setting and read as follows. Given q, p, ℓ, L, define the *confusability set* $\mathcal{K}_{(p,\ell,L)_q}$ as the set of types[9] (cf. [46, Definition 15]) of all "confusable" L-tuple of codewords in the sense that they can fit into a certain list-recovery ball (cf. [46, Definition 3]) of radius np. Specifically,

$$\mathcal{K}_{(p,\ell,L)_q} \coloneqq \left\{ \sum_{\mathcal{Y}\in\binom{[q]}{\ell}} P_{X_1,\cdots,X_L,Y=\mathcal{Y}} \in \Delta\left([q]^L\right) : \begin{array}{c} P_{X_1,\cdots,X_L,Y} \in \Delta\left([q]^L \times \binom{[q]}{\ell}\right) \\ \forall i \in [L], \quad \sum\limits_{\substack{(x,\mathcal{Y})\in[q]\times\binom{[q]}{\ell} \\ x\notin\mathcal{Y}}} P_{X_i,Y}(x,\mathcal{Y}) \leqslant p \end{array} \right\}.$$

In the above definition, we use the notation $\sum_b P_{A,B=b}$ to denote the marginalization of $P_{A,B}$ onto the first variable A, and use $P_{X_i,Y}$ to denote the marginal of $P_{X_1,\cdots,X_i,Y}$ on (X_i, Y). It is not hard to verify that the confusability set is (i) "increasing" in p in the sense that $\mathcal{K}_{(p,\ell,L)_q} \subset \mathcal{K}_{(p',\ell,L)_q}$ if $p \leqslant p'$, and (ii) convex. Define also the convex cone of *completely positive (CP) tensors* of order L, i.e., tensors that can be written as a sum of *element-wise non-negative* rank-one tensors:

$$\mathsf{CP}_q^{\otimes L} \coloneqq \left\{ \sum_{i=1}^{k} \boldsymbol{p}_i^{\otimes L} \in (\mathbb{R}_{\geqslant 0}^q)^{\otimes L} : k \in \mathbb{Z}_{\geqslant 1}, (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_k) \in (\mathbb{R}_{\geqslant 0}^q)^k \right\}.$$

It is proved in [52] that the zero-rate threshold $p_*(q, \ell, L)$ can be expressed as the smallest p such that all completely positive distributions are confusable:

$$p_*(q, \ell, L) = \inf\left\{ p \in [0,1] : \mathsf{CP}_q^{\otimes L} \cap \Delta([q]^L) \subset \mathcal{K}_{(p,\ell,L)_q} \right\}. \tag{10}$$

The above characterization is *single-letter* in the sense that it is independent of the blocklength n. For q, ℓ, L independent of n (which is assumed to be the case in the current paper), the optimization problem on the RHS of Equation (10) can be solved in constant time. However, it does not immediately provide an explicit formula of $p_*(q, \ell, L)$ and analytically solving the optimization problem does not appear easy to the authors. On the other hand, the characterization $p_*(q, \ell, L) = 1 - \frac{1}{L}\mathbb{E}\left[\mathsf{pl}_\ell(X_1, \cdots, X_L)\right]$ (where the expectation is over $(X_1, \cdots, X_L) \sim \mathrm{Unif}([q])^{\otimes L}$, cf. Equation (1)) in this paper can be seen as the explicit solution to the optimization problem, though the way it is obtained is *not* by solving the latter problem per se. Instead, we prove the characterization from the first principle by leveraging specific structures of list-recovery. We hope that our characterization can shed light on the geometry of the high-dimensional polytopes – the confusability set and the set of CP distributions – involved in the characterization in Equation (10).

1.3 Organization

We state our main results in Section 2. We discuss the flaw in Blinovsky's proof in Section 3. We summarize our results and state open problems in Section 4. Additional notation, definitions, preliminary results and missing proofs can be found in [46].

[9] More precisely, the confusability set is the *closure* of the set of types of all confusable codeword tuples, since types are dense in distributions.

2 Main results

2.1 q-ary list-decoding

Define $f_{q,L}\colon \Delta([q]) \to \mathbb{R}_{\geqslant 0}$ as

$$f_{q,L}(P) := \mathop{\mathbb{E}}_{(X_1,\ldots,X_L)\sim P^{\otimes L}} [\mathsf{pl}(X_1,\ldots,X_L)] \tag{11}$$

for $P \in \Delta([q])$.

For $w \in [0,1]$, let $P_{q,w} \in \Delta([q])$ denote the following probability vector:

$$P_{q,w} := \left(\frac{w}{q-1},\ldots,\frac{w}{q-1},1-w\right). \tag{12}$$

Define $g_{q,L}\colon [0,1] \to \mathbb{R}_{\geqslant 0}$ as

$$g_{q,L}(w) := f_{q,L}(P_{q,w}). \tag{13}$$

▶ **Definition 1** (Majorization). *Let $\boldsymbol{a},\boldsymbol{b} \in \mathbb{R}^d$. Let $\boldsymbol{a}^{\downarrow},\boldsymbol{b}^{\downarrow} \in \mathbb{R}^d$ denote the vectors obtained by sorting the elements in $\boldsymbol{a}$ and $\boldsymbol{b}$ in descending order, respectively. We say that $\boldsymbol{a}$* majorizes *$\boldsymbol{b}$, written as $\boldsymbol{a} \succeq \boldsymbol{b}$, if*

$$\sum_{i=1}^{k} a_i^{\downarrow} \geqslant \sum_{i=1}^{k} b_i^{\downarrow}$$

for every $k \in [d]$, and

$$\sum_{i=1}^{d} a_i = \sum_{i=1}^{d} b_i.$$

▶ **Definition 2** (Schur convexity). *A function $f\colon \mathbb{R}^d \to \mathbb{R}$ is called* Schur-convex *if $f(\boldsymbol{x}) \geqslant f(\boldsymbol{y})$ for every $\boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^d$ such that $\boldsymbol{x} \succeq \boldsymbol{y}$ (in the sense of Definition 1).*

▶ **Theorem 3** (Schur convexity of $f_{q,L}$). *For any $q \in \mathbb{Z}_{\geqslant 2}$ and $L \in \mathbb{Z}_{\geqslant 2}$, the function $f_{q,L}\colon \Delta([q]) \to \mathbb{R}_{\geqslant 0}$ defined in Equation (11) is Schur convex.*

Proof. See [46, Sec. 4]. ◀

▶ **Theorem 4** (Convexity of $g_{q,L}$). *For any $q \in \mathbb{Z}_{\geqslant 2}$ and $L \in \mathbb{Z}_{\geqslant 2}$, the function $g_{q,L}\colon [0,1] \to \mathbb{R}_{\geqslant 0}$ defined in Equation (13) is convex in the interval $[0,(q-1)/q]$.*

Proof. See [46, Sec. 5]. ◀

▶ Remark 5. In the binary case (i.e., $q = 2$), understanding the functions $f_{2,L}$ and $g_{2,L}$ is an easier task. In fact, $f_{2,L}$ collapses to a univariate function and coincides with $g_{2,L}$. It can be computed [8, Eqn. (2.15) and (2.16)] that for $L = 2k, 2k+1$,

$$p_*(2,L;w) := 1 - \frac{1}{L} g_{2,L}(w) = \sum_{i=1}^{k} \frac{\binom{2i-2}{i-1}}{i} (w(1-w))^i,$$

and

$$\frac{\partial^2}{\partial w^2} p_*(2,L;w) = -k\binom{2k}{k}(w(1-w))^{k-1}.$$

The concavity (see also [43, Lemma 8]) and monotonicity of $p_*(2,L;w)$ immediately follow. Such explicit computation cannot be performed in the $q > 2$ case (and for list-recovery) and we have to work with summations like in [46, Lemma 14]. Other approaches to arguing monotonicity such as induction [1, Lemma 8(d)] do not seem to work well either for larger q.

As convexity only holds in the interval $[0,(q-1)/q]$, we will also require the following monotonicity properties, which follow easily from the Schur convexity of $f_{q,L}$.

▶ **Lemma 6.** *For any $q \in \mathbb{Z}_{\geqslant 2}$ and $L \in \mathbb{Z}_{\geqslant 2}$, the function $g_{q,L}\colon [0,1] \to \mathbb{R}_{\geqslant 0}$ defined in Equation (13) is non-increasing on $[0,(q-1)/q]$ and non-decreasing on $[(q-1)/q,1]$.*

Proof. See [46, Appendix B]. ◀

Define

$$p_*(q,L;w) \coloneqq 1 - \frac{1}{L} g_{q,L}(w). \tag{14}$$

▶ **Theorem 7** (Plotkin bound for q-ary list-decoding)**.** *Fix any $q \in \mathbb{Z}_{\geqslant 2}$ and $L \in \mathbb{Z}_{\geqslant 2}$. Let $\mathcal{C} \subset [q]^n$ be an arbitrary $(p,L)_q$-list-decodable code with $p = p_*\left(q,L;\frac{q-1}{q}\right) + \tau$ for any constant $\tau \in (0,1)$. Then there exists a constant $M_* = M_*(q,L,\tau)$ independent of n such that $|\mathcal{C}| \leqslant M_*$. As a consequence, in particular, we have*

$$p_*(q,L) \leqslant p_*\left(q,L;\frac{q-1}{q}\right) = 1 - \frac{1}{L} g_{q,L}\left(\frac{q-1}{q}\right).$$

Proof. The proof of this theorem can be found in [46, Sec. 6]. Specifically, a theorem (cf. [46, Theorem 16]) of the above kind will be first proved for *approximately constant-weight* codes in which all codewords have approximately the same Hamming weight. This theorem can then be used to prove Theorem 7 above (see [46, Corollary 18] for a more quantitative version) by partitioning a general (weight-unconstrained) code into a constant number of almost constant-weight subcodes. ◀

The upper bound on the zero-rate threshold in Theorem 7 is in fact sharp. It turns out that positive rate $(p,L)_q$-list-decodable codes exist for any p strictly smaller than the bound $1 - \frac{1}{L} g_{q,L}\left(\frac{q-1}{q}\right)$ in Theorem 7. Indeed, Blinovsky [6] proved the following lower bound on the $(p,L)_q$-list-decoding capacity which remains the best known to date. It can also be implied by our lower bound (Theorem 14 below) for list-recovery upon setting $\ell = 1$.

▶ **Theorem 8** ([6, Sec. 2])**.** *For any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$ and $0 \leqslant p < p_*\left(q,L;\frac{q-1}{q}\right)$, the following lower bound on the $(p,L)_q$-list-decoding capacity holds:*

$$C_{(p,L)_q} \geqslant \frac{L}{L-1} - \frac{1}{L-1}\left\{\lambda_* p + \log_q\left[\sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \exp_q\left(-\lambda_*\left(1 - \frac{1}{L}\max\{\boldsymbol{a}\}\right)\right)\right]\right\},$$

where $\lambda_* = \lambda_*(q,L,p)$ is the solution to the following equation

$$p = \frac{\sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \exp_q\left(-\lambda_*\left(1 - \frac{1}{L}\max\{\boldsymbol{a}\}\right)\right)\left(1 - \frac{1}{L}\max\{\boldsymbol{a}\}\right)}{\sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \exp_q\left(-\lambda_*\left(1 - \frac{1}{L}\max\{\boldsymbol{a}\}\right)\right)}.$$

Blinovsky's lower bound is plotted in Figure 1d. It is not hard to verify that the lower bound above vanishes at

$$p = q^{-L} \sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \left(1 - \frac{1}{L}\max\{\boldsymbol{a}\}\right),$$

and the corresponding λ_* equals 0.

Theorems 7 and 8 together pin down the exact value of $p_*(q,L)$ shown in the following corollary.

▶ **Corollary 9.** *For any $q \in \mathbb{Z}_{\geqslant 2}$ and $L \in \mathbb{Z}_{\geqslant 2}$, the zero-rate threshold $p_*(q, L)$ for $(p, L)_q$-list-decoding is given by*

$$p_*(q, L) = p_*\left(q, L; \frac{q-1}{q}\right) = 1 - \frac{1}{L} g_{q,L}\left(\frac{q-1}{q}\right) = q^{-L} \sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \left(1 - \frac{1}{L}\max\{\boldsymbol{a}\}\right). \tag{15}$$

From now on, we will use $p_*(q, L)$ to denote the RHS of Equation (15).

▶ **Theorem 10** (Elias–Bassalygo bound for q-ary list-decoding)**.** *Fix any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$ and $0 \leqslant p < p_*(q, L)$. Then the $(p, L)_q$-list-decoding capacity can be upper bounded as $C_{(p,L)_q} \leqslant 1 - H_q(w_{q,L})$ where $w_{q,L}$ is the solution to the equation $p_*(q, L; w) = p$ in $w \in [0, (q-1)/q]$.*

Proof. The above theorem is implied by [46, Theorem 19] proved in [46, Sec. 7]. The latter theorem shows that for any $(p, L)_q$-list-decodable code $\mathcal{C} \subset [q]^n$ with $p < p_*(q, L)$ and any sufficiently small constant $\tau > 0$, $|\mathcal{C}|$ is at most $B \cdot n^{1.5} \cdot q^{n(1-H_q(w_{q,L,\tau}))}$, where $B = B(q, L, \tau)$ is a constant and $w_{q,L,\tau}$ is the solution to $p_*(q, L; w) = p - \tau$. Taking $\tau \to 0$ and neglecting polynomial factors, we obtain the upper bound on the list-decoding capacity. ◀

The above upper bound is plotted in Figure 1d.

2.2 List-recovery

Define $f_{q,L,\ell} \colon \Delta([q]) \to \mathbb{R}_{\geqslant 0}$ as

$$f_{q,L,\ell}(P) \coloneqq \mathop{\mathbb{E}}_{(X_1,\ldots,X_L) \in P^{\otimes L}} [\mathsf{pl}_\ell(X_1, \ldots, X_L)] \tag{16}$$

for $P \in \Delta([q])$. Define $g_{q,L,\ell} \colon [0, 1] \to \mathbb{R}_{\geqslant 0}$ as

$$g_{q,L,\ell}(w) \coloneqq f_{q,L,\ell}(P_{q,\ell,w}), \tag{17}$$

where the distribution $P_{q,\ell,w} \in \Delta([q])$ is defined as

$$P_{q,\ell,w}(i) = \begin{cases} \frac{w}{q-\ell}, & 1 \leqslant i \leqslant q - \ell \\ \frac{1-w}{\ell}, & q - \ell + 1 \leqslant i \leqslant q \end{cases}. \tag{18}$$

▶ **Theorem 11** (Schur convexity of $f_{q,L,\ell}$)**.** *For any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$ and integer $1 \leqslant \ell \leqslant q-1$, the function $f_{q,L,\ell} \colon \Delta([q]) \to \mathbb{R}_{\geqslant 0}$ defined in Equation (16) is Schur convex.*

Proof. See [46, Sec. 8]. ◀

▶ **Theorem 12** (Convexity of $g_{q,L,\ell}$)**.** *For any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$ and integer $2 \leqslant \ell \leqslant q-1$, the function $g_{q,L,\ell} \colon \Delta([q]) \to \mathbb{R}_{\geqslant 0}$ defined in Equation (17) is convex in the interval $w \in [0, 1]$.*

Proof. See [46, Sec. 9]. ◀

Define

$$p_*(q, \ell, L; w) \coloneqq 1 - \frac{1}{L} g_{q,L,\ell}(w). \tag{19}$$

▶ **Theorem 13** (Plotkin bound for list-recovery). *Fix any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$ and integer $2 \leqslant \ell \leqslant q-1$. Let $\mathcal{C} \subset [q]^n$ be an arbitrary $(p, \ell, L)_q$-list-recoverable code with $p = p_* \left(q, \ell, L; \frac{q-\ell}{q}\right) + \tau$ for any constant $\tau \in (0,1)$. Then there exists a constant $M_* = M_*(q, \ell, \tau)$ independent of n such that $|\mathcal{C}| \leqslant M_*$. This implies, in particular,*

$$p_*(q, \ell, L) \leqslant p_* \left(q, \ell, L; \frac{q-\ell}{q}\right) = 1 - \frac{1}{L} g_{q,L,\ell} \left(\frac{q-\ell}{q}\right).$$

Proof. The proof structure is similar to that of Theorem 7. We first prove the analogous the statement for almost constant-weight codes (in which all codewords have approximately the same list-recovery weight) in [46, Theorem 20] and then pass to general codes by weight partitioning (cf. [46, Corollary 21]). Since the technical proofs bear many similarities to those in the list-decoding case, we only present proof sketches in [46, Sec. 10]. ◀

To complement Theorem 13, we prove in [46, Sec. 12] the following lower bound on the $(p, \ell, L)_q$-list-recovery capacity. To the best of our knowledge, this is the first bound for list-recovery with q, ℓ, L all being *constants* (independent of p and n). We believe that improving it likely requires novel techniques beyond expurgation.

▶ **Theorem 14.** *For any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$, integer $2 \leqslant \ell \leqslant q-1$ and $0 \leqslant p < p_* \left(q, \ell, L; \frac{q-\ell}{q}\right)$, the following lower bound on the $(p, \ell, L)_q$-list-recovery capacity holds:*

$$C_{(p,\ell,L)_q} \geqslant \frac{L}{L-1} - \frac{1}{L-1} \left\{ \lambda_* p + \log_q \left[\sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \exp_q \left(-\lambda_* \left(1 - \frac{1}{L} \max_\ell \{\boldsymbol{a}\} \right) \right) \right] \right\},$$

where $\lambda_* = \lambda_*(q, \ell, L, p)$ is the solution to the following equation

$$p = \frac{\sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \exp_q \left(-\lambda_* \left(1 - \frac{1}{L} \max_\ell \{\boldsymbol{a}\} \right) \right) \left(1 - \frac{1}{L} \max_\ell \{\boldsymbol{a}\} \right)}{\sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \exp_q \left(-\lambda_* \left(1 - \frac{1}{L} \max_\ell \{\boldsymbol{a}\} \right) \right)}.$$

Similar to the list-decoding case (Theorem 8), the above lower bound vanishes at

$$p = q^{-L} \sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \left(1 - \frac{1}{L} \max_\ell \{\boldsymbol{a}\} \right),$$

and the corresponding λ_* equals 0.

Theorems 13 and 14 jointly determine the value of $p_*(q, \ell, L)$ shown in the corollary below.

▶ **Corollary 15.** *For any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$ and integer $2 \leqslant \ell \leqslant q-1$, the zero-rate threshold $p_*(q, \ell, L)$ for $(p, \ell, L)_q$-list-recovery is given by*

$$\begin{aligned} p_*(q, \ell, L) &= p_* \left(q, \ell, L; \frac{q-\ell}{q}\right) = 1 - \frac{1}{L} g_{q,L,\ell} \left(\frac{q-\ell}{q}\right) \\ &= q^{-L} \sum_{\boldsymbol{a} \in \mathcal{A}_{q,L}} \binom{L}{\boldsymbol{a}} \left(1 - \frac{1}{L} \max_\ell \{\boldsymbol{a}\} \right). \end{aligned} \tag{20}$$

From now on, we use $p_*(q, \ell, L)$ to refer to the same quantity as the RHS of Equation (20).

▶ **Theorem 16** (Elias–Bassalygo bound for list-recovery). *Fix any $q \in \mathbb{Z}_{\geqslant 2}$, $L \in \mathbb{Z}_{\geqslant 2}$, integer $2 \leqslant \ell \leqslant q-1$ and $0 \leqslant p < p_*(q, \ell, L)$. Then the $(p, \ell, L)_q$-list-recovery capacity can be upper bounded as $C_{(p,\ell,L)_q} \leqslant 1 - H_{q,\ell}(w_{q,\ell,L})$ where $w_{q,\ell,L}$ is the solution to the equation $p_*(q, \ell, L; w) = p$ in $w \in [0, (q-\ell)/q]$.*

Proof. Parallel to Theorem 10, the above theorem is immediately implied by a finite-blocklength version [46, Theorem 22] (analogous to [46, Theorem 19]) whose full proof is presented in [46, Sec. 11]. ◀

3 Discussion of Blinovsky's results [6, 7]

As mentioned in Section 1, part of the motivation of this work is to fill in the gaps in the proofs in [6, 7] for q-ary list-decoding. We discuss in detail below the issues therein. The main result in [6] is a Plotkin bound (as our Theorem 7) for an arbitrary q-ary list-decodable code $\mathcal{C} \subset [q]^n$. For the sake of brevity, we assume in the proceeding discussion that $\mathcal{C}$ is w-constant weight. Additional bookkeeping is needed to handle small deviations in the weight, as we did in the proof of [46, Theorem 16]. The skeleton of the proof in [6] follows Blinovsky's proof in the *binary* case [5] which we adopt here as well: (i) pass to an (approximately) equi-coupled subcode $\mathcal{C}' = \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_M\} \subset \mathcal{C}$ using a Ramsey reduction; (ii) handle asymmetric coupling using Komlós's argument (and its order-L generalization [9]); (iii) prove an upper bound on the size M of the subcode $\mathcal{C}'$ using a double-counting argument. In completing the double-counting argument, one is required to upper bound the average radius (averaged over all L-lists in the subcode) by the zero-rate threshold $p_*(q, L; w) = 1 - \frac{1}{L} g_{q,L}(w)$:

$$\frac{1}{M^L} \sum_{(i_1,\cdots,i_L)\in[M]^L} \overline{\mathrm{rad}}(\boldsymbol{x}_{i_1}, \cdots, \boldsymbol{x}_{i_L}) = \sum_{k=1}^{n} \left(1 - \frac{1}{L} f_{q,L}(P_k)\right) \leqslant n \left(1 - \frac{1}{L} g_{q,L}(w)\right), \quad (21)$$

where $\overline{\mathrm{rad}}$ is defined in [46, Definition 10] and $P_k \in \Delta([q])$ is the empirical distribution of the k-th column of $\mathcal{C}' \in [q]^{M\times n}$. The equality in Equation (21) is by elementary algebraic manipulations (see [46, Eqn. (58)] for details). To show the inequality in Equation (21), we need the following properties of the functions $f_{q,L}$ and $g_{q,L}$:

1. For any $P = (p_1, \cdots, p_q) \in \Delta([q])$, we have $f_{q,L}(P) \geqslant g_{q,L}(1-p_q)$. In words, uniformizing P except one entry will only make $f_{g,L}$ no larger.
2. $g_{q,L}$ is convex as a univariate real-valued function on $[0, (q-1)/q]$.

If these properties hold, one can deduce [46, Eqn. (59) and (61)] from which Equation (21) follows. However, we observe that the proofs in [6, 7] for both properties above are problematic.

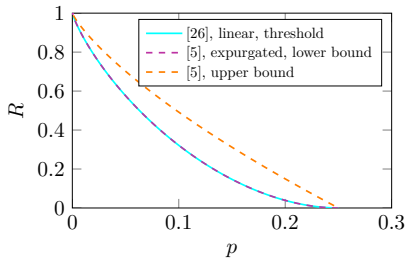
To show Item 1 above, the idea in [6] is to show instead monotonicity of $f_{q,L}$ under the so-called *Robin Hood operation* which averages two distinct entries of P. Specifically, [6] attempts to show

$$f_{q,L}(p_1, \cdots, p_i, \cdots, p_j, \cdots, p_q) \geqslant f_{q,L}\left(p_1, \cdots, \frac{p_i + p_j}{2}, \cdots, \frac{p_i + p_j}{2}, \cdots, p_q\right), \quad (22)$$

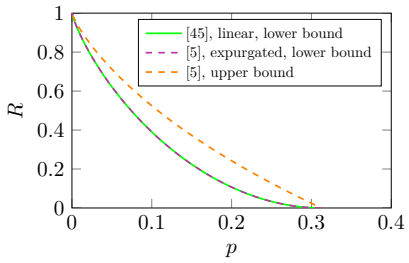
for any $1 \leqslant i < j \leqslant q$. This suffices since a sequence of Robin Hood operations can turn P into $P_{q,1-p_q}$ (defined in Equation (12)). [6] then proceeds to show Equation (22) by checking the derivative of a certain function related to the Robin Hood operation. Specifically, fix $(p_k)_{k\in[q]\setminus\{i,j\}}$ and assume $p_i + p_j = c$ (or equivalently $\sum_{k\in[q]\setminus\{i,j\}} p_i = 1 - c$) for some constant $0 \leqslant c \leqslant 1$. Consider the function $F_{q,L} \colon [0, c] \to \mathbb{R}$ defined as:

$$F_{q,L}(p) = f_{q,L}(p_1, \cdots, p, \cdots, c - p, \cdots, p_q),$$

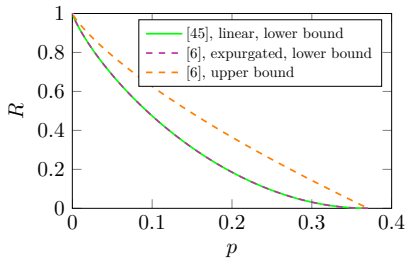
i.e., $f_{q,L}$ evaluated at P with $p_i = p, p_j = c - p$. The proof of Equation (22) is reduced to proving $F'_{q,L}(p) \leqslant 0$ for $p \in [0, c/2]$ and $F'_{q,L}(p) \geqslant 0$ for $p \in [c/2, c]$. If true, it implies that $f_{q,L}(P)$ is minimized at $p_i = p_j = c/2$ with fixed $(p_k)_{k\in[q]\setminus\{i,j\}}$. However, we note that the



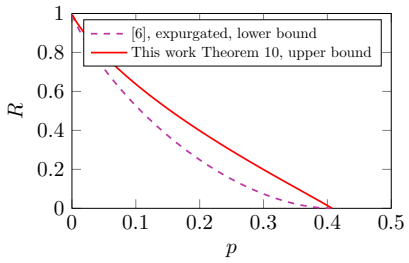
(a) $q = 2, \ell = 1, L = 3$ (list-decoding).



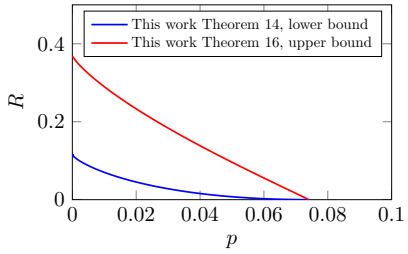
(b) $q = 2, \ell = 1, L = 4$ (list-decoding).



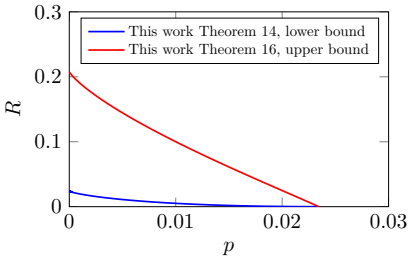
(c) $q = 3, \ell = 1, L = 3$ (list-decoding).



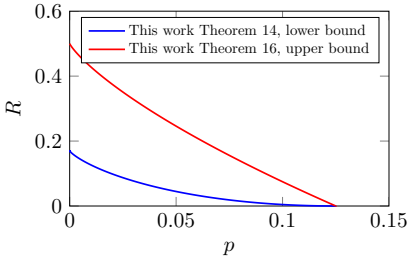
(d) $q = 3, \ell = 1, L = 4$ (list-decoding).



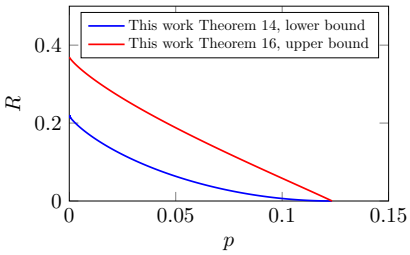
(e) $q = 3, \ell = 2, L = 3$ (list-recovery, 3-hashing).



(f) $q = 4, \ell = 3, L = 4$ (list-recovery, 4-hashing).



(g) $q = 4, \ell = 2, L = 3$ (list-recovery, $(4, 3)$-hashing).



(h) $q = 3, \ell = 2, L = 5$ (list-recovery).

Figure 1 Plots of upper and lower bounds in [5, 6, 26, 45] and this work for various values of $q \geqslant 2, 1 \leqslant \ell \leqslant q - 1, L \geqslant 2$.

expression of $F'_{q,L}(p)$ (see the second displayed equation on page 27 of [6]) is incorrect. Upon correcting it, we do not see an easy way to argue its non-positivity/-negativity. In particular, the claim in [6] that $F'_{q,L}(p)$, as a sum of multiple terms, is *term-wise* non-positive/-negative can be in general falsified by counterexamples.

The proof (attempt) of Item 2 is deferred to a subsequent paper [7]. The methodology thereof is similar to ours, i.e., verifying $g''_{q,L} \geqslant 0$. However, the expression of $g''_{q,L}$ in [7] is not exactly correct (see the first displayed equation on page 36 of [7] and compare it with ours in [46, Eqn. (34)][10]) and we have trouble verifying the case analysis of the values of $G(\cdot)$ (see [46, Eqn. (35)] in our notation, denoted by $\gamma(\cdot)$ in [7]) following that expression.

In contrast to Blinovsky's approach [6, 7], we deduce the monotonicity property of $f_{q,L}$ (cf. Item 1 above) from a stronger property: Schur convexity (cf. Theorem 3). Also, we believe that our proof of the convexity of $g_{q,L}$ (cf. Item 2 above) is cleaner, more transparent and easier to verify. Both results can be extended to list-recovery setting. Another advantage is that the monotonicity property of $g_{q,L}$ (specifically, $g_{q,L}$ is non-increasing in $[0, (q-1)/q]$ and non-decreasing in $[(q-1)/q, 1]$) which is needed in the proof of the Plotkin bound appears to be a simple consequence of the Schur convexity of $f_{q,L}$ (see Lemma 6). In [7], this is proved by checking the first derivative of $g_{q,L}$ which involves somewhat cumbersome calculations and case analysis.

4 Conclusion

In this work, we addressed the basic question of determining the maximum achievable decoding radius for positive rate list-recoverable codes, i.e., we pinned down the list-recovery zero-rate threshold. We then adapted known techniques to show that codes correcting more errors must in fact have *constant* size. Subsequently, we transferred this bound to give upper bounds on the rate of list-recoverable codes for all values of decoding radius.

As we apply general Ramsey-theoretic tools in bounding the size of list-recoverable codes in the zero-rate regime, our dependence on the corresponding parameters is quite poor, and indeed, we made no efforts to optimize these constants. However, for list-decodable binary codes in the zero-rate, a recent work of Alon, Bukh and Polyanskiy [1] derived new (and, in some cases, tight) upper bounds on their size. Obtaining similarly improved size upper bounds for q-ary list-decodable/-recoverable codes in the zero-rate regime therefore appears to be a natural next step.

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 Erdal Arikan. Upper bound on the zero-error list-coding capacity. *Information Theory, IEEE Transactions on*, 40:1237–1240, August 1994. doi:10.1109/18.335947.

3 László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complex.*, 3:307–318, 1993. doi:10.1007/BF01275486.

4 L. A. Bassalygo. New upper bounds for error-correcting codes. *Probl. of Info. Transm.*, 1:32–35, 1965.

[10] Note that the function considered in [7] is, in our notation, $1 - \frac{1}{L} g_{q,L}(w)$ instead of $g_{q,L}(w)$ per se as considered in Theorem 4.

5 Vladimir M Blinovsky. Bounds for codes in the case of list decoding of finite volume. *Problems of Information Transmission*, 22:7–19, 1986.

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

7 Vladimir M Blinovsky. On the convexity of one coding-theory function. *Problems of Information Transmission*, 44:34–39, 2008.

8 Volodia Blinovsky. *Asymptotic combinatorial coding theory*, volume 415 of *The Kluwer International Series in Engineering and Computer Science*. Kluwer Academic Publishers, Boston, MA, 1997. doi:10.1007/978-1-4615-6193-4.

9 Marco Bondaschi and Marco Dalai. A revisitation of low-rate bounds on the reliability function of discrete memoryless channels for list decoding. *IEEE Transactions on Information Theory*, 68(5):2829–2838, 2022. doi:10.1109/TIT.2022.3145318.

10 Simone Costa and Marco Dalai. New bounds for perfect k-hashing. *CoRR*, abs/2002.11025, 2020. arXiv:2002.11025.

11 M. Dalai, V. G. Carnegie, and J. Radhakrishnan. An improved bound on the zero-error list-decoding capacity of the 4/3 channel. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1658–1662, June 2017. doi:10.1109/ISIT.2017.8006811.

12 Stefano Della Fiore, Simone Costa, and Marco Dalai. Improved bounds for (b, k)-hashing. *IEEE Transactions on Information Theory*, 68(8):4983–4997, 2022. doi:10.1109/TIT.2022.3167608.

13 Philippe Delsarte. An algebraic approach to the association schemes of coding theory. *Philips Res. Rep. Suppl.*, 10:vi+–97, 1973.

14 Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Nearly optimal pseudorandomness from hardness. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1057–1068. IEEE, 2020.

15 Dean Doron and Mary Wootters. High-probability list-recovery, and applications to heavy hitters. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

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

17 Peter Elias. Error-correcting codes for list decoding. *IEEE Transactions on Information Theory*, 37(1):5–12, 1991.

18 M. Fredman and J. Komlós. On the size of separating systems and families of perfect hash functions. *SIAM Journal on Algebraic Discrete Methods*, 5(1):61–68, 1984. doi:10.1137/0605009.

19 Edgar N Gilbert. A comparison of signalling alphabets. *The Bell System Technical Journal*, 31(3):504–522, 1952.

20 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.

21 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.

22 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.

23 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.

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

25 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*, 68(2):923–939, 2022. doi:10.1109/TIT.2021.3127126.

26 Venkatesan Guruswami, Jonathan Mosheiff, Nicolas Resch, Shashwat Silas, and Mary Wootters. Threshold rates for properties of random codes. *IEEE Transactions on Information Theory*, 68(2):905–922, 2022. doi:10.1109/TIT.2021.3123497.

27 Venkatesan Guruswami and Andrii Riazanov. Beating Fredman-Komlós for Perfect k-Hashing. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132, pages 92:1–92:14, Dagstuhl, Germany, 2019. doi:10.4230/LIPIcs.ICALP.2019.92.

28 Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory, January 31, 2022. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/web-coding-book.pdf.

29 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.

30 Iftach Haitner, Yuval Ishai, Eran Omri, and Ronen Shaltiel. Parallel hashing via list recoverability. In *Annual Cryptology Conference*, pages 173–190. Springer, 2015.

31 Justin Holmgren, Alex Lombardi, and Ron D Rothblum. Fiat–shamir via list-recoverable codes (or: parallel repetition of gmw is not zero-knowledge). In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 750–760, 2021.

32 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.

33 Jeffrey C Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

34 J. Körner. Coding of an information source having ambiguous alphabet and the entropy of graphs. In *Transactions of the Sixth Prague Conference on Information Theory, Statistical Decision Functions, Random Processes (Tech Univ., Prague, 1971; dedicated to the memory of Antonín Špaček)*, pages 411–425. Academia, Prague, 1973.

35 J. Körner. Fredman-komlós bounds and information theory. *SIAM Journal on Algebraic Discrete Methods*, pages 560–570, 1986.

36 J. Körner and K. Marton. New bounds for perfect hashing via information theory. *European Journal of Combinatorics*, 9(6):523–530, 1988. doi:10.1016/S0195-6698(88)80048-9.

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

38 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.

39 Robert J. McEliece, Eugene R. Rodemich, Howard Rumsey, Jr., and Lloyd R. Welch. New upper bounds on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inform. Theory*, IT-23(2):157–166, 1977. doi:10.1109/tit.1977.1055688.

40 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*, pages 458–469. IEEE Computer Soc., Los Alamitos, CA, [2020] ©2020. doi:10.1109/FOCS46700.2020.00050.

41 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.

42 Morris Plotkin. Binary codes with specified minimum distance. *IRE Transactions on Information Theory*, 6(4):445–450, 1960.

43 Yury Polyanskiy. Upper bound on list-decoding radius of binary codes. *IEEE Transactions on Information Theory*, 62(3):1119–1128, 2016.

44 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.

45 Nicolas Resch and Chen Yuan. Threshold rates of code ensembles: Linear is best. In Mikolaj Bojanczyk, Emanuela Merelli, and David P. Woodruff, editors, *49th International Colloquium on Automata, Languages, and Programming, ICALP 2022, July 4-8, 2022, Paris, France*, volume 229 of *LIPIcs*, pages 104:1–104:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ICALP.2022.104.

46 Nicolas Resch, Chen Yuan, and Yihan Zhang. Zero-rate thresholds and new capacity bounds for list-decoding and list-recovery. *CoRR*, abs/2210.07754, 2022. doi:10.48550/arXiv.2210.07754.

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

48 RR Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741, 1957.

49 Lloyd R. Welch, Robert J. McEliece, and Howard Rumsey, Jr. A low-rate improvement on the Elias bound. *IEEE Trans. Inform. Theory*, IT-20:676–678, 1974. doi:10.1109/tit.1974.1055279.

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

51 Chaoping Xing and Chen Yuan. Beating the probabilistic lower bound on perfect hashing. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 33–41. SIAM, 2021. doi:10.1137/1.9781611976465.3.

52 Yihan Zhang, Amitalok J. Budkuley, and Sidharth Jaggi. Generalized List Decoding. In Thomas Vidick, editor, *11th Innovations in Theoretical Computer Science Conference (ITCS 2020)*, volume 151 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 51:1–51:83, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.ITCS.2020.51.