On List Recovery of High-Rate Tensor Codes

Swastik Kopparty
Department of Mathematics and Department of Computer Science, Rutgers University, NJ, USA
swastik.kopparty@gmail.com

Nicolas Resch
Department of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA
nresch@cs.cmu.edu

Noga Ron-Zewi
Department of Computer Science, University of Haifa, Israel
noga@cs.haifa.ac.il

Shubhangi Saraf
Department of Mathematics and Department of Computer Science, Rutgers University, NJ, USA
shubhangi.saraf@gmail.com

Shashwat Silas
Department of Computer Science, Stanford University, CA, USA
silas@stanford.edu

Abstract

We continue the study of list recovery properties of high-rate tensor codes, initiated by Hemenway, Ron-Zewi, and Wootters (FOCS'17). In that work it was shown that the tensor product of an efficient (poly-time) high-rate globally list recoverable code is *approximately* locally list recoverable, as well as globally list recoverable in *probabilistic* near-linear time. This was used in turn to give the first capacity-achieving list decodable codes with (1) local list decoding algorithms, and with (2) *probabilistic* near-linear time global list decoding algorithms. This also yielded constant-rate codes approaching the Gilbert-Varshamov bound with *probabilistic* near-linear time global unique decoding algorithms.

In the current work we obtain the following results:

1. The tensor product of an efficient (poly-time) high-rate globally list recoverable code is globally list recoverable in *deterministic* near-linear time. This yields in turn the first capacity-achieving list decodable codes with *deterministic* near-linear time global list decoding algorithms. It also gives constant-rate codes approaching the Gilbert-Varshamov bound with *deterministic* near-linear time global unique decoding algorithms.
2. If the base code is additionally locally correctable, then the tensor product is (genuinely) locally list recoverable. This yields in turn (non-explicit) constant-rate codes approaching the Gilbert-Varshamov bound that are *locally correctable* with query complexity and running time $N^{o(1)}$. This improves over prior work by Gopi et. al. (SODA'17; IEEE Transactions on Information Theory'18) that only gave query complexity N^{ε} with rate that is exponentially small in $1/\varepsilon$.
3. A nearly-tight combinatorial lower bound on output list size for list recovering high-rate tensor codes. This bound implies in turn a nearly-tight lower bound of $N^{\Omega(1/\log\log N)}$ on the product of query complexity and output list size for locally list recovering high-rate tensor codes.

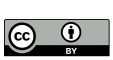
2012 ACM Subject Classification Mathematics of computing → Coding theory; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Coding theory, Tensor codes, List-decoding and recovery, Local codes

Digital Object Identifier 10.4230/LIPIcs.APPROX-RANDOM.2019.68

Category RANDOM

Related Version https://eccc.weizmann.ac.il/report/2019/080/



Funding *Swastik Kopparty*: Research supported in part by NSF grants CCF-1253886, CCF-1540634, CCF-1814409 and CCF-1412958, and BSF grant 2014359. Some of this research was done while visiting the Institute for Advanced Study.
Nicolas Resch: Research supported in part by NSF-BSF grant CCF-1814629 and 2017732, NSERC grant CGSD2-502898, NSF grants CCF- 1422045, CCF-1527110, CCF-1618280, CCF-1814603, CCF-1910588, NSF CAREER award CCF-1750808 and a Sloan Research Fellowship.
Noga Ron-Zewi: Research supported in part by NSF-BSF grant CCF-1814629 and 2017732.
Shubhangi Saraf: Research supported in part by NSF grants CCF-1350572, CCF-1540634 and CCF-1412958, BSF grant 2014359, a Sloan research fellowship and the Simons Collaboration on Algorithms and Geometry. Some of this research was done while visiting the Institute for Advanced Study.
Shashwat Silas: Research supported in part by NSF-BSF grant CCF-1814629 and 2017732 and a Google Fellowship in the School of Engineering at Stanford.

1 Introduction

Error-correcting codes enable protection of data from errors. They allow one to encode a message so that even after some symbols of the encoding get changed, the original message can still be recovered.

Formally, an *error-correcting code* of *blocklength* n over a finite alphabet Σ is a subset $C \subseteq \Sigma^n$. If k is such that $|C| = |\Sigma|^k$, then a k symbol *message* can be encoded using this code. The redundancy of the code is measured by the *rate* $\rho = k/n$ (so that $|C| = |\Sigma|^{\rho n}$). The robustness to errors is measured by its *relative distance* δ, defined to be the minimum, over all distinct $x, y \in C$, of the relative Hamming distance $\mathrm{dist}(x, y)$. A basic but important observation is that for codes with relative distance δ, for every $w \in \Sigma^n$, there is at most one codeword $c \in C$ for which $\mathrm{dist}(w, c) < \delta/2$. Finding this codeword given w is the algorithmic problem of *unique decoding* C upto half the minimum distance.

Given this setup, we now state some central goals of coding theory. First, we would like to understand the best possible *tradeoffs* for ρ and δ that are achievable. Next, we would like to have *explicit constructions* of codes that achieve this best possible tradeoff. Finally, we would like *efficient algorithms* for decoding such optimal codes upto half their minimum distance – this would give codes correcting the maximum possible fraction of (worst-case) errors for their rate.

For the case of $|\Sigma| = 2$ (the binary alphabet), the *Gilbert-Varshamov bound* states that for all $\delta \leq 1/2$ and $\gamma > 0$ there exist codes with $n \to \infty$ for which[1] $\rho \geq 1 - H_2(\delta) - \gamma$. In fact, a random linear code satisfies this with high probability. The Gilbert-Varshamov bound is the best known tradeoff in the setting where $\delta = \Omega(1)$, and surprisingly, it is not known to be tight. Furthermore, despite their abundance, we do not know how to explicitly construct codes achieving the Gilbert-Varshamov bound.

For growing alphabets, $|\Sigma| = \omega(1)$, the picture is almost completely understood. We know that the best tradeoff achievable is $\rho = 1 - \delta - \gamma$, and furthermore we know how to explicitly construct codes achieving this tradeoff that can be efficiently unique decoded upto half their minimum distance.

[1] Here H_2 is the binary entropy function.

1.1 The cast

In recent years, several important variations of the problem of unique decoding have been considered. We will need many of these, so we give below a quick and gentle introduction (without formal definitions).

List decoding

In list decoding we attempt to decode from an even larger fraction α of errors than $\delta/2$ – now there may be more than one nearby codeword, and our goal is to find the *list* of all of them. A basic limitation is that efficient list decoding is only possible if the number of nearby codewords is guaranteed to be polynomially bounded.

Unlike the case of unique decoding, the optimal tradeoff between the rate ρ and the *list decoding radius* α (for polynomial-size lists) is known for all alphabet sizes. The optimal rate for a given α is known as the *list decoding capacity*. For $|\Sigma| = 2$, the list decoding capacity is $\rho = 1 - H_2(\alpha) - \gamma$, while for $|\Sigma| = \omega(1)$, the list decoding capacity is $\rho = 1 - \alpha - \gamma$. Over large alphabets, this tradeoff can be achieved by explicit codes with efficient list decoding algorithms [21] (see also [27] for the state of the art). Over binary alphabet, we do not know how to explicitly construct codes achieving list decoding capacity.

List recovery

List recovery is a generalization of list decoding where we are given a small list of candidate alphabet symbols at each coordinate (these lists are called the *input lists*) and the goal is to find the *output list* of all codewords that are consistent with many of these input lists. In other words, we want all codewords such that for a $(1 - \alpha)$-fraction of coordinates, the symbol of the codeword at that coordinate lies within the input list for that coordinate (we call these the "nearby codewords"). When the input list size is 1, then list recovery is the same as list decoding.

Local decoding

In local decoding, we want to unique decode in sublinear time. Standard decoding has linear output size, so we need to aim lower. For a given $w \in \Sigma^n$ and a given message coordinate $i \in [k]$, we only ask to recover symbol i of the message underlying the codeword c near w. We would like to run in sublinear time (and hence use only a sublinear number of queries to w), so we allow the algorithm to use randomness and allow a small probability of error.

Local correction is a variation of local decoding where one is required to recover *codeword* symbols as opposed to message symbols. In *approximate* local decoding (local correction, resp.) one is only required to recover correctly *most* of the message (codeword, resp.) coordinates.

Local list decoding

Local list decoding combines the notions of local decoding and list decoding. We are given some $w \in \Sigma^n$, and the goal is that for any nearby codeword, one can in sublinear time recover the ith symbol of the message corresponding to the codeword for any $i \in [k]$. In order to make this precise, the local list decoding algorithm first does some preprocessing and then produces as output a collection of algorithms A_j. For any nearby codeword c, with high

probability one of these algorithms corresponds to it.[2] These algorithms then behave like local decoding algorithms. On input $i \in [k]$, if the algorithm corresponded to a codeword c, then by making queries to only a sublinear number of coordinates, the algorithm with high probability outputs the correct value of the ith symbol of the message corresponding to c.

The above definition of local list decoding can be extended to *local list recovery* in a straightforward way where now the algorithms A_j correspond to all codewords that agree with most of the input lists. As above, we can also define a local correction version of local list decoding (or local list recovery) where the algorithms A_j are required to recover codeword symbols as opposed to message symbols. Finally, we can also define approximate local list decoding (or local list recovery) where the algorithms A_j are only required to recover correctly most of the message (or codeword in the local correction version) coordinates.

The context

The starting point for this paper is the recent result of [23] on high-rate list recoverable tensor codes, and its corollaries. Tensoring is a natural operation on codes that significantly enhances their local properties [5, 34, 9, 10, 15, 6, 7, 37, 28, 36, 26].

The main technical result of [23] was that the tensor product of an efficient (poly-time) high-rate globally list recoverable code is *approximately* locally list recoverable (in either the local decoding or local correction version). They then observed that the "approximately" modifier can be eliminated by pre-encoding the tensor product with a locally decodable code. This gave the first construction of codes with rate arbitrarily close to 1 that are locally list recoverable from an $\Omega(1)$ fraction of errors (however, only in the local decoding version). Finally, using the expander-based distance amplification method of [2, 3] (specialized to the setting of local list recovery [18, 17]), this gave the first capacity-achieving locally list recoverable (and in particular, list decodable) codes with sublinear (and in fact $N^{\tilde{O}(1/\log\log N)}$) query complexity and running time (once more, in the local decoding version).

The above result also yielded further consequences for global decoding. Specifically, [23] observed that the approximate local list recovery algorithm for tensor codes naturally gives a *probabilistic* near-linear time global list recovery algorithm. Once more, using the expander-based distance amplification method of [2, 3, 18], this gave the first capacity-achieving list recoverable (and in particular, list decodable) codes with *probabilistic* near-linear time global list recovery algorithms. Finally, via the random concatenation method of [33, 19], this yielded in turn a (randomized) construction of constant-rate binary codes approaching the Gilbert-Varshamov bound with a *probabilistic* near-linear time algorithm for global unique decoding upto half the minimum distance.

One could potentially hope (following [17] which implemented a local version of [33, 19]) for an analogous result that would give constant-rate codes approaching the Gilbert-Varshamov bound that are locally correctable (or locally decodable) with query complexity and running time $N^{o(1)}$. However, what prevented [23] from obtaining such a result was the fact that their capacity-achieving locally list recoverable codes only worked in the local decoding version (i.e., they were only able to recover message coordinates).

[2] Some of these algorithms A_j might not correspond to any codeword and might output garbage. Later in the paper we define local list decoding to not allow these garbage producing A_j's. Eliminating the garbage can be easily done if the underlying code is also *locally testable*, and in this case the stronger notion can be achieved.

1.2 Results

We revisit the technique of [23] and show the following.

- The tensor product of an efficient (poly-time) high-rate globally list recoverable code is globally list recoverable in *deterministic* near-linear time. Plugging this into the machinery of [2, 3, 18], we get the first capacity-achieving list recoverable (and in particular, list decodable) codes with *deterministic* near-linear time global list recovery algorithms. Plugging this into the machinery of [33, 19], yields in turn constant-rate binary codes (with a randomized construction) approaching the Gilbert-Varshamov bound with *deterministic* near-linear time global unique decoding algorithms.

  Our deterministic global list recovery algorithm is obtained by derandomizing the random choices of the [23] algorithm using appropriate samplers.
- An instantiation of the base code to produce tensor product codes which are themselves genuinely locally list recoverable (i.e., not just approximately locally list recoverable) in the *local correction version*. Once more, plugging this into the machinery of [2, 3, 17], we get capacity-achieving locally list recoverable codes, but now in the *local correction version*. This now plugs in turn into the machinery of [33, 19, 17] to give constant-rate binary codes (with a randomized construction) approaching the Gilbert-Varshamov bound that are locally decodable with query complexity and running time $N^{o(1)}$. This improves over prior work [17] that only gave query complexity N^{ε} with rate that is exponentially small in $1/\varepsilon$.

  We obtain our result by taking the base code to be the *intersection* of an efficient (poly-time) high-rate globally list recoverable code and a high-rate locally correctable code. Assuming both codes are linear, we have that the intersection is a high-rate code that is both! The result of [23] already guarantees that this tensor product is approximately locally list recoverable (in the local correction version), and we use the fact that the tensor product of a locally correctable codes is also locally correctable [37] to remove the "approximately" modifier.[3]
- A combinatorial lower bound showing the limitations on the list recoverability of high-rate tensor codes. Specifically, we show that when the rate of the base code is high, every t-wise tensor product of this code has output list size *doubly-exponential* in t. This means that taking t to be more than $\log \log N$ leads to superpolynomial output list size, precluding the possibility of efficient list recovery.

  Instantiating this appropriately, this implies in turn that there is a base code such that for every tensor power with block length N, the product of the query complexity and output list size for local list recovery is at least $N^{\Omega(1/\log \log N)}$. We note that in contrast, it could be that for every base code, there is a tensor power with block length N for which local correction can be done with query complexity $O(1)$.

  A key observation that we use is that a high-rate code has many codewords with pairwise-disjoint supports. We combine this along with other linear-algebraic arguments to design a list recovery instance for the tensor product of a high-rate code which has many codewords that are consistent with it.

Below we give formal statements of our results. For formal definitions of the various notions of decoding in the following theorem statements, see Section 2.

[3] To eliminate "garbage" we also use the fact that the tensor product is locally testable [37].

1.2.1 Deterministic near-linear time global list recovery

Our first main result shows that the tensor product of an efficient (poly-time) high-rate globally list recoverable code is globally list recoverable in *deterministic* near-linear time. In the theorem statement, one should think of all parameters δ, α, L, t, and consequently also s, as constants (or more generally, as slowly increasing/decreasing functions of n). In that case, the theorem says that if $C \subseteq \mathbb{F}^n$ is (α, ℓ, L)-globally list recoverable deterministically in time $T = \mathrm{poly}(n)$, then the t-iterated tensor product $C^{\otimes t}$ of length $N := n^t$ is $(\Omega(\alpha), \ell, L^{O(1)})$-globally list recoverable deterministically in time $O(n^t \cdot T) = n^{t+O(1)} = N^{1+O(1/t)}$.

▶ **Theorem 1** (Deterministic near-linear time list recovery of high-rate tensor codes)**.** *The following holds for any $\delta, \alpha > 0$, and $s = \mathrm{poly}(1/\delta, 1/\alpha)$. Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ that is (α, ℓ, L)-globally list recoverable deterministically in time T. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(\alpha \cdot s^{-t^2}, \ell, L^{s^{t^3} \cdot L^t})$-globally list recoverable deterministically in time $n^t \cdot T \cdot L^{s^{t^3} \cdot L^t}$.*

Applying the expander-based distance amplification method of [2, 3, 18] on the codes given by the above theorem, we obtain the first capacity-achieving list recoverable (and in particular, list decodable) codes with *deterministic* near-linear time global list recovery algorithms.

▶ **Corollary 2** (Deterministic nearly-linear time capacity-achieving list recoverable codes)**.** *For any constants $\rho \in [0, 1]$, $\gamma > 0$, and $\ell \geq 1$ there exists an infinite family of codes $\{C_N\}_N$, where C_N has block length N, alphabet size $N^{o(1)}$, rate ρ, and is $(1 - \rho - \gamma, \ell, N^{o(1)})$-globally list recoverable deterministically in time $N^{1+o(1)}$.*

Applying the random concatenation method of [33, 19], the above corollary yields in turn constant-rate codes approaching the Gilbert-Varshamov bound with *deterministic* near-linear time global unique decoding algorithms.

▶ **Corollary 3** (Deterministic near-linear time unique decoding up to the GV bound)**.** *For any constants $\rho \in [0, 0.02]$ and $\gamma > 0$ there exists an infinite family of binary linear codes $\{C_N\}_N$, where C_N has block length N and rate ρ, and is globally uniquely decodable deterministically from $\frac{H_2^{-1}(1-\rho)-\gamma}{2}$-fraction of errors in time $N^{1+o(1)}$.*

1.2.2 Local list recovery

Our second main result shows that if the base code is *both* globally list recoverable and locally correctable, then the tensor product is (genuinely) locally list recoverable (in the local correction version).

▶ **Theorem 4** (Local list recovery of high-rate tensor codes)**.** *The following holds for any $\delta, \alpha > 0$, and $s = \mathrm{poly}(1/\delta, 1/\alpha)$. Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ that is (α, ℓ, L)-globally list recoverable, and locally correctable from $(\delta/2)$-fraction of errors with query complexity Q, and $t \geq 3$. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(\alpha \cdot s^{-t^3}, \ell, L^{s^{t^3} \cdot \log^t L})$-locally list recoverable with query complexity $n^{O(1)} \cdot Q^{O(t)} \cdot L^{s^{t^3} \cdot \log^t L}$.*

Once more, applying the expander-based distance amplification method of [2, 3, 18, 17], as well as the random concatenation method of [33, 19, 17], the above theorem yields constant-rate codes approaching the Gilbert-Varshamov bound that are *locally correctable* with query complexity $N^{o(1)}$.

▶ **Corollary 5** (Local correction up to the GV bound). *For any constants $\rho \in [0, 0.02]$ and $\gamma > 0$ there exists an infinite family of binary linear codes $\{C_N\}_N$, where C_N has block length N and rate ρ, and is locally correctable from $\frac{H_2^{-1}(1-\rho)-\gamma}{2}$-fraction of errors with query complexity $N^{o(1)}$.*

1.2.3 Combinatorial lower bound on output list size

Our final main result shows a nearly-tight combinatorial lower bound on output list size for list recovering high-rate tensor codes.

▶ **Theorem 6** (Output list size for list recovering high-rate tensor codes). *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of rate $1-\gamma$, and that $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(0, \ell, L)$-list recoverable. Then $L \geq \ell^{1/\gamma^t}$.*

The above bound can be instantiated concretely as follows.

▶ **Corollary 7.** *For any $\delta > 0$ and $\ell > 1$ there exists $L > 1$ such that the following holds for any sufficiently large n. There exists a linear code $C \subseteq \mathbb{F}^n$ of relative distance δ that is $(\Omega(\delta), \ell, L)$-list recoverable, but $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is only $(0, \ell, L')$-list recoverable for $L' \geq \exp((2\delta)^{-(t-3/2)} \cdot \sqrt{\log L})$.*

Finally, we also obtain a nearly-tight lower bound of $N^{\Omega(1/\log\log N)}$ on the product of query complexity and output list size for locally list recovering high-rate tensor codes.

▶ **Corollary 8.** *For any $\delta > 0$ and sufficiently large n there exists a linear code $C \subseteq \mathbb{F}^n$ of relative distance δ such that the following holds. Suppose that $C^{\otimes t} \subseteq \mathbb{F}^N$ is $(\frac{1}{N}, 2, L)$-locally list recoverable with query complexity Q. Then $Q \cdot L \geq N^{\Omega_\delta(1/\log\log N)}$.*

2 Preliminaries

For a prime power q we denote by $\mathbb{F}_q$ the finite field of q elements. For any finite alphabet Σ and for any pair of strings $x, y \in \Sigma^n$, the relative distance between x and y is the fraction of coordinates $i \in [n]$ on which x and y differ, and is denoted by $\mathrm{dist}(x, y) := |\{i \in [n] : x_i \neq y_i\}|/n$. For a subset $Y \subseteq \Sigma^n$, we denote by $\mathrm{dist}(x, Y)$ the minimum relative distance of a string $y \in Y$ from x. For a positive integer ℓ we denote by $\binom{\Sigma}{\ell}$ the collection of all subsets of Σ of size ℓ and by $\binom{\Sigma}{\leq \ell}$ the collection of all nonempty subsets of Σ of size at most ℓ. For any string $x \in \Sigma^n$ and tuple $S \in \binom{\Sigma}{\leq \ell}^n$ we denote by $\mathrm{dist}(x, S)$ the fraction of coordinates $i \in [n]$ for which $x_i \notin S_i$, that is, $\mathrm{dist}(x, S) := |\{i \in [n] : x_i \notin S_i\}|/n$. For a string $x \in \Sigma^n$ and a subset $T \subseteq [n]$, we use $x|_T \in \Sigma^{|T|}$ to denote the restriction of x to the coordinates in T. Throughout the paper, we use $\exp(n)$ to denote $2^{\Theta(n)}$, and whenever we use log, it is base 2, unless noted otherwise.

2.1 Error-correcting codes

An error-correcting code is simply a subset $C \subseteq \Sigma^n$. We call Σ the alphabet of the code, and n its block length. The elements of C are called codewords. If $\mathbb{F}$ is a finite field and Σ is a vector space over $\mathbb{F}$, we say that a code $C \subseteq \Sigma^n$ is $\mathbb{F}$-linear if it is an $\mathbb{F}$-linear subspace of the $\mathbb{F}$-vector space Σ^n. If $\Sigma = \mathbb{F}$, we simply say that C is linear.

The rate of a code is the ratio $\rho := \frac{\log|C|}{\log(|\Sigma|^n)}$, which for $\mathbb{F}$-linear codes equals $\frac{\dim_{\mathbb{F}}(C)}{n \cdot \dim_{\mathbb{F}}(\Sigma)}$. The relative distance $\mathrm{dist}(C)$ of C is the minimum $\delta > 0$ such that for every pair of distinct codewords $c_1, c_2 \in C$ it holds that $\mathrm{dist}(c_1, c_2) \geq \delta$. We denote by $\Delta(C) := \mathrm{dist}(C) \cdot n$ the (absolute) distance of C.

The best known general trade-off between rate and distance of codes is the Gilbert-Varshamov bound, attained by random (linear) codes. For $x \in [0,1]$ let

$$H_q(x) = x \log_q(q-1) + x \log_q(1/x) + (1-x)\log_q(1/(1-x))$$

denote the q-ary entropy function.

▶ **Theorem 9** (Gilbert-Varshamov (GV) bound, [12, 35]). *For any prime power q, $\delta \in (0, 1-\frac{1}{q})$, and $\rho \in (0, 1-H_q(\delta))$, a random linear code $C \subseteq \mathbb{F}_q^n$ of rate ρ has relative distance at least δ with probability $1-\exp(-n)$.*

▶ **Corollary 10.** *For any $\rho \in [0,1]$ and $\gamma > 0$, and prime power $q \geq 2^{H_2(1-\rho-\gamma)/\gamma}$, a random linear code $C \subseteq \mathbb{F}_q^n$ of rate ρ has relative distance at least $1-\rho-\gamma$ with probability $1-\exp(-n)$.*

An encoding map for C is a bijection $E_C : \Sigma^k \to C$, where $|\Sigma|^k = |C|$. We call the elements in the domain of E_C messages, and k the message length. We say that C is encodable in time T if an encoding map for C can be computed in time T. For a code $C \subseteq \Sigma^n$ of relative distance δ and a given parameter $\alpha < \delta/2$, we say that C is decodable from α-fraction of errors in time T if there exists an algorithm, running in time T, that given a received word $w \in \Sigma^n$, computes the unique codeword $c \in C$ (if any) which satisfies $\mathrm{dist}(c,w) \leq \alpha$.

▶ **Proposition 11** (Reed-Solomon codes, [29, 8]). *For any prime power q and integers $k \leq n \leq q$, there exists a linear code $C \subseteq \mathbb{F}_q^n$ of rate $\rho := k/n$ and relative distance at least $1-\rho$ that is encodable and decodable from $\frac{1-\rho}{2}$-fraction of errors in time $\mathrm{poly}(n, \log q)$.*

Let $C \subseteq \mathbb{F}^n$ be a linear code of dimension k. A generating matrix for C is an $n \times k$ matrix G such that $\mathrm{Im}(G) = C$. A parity-check matrix for C is an $(n-k) \times n$ matrix H such that $\ker(H) = C$. The dual code $C^\perp \subseteq \mathbb{F}^n$ is given by

$$C^\perp = \{y \in \mathbb{F}^n \mid \langle y, c\rangle = 0 \ \forall c \in C\}.$$

It is well-known that $C^{\perp\perp} = C$, and that a matrix G is a generating matrix for C if and only if G^T is a parity-check matrix for $C^\perp$.

2.2 List recoverable codes

List recovery is a generalization of the standard error-correction setting where each entry w_i of the received word w is replaced with a list S_i of ℓ possible symbols of Σ. Formally, for $\alpha \in [0,1]$ and integers ℓ, L we say that a code $C \subseteq \Sigma^n$ is (α, ℓ, L)-list recoverable if for any tuple $S \in \binom{\Sigma}{\leq \ell}^n$ there are at most L different codewords $c \in C$ so that $\mathrm{dist}(c, S) \leq \alpha$. We say that C is (α, L)-list decodable if it is $(\alpha, 1, L)$-list recoverable.

▶ **Corollary 12** ([24], Corollary 2.2). *For any $\rho \in [0,1]$, $\gamma > 0$, and $\ell \geq 1$, and for sufficiently large prime power q, a random linear code $C \subseteq \mathbb{F}_q^n$ of rate ρ is $(1-\rho-\gamma, \ell, q^{O(\ell/\gamma)})$-list recoverable with probability $1-\exp(-n)$.*

We say that C is (α, ℓ, L)-list recoverable in time T if there exists an algorithm, running in time T, that given a tuple $S \in \binom{\Sigma}{\ell}^n$, returns all codewords $c \in C$ (if any) which satisfy $\mathrm{dist}(c, S) \leq \alpha$. The following theorem from [22, 20, 23] gives a family of high-rate linear codes which are efficiently list recoverable with constant alphabet size and nearly-constant output list size.

▶ **Theorem 13** ([24], Theorem A.1). *There exists an absolute constant b_0 so that the following holds. For any $\gamma > 0$, $\ell \geq 1$, $q \geq \ell^{b_0/\gamma}$ that is an even power of a prime*[4]*, and integer $n \geq q^{b_0 \ell/\gamma}$, there exists a linear code $C \subseteq \mathbb{F}_q^n$ of rate $1-\gamma$ and relative distance $\Omega(\gamma^2)$ that is $(\Omega(\gamma^2), \ell, L)$-list recoverable for $L = q^{q^{(\ell/\gamma)\cdot \exp(\log^* n)}}$. Moreover, C can be encoded in time $\mathrm{poly}(n, \log q)$ and list recovered in time $\mathrm{poly}(n, L)$.*

2.3 Local codes

2.3.0.1 Locally testable codes

Intuitively, a code is said to be locally testable [11, 30, 16] if, given a string $w \in \Sigma^n$, it is possible to determine whether w is a codeword of C, or rather far from C, by reading only a small part of w. For our purposes, we shall also require an additional *tolerance* property of determining whether w is sufficiently close to the code.

▶ **Definition 14** (Tolerant locally testable code (Tolerant LTC)). *We say that a code $C \subseteq \Sigma^n$ is (Q, α, β)-tolerantly locally testable if there exists a randomized algorithm A that satisfies the following requirements:*

- ***Input:*** *A gets oracle access to a string $w \in \Sigma^n$.*
- ***Query complexity:*** *A makes at most Q queries to the oracle w.*
- ***Completeness:*** *If $\mathrm{dist}(w, C) \leq \alpha$, then A accepts with probability at least $\frac{2}{3}$.*
- ***Soundness:*** *If $\mathrm{dist}(w, C) \geq \beta$, then A rejects with probability at least $\frac{2}{3}$.*

▶ Remark 15. The definition requires $0 \leq \alpha < \beta \leq 1$. The above success probability of $\frac{2}{3}$ can be amplified using sequential repetition, at the cost of increasing the query complexity. Specifically, amplifying the success probability to $1 - \exp(-t)$ requires increasing the query complexity by a multiplicative factor of $O(t)$.

Locally correctable codes

Intuitively, a code is said to be locally correctable [4, 32, 25] if, given a codeword $c \in C$ that has been corrupted by some errors, it is possible to decode any coordinate of c by reading only a small part of the corrupted version of c.

▶ **Definition 16** (Locally correctable code (LCC)). *We say that a code $C \subseteq \Sigma^n$ is (Q, α)-locally correctable if there exists a randomized algorithm A that satisfies the following requirements:*

- ***Input:*** *A takes as input a coordinate $i \in [n]$, and also gets oracle access to a string $w \in \Sigma^n$ that is α-close to a codeword $c \in C$.*
- ***Query complexity:*** *A makes at most Q queries to the oracle w.*
- ***Output:*** *A outputs c_i with probability at least $\frac{2}{3}$.*

▶ Remark 17. The definition requires $\alpha < \mathrm{dist}(C)/2$. The above success probability of $\frac{2}{3}$ can be amplified using sequential repetition, at the cost of increasing the query complexity. Specifically, amplifying the success probability to $1 - \exp(-t)$ requires increasing the query complexity by a multiplicative factor of $O(t)$.

[4] That is, q is of the form p^{2t} for a prime p and for an integer t.

Locally list recoverable codes

The following definition from [14, 32, 17] generalizes the notion of locally correctable codes to the setting of list decoding/recovery. In this setting, the local list recovery algorithm is required to output in an implicit sense all codewords that are consistent with most of the input lists.

▶ **Definition 18** (Locally list recoverable code). *We say that a code $C \subseteq \Sigma^n$ is $(Q, \alpha, \varepsilon, \ell, L)$-locally list recoverable if there exists a randomized algorithm A that satisfies the following requirements:*

- ***Input:*** *A gets oracle access to a string $S \in \binom{\Sigma}{\leq \ell}^n$.*
- ***Query complexity:*** *A makes at most Q queries to the oracle S.*
- ***Output:*** *A outputs L randomized algorithms $A_1, \ldots, A_L$, where each A_j takes as input a coordinate $i \in [n]$, makes at most Q queries to the oracle S, and outputs a symbol in Σ.*
- ***Completeness:*** *For any codeword $c \in C$ which satisfies $\mathrm{dist}(c, S) \leq \alpha$, with probability at least $1 - \varepsilon$ over the randomness of A, the following event happens: there exists some $j \in [L]$ such that for all $i \in [n]$,*

$$\Pr[A_j(i) = c_i] \geq \frac{2}{3}, \tag{1}$$

  where the probability is over the internal randomness of A_j.
- ***Soundness:*** *With probability at least $1 - \varepsilon$ over the randomness of A, the following event happens: for every $j \in [L]$, there exists some $c \in C$ such that for all $i \in [n]$,*

$$\Pr[A_j(i) = c_i] \geq \frac{2}{3},$$

  where the probability is over the internal randomness of A_j.

We say that A has preprocessing time T_{pre} if A outputs the description of the algorithms $A_1, \ldots, A_L$ in time at most T_{pre}, and has running time T if each A_j has running time at most T. As before, we say that the code C is $(Q, \alpha, \varepsilon, L)$-locally list decodable if it is $(Q, \alpha, \varepsilon, 1, L)$-locally list recoverable.

2.4 Tensor codes

In this paper we study the list recovery properties of the high-rate tensor product codes, defined as follows.

▶ **Definition 19** (Tensor product codes). *Let $C_1 \subseteq \mathbb{F}^{n_1}$, $C_2 \subseteq \mathbb{F}^{n_2}$ be linear codes. Their tensor product code $C_1 \otimes C_2 \subseteq \mathbb{F}^{n_1 \times n_2}$ consists of all matrices $M \in \mathbb{F}^{n_1 \times n_2}$ such that all the rows of M are codewords of C_2 and all the columns are codewords of C_1.*

3 Deterministic near-linear time global list recovery

3.1 Deterministic near-linear time list recovery of high-rate tensor codes

In this section we prove Theorem 1, restated bollow, which shows that the tensor product of an efficient (poly-time) high-rate globally list recoverable code is globally list recoverable in *deterministic* near-linear time.

▶ **Theorem 1** (Deterministic near-linear time list recovery of high-rate tensor codes)**.** *The following holds for any $\delta, \alpha > 0$, and $s = \mathrm{poly}(1/\delta, 1/\alpha)$. Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ that is (α, ℓ, L)-globally list recoverable deterministically in time T. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(\alpha \cdot s^{-t^2}, \ell, L^{s^{t^3} \cdot L^t})$-globally list recoverable deterministically in time $n^t \cdot T \cdot L^{s^{t^3} \cdot L^t}$.*

Theorem 1 follows by applying the lemma below iteratively.

▶ **Lemma 20.** *The following holds for any $\delta, \alpha, \delta_{\mathrm{dec}}, \delta'_{\mathrm{dec}} > 0$, and $\bar{s} = \mathrm{poly}(1/\delta, 1/\alpha, 1/\delta_{\mathrm{dec}}, 1/\delta'_{\mathrm{dec}})$.*

Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ that is (α, ℓ, L)-globally list recoverable deterministically in time T, and $C' \subseteq \mathbb{F}^{n'}$ is a linear code that is (α', ℓ, L')-globally list recoverable deterministically in time T'. Suppose furthermore that C, C' are uniquely decodable deterministically from $\delta_{\mathrm{dec}}, \delta'_{\mathrm{dec}}$-fraction of errors in times $T_{\mathrm{dec}}, T'_{\mathrm{dec}}$, respectively.

Then $C \otimes C' \subseteq \mathbb{F}^{n \times n'}$ is $(\alpha'/\bar{s}, \ell, (L')^{\bar{s} \cdot L/(\alpha')^2})$-globally list recoverable deterministically in time

$$(L')^{\bar{s} \cdot L/(\alpha')^2} \cdot n \cdot (n' \cdot (T + T_{\mathrm{dec}}) + n \cdot T'_{\mathrm{dec}} + T') \,.$$

We now sketch the proof of Lemma 20. Our plan is to derandomize the approximate local list recovery algorithm for high-rate tensor codes of [23]. Recall that an approximate local list recovery algorithm (local correction version) is a randomized algorithm A that outputs a collection of (without loss of generality, deterministic) local algorithms A_j satisfying the following: for any codeword c that is consistent with most of the input lists, with high probability (over the randomness of A) one of the local algorithms A_j locally corrects most of the coordinates of c.

As observed in [23], an approximate local list recovery algorithm naturally gives a *probabilistic* near-linear time *global* list recovery algorithm as follows. First run the algorithm A to obtain the collection of local algorithms A_j. Then for each A_j, output a codeword that is obtained by applying A_j on each codeword coordinate, and then uniquely decoding the resulting word to the closest codeword. The guarantee now is that any codeword that is consistent with most of the input lists will be output with high probability.

To derandomize the probabilistic global algorithm described above, we note that the preprocessing algorithm A in [23] produces the collection of local algorithms A_j by choosing a random subset of rows in the tensor product,[5] that is chosen uniformly at random amongst all subsets of the appropriate size. We then observe that this subset can be alternatively chosen using a randomness-efficient *sampler* without harming much the performance. Finally, since the sampler uses a small amount of randomness (logarithmic in the blocklength of C), we can afford to iterate over all seeds and return the union of all output lists. This gives a *deterministic* near-linear time global list recovery algorithm that outputs all codewords that are consistent with most of the input lists.

3.1.1 Samplers

We start by defining the appropriate samplers we use.

[5] In [23], the role of columns and rows is swapped.

▶ **Definition 21** ((averaging) sampler). *An (n, η, γ)-sampler with randomness r and sample size m is a randomized algorithm that tosses r random coins and outputs a subset $I \subseteq [n]$ of size m such that the following holds. For any function $f : [n] \to [0, 1]$, with probability at least $1 - \eta$ over the choice of I,*

$$\left|\mathbb{E}_{i \in I}\left[f(i)\right] - \mathbb{E}_{i \in [n]}\left[f(i)\right]\right| \leq \gamma.$$

We shall use the following construction from Goldreich [13].

▶ **Theorem 22** ([13], Corollary 5.6). *For any $\eta, \gamma > 0$ and integer n, there exists an (n, η, γ)-sampler with randomness $\log(n/\gamma)$, sample size $O\left(1/(\eta\gamma^2)\right)$, and running time $\mathrm{poly}(\log n, 1/\eta, 1/\gamma)$.*

In what follows, let Γ denote the (n, η, γ)-sampler promised by the above theorem, where we set $\eta := \frac{0.1}{L} \cdot \frac{\delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}}{3}$ and $\gamma := \alpha' \cdot \frac{\delta \cdot \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}}{24}$. Let $r := \log(n/\gamma) \leq \log(n \cdot \bar{s}/\alpha')$ and $m := O(1/(\eta\gamma^2)) \leq L \cdot \bar{s}/(\alpha')^2$ denote the randomness and sample size of Γ, respectively (assuming that $\bar{s}$ is a sufficiently large polynomial).

3.1.2 Randomness-efficient algorithm

We first describe a randomness-efficient global list recovery algorithm $\tilde{A}$ for $C \otimes C'$ that is obtained by replacing the choice of a uniform random subset of rows made in [23] with a sample from Γ. We will later observe that the randomness can be eliminated by iterating over all seeds of Γ and returning the union of all output lists.

The algorithm $\tilde{A}$ behaves as follows. First, it uses Γ to sample a subset of m rows $I = \{i_1, \ldots, i_m\} \subseteq [n]$. Then for $k = 1, \ldots, m$, it runs the list recovery algorithm A' for C' on the i_k-th row $S|_{\{i_k\} \times [n']}$; let $\mathcal{L}'_{i_1}, \mathcal{L}'_{i_2}, \ldots, \mathcal{L}'_{i_m} \subseteq C'$ denote the lists output by A' on each of the rows in I. Finally, for any choice of codewords $c'_1 \in \mathcal{L}'_{i_1}, c'_2 \in \mathcal{L}'_{i_2}, \ldots, c'_m \in \mathcal{L}'_{i_m}$, the algorithm $\tilde{A}$ outputs a codeword $\tilde{c} \in C \otimes C'$ that is obtained as follows.

For each column $j \in [n']$, the algorithm $\tilde{A}$ runs the list recovery algorithm A for C on the j-th column $S|_{[n] \times \{j\}}$; let $\mathcal{L}_1, \mathcal{L}_2, \ldots \mathcal{L}_{n'} \subseteq C$ denote the lists output by A on each of the n' columns. Then the algorithm $\tilde{A}$ chooses for each column $j \in [n']$ the codeword $c_j \in \mathcal{L}_j$ whose restriction to I is closest to $((c'_1)_j, (c'_2)_j \ldots, (c'_m)_j)$ (i.e., the restriction of $c'_1, c'_2, \ldots, c'_m$ to the j-th column). Finally, the algorithm $\tilde{A}$ sets the value of each column $j \in [n']$ to c_j, and uniquely decodes the resulting word $\tilde{c}_0$ to the nearest codeword $\tilde{c} \in C \otimes C'$, assuming there is one at distance at most $\delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}$. If $\mathrm{dist}(\tilde{c}, S) \leq \alpha'/\bar{s}$, then $\tilde{A}$ includes $\tilde{c}$ in the output list $\tilde{\mathcal{L}}$. The formal description is given in Algorithm 1.

3.2 Deterministic nearly-linear time capacity-achieving list recoverable codes

In this section we prove the following lemma which implies Corollary 2 from the introduction.

▶ **Lemma 23.** *For any constants $\rho \in [0, 1]$, $\gamma > 0$, and $\ell \geq 1$ there exists an infinite family of codes $\{C_N\}_N$ that satisfy the following.*

- *C_N is an $\mathbb{F}_2$-linear code of block length N and alphabet size $N^{o(1)}$.*
- *C_N has rate ρ and relative distance at least $1 - \rho - \gamma$.*
- *C_N is $(1 - \rho - \gamma, \ell, N^{o(1)})$-globally list recoverable deterministically in time $N^{1+o(1)}$.*
- *C_N is encodable deterministically in time $N^{1+o(1)}$.*

Algorithm 1 The randomness-efficient global list recovery algorithm $\tilde{A}$ for $C \otimes C'$.

function $\tilde{A}(S \in \binom{\mathbb{F}}{\leq \ell}^{n \times n'})$
  Sample $I = \{i_1, \ldots, i_m\} \subseteq [n]$ of size m using sampler Γ.
  for $k = 1, \ldots, m$ **do**
    Run the list recovery algorithm A' for C' on the i_k-th row $S|_{\{i_k\} \times [n']}$, and let $\mathcal{L}'_{i_k} \subseteq C'$ be the list of codewords output by A'.
  end for
  Initialize $\tilde{c}_0 \in \mathbb{F}^{n \times n'}$, $\tilde{\mathcal{L}} \leftarrow \emptyset$.
  for any choice of codewords $c'_1 \in \mathcal{L}'_{i_1}, c'_2 \in \mathcal{L}'_{i_2}, \ldots, c'_m \in \mathcal{L}'_{i_m}$ **do**
    for $j \in [n']$ **do**
      Run the list recovery algorithm A for C on the j-th column $S|_{[n] \times \{j\}}$, and let $\mathcal{L}_j \subseteq C$ be the list of codewords output by A.
      Choose a codeword $c_j \in \mathcal{L}_j$ for which $c_j|_I$ is closest to $((c'_1)_j, (c'_2)_j, \ldots, (c'_m)_j)$ (breaking ties arbitrarily).
      Set the j-th column of $\tilde{c}_0$ to c_j.
    end for
    Uniquely decode $\tilde{c}_0$ from $(\delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}})$-fraction of errors, and let $\tilde{c} \in C \otimes C'$ be the resulting codeword (if exists). If $\mathrm{dist}(\tilde{c}, S) \leq \alpha'/\bar{s}$, add $\tilde{c}$ to $\tilde{\mathcal{L}}$.
  end for
end function

To prove the above lemma, we first use Theorem 1 to obtain deterministic nearly-linear time *high-rate* list recoverable codes, and then use the Alon-Edmonds-Luby (AEL) distance amplification method [2, 3] to turn these codes into deterministic nearly-linear time *capacity-achieving* list recoverable codes.

3.3 Deterministic near-linear time unique decoding up to the GV bound

In this section we prove the following lemma which implies Corollary 3 from the introduction.

▶ **Lemma 24.** *For any constants $\rho \in [0, 0.02]$ and $\gamma > 0$ there exists an infinite family of binary linear codes $\{C_N\}_N$, where C_N has block length N and rate ρ, and is globally uniquely decodable deterministically from $\frac{H_2^{-1}(1-\rho)-\gamma}{2}$-fraction of errors in time $N^{1+o(1)}$.*

Furthermore, there exists a randomized algorithm which, on input N, runs in time $N^{1+o(1)}$ and outputs with high probability a description of a code C_N with the properties above. Given the description, the code C_N can be encoded deterministically in time $N^{1+o(1)}$.

To prove the above lemma, we rely on a lemma from [33, 23] which says that one can turn a code that approximately satisfies the Singleton bound into one that approximately satisfies the GV bound via random concatenation.

4 Local list recovery

4.1 Local list recovery of high-rate tensor codes

In this section we prove the following lemma which implies Theorem 4 from the introduction.

▶ **Lemma 25.** *The following holds for any $\delta, \alpha, \varepsilon > 0$ and $s = \mathrm{poly}(1/\delta, 1/\alpha)$. Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ that is (α, ℓ, L)-globally list recoverable, and $(Q, \delta/2)$-locally correctable, and $t \geq 3$. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(\tilde{Q}, \alpha \cdot s^{-t^3}, \varepsilon, \ell, L^{s^{t^3} \cdot \log^t L} \cdot \log(1/\varepsilon))$-locally list recoverable for*

$$\tilde{Q} = n^3 \cdot (Q \log Q)^t \cdot L^{s^{t^3} \cdot \log^t L} \cdot \log^2(1/\varepsilon).$$

Moreover, if C is globally list recoverable in time $\mathrm{poly}(n)$, locally correctable in time T, and globally decodable for $(\delta/2)$-fraction of errors in time $\mathrm{poly}(n)$, then the local list recovery algorithm for $C^{\otimes t}$ has preprocessing time $\mathrm{poly}(n) \cdot L^{s^{t^3} \cdot \log^t L} \cdot \log^2(1/\varepsilon)$ and running time $\mathrm{poly}(n) \cdot (T \log T)^t \cdot (s^{t^3} \log^t L)$.

The above lemma relies on the following lemma from [23] which says that the tensor product of a high-rate globally list recoverable code (which is not necessarily locally correctable) is *approximately* locally list recoverable. Approximate local list recovery is a relaxation of local list recovery, where the local algorithms in the output list are not required to recover *all* the codeword coordinates, but only *most* of them. Formally, a β-approximately $(Q, \alpha, \varepsilon, \ell, L)$-locally list recoverable code $C \subseteq \Sigma^n$ satisfies all the requirements of Definition 18, except that the requirement (1) is replaced with the relaxed condition that

$$\Pr_{i \in [n]} \left[A_j(i) = c_i \right] \geq 1 - \beta, \tag{2}$$

where the probability is over the choice of uniform random $i \in [n]$, and the soundness requirement is eliminated.

▶ **Lemma 26** (Approximate local list recovery of high-rate tensor codes, [24], Lemma 4.1)**.** *The following holds for any $\delta, \alpha, \beta, \varepsilon > 0$ and $s = \mathrm{poly}(1/\delta, 1/\alpha, 1/\beta)$. Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ that is (α, ℓ, L)-globally list recoverable. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is β-approximately $(n \cdot (s^{t^2} \log^t L), \alpha \cdot s^{-t^2}, \varepsilon, \ell, L^{s^{t^2} \cdot \log^t L} \cdot \log(1/\varepsilon))$-locally list recoverable.*

Moreover, if C is globally list recoverable in time $\mathrm{poly}(n)$, then the approximate local list recovery algorithm for $C^{\otimes t}$ has preprocessing time $\log(n) \cdot L^{s^{t^2} \cdot \log^t L} \cdot \log(1/\varepsilon)$ and running time $\mathrm{poly}(n) \cdot (s^{t^2} \log^t L)$.

To turn the approximate local list recovery algorithm given by the above lemma into a local list recovery algorithm we shall use the fact that the tensor product of a locally correctable code is also locally correctable with slightly worse parameters. A similar observation was made in [37, Proposition 3.15.].

▶ **Lemma 27** (Local correction of tensor codes)**.** *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code that is (Q, α)-locally correctable. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(\big(O(Q \log Q)\big)^t, \alpha^t)$-locally correctable.*

Moreover, if C is locally correctable in time T, then the local correction algorithm for $C^{\otimes t}$ runs in time $(O(T \log T))^t$.

To guarantee the soundness property we shall also use the following lemma which says that high-rate tensor codes are tolerantly locally testable.

▶ **Lemma 28** (Tolerant local testing of high-rate tensor codes)**.** *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ, and $t \geq 3$. Then $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(n^2 \cdot \delta^{-O(t)}, \delta^{O(t)}, (\delta/2)^t)$-tolerantly locally testable.*

Moreover, if C is globally decodable from $(\delta/2)$-fraction of errors in time T, then the tolerant local testing algorithm for $C^{\otimes t}$ runs in time $T \cdot n \cdot \delta^{-O(t)}$.

Finally, we show a general transformation that turns an approximately locally list recoverable code that is also locally correctable and tolerantly locally testable into a (genuinely) locally list recoverable code.

▶ **Lemma 29.** *Suppose that $C \subseteq \Sigma^n$ is a β-approximately $(Q, \alpha, \varepsilon, \ell, L)$-locally list recoverable code that is also $(Q_{\text{corr}}, \gamma)$-locally correctable and $(Q_{\text{test}}, \beta, \gamma)$-tolerantly locally testable. Then C is $(\tilde{Q}, \alpha, 2\varepsilon, \ell, L)$-locally list recoverable for*

$$\tilde{Q} = \max\{Q \cdot Q_{\text{test}} \cdot O(|L| \log(|L|/\varepsilon)), Q \cdot Q_{\text{corr}}\}.$$

Moreover, if the approximate local list recovery algorithm has preprocessing time T_{pre} and running time T, and the local correction and tolerant local testing algorithms run in times $T_{\text{test}}, T_{\text{corr}}$, respectively, then the local list recovery algorithm has preprocessing time $T_{\text{pre}} + T \cdot T_{\text{test}} \cdot O(|L| \log(|L|/\varepsilon))$ and running time $T \cdot T_{\text{corr}}$.

4.2 Capacity-achieving locally list recoverable codes

In this section we prove the following lemma which shows the existence of capacity-achieving locally list recoverable codes. An analogous lemma was proven in [24, Lemma 5.3], however only for local decoding *message* coordinates, and without the soundness property. The fact that we are able to locally correct *codeword* coordinates, as well as guarantee the soundness property, will be crucial for our GV bound local correction application.

▶ **Lemma 30.** *For any constants $\rho \in [0, 1]$, $\gamma > 0$, $\varepsilon > 0$, and $\ell \geq 1$ there exists an infinite family of codes $\{C_N\}_N$ that satisfy the following.*

- *C_N is an $\mathbb{F}_2$-linear code of block length N and alphabet size $N^{o(1)}$.*
- *C_N has rate ρ and relative distance at least $1 - \rho - \gamma$.*
- *C_N is $(N^{o(1)}, 1 - \rho - \gamma, \varepsilon, \ell, N^{o(1)})$-locally list recoverable with preprocessing and running time $N^{o(1)}$.*
- *C_N is encodable in time $N^{1+o(1)}$.*

As in the proof of Lemma 23, we first use Lemma 25 to obtain *high-rate* locally list recoverable codes, and then use the Alon-Edmonds-Luby (AEL) distance amplification method [2, 3] to turn these codes into *capacity-achieving* locally list recoverable codes. However, this time we use a version of the AEL method for *local* list recovery from [17].

4.3 Local correction up to the GV bound

In this section we prove the following lemma which implies Corollary 5 from the introduction.

▶ **Lemma 31.** *For any constants $\rho \in [0, 0.02]$ and $\gamma > 0$ there exists an infinite family of binary linear codes $\{C_N\}_N$, where C_N has block length N and rate ρ, and is locally correctable from $\frac{H_2^{-1}(1-\rho)-\gamma}{2}$-fraction of errors with query complexity $N^{o(1)}$.*

Furthermore,

- *The local correction algorithm for C_N runs in time $N^{o(1)}$.*
- *There exists a randomized algorithm which, on input N, runs in time $N^{1+o(1)}$ and outputs with high probability a description of a code C_N with the properties above. Given the description, the code C_N can be encoded deterministically in time $N^{1+o(1)}$.*

The proof is analogous to that of Lemma 24 and relies on concatenation.

▶ **Lemma 32** (Concatenation for local list recovery). *Suppose that $C \subseteq (\Sigma^{\rho' \cdot t})^n$ is $(Q, \alpha, \varepsilon, \ell, L)$-locally list recoverable, and $C_{\mathrm{con}} \subseteq \Sigma^{tn}$ is a code obtained from C by applying a code $C^{(i)} \subseteq \Sigma^t$ of rate ρ' on each coordinate $i \in [n]$ of C. Suppose furthermore that at least $(1-\gamma)$-fraction of the codes $C^{(i)}$ are (α', ℓ', ℓ)-globally list recoverable. Then C_{con} is $(Q \cdot t, (\alpha - \gamma) \cdot \alpha', \varepsilon, \ell', L)$-locally list recoverable.*

Moreover, if the local list recovery algorithm for C has preprocessing time T_{pre} and running time T, and each $C^{(i)}$ can be globally list recovered in time T', then the local list recovery algorithm for C_{con} has preprocessing time $T_{\mathrm{pre}} + Q \cdot T'$ and running time $T + Q \cdot T'$.

References

1 Noga Alon, László Babai, and Alon Itai. A fast and simple randomized parallel algorithm for the maximal independent set problem. *Journal of algorithms*, 7(4):567–583, 1986.

2 Noga Alon, Jeff Edmonds, and Michael Luby. Linear Time Erasure Codes with Nearly Optimal Recovery. In *proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 512–519. IEEE Computer Society, 1995.

3 Noga Alon and Michael Luby. A linear time erasure-resilient code with nearly optimal recovery. *IEEE Transactions on Information Theory*, 42(6):1732–1736, 1996.

4 László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking Computations in Polylogarithmic Time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–31. ACM Press, 1991. doi:10.1145/103418.103428.

5 Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Structures and Algorithms*, 28(4):387–402, 2006. doi:10.1002/rsa.20120.

6 Eli Ben-Sasson and Michael Viderman. Tensor Products of Weakly Smooth Codes are Robust. *Theory of Computing*, 5(1):239–255, 2009.

7 Eli Ben-Sasson and Michael Viderman. Composition of semi-LTCs by two-wise tensor products. *Computational Complexity*, 24(3):601–643, 2015. doi:10.1007/s00037-013-0074-8.

8 E. R. Berlekamp and L. Welch. Error correction of algebraic block codes. US Patent Number 4,633,470.

9 Don Coppersmith and Atri Rudra. On the robust testability of tensor products of codes. ECCC TR05-104, 2005. URL: https://eccc.weizmann.ac.il/eccc-reports/2005/TR05-104/index.html.

10 Irit Dinur, Madhu Sudan, and Avi Wigderson. Robust local testability of tensor products of LDPC codes. In *proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pages 304–315. Springer, 2006.

11 Katalin Friedl and Madhu Sudan. Some Improvements to Total Degree Tests. In *proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems (ISTCS)*, pages 190–198. IEEE Computer Society, 1995.

12 Edgar N. Gilbert. A comparision of signalling alphabets. *Bell System Technical Journal*, 31:504–522, 1952.

13 Oded Goldreich. A sample of samplers: A computational perspective on sampling. *def*, 1:2n, 1997.

14 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 Press, 1989.

15 Oded Goldreich and Or Meir. The tensor product of two good codes is not necessarily locally testable. *Information Processsing Letters*, 112(8-9):351–355, 2012.

16 Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost linear length. *Journal of ACM*, 53(4):558–655, 2006.

17 Sivakanth Gopi, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally Testable and Locally Correctable Codes approaching the Gilbert-Varshamov Bound. *IEEE Transactions on Information Theory*, 64(8):5813–5831, 2018.

18 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 34th Annual ACM Symposium on Theory of Computing (STOC)*, pages 812–821. ACM Press, 2002.

19 Venkatesan Guruswami and Piotr Indyk. Efficiently decodable codes meeting Gilbert-Varshamov bound for low rates. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithm (SODA)*, pages 756–757. SIAM, 2004. URL: `http://dl.acm.org/citation.cfm?id=982792`.

20 Venkatesan Guruswami and Swastik Kopparty. Explicit subspace designs. *Combinatorica*, 36(2):161–185, 2016. `doi:10.1007/s00493-014-3169-1`.

21 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.

22 Venkatesan Guruswami and Chaoping Xing. List decoding Reed-Solomon, Algebraic-Geometric, and Gabidulin subcodes up to the Singleton bound. In *Proceedings of the 45th annual ACM symposium on Theory of Computing (STOC)*, pages 843–852. ACM, 2013.

23 Brett Hemenway, Noga Ron-Zewi, and Mary Wootters. Local List Recovery of High-Rate Tensor Codes & Applications. In *Proceedings of the 58th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2017.

24 Brett Hemenway, Noga Ron-Zewi, and Mary Wootters. Local List Recovery of High-rate Tensor Codes & Applications. *Electronic Colloquium on Computational Complexity (ECCC)*, 24:104 (revision 1), 2017.

25 Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *STOC '00: Proceedings of the 32nd Annual Symposium on the Theory of Computing*, pages 80–86, 2000. `doi:10.1145/335305.335315`.

26 Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-Rate Locally Correctable and Locally Testable Codes with Sub-Polynomial Query Complexity. *Journal of ACM*, 64(2):11:1–11:42, 2017. `doi:10.1145/3051093`.

27 Swastik Kopparty, Noga Ron-Zewi, Shubhangi Saraf, and Mary Wootters. Improved List Decoding of Folded Reed-Solomon and Multiplicity Codes. In *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. IEEE Computer Society, 2018.

28 Or Meir. Combinatorial Construction of Locally Testable Codes. *SIAM Journal on Computing*, 39(2):491–544, 2009.

29 Irving S. Reed and Gustave Solomon. Polynomial Codes over Certain Finite Fields. *SIAM Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

30 Ronitt Rubinfeld and Madhu Sudan. Robust Characterization of Polynomials with Applications to Program Testing. *SIAM Journal of Computing*, 25(2):252–271, 1996.

31 Atri Rudra and Mary Wootters. Average-radius list-recoverability of random linear codes. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 644–662. SIAM, 2018.

32 Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom Generators without the XOR Lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. `doi:10.1006/jcss.2000.1730`.

33 Christian Thommesen. The existence of binary linear concatenated codes with Reed - Solomon outer codes which asymptotically meet the Gilbert- Varshamov bound. *IEEE Trans. Information Theory*, 29(6):850–853, 1983. `doi:10.1109/TIT.1983.1056765`.

34 Paul Valiant. The tensor product of two codes is not necessarily robustly testable. In *proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pages 472–481. Springer, 2005.

35 R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akadamii Nauk*, pages 739–741, 1957.

36 Michael Viderman. Strong LTCs with inverse poly-log rate and constant soundness. In *proceedings of the 54th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 330–339. IEEE Computer Society, 2013.

37 Michael Viderman. A combination of testability and decodability by tensor products. *Random Structures and Algorithms*, 46(3):572–598, 2015.

A Combinatorial lower bound on output list size

In this appendix, we first provide a *combinatorial* lower bound on the output list size for list recovering a high-rate tensor product $C^{\otimes t}$, even in the noiseless setting. In particular, we show that the output list size must be doubly-exponential in t. From this, we are able to deduce certain corollaries demonstrating that our algorithms nearly achieve optimal parameters.

Recall that given vectors $v_1 \in \mathbb{F}^{n_1}, v_2 \in \mathbb{F}^{n_2}, \ldots, v_t \in \mathbb{F}^{n_t}$, their tensor product $v_1 \otimes v_2 \otimes \cdots \otimes v_t$ is the t-dimensional box whose value in the $(i_1, i_2, \ldots, i_t) \in n_1 \times n_2 \cdots \times n_t$ coordinate is given by the product

$$(v_1 \otimes v_2 \otimes \cdots \otimes v_t)_{i_1,i_2,\ldots,i_t} = (v_1)_{i_1} \cdot (v_2)_{i_2} \cdots (v_t)_{i_t} \ .$$

For the special case of $t = 2$, the tensor product $v \otimes u$ can be thought of as the outer product vu^T.

We also record the following standard fact regarding tensor products.

▶ **Proposition 33.** *Let $v_1, \ldots, v_{t_1} \in \mathbb{F}^{n_1}$ and $u_1, \ldots, u_{t_2} \in \mathbb{F}^{n_2}$ be sets of linearly independent vectors. Then the collection $\{v_i \otimes u_j \mid i \in [t_1], j \in [t_2]\}$ is linearly independent in $\mathbb{F}^{n_1 \times n_2}$.*

A.1 Output list size for list recovering high-rate tensor codes

In this section we prove Theorem 6 from the introduction, which we restate here for convenience.

▶ **Theorem 6** (Output list size for list recovering high-rate tensor codes)**.** *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of rate $1-\gamma$, and that $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is $(0, \ell, L)$-list recoverable. Then $L \geq \ell^{1/\gamma^t}$.*

To prove this theorem, we first prove the following proposition. Informally speaking, we iteratively apply the Singleton bound to conclude that linear codes of rate $1-\gamma$ contain about $1/\gamma$ codewords with pairwise disjoint supports. Recall that, for a vector $v \in \mathbb{F}^n$, the support of v is $\mathrm{Supp}(v) = \{i \in [n] \mid v_i \neq 0\}$.

▶ **Proposition 34.** *Let $C \subseteq \mathbb{F}^n$ be a subspace of dimension k, and let r be a positive integer. Suppose that*

$$\left(1 - \tfrac{1}{r}\right) \cdot n + 1 \leq k \ . \tag{3}$$

Then there exist non-zero vectors $c_1, \ldots, c_r \in C$ such that for all $i \neq j$, $\mathrm{Supp}(c_i) \cap \mathrm{Supp}(c_j) = \emptyset$.

Proof. Let $m := n - k + 1$, and note that Condition (3) is equivalent to

$$(r-1)m \leq k - 1 \ .$$

Take a basis for C of the form $(e_1, u_1), \ldots, (e_k, u_k)$, where $e_i \in \mathbb{F}^k$ is the ith standard basis vector, and $u_1, \ldots, u_k \in \mathbb{F}^{n-k}$ are vectors. For $j = 1, \ldots, r-1$, we can find a nontrivial linear combination of the vectors $u_{(j-1)\cdot m+1}, \ldots, u_{j \cdot m}$ summing to zero, as they

are a (multi-)set of $m = n - k + 1$ vectors lying in $\mathbb{F}^{n-k}$. Taking this linear combination of $(e_{(j-1)\cdot m+1}, u_{(j-1)\cdot m+1}), \ldots, (e_{j\cdot m}, u_{j\cdot m})$, we obtain a nonzero vector whose support is contained in the interval $\{(j-1)\cdot m+1, \ldots, j\cdot m\}$; denote this vector by c_j. In this manner, we obtain $r-1$ nonzero vectors $c_1, \ldots, c_{r-1} \in C$ with pairwise disjoint support. Finally, we may add the vector $c_r := (e_k, u_k)$ to this collection, yielding r vectors, as desired. ◀

Next we prove Theorem 6, based on the above proposition.

Proof of Theorem 6. Let $r := 1/\gamma$, and recall wish to come up with ℓ^{r^t} codewords in $C^{\otimes t}$ that are contained in the output list for appropriately chosen input lists.

In order to accomplish this, we first use Proposition 34 to obtain a subset $C' \subseteq C$ of r nonzero codewords with pairwise disjoint support. We then consider the subset $C'' \subseteq C^{\otimes t}$ containing all tensor products $c_1 \otimes c_2 \otimes \cdots \otimes c_t$ of t (not necessarily distinct) codewords $c_1, \ldots, c_t \in C'$, and our main observation is that all these r^t tensor products are also nonzero with pairwise disjoint support. Finally, we let $B \subseteq \mathbb{F}$ be an arbitrary subset of size ℓ, and consider the subset $\bar{C} \subseteq C^{\otimes t}$ containing all linear combinations of codewords in C'' with coefficients in B. Since all codewords in C'' are nonzero with pairwise disjoint support, they are in particular linearly independent, so the set $\bar{C}$ contains ℓ^{r^t} distinct codewords in $C^{\otimes t}$.

Moreover, since codewords in C'' have pairwise disjoint support, for each coordinate $(i_1, \ldots, i_t) \in [n]^t$, there is at most one codeword $c \in C''$ for which $c_{i_1,\ldots,i_t}$ is nonzero. Therefore this is the only term which can contribute nontrivially to the value in the $(i_1, \ldots, i_t)$ coordinate of a codeword in $\bar{C}$. So we can let the corresponding input list $S_{i_1,\ldots,i_t}$ contain all ℓ multiples of $c_{i_1,\ldots,i_t}$ by elements in B. Details follow.

Since C has rate $1-\gamma$, it has dimension $k = (1-\gamma)n$, and so Proposition 34 guarantees the existence of a subset $C' \subseteq C$ of $r = 1/\gamma$ nonzero codewords with pairwise disjoint support.

Next we let

$$C'' := \left\{c_1 \otimes c_2 \otimes \cdots \otimes c_t \mid c_1, c_2, \ldots, c_t \in C'\right\}$$

be the subset of $C^{\otimes t}$ containing all tensor products of t (not necessarily distinct) codewords in C'. Since all codewords in C' are nonzero, their t-wise tensor products are nonzero as well.

To see that all codewords in C'' have pairwise disjoint support, suppose that $c = c_1 \otimes c_2 \otimes \ldots \otimes c_t \in C''$, and $(i_1, i_2, \ldots, i_t) \in \mathrm{Supp}(c)$. Then

$$0 \neq c_{i_1,i_2,\ldots,i_t} = (c_1)_{i_1} \cdot (c_2)_{i_2} \cdots (c_t)_{i_t},$$

so we must have that $(c_1)_{i_1}, (c_2)_{i_2}, \ldots, (c_t)_{i_t}$ are all nonzero. We conclude that

$$\mathrm{Supp}(c) \subseteq \mathrm{Supp}(c_1) \times \mathrm{Supp}(c_2) \times \cdots \times \mathrm{Supp}(c_t).$$

Now, suppose that $c = c_1 \otimes \ldots \otimes c_t$, $c' = c'_1 \otimes \ldots \otimes c'_t$ are a pair of codewords in C'' with $c_j \neq c'_j$ for some $j \in [t]$. Since all codewords in C' have pairwise disjoint support it must hold that $\mathrm{Supp}(c_j) \cap \mathrm{Supp}(c'_j) = \emptyset$, and we conclude that $\mathrm{Supp}(c) \cap \mathrm{Supp}(c') = \emptyset$.

Now, let $B \subseteq \mathbb{F}$ be an arbitrary subset of size ℓ, and let

$$\bar{C} := \left\{ \sum_{c \in C''} \beta_c \cdot c \;\middle|\; \beta_c \in B \text{ for all } c \in C'' \right\}$$

be the subset of $C^{\otimes t}$ containing all linear combinations of codewords in C'' with coefficients in B. Since all codewords in C'' are nonzero with pairwise disjoint support, they are in particular linearly independent in $\mathbb{F}^{n^t}$,[6] so the set $\bar{C}$ contains ℓ^{r^t} distinct codewords in $C^{\otimes t}$.

[6] This also follows from the fact that all codewords in C'' are linearly independent together with Proposition 33.

Finally, we wish to define input lists $S_{i_1,\dots,i_t}$ for any coordinate $(i_1,\dots,i_t) \in [n]^t$ so that for any codeword $c \in \bar{C}$, and for any coordinate $(i_1,\dots,i_t) \in [n]^t$, it holds that $c_{i_1,\dots,i_t} \in S_{i_1,\dots,i_t}$.

To this end, we observe that since codewords in C'' have pairwise disjoint support, for each coordinate $(i_1,\dots,i_t) \in [n]^t$, there is at most one codeword $c \in C''$ for which $c_{i_1,\dots,i_t}$ is nonzero. Therefore this is the only term which can contribute nontrivially to the value in the $(i_t,\dots,i_t)$ coordinate of a codeword in $\bar{C}$. So we can define the corresponding input list $S_{i_1,\dots,i_t}$ as

$$S_{i_1,\dots,i_t} := \{\beta \cdot c_{i_1,\dots,it} \mid \beta \in B\}$$

if such a codeword c exists, and as $S_{i_1,\dots,i_t} = \{0\}$ otherwise. Note that each set $S_{i_1,\dots,i_t}$ has size at most ℓ, and that they satisfy the required property.

This yields a set of ℓ^{r^t} codewords from $C^{\otimes t}$ that are contained in the output list for the input list tuple S defined above, proving the theorem. ◀

A.2 Concrete lower bound on output list size

In this section, we demonstrate a setting of parameters that yields Corollary 7 from the introduction, restated below.

▶ **Corollary 7.** *For any $\delta > 0$ and $\ell > 1$ there exists $L > 1$ such that the following holds for any sufficiently large n. There exists a linear code $C \subseteq \mathbb{F}^n$ of relative distance δ that is $(\Omega(\delta), \ell, L)$-list recoverable, but $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is only $(0, \ell, L')$-list recoverable for $L' \geq \exp((2\delta)^{-(t-3/2)} \cdot \sqrt{\log L})$.*

We use the following result on the list-recoverability of random linear codes from [31].

▶ **Theorem 35** ([31], Corollary 3.3)**.** *There exists an absolute constant b_0 so that the following holds. For any $\gamma > 0$, $\ell \geq 1$, and a prime power $q \geq \ell^{b_0/\gamma}$, a random linear code $C \subseteq \mathbb{F}_q^n$ of rate $1-\gamma$ is $(\Omega(\gamma), \ell, L)$-list recoverable for*

$$L \leq \left(\frac{q\ell}{\gamma}\right)^{(\log \ell)/\gamma} \cdot \exp\left(\frac{\log^2 \ell}{\gamma^3}\right)$$

with probability $1 - \exp(-n)$.

Proof of Corollary 7. Let $C \subseteq \mathbb{F}_q^n$ be the linear code given by Theorem 35 of rate $1-2\delta$ and $q = \ell^{O(1/\delta)}$ that is $(\Omega(\delta), \ell, L)$-list recoverable for $L = \exp((\log^2 \ell)/\delta^3)$, or equivalently, $\ell = \exp(\delta^{3/2} \cdot \sqrt{\log L})$. By Corollary 10, we may further assume that the code C has relative distance at least δ. Now, by Theorem 6 we have that $L' \geq \ell^{(2\delta)^{-t}} = \exp((2\delta)^{-(t-3/2)} \cdot \sqrt{\log L})$. ◀

A.3 Lower bound for local list recovering

We now prove Corollary 8 from the introduction, restated below.

▶ **Corollary 8.** *For any $\delta > 0$ and sufficiently large n there exists a linear code $C \subseteq \mathbb{F}^n$ of relative distance δ such that the following holds. Suppose that $C^{\otimes t} \subseteq \mathbb{F}^N$ is $(\frac{1}{N}, 2, L)$-locally list recoverable with query complexity Q. Then $Q \cdot L \geq N^{\Omega_\delta(1/\log\log N)}$.*

We first show the following lemma which says that a locally list decodable (and in particular locally list recoverable) code with output list size L and query complexity Q is also locally correctable with query complexity roughly $Q \cdot L$.

▶ **Lemma 36.** *Suppose that $C \subseteq \Sigma^n$ is a code of relative distance δ that is $(Q, \alpha, 0.1, L)$-locally list decodable for $\alpha < \delta/2$. Then C is $\left(O\left(Q \cdot L \cdot \frac{\log^2 n}{(\delta/2-\alpha)^2}\right), \alpha\right)$-locally correctable.*

So to prove Corollary 8, it is enough to show a lower bound on the query complexity for local correcting $C^{\otimes t}$, assuming that the output list for list recovering $C^{\otimes t}$ is small. To show such a lower bound, we first observe that for any linear code C, the (absolute) distance of $C^\perp$ is a lower bound on the query complexity for local correcting C.

▶ **Lemma 37.** *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code that is $(Q, \frac{1}{n})$-locally correctable. Then $Q \geq \Delta(C^\perp) - 2$.*

We prove the above lemma in Section A.3.1. To apply this lemma to $C^{\otimes t}$ we further observe that the tensor product preserves the dual distance of the base code.

▶ **Lemma 38.** *Suppose that $C_1 \subseteq \mathbb{F}^{n_1}$, $C_2 \subseteq \mathbb{F}^{n_2}$ are linear codes, and that $C_1^\perp, C_2^\perp$ have distances Δ_1, Δ_2, respectively. Then $(C_1 \otimes C_2)^\perp$ has distance $\min\{\Delta_1, \Delta_2\}$. In particular, if $C \subseteq \mathbb{F}^n$ is a linear code, and $C^\perp$ has distance Δ, then $(C^{\otimes t})^\perp$ has distance Δ for any $t \geq 1$.*

We prove the above lemma in Section A.3.2. We now proceed to the proof of Corollary 8.

Proof of Corollary 8. Let $C \subseteq \mathbb{F}^n$ be a random linear code of rate $1 - 2\delta$. By Corollary 10, for sufficiently large field size, the code C will have relative distance at least δ with high probability. Moreover, since $C^\perp$ has rate 2δ, by the same corollary we also have that $C^\perp$ has relative distance at least $1 - 3\delta$ with high probability. We conclude for any sufficiently large n the existence of a linear code $C \subseteq \mathbb{F}^n$ of rate $1 - 2\delta$ and relative distance at least δ such that $C^\perp$ has relative distance at least $1 - 3\delta$.

Next observe that for the code $C^{\otimes t}$ to be $(Q, \frac{1}{N}, 0.1, 2, L)$-locally list recoverable, it in particular must be $(0, 2, L)$-list recoverable, so the lower bound from Theorem 6 implies that $L \geq 2^{1/(2\delta)^t}$. Now, if $2^{1/(2\delta)^t} \geq N$ then we have that $Q \cdot L \geq 2^{1/(2\delta)^t} \geq N$, and we are done. So we may assume that $2^{1/(2\delta)^t} < N$ which implies in turn that $t = O_\delta(\log \log N)$ and $n = N^{1/t} = N^{\Omega_\delta(1/\log \log N)}$.

Moreover, as we have assumed we have a $(Q, \frac{1}{N}, 0.1, 2, L)$-local list recovery algorithm for $C^{\otimes t}$, we also have a $(Q, \frac{1}{N}, 0.1, L)$-local list decoding algorithm for $C^{\otimes t}$. Lemma 36 then promises that we have a $(O(Q \cdot L \cdot \frac{\log^2 N}{(\delta^t/2 - 1/N)^2}), \frac{1}{N})$-local correction algorithm for $C^{\otimes t}$. Now, by Lemma 38 we have that $(C^{\otimes t})^\perp$ has (absolute) distance at least $(1 - 3\delta)n$, and consequently Lemma 37 implies that

$$O\left(Q \cdot L \cdot \frac{\log^2 N}{(\delta^t/2 - \frac{1}{N})^2}\right) \geq (1 - 3\delta)n - 2 = N^{\Omega_\delta(1/\log \log N)} .$$

This implies $Q \cdot L \geq N^{\Omega_\delta(1/\log \log N)}$, as desired. ◀

A.3.1 Dual distance is a lower bound on query complexity – proof of Lemma 37

First, we recall the standard fact that (absolute) dual distance Δ implies that the uniform distribution over the code is $(\Delta - 1)$-wise independent.

▶ **Proposition 39** ([1])**.** *Suppose that $C \subseteq \mathbb{F}_q^n$ is a linear code, and that $C^\perp$ has (absolute) distance Δ. Then for all $1 \leq i_1 < \cdots < i_s \leq n$ with $s < \Delta$, and all $a_1, \ldots, a_s \in \mathbb{F}_q$,*

$$\Pr_{c \in C}\left[c_{i_1} = a_1 \wedge \cdots \wedge c_{i_s} = a_s\right] = \frac{1}{q^s}.$$

In what follows let $\Delta := \Delta(C^\perp)$, and let q denote the alphabet size of C. Now, making use of Yao's principle, it suffices to show a distribution $\mathcal{D}$ over vectors w at absolute distance at most 1 from C such that the following holds. For any *deterministic* algorithm making at most $\Delta - 2$ queries to its input w sampled according to $\mathcal{D}$, the probability that it correctly computes c_1 is at most $1/3$, where c is the unique codeword in C at absolute distance at most 1 from w. We will in fact show that no deterministic query algorithm can correctly compute c_1 with probability greater than $1/q$.

Let $\mathcal{D}$ denote the distribution that samples $c \in C$ uniformly at random and then sets $c_1 = 0$. Let A be a deterministic algorithm making at most $\Delta - 2$ queries, and let $j_1, \ldots, j_s \in [n]$ denote the queries made by A, where we assume $s \leq \Delta - 2$. Note that querying 1 does not help A, as it will always read 0. Hence, without loss of generality, $1 \notin \{j_1, \ldots, j_s\}$.

Now, by Proposition 39 and Bayes' rule, for any $b_1, \ldots, b_s, a \in \mathbb{F}_q$,

$$\Pr_{c \in C}[c_1 = a | c_{j_1} = b_1, \ldots, c_{j_s} = b_s] = \frac{\Pr[c_1 = a, c_{j_1} = b_1, \ldots, c_{j_s} = b_s]}{\Pr[c_{j_1} = b_1, \ldots, c_{j_s} = b_s]} = \frac{q^{-(s+1)}}{q^{-s}} = \frac{1}{q} .$$

Additionally, observe that the distribution of the tuple $(c_{j_1}, \ldots, c_{j_s})$ is the same if c is a uniformly random codeword from C or if it is sampled according to $\mathcal{D}$.

Hence, if we think of the query algorithm as implementing a (deterministic) function $g : \mathbb{F}_q^s \to \mathbb{F}_q$ from the responses to its queries to its guess for c_1, regardless of the responses $b_1, \ldots, b_s$ to the queries, we have

$$\Pr_{w \in \mathcal{D}}[c_1 = g(b_1, \ldots, b_s) | w_{j_1} = b_1, \ldots, w_{j_s} = b_s] = \frac{1}{q} ,$$

where c is the unique codeword in C for which $\mathrm{dist}(c, w) \leq \frac{1}{n}$. That is, the query algorithm will not be able to guess c_1 with probability greater than $1/q$, as claimed.

A.3.2 Tensor product preserves dual distance – proof of Lemma 38

First note that we clearly have that $\Delta((C_1 \otimes C_2)^\perp) \leq \min\{\Delta_1, \Delta_2\}$: for example, the matrix whose first column is a vector from $C_1^\perp$ of weight Δ_1 and all other columns are 0 gives a matrix in $(C_1 \otimes C_2)^\perp$ of weight Δ_1, and similarly a matrix in $(C_1 \otimes C_2)^\perp$ of weight Δ_2 can be constructed. We now establish the opposite inequality of $\Delta((C_1 \otimes C_2)^\perp) \geq \min\{\Delta_1, \Delta_2\}$.

It is well-known (and not hard to show) that the (absolute) distance of a code C is the minimum number of linearly dependent columns in a parity-check matrix for C. Furthermore, by duality we have that if G is a generating matrix for C then G^T is a parity-check matrix for $C^\perp$. We conclude that the distance of $C^\perp$ is the minimum number of linearly dependent rows in a generating matrix for C.

Let G_1, G_2 be generating matrices for C_1, C_2, respectively, and note that by the above, any collection of $t_1 < \Delta_1, t_2 < \Delta_2$ rows of G_1, G_2, respectively, are linearly independent. Next recall that $G_1 \otimes G_2$ is a generating matrix for $C_1 \otimes C_2$, and so it suffices to show that for any $t < \min\{\Delta_1, \Delta_2\}$, any collection of t rows of $G_1 \otimes G_2$ are linearly independent.

Let $u_1, u_2, \ldots, u_{n_1}$ and $v_1, v_2, \ldots, v_{n_2}$ denote the rows of G_1, G_2, respectively, and note that each row in $G_1 \otimes G_2$ is of the form $u_i \otimes v_j$ for some $i \in [n_1], j \in [n_2]$. Fix $t < \min\{\Delta_1, \Delta_2\}$, and suppose that $u_{i_1} \otimes v_{j_1}, u_{i_2} \otimes v_{j_2}, \ldots, u_{i_t} \otimes v_{j_t}$ is a collection of t rows of $G_1 \otimes G_2$. Then by the above we have that both collections $u_{i_1}, u_{i_2}, \ldots, u_{i_t}$ and $v_{j_1}, v_{j_2}, \ldots, v_{j_t}$ are linearly independent (ignoring duplications). Proposition 33 implies in turn that the collection $u_{i_1} \otimes v_{j_1}, u_{i_2} \otimes v_{j_2}, \ldots, u_{i_t} \otimes v_{j_t}$ are also linearly independent which concludes the proof of the lemma.