Electronic Colloquium on Computational Complexity, Report No. 80 (2019)

On List Recovery of High-Rate Tensor Codes

Swastik Kopparty* Nicolas Resch † Noga Ron-Zewi‡ Shubhangi Saraf§
Shashwat Silas ¶

June 1, 2019

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 was 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 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.

*Department of Mathematics and Department of Computer Science, Rutgers University. swastik.kopparty@gmail.com.

†Department of Computer Science, Carnegie Mellon University. nresch@andrew.cmu.edu.

‡Department of Computer Science, University of Haifa. noga@cs.haifa.ac.il.

§Department of Mathematics and Department of Computer Science, Rutgers University. shubhangi.saraf@gmail.com.

¶Department of Computer Science, Stanford University. silas@stanford.edu .

ISSN 1433-8092

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.

1.1 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 [GR08] (see also [KRSW18]

[1] Here H_2 is the binary entropy function.

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.

1.2 The context

The starting point for this paper is the recent result of [HRW17a] on high-rate list recoverable tensor codes, and its corollaries. Tensoring is a natural operation on codes that significantly enhances their local properties [BS06, Val05, CR05, DSW06, GM12, BV09, BV15, Vid15, Mei09, Vid13, KMRS17].

[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.

The main technical result of [HRW17a] 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 [AEL95, AL96] (specialized to the setting of local list recovery [GI02, GKO$^+$18]), 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, [HRW17a] 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 [AEL95, AL96, GI02], 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 [Tho83, GI04], 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 [GKO$^+$18] which implemented a local version of [Tho83, GI04]) 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 [HRW17a] 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).

1.3 Results

We revisit the technique of [HRW17a] 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 [AEL95, AL96, GI02], 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 [Tho83, GI04], 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 [HRW17a] 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 [AEL95, AL96, GKO$^+$18], we get capacity-achieving locally list recoverable codes, but now in the *local correction version*. This now plugs in turn into the machinery of [Tho83, GI04, GKO$^+$18] 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 [GKO+18] 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 [HRW17a] 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 [Vid15] 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.

1.3.1 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 1.1 (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 $n^t \cdot T \cdot L^{s^{t^3} \cdot L^t}$.*

[3] To eliminate 'garbage' we also use the fact that the tensor product is locally testable [Vid15].

Applying the expander-based distance amplification method of [AEL95, AL96, GI02] 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 1.2 (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 [Tho83, GI04], the above corollary yields in turn constant-rate codes approaching the Gilbert-Varshamov bound with *deterministic* near-linear time global unique decoding algorithms.

Corollary 1.3 (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 $N^{1+o(1)}$.*

1.3.2 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 1.4 (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 [AEL95, AL96, GI02, GKO+18], as well as the random concatenation method of [Tho83, GI04, GKO+18], the above theorem yields constant-rate codes approaching the Gilbert-Varshamov bound that are *locally correctable* with query complexity $N^{o(1)}$.

Corollary 1.5 (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 $N^{o(1)}$.*

1.3.3 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 1.6 (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 1.7. *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 1.8. *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 2.1 (Gilbert-Varshamov (GV) bound, [Gil52, Var57])**.** *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 2.2. *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 $\mathrm{dist}(c, w) \leq \alpha$.

Fact 2.3 (Reed-Solomon codes, [RS60, BW])**.** *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.

Theorem 2.4 ([Gur01], Theorem 5.3)**.** *For any prime power q, $\alpha \in (0, 1 - \frac{1}{q})$, $\rho \in (0, 1 - H_q(\alpha) - 1/\log_q(L+1))$, a random linear code $C \subseteq \mathbb{F}_q^n$ of rate ρ is (α, L)-list decodable with probability* $1 - \exp(-n)$.

Theorem 2.5 ([Gur01], Lemma 9.6)**.** *For any prime power q, integers $1 \leq \ell \leq q$ and $L > \ell$, $\alpha \in (0, 1)$, and $\rho \in [0, 1]$ which is at most*

$$\frac{1}{\log q} \cdot \left[(1 - \alpha) \cdot \log(q/\ell) - H_2(\alpha) - H_2(\ell/q) \cdot \frac{q}{\log_q(L+1)} \right],$$

a random linear code $C \subseteq \mathbb{F}_q^n$ of rate ρ is (α, ℓ, L)-list recoverable with probability $1 - \exp(-n)$.

Corollary 2.6 ([HRW17b], 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 [GX13, GK16, HRW17a] gives a family of high-rate linear codes which are efficiently list recoverable with constant alphabet size and nearly-constant output list size.

Theorem 2.7 ([HRW17b], 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

Locally testable codes. Intuitively, a code is said to be locally testable [FS95, RS96, GS06] 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 2.8 (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 2.9. 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 [BFLS91, STV01, KT00] 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 2.10 (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 2.11. 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 [GL89, STV01, GKO+18] 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 2.12 (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.

Remark 2.13. The above definition of locally list recoverable code defers from that given in [HRW17a, Definition 4.5] in two ways. First, our definition requires that the local algorithms $A_1, \ldots, A_L$ in the output list of A locally decode codeword coordinates as opposed to message coordinates. Second, following [GKO+18], we require an additional soundness property that guarantees that with high probability, each local algorithm in the output list locally decodes a true codeword. These two requirements will be crucial for our GV bound local correction application (Corollary 1.5).

2.4 Tensor codes

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

Definition 2.14 (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.

The following are some well-known facts about the tensor product operation, and its effect on the classical parameters of a code.

Fact 2.15. *Suppose that $C_1 \subseteq \mathbb{F}^{n_1}$, $C_2 \subseteq \mathbb{F}^{n_2}$ are linear codes of rates ρ_1, ρ_2 and relative distances δ_1, δ_2 respectively. Then the tensor product code $C_1 \otimes C_2 \subseteq \mathbb{F}^{n_1 \times n_2}$ is a linear code of rate $\rho_1 \cdot \rho_2$ and relative distance $\delta_1 \cdot \delta_2$.*

Moreover, if C_1, C_2 are encodable in times T_1, T_2, respectively, then $C_1 \otimes C_2$ is encodable in time $n_1 T_2 + n_2 T_1$, and if C_1, C_2 are decodable from α_1, α_2-fraction of errors in times T_1, T_2, respectively, then $C_1 \otimes C_2$ is decodable from $(\alpha_1 \cdot \alpha_2)$-fraction of errors in time $n_1 T_2 + n_2 T_1$.

For a linear code C, let $C^{\otimes 1} := C$ and $C^{\otimes t} := C \otimes C^{\otimes (t-1)}$. By induction on t we have the following.

Corollary 2.16. *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of rate ρ and relative distance δ. Then the tensor product code $C^{\otimes t} \subseteq \mathbb{F}^{n^t}$ is a linear code of rate ρ^t and relative distance δ^t.*

Moreover, if C is encodable in time T then $C^{\otimes t}$ is encodable in time $t \cdot n^{t-1} \cdot T$, and if $C^{\otimes t}$ is decodable from α-fraction of errors in time T then $C^{\otimes t}$ is decodable from α^t-fraction of errors in time $t \cdot n^{t-1} \cdot T$.

For a pair of matrices $G_1 \in \mathbb{F}^{n_1 \times k_1}$ and $G_2 \in \mathbb{F}^{n_2 \times k_2}$, their *tensor product* $G_1 \otimes G_2$ is the $(n_1 \cdot n_2) \times (k_1 \cdot k_2)$-matrix over $\mathbb{F}$ with entries

$$(G_1 \otimes G_2)_{(i_1,i_2),(j_1,j_2)} = (G_1)_{i_1,j_1} \cdot (G_2)_{i_2,j_2}$$

for every $i_1 \in [n_1]$, $i_2 \in [n_2]$, $j_1 \in [k_1]$, and $j_2 \in [k_2]$.

Fact 2.17. *Suppose that G_1, G_2 are generating matrices of linear codes $C_1 \subseteq \mathbb{F}^{n_1}, C_2 \subseteq \mathbb{F}^{n_2}$, respectively. Then the tensor product $G_1 \otimes G_2$ is a generating matrix of $C_1 \otimes C_2$.*

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 1.1, 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 1.1 (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 1.1 follows by applying the lemma below iteratively.

Lemma 3.1. *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

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

Before we prove the above lemma, we first show how it implies Theorem 1.1.

Proof of Theorem 1.1. We start with the code C, and iteratively tensor with a new copy of C for $t-1$ times. Specifically, we initially set $C' := C$, and at each step we apply Lemma 3.1 with the code C' being the code constructed so far, and the code C being a new copy of C.

On each iteration, we can set in Lemma 3.1 $\delta_{\mathrm{dec}} := \min\{\alpha, \delta/2\}$ and $T_{\mathrm{dec}} := T$ since the code C can be uniquely decoded from δ_{dec}-fraction of errors by running the list recovery algorithm for C on the received word, and returning the codeword from the output list that is closest to the received word. Moreover, by Corollary 2.16, on the i-th iteration we can set $\delta'_{\mathrm{dec}} := \delta^t_{\mathrm{dec}}$ and $T'_{\mathrm{dec}} := i \cdot n^{i-1} \cdot T$. We conclude that on each iteration we can apply Lemma 3.1 with $\bar{s} := s^t$ for $s = \mathrm{poly}(1/\delta, 1/\alpha)$.

In the above setting of parameters, we have that the list recovery radius of $C^{\otimes t}$ is at least $\tilde{\alpha} := \alpha/\bar{s}^t = \alpha/s^{t^2}$, and that the output list size is at most $\tilde{L} := L^{\bar{s}^t \cdot L^t/\tilde{\alpha}^{2t}} \leq L^{s^{O(t^3)} \cdot L^t}$. Finally, on the i-th iteration the running time is increased by an additive factor of $n' \cdot (T + T_{\mathrm{dec}}) + n \cdot T'_{\mathrm{dec}} = O(i \cdot n^i \cdot T)$, and then by a multiplicative factor of at most $\tilde{L} \cdot n$, yielding a total running time of at most

$$\sum_{i=1}^{t-1} O(i \cdot n^i \cdot T) \cdot \left(\tilde{L} \cdot n\right)^{t-i} \cdot T \leq L^{s^{O(t^3)} \cdot L^t} \cdot n^t \cdot T.$$

So the desired conclusion holds by slightly enlarging the size of the polynomial s. □

We now proceed to the proof of Lemma 3.1. Our plan is to derandomize the approximate local list recovery algorithm for high-rate tensor codes of [HRW17a]. 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 [HRW17a], 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 [HRW17a] 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.

[5] In [HRW17a], the role of columns and rows is swapped.

3.1.1 Samplers

We start by defining the appropriate samplers we use.

Definition 3.2 ((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 [Gol97].

Theorem 3.3 ([Gol97], 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 [HRW17a] 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.

3.1.3 Output list size, randomness, and running time

The output list size is at most the number of choices of $c'_1 \in \mathcal{L}'_1, c'_2 \in \mathcal{L}'_2, \ldots, c'_m \in \mathcal{L}'_m$ which is $(L')^m \leq (L')^{L \cdot \bar{s}/(\alpha')^2}$, and the randomness is $r \leq \log\left(n \cdot \bar{s}/\alpha'\right)$. As to running time, the algorithm $\tilde{A}$ invokes the sampler Γ, followed by m invocations of the list recovery algorithm A' for C', and $(L')^m \cdot n'$ invocations of the list recovery algorithm A for C. Finally, it invokes $(L')^m$ times the

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

unique decoding algorithm for $C \otimes C'$ which can be implemented to run in time $n \cdot T'_{\mathrm{dec}} + n' \cdot T_{\mathrm{dec}}$ by Fact 2.15. Thus the total running time is at most

$$\begin{aligned} &\mathrm{poly}(\log n, m) + m \cdot T' + (L')^m \cdot n' \cdot T + (L')^m \cdot (n \cdot T'_{\mathrm{dec}} + n' \cdot T_{\mathrm{dec}}) \\ &\leq (L')^{\bar{s} \cdot L/(\alpha')^2} \cdot \left(n' \cdot (T + T_{\mathrm{dec}}) + n \cdot T'_{\mathrm{dec}} + T'\right), \end{aligned}$$

where the inequality holds for a sufficiently large polynomial $\bar{s}$.

3.1.4 Correctness

Next we establish the following.

Claim 3.4. *Suppose that $\tilde{c} \in C \otimes C'$ has $\mathrm{dist}(\tilde{c}, S) \leq \alpha'/\bar{s}$. Then with probability at least $2/3$, the codeword $\tilde{c}$ is included in $\tilde{\mathcal{L}}$.*

Note that the above claim in particular implies that there are at most $O((L')^m)$ codewords $\tilde{c} \in C \otimes C'$ with $\mathrm{dist}(\tilde{c}, S) \leq \alpha'/\bar{s}$. To prove the above claim, it is enough to show that with probability at least $2/3$ over the choice of $I = \{i_1, \ldots, i_m\}$, there exists a choice of $c'_1 \in \mathcal{L}'_{i_1}, c'_2 \in \mathcal{L}'_{i_2}, \ldots, c'_m \in \mathcal{L}'_{i_m}$ such that at the iteration corresponding to $c'_1, c'_2, \ldots, c'_m$ the word $\tilde{c}_0$ satisfies that $\mathrm{dist}(\tilde{c}_0, \tilde{c}) \leq \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}$. Once we establish this, the unique decoding algorithm for $C \otimes C'$ will successfully decode $\tilde{c}$ from $\tilde{c}_0$.

For a row $i \in [n]$, let $\hat{c}_i$ be the codeword in $\mathcal{L}'_i$ that is closest to the i-th row of $\tilde{c}$ (breaking ties arbitrarily), that is, the codeword $\hat{c}_i \in \mathcal{L}'_i$ for which $\mathrm{dist}(\hat{c}_i, \tilde{c}|_{\{i\} \times [n']})$ is minimal. We will show that with probability at least $2/3$ over the choice of $I = \{i_1, \ldots, i_m\}$, at the iteration corresponding to the choice of $\hat{c}_{i_1} \in \mathcal{L}'_{i_1}, \hat{c}_{i_2} \in \mathcal{L}'_{i_2}, \ldots, \hat{c}_{i_m} \in \mathcal{L}'_{i_m}$, the word $\tilde{c}_0$ will satisfy that $\mathrm{dist}(\tilde{c}_0, \tilde{c}) \leq \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}$.

Following [HRW17a], to establish the above, we show that with high probability over the choice of I, a large fraction of the columns $j \in [n']$ are "good", in the sense that $\tilde{c}_0$ and $\tilde{c}$ agree on all of these columns at the iteration corresponding to the choice of $\hat{c}_{i_1} \in \mathcal{L}'_{i_1}, \hat{c}_{i_2} \in \mathcal{L}'_{i_2}, \ldots, \hat{c}_{i_m} \in \mathcal{L}'_{i_m}$. In what follows, let $\tilde{c}_1, \tilde{c}_2 \ldots, \tilde{c}_{n'}$ denote the columns of $\tilde{c}$.

Definition 3.5 (Good column)**.** Let $I = \{i_1, \ldots, i_m\} \subseteq [n]$ be a subset of m rows. We say that a column $j \in [n']$ is good with respect to I if it satisfies the following properties:

1. The codeword $\tilde{c}$ is consistent with all but an α-fraction of the input lists on column j, that is, $\mathrm{dist}(\tilde{c}_j, S|_{[n]\times\{j\}}) \leq \alpha$.
2. Let $\mathcal{L}_j$ denote the list of all codewords in C that are consistent with all but an α-fraction of the input lists on column j. Then for any $c \in \mathcal{L}_j \setminus \{\tilde{c}_j\}$ it holds that $\mathrm{dist}(c|_I, \tilde{c}_j|_I) > \delta/2$.
3. $\mathrm{dist}\left(\tilde{c}_j|_I, ((\hat{c}_{i_1})_j, \ldots, (\hat{c}_{i_m})_j)\right) \leq \delta/4$.

Claim 3.6 below shows that at the iteration corresponding to the choice of $\hat{c}_{i_1} \in \mathcal{L}'_{i_1}, \hat{c}_{i_2} \in \mathcal{L}'_{i_2}, \ldots, \hat{c}_{i_m} \in \mathcal{L}'_{i_m}$, $\tilde{c}_0$ and $\tilde{c}$ agree on all of the good columns. Claim 3.7 complements this by showing that with probability at least $2/3$ over the choice of I, at least a $(1 - \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}})$-fraction of the columns are good with respect to I. The combination of these claims yields the desired conclusion.

Claim 3.6. *Let $I = \{i_1, \ldots, i_m\} \subseteq [n]$ be a subset of m rows, and suppose that a column $j \in [n']$ is good with respect to I. Then at the iteration corresponding to the choice of $\hat{c}_{i_1} \in \mathcal{L}'_{i_1}, \hat{c}_{i_2} \in \mathcal{L}'_{i_2}, \ldots, \hat{c}_{i_m} \in \mathcal{L}'_{i_m}$ it holds that $\tilde{c}_0|_{[n]\times\{j\}} = \tilde{c}_j$.*

Proof. By Property (1) in the definition of a good column, $\tilde{c}$ is consistent with all but an α-fraction of the input lists on column j, and so $\tilde{c}_j \in \mathcal{L}_j$. By Property (3),

$$\mathrm{dist}\left(\tilde{c}_j|_I, ((\hat{c}_{i_1})_j, \ldots, (\hat{c}_{i_m})_j)\right) \leq \delta/4.$$

On the other hand, by Property (2) for any other codeword $c \in \mathcal{L}_j$ we have that

$$\mathrm{dist}\left(c|_I, ((\hat{c}_{i_1})_j, \ldots, (\hat{c}_{i_m})_j)\right) \geq \mathrm{dist}\left(\tilde{c}_j|_I, c|_I\right) - \mathrm{dist}\left(\tilde{c}_j|_I, ((\hat{c}_{i_1})_j, \ldots, (\hat{c}_{i_m})_j)\right) > \delta/4.$$

Thus, $\tilde{c}_j$ is the codeword in $\mathcal{L}_j$ whose restriction to I is closest to $((\hat{c}_{i_1})_j, \ldots, (\hat{c}_{i_m})_j)$, and so the algorithm $\tilde{A}$ will set $c_j := \tilde{c}_j$ at the iteration corresponding to the choice of $\hat{c}_{i_1} \in \mathcal{L}'_{i_1}, \hat{c}_{i_2} \in \mathcal{L}'_{i_2}, \ldots, \hat{c}_{i_m} \in \mathcal{L}'_{i_m}$. Consequently, the j-th column of $\tilde{c}_0$ will be set to the j-th column of $\tilde{c}$. □

Claim 3.7. *With probability at least $2/3$ over the choice of I, at least a $(1 - \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}})$-fraction of the columns are good with respect to I.*

For the proof of the above claim we shall also use the notion of a "good row".

Definition 3.8 (Good Row)**.** A row $i \in [n]$ is good if the codeword $\tilde{c}$ is consistent with all but an α'-fraction of the input lists row i, that is, $\mathrm{dist}(\tilde{c}|_{\{i\}\times[n']}, S|_{\{i\}\times[n']}) \leq \alpha'$.

We claim that with high probability over the choice of I, a large fraction of the rows in I are good.

Claim 3.9. *With probability at least 0.9 over the choice of I, at least a $\left(1 - \frac{\delta \cdot \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}}{12}\right)$-fraction of the rows in I are good.*

Proof of Claim 3.9. For $i \in [n]$, let $f(i) := \mathrm{dist}(\tilde{c}|_{\{i\}\times[n']}, S|_{\{i\}\times[n']})$, and note that by the sampling property of Γ, with probability at least 0.9 over the choice of I we have that

$$\mathbb{E}_{i\in I}\left[f(i)\right] \leq \mathbb{E}_{i\in[n]}\left[f(i)\right] + \gamma = \mathrm{dist}(\tilde{c}, S) + \gamma \leq \alpha' \cdot \frac{\delta \cdot \delta_{\mathrm{dec}} \cdot \delta'_{\mathrm{dec}}}{12},$$

where the last inequality holds by assumption that $\gamma = \alpha' \cdot \frac{\delta\cdot\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{24}$ and $\mathrm{dist}(\tilde{c}, S) \leq \alpha' \cdot \frac{\delta\cdot\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{24}$ (which holds assuming that $\bar{s}$ is a sufficiently large polynomial). An averaging argument yields that in this case, for at least a $(1-\frac{\delta\cdot\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{12})$-fraction of the rows $i \in I$ it holds that $\mathrm{dist}(\tilde{c}|_{\{i\}\times[n']}, S|_{\{i\}\times[n']}) = f(i) \leq \alpha'$. □

Finally, we provide the proof of Claim 3.7.

Proof of Claim 3.7. We will show that each of the three properties in the definition of a good column holds for at least a $(1 - \frac{\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3})$-fraction of the columns with probability at least 0.9 over the choice of I. The claim will then follow by a union bound over the choice of I and the fraction of bad columns.

Property (1): Assuming that $\mathrm{dist}(\tilde{c}, S) \leq \frac{\alpha\cdot\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3}$ (which once more holds assuming that $\bar{s}$ is a sufficiently large polynomial), an averaging argument implies that for at least a $(1-\frac{\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3})$-fraction of the columns $j \in [n']$ it holds that $\mathrm{dist}\left(\tilde{c}_j, S|_{[n]\times\{j\}}\right) \leq \alpha$.

Property (2): Fix $j \in [n']$ and $c \in \mathcal{L}_j \setminus \{\tilde{c}_j\}$, and note that $\mathrm{dist}(c, \tilde{c}_j) \geq \delta$ since C has relative distance δ. For $i \in [n]$, let

$$f(i) := \begin{cases} 1, & \text{if } c_i = (\tilde{c}_j)_i \\ 0, & \text{otherwise.} \end{cases},$$

and note that by the sampling property of Γ, with probability at least $1 - \frac{0.1}{L} \cdot \frac{\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3}$ over the choice of I we have that

$$\mathrm{dist}(c|_I, \tilde{c}_j|_I) = \mathbb{E}_{i\in I}\left[f(i)\right] \geq \mathbb{E}_{i\in[n]}\left[f(i)\right] - \gamma = \mathrm{dist}(c, \tilde{c}_j) - \gamma > \delta/2,$$

where the last inequality follows by choice of $\gamma < \delta/2$. Hence, by a union bound, with probability at least $1 - 0.1 \cdot \frac{\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3}$ over the choice of I, we have $\mathrm{dist}(c|_I, \tilde{c}_j|_I) > \delta/2$ for all $c \in \mathcal{L}_j \setminus \{\tilde{c}_j\}$.

Finally, by an averaging argument we conclude that with probability at least 0.9 over the choice of I, at least a $(1 - \frac{\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3})$-fraction of the columns $j \in [n']$ satisfy Property (2).

Property (3): By Claim 3.9, with probability at least 0.9 over the choice of $I = \{i_1, \ldots, i_m\}$, at least a $(1 - \frac{\delta\cdot\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{12})$-fraction of the rows in I are good, where for a good row $i_k \in I$ we have that $\tilde{c}|_{\{i_k\}\times[n']} \in \mathcal{L}'_{i_k}$, and so $\hat{c}_{i_k} = \tilde{c}|_{\{i_k\}\times[n']}$. Assuming this is the case, we have that $\tilde{c}$ agrees with $(\hat{c}_{i_1}, \ldots, \hat{c}_{i_m})$ on at least a $\left(1 - \frac{\delta\cdot\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{12}\right)$-fraction of the points in $I\times[n']$, and so by averaging for at least a $(1-\frac{\delta_{\mathrm{dec}}\cdot\delta'_{\mathrm{dec}}}{3})$-fraction of the columns $j \in [n']$ it holds that $\mathrm{dist}\left(\tilde{c}_j|_I, ((\hat{c}_{i_1})_j, \ldots, (\hat{c}_{i_m})_j)\right) \leq \delta/4$. □

3.1.5 Deterministic algorithm

Lastly, to obtain a *deterministic* global list recovery algorithm, we simply iterate over the randomness of Γ, and output the union of all output lists. This increases the running time by a multiplicative factor of $2^r = n \cdot \bar{s}/\alpha'$. Moreover, Claim 3.4 guarantees that any codeword that is consistent with all but $(\alpha'/\bar{s})$-fraction of the input lists will be output in one of the invocations, and consequently will be included in the final output list (which is of size at most $(L')^{\bar{s} \cdot L/(\alpha')^2}$ by the same claim).

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

In this section we prove the following lemma which implies Corollary 1.2 from the introduction.

Lemma 3.10. *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 $N^{1+o(1)}$.*

To prove the above lemma, we first use Theorem 1.1 to obtain deterministic nearly-linear time *high-rate* list recoverable codes, and then use the Alon-Edmonds-Luby (AEL) distance amplification method [AEL95, AL96] to turn these codes into deterministic nearly-linear time *capacity-achieving* list recoverable codes. Specifically, we shall use the following version of the AEL method for list recovery from [GI02] which roughly says the following. Given an efficient "outer" code C of rate approaching 1 that is list recoverable from a tiny fraction of errors, and a small "inner" code C' that is a (possibly non-efficient) capacity-achieving list recoverable code, they can be combined to get a new code C_{AEL} that on the one hand, inherits the tradeoff between rate and error correction that C' enjoys, and on the other hand, is almost as efficient as C is.

Lemma 3.11 (Distance amplification for list recovery, [GI02], Lemma 6)**.** *There exists an absolute constant b_0 such that the following holds for any $\delta, \alpha, \gamma > 0$ and $t \geq (\delta \cdot \alpha \cdot \gamma)^{-b_0}$.*

Suppose that $C \subseteq (\Sigma^{\rho \cdot t})^n$ is an outer code of rate $1 - \gamma$ and relative distance δ that is (α, ℓ, L)-globally list recoverable in time T, and $C' \subseteq \Sigma^t$ is an inner code of rate ρ and relative distance $1 - \rho - \gamma$ that is $(1 - \rho - \gamma, \ell', \ell)$-globally list recoverable in time T'. Then there exists a code $C_{AEL} \subseteq (\Sigma^t)^n$ of rate $\rho - \gamma$ and relative distance $1 - \rho - 2\gamma$ that is $(1 - \rho - 2\gamma, \ell', L)$-globally list recoverable in time $T + n \cdot (T' + \mathrm{poly}(t, \log n))$.

Moreover,

- *If C, C' have encoding times T, T', respectively, then C_{AEL} has encoding time $T + n \cdot (T' + \mathrm{poly}(t, \log n))$.*
- *If C, C' are $\mathbb{F}$-linear then so is C_{AEL}.*

Remark 3.12. Lemma 6 in [GI02] is stated for the special case of $\ell' = 1$, and for a more specific choice of list recovery radii and running times. Also, it does not mention explicitly relative distance, encoding time, and linearity. However, these can be deduced from the proof of the lemma, combined with the expander graph construction described in [KMRS17, Lemma 2.12] (See also [GKO+18, Lemma 5.4] for a similar transformation for the setting of local list recovery).

Next we prove Lemma 3.10, based on Theorem 1.1 and Lemma 3.11.

Proof of Lemma 3.10. We shall first apply Theorem 1.1 on a suitable base code C to obtain a deterministic nearly-linear time high-rate list recoverable code C', and then use the transformation given by Lemma 3.11 to obtain a deterministic nearly-linear time capacity-achieving list recoverable code C''.

Base code C: The code C will be the efficient high-rate list recoverable code given by Theorem 2.7, in an appropriate setting of parameters.

Specifically, in what follows, we let $\beta := (\log\log\log N)^{-o(1)}$ (where the $o(1)$ term in the exponent is an arbitrarily slowly decreasing function of N), and we choose the block length of C to be N^β, and the rate to be $1-\gamma\beta/4$. As we will see in a moment, the rationale for these choices is that if we raise C to the tensor power of $1/\beta$, Theorem 1.1 will yield a code of block length N with running time $N^{1+O(\beta)} = N^{1+o(1)}$ and rate greater than $1-\gamma$.

Theorem 2.7 then guarantees, for any constant $\ell' \geq 1$, the existence of a linear code C as above that has relative distance $(\log\log\log N)^{-o(1)}$, and is $((\log\log\log N)^{-o(1)}, \ell', \exp\exp((\log\log\log N)^{o(1)}))$-globally list recoverable in time $N^{O(\beta)}$, provided that the alphabet size is sufficiently large even power of a prime $\exp((\log\log\log N)^{o(1)})$.

High-rate list recoverable code C': Let C' be the code obtained by raising C to a tensor power of $1/\beta = (\log\log\log N)^{o(1)}$. Then C' has block length N, alphabet size $\exp((\log\log\log N)^{o(1)})$, rate at least $1-\gamma/4$, and relative distance $\exp(-(\log\log\log N)^{o(1)})$. Furthermore, by Theorem 1.1, it is $(\exp(-(\log\log\log N)^{o(1)}), \ell', N^{o(1)})$-globally list recoverable deterministically in time $N^{1+O(\beta)} = N^{1+o(1)}$.

Capacity-achieving list recoverable code C'': Let C'' be the code obtained by applying Lemma 3.11 with the outer code being the code C' constructed so far, and the inner code being a capacity-achieving list recoverable code D'' of rate $\rho+\gamma/4$ and relative distance at least $1-\rho-\gamma/2$.

Corollaries 2.2 and 2.6 guarantee the existence of a code D'' as above that is $(1-\rho-\gamma/2, \ell, \ell')$-globally list recoverable for some constant ℓ', provided that the alphabet size is a sufficiently large constant prime power, and the block length is sufficiently large. To satisfy the conditions of Lemma 3.11, we further require that the block length of D'' is sufficiently large $\exp((\log\log\log N)^{o(1)})$, and that the alphabet size of C' is $\exp\exp((\log\log\log N)^{o(1)})$—the size of D''—which can be achieved by grouping together consecutive symbols of C'.

Lemma 3.11 then implies that C'' is a code of block length N, alphabet size $N^{o(1)}$, rate ρ, and relative distance $1-\rho-\gamma$, that is $(1-\rho-\gamma, \ell, N^{o(1)})$-globally list recoverable deterministically in time $N^{1+o(1)}$ (using brute-force decoding of inner code).

Finally, it can be verified that encoding time is as claimed, and that all codes in the process can be taken to be $\mathbb{F}_2$-linear, and all transformations preserve $\mathbb{F}_2$-linearity, so the final code can be guaranteed to be $\mathbb{F}_2$-linear as well. □

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

In this section we prove the following lemma which implies Corollary 1.3 from the introduction.

Lemma 3.13. *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 the following lemma from [Tho83, HRW17a] 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. In what follows let $\theta(x) := 1 - H_2(1 - 2^{x-1})$ for $x \in [0, 1]$.

Claim 3.14 ([GR10], Lemma 2.2)**.** *$\theta(x) \le x$ for all $x \in [0, 1]$.*

Lemma 3.15 (Random concatenation, [HRW17c], Lemma 7.3)**.** *There exists an absolute constant b_0 such that the following holds for any $\gamma > 0$, $\rho' \in [0, 1]$, $\rho \in \left[0, \frac{\theta(\rho')-\gamma/2}{\rho'}\right]$, and $t \ge \frac{b_0}{\gamma^2 \cdot (1-\rho)}$. Suppose that $C \subseteq (\mathbb{F}_2^{\rho' \cdot t})^n$ is an $\mathbb{F}_2$-linear code of rate ρ and relative distance $1 - \rho - \frac{\gamma^2}{b_0}$, and $C_{con} \subseteq \mathbb{F}_2^{tn}$ is a code obtained from C by applying a random linear code $C^{(i)} \subseteq \mathbb{F}_2^t$ of rate ρ' on each coordinate $i \in [n]$ of C independently. Then C_{con} has relative distance at least $H_2^{-1}(1 - \rho \cdot \rho') - \gamma$ with probability $1 - \exp(-n)$.*

We shall also use the following lemma that states the effect of concatenation on list recovery properties.

Lemma 3.16 (Concatenation for list recovery, [HRW17c], Lemma 7.4)**.** *Suppose that $C \subseteq (\Sigma^{\rho' \cdot t})^n$ is (α, ℓ, L)-globally list recoverable in time T, and $C_{con} \subseteq \Sigma^{tn}$ 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 in time T'. Then C_{con} is $((\alpha - \gamma) \cdot \alpha', \ell', L)$-globally list recoverable in time $T + n \cdot T'$.*

Next we prove Lemma 3.13, based on Lemma 3.10 and the above Lemmas 3.15 and 3.16.

Proof of Lemma 4.12. We apply random concatenation on the deterministic nearly-linear time capacity-achieving list recoverable code C given by Lemma 3.10. By Lemma 3.15, the resulting code $\tilde{C}$ will approach the Gilbert-Varshmaov bound with high probability, while by Lemma 3.16, the code $\tilde{C}$ will also be nearly-linear time list recoverable (and in particular, list decodable) with high probability. Thus, whenever the list decoding radius exceeds half the minimum distance (which turns to be the case whenever the rate is smaller than 0.02), the code $\tilde{C}$ can be uniquely decoded from half the minimum distance in near-linear time by first running the list decoding algorithm, and then choosing the codeword from the output list that is closest to the received word. Details follow.

The code C: Let b_0 be the absolute constant guaranteed by Lemma 3.15, and apply Lemma 3.10 with rate $\rho_0 := \frac{\rho}{\theta^{-1}(\rho+\gamma/2)}$ (noting that this is at most 1 since $\theta(x) \le x$ for all $x \in [0,1]$, and θ is monotonically increasing in $[0,1]$), proximity parameter $\gamma_0 := \gamma^2/b_0$, and input list size $\ell_0 := 2^{1/\gamma}$. Lemma 3.10 then guarantees, for infinite number of N's, the existence of an $\mathbb{F}_2$-linear code C of block length N, alphabet size $N^{o(1)}$, rate ρ_0, and relative distance $1-\rho_0-\gamma_0$, that is $(1-\rho_0-\gamma_0, \ell_0, N^{o(1)})$-globally list recoverable deterministically in time $N^{1+o(1)}$.

The code $\tilde{C}$: Let $\tilde{C} \subseteq \mathbb{F}_2^{tN}$ be a binary linear code obtained from C by applying a random linear code $C^{(i)} \subseteq \mathbb{F}_2^t$ of rate $\rho' := \theta^{-1}(\rho+\gamma/2)$ on each coordinate $i \in [n]$ of C independently. Then the code $\tilde{C}$ has rate ρ, and by Lemma 3.15 it also has relative distance at least $H_2^{-1}(1-\rho)-\gamma$ with probability $1-\exp(-N)$. Moreover, by Theorem 2.4, each $C^{(i)}$ is $(H_2^{-1}(1-\rho'-\gamma), 2^{1/\gamma})$-list decodable with probability $1-o(1)$, so with probability $1-\exp(-N)$ this property holds for at least $(1-\gamma^2/b_0)$-fraction of the $C^{(i)}$'s. Lemma 3.16 implies in turn that the code $\tilde{C}$ is $(\tilde{\alpha}, N^{o(1)})$-globally list decodable in time $N^{1+o(1)}$ (using brute-force decoding of inner codes $C^{(i)}$) for

$$\tilde{\alpha} = (1-\rho_0-2\gamma^2/b_0) \cdot H_2^{-1}(1-\rho'-\gamma).$$

Decoding: Next assume that the list decoding radius $\tilde{\alpha}$ exceeds the desired decoding radius, i.e.,

$$(1-\rho_0-2\gamma^2/b_0) \cdot H_2^{-1}(1-\rho'-\gamma) \ge \frac{H_2^{-1}(1-\rho)-\gamma}{2}, \tag{2}$$

where $\rho_0 := \frac{\rho}{\theta^{-1}(\rho+\gamma/2)}$ and $\rho' := \theta^{-1}(\rho+\gamma/2)$. It was shown in [Rud07, Section 4.4] that this is indeed the case whenever $\rho \le 0.02$ and γ is a sufficiently small constant.

Assuming that (2) holds, one can globally uniquely decode $\tilde{C}$ up to half the minimum distance in time $N^{1+o(1)}$ by list decoding $\tilde{C}$, and outputting the codeword in the output list that is closest to the received word. □

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 1.4 from the introduction.

Lemma 4.1. *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 $t \ge 3$. 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 [HRW17a] 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 2.12, except that the requirement (1) is replaced with the relaxed condition that

$$\Pr_{i \in [n]} \left[A_j(i) = c_i \right] \geq 1 - \beta, \tag{3}$$

where the probability is over the choice of uniform random $i \in [n]$,[6] and the soundness requirement is eliminated.

Lemma 4.2 (Approximate local list recovery of high-rate tensor codes, [HRW17b], 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 [Vid15, Proposition 3.15.], but for completeness we provide a full proof below in Section 4.1.1.

Lemma 4.3 (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 $\left(\left(O(Q \log Q)\right)^t, \alpha^t\right)$-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. We prove this lemma in Section 4.1.2, based on a robust local testing procedure for high-rate tensor codes given in [Vid15].

Lemma 4.4 (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.

[6] A simple averaging argument shows that in the case of approximate local list recovery, each of the local algorithms $A_1, \ldots, A_L$ can be assumed to be deterministic.

Lemma 4.5. *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 $T \cdot T_{\text{corr}}$.

Proof. First note that by Remark 2.9, we may assume that the tolerant local testing algorithm A_{test} fails with probability at most $\varepsilon/|L|$, at the cost of increasing the query complexity and running time by a multiplicative factor of $O(\log(|L|/\varepsilon))$.

The local list recovery algorithm $\tilde{A}$ first runs the approximate local list recovery algorithm A, let $A_1, A_2, \ldots, A_L$ be the (deterministic) output local algorithms. Then for any $j = 1, \ldots |L|$, the local list recovery algorithm $\tilde{A}$ runs the tolerant local testing algorithm A_{test} on A_j, and outputs $A_{\text{corr}}(A_j)$ if and only if the test passes, where A_{corr} is the local correction algorithm.

It can be verified that query complexity, output list size, and running times are as claimed. For completeness, suppose that $c \in C$ satisfies $\text{dist}(c, S) \leq \alpha$. Then with probability at least $1 - \varepsilon$ the approximate local list recovery algorithm A will output some A_j for which $\text{dist}(A_j, c) \leq \beta$. Consequently, the tolerant local testing algorithm A_{test} will accept A_j with probability at least $1 - \varepsilon$. So we conclude that with probability at least $1 - 2\varepsilon$ the local algorithm $A_{\text{corr}}(A_j)$ will be included in the output list of $\tilde{A}$, and furthermore, by properties of A_{corr} it will be consistent with the codeword c.

For soundness, suppose that $A_{\text{corr}}(A_j)$ is not consistent with some codeword $c \in C$. Then by properties of A_{corr}, it holds that $\text{dist}(A_j, C) > \gamma$. But in this case the tolerant local testing algorithm A_{test} will reject A_j with probability at least $1 - \varepsilon/|L|$. So by union bound, with probability at least $1 - \varepsilon$, each local algorithm in the output list of $\tilde{A}$ is consistent with some codeword $c \in C$. □

Next we prove Lemma 4.1 based on the above transformation and Lemmas 4.2, 4.3, and 4.4.

Proof of Lemma 4.1. By Lemma 4.3 the tensor product code $C^{\otimes t}$ is $((O(Q \log Q))^t, (\delta/2)^t)$-locally correctable, and by Lemma 4.4 it is $(n^2 \cdot \delta^{-O(t)}, \delta^{b_0 t}, (\delta/2)^t)$-tolerantly locally testable for some absolute constant b_0. Moreover, by Lemma 4.2 the tensor product code $C^{\otimes t}$ is $(\delta^{b_0 t})$-approximately $(n \cdot (s^{t^3} \log^t L), \alpha \cdot s^{-t^3}, \varepsilon/2, \ell, L^{s^{t^3} \cdot \log^t L} \cdot \log(1/\varepsilon))$-locally list recoverable. Finally, Lemma 4.5 implies that $C^{\otimes t}$ 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

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

Running times follow similarly. □

4.1.1 Local correction of tensor codes – proof of Lemma 4.3

Lemma 4.3 can be easily deduced from the following lemma using induction.

Lemma 4.6. *Suppose that $C \subseteq \mathbb{F}^n$, $C' \subseteq \mathbb{F}^{n'}$ are linear codes that are (Q, α), (Q', α')-locally correctable, respectively. Then $C \otimes C' \subseteq \mathbb{F}^{n \times n'}$ is $(Q \cdot O(Q' \log Q'), \alpha \cdot \alpha')$-locally correctable.*

Moreover, if C, C' are locally correctable in times T, T', respectively, then the local correction algorithm for $C \otimes C'$ runs in time $T \cdot O(T' \log T')$.

Proof. First note that by Remark 2.11, we may assume that the local correction algorithm A' for C' fails with probability at most $1/6$, at the cost of increasing the query complexity and running time by some multiplicative constant b_0. Similarly, we may also assume that the local correction algorithm A for C fails with probability at most $1/(6b_0Q')$, at the cost of increasing the query complexity and running time by a multiplicative factor of $O(\log Q')$.

Let $w \in \mathbb{F}^{n \times n'}$ be a string that is $(\alpha \cdot \alpha')$-close to some codeword $c \in C \otimes C'$. Recall that the local correction algorithm $\tilde{A}$ for $C \otimes C'$ is given as input a codeword coordinate $(i, j) \in [n] \times [n']$ in the tensor product code $C \otimes C'$, is allowed to query the received word w at every coordinate of $C \otimes C'$, and must produce a guess for $c_{i,j}$, the codeword value indexed by (i, j).

To this end, the local correction algorithm $\tilde{A}$ for $C \otimes C'$ first runs the local correction algorithm A' for C' on input $j \in [n']$, let $J = \{j_1, \ldots, j_m\} \subseteq [n']$ be the set of query locations for $m := b_0 \cdot Q'$. Next for each query location $j_r \in J$, the algorithm $\tilde{A}$ obtains a guess for the symbol at position (i, j_r) by running the local correction algorithm A for C on input i with oracle access to the column j_r. Let v_r be the guess for the symbol at position (i, j_r) produced by A. At this point we have candidate symbols $(v_1, \ldots, v_m)$ for all positions in $\{i\} \times J$. Finally, the algorithm $\tilde{A}$ responds with the output of A' on query locations $j_1, \ldots, j_m$ and values $v_1, \ldots, v_m$. The formal description of the local correction algorithm $\tilde{A}$ is given in Algorithm 2.

Algorithm 2 The local correction algorithm $\tilde{A}$ for $C \otimes C'$.

function $\tilde{A}((i, j) \in [n] \times [n'])$
▷ $\tilde{A}$ receives oracle access to a matrix $w \in \mathbb{F}^{n \times n'}$.
Run the local correction algorithm A' for C' on input j, let $J = \{j_1, \ldots, j_m\} \subseteq [n']$ be the query locations for $m = b_0 \cdot Q'$.
for $r = 1, \ldots, m$ **do**
Run the local correction algorithm A for C on input i and oracle access to the j_r-th column $w|_{[n] \times \{j_r\}}$.
Let $v_r \leftarrow A(i)$.
▷ v_r is a candidate for the symbol at position $(i, j_r) \in [n] \times [n']$.
end for
▷ At this point, we have candidate symbols $(v_1, \ldots, v_m)$ for every position in $\{i\} \times J$.
Let v be the output of A' on query locations $j_1, \ldots, j_m$ and values $v_1, \ldots, v_m$.
Return: v
end function

The algorithm $\tilde{A}$ invokes the algorithm A' once, followed by $m = O(Q')$ invocations of the algorithm A. Thus, the query complexity of $\tilde{A}$ is

$$O(Q') + m \cdot O(Q \cdot \log Q') = Q \cdot O(Q' \log Q'),$$

and the running time is

$$O(T') + m \cdot O(T \cdot \log Q') = O(T') + T \cdot O(Q' \log Q') = T \cdot O(T' \log T').$$

As for correctness, recall that by assumption the received word w is $(\alpha \cdot \alpha')$-close to the codeword $c \in C \otimes C'$. Let us call a column 'good' if w and c are α-close on this column, and note that by Markov's inequality, at least $(1-\alpha')$-fraction of the columns are good. Furthermore, by our assumptions on each good column the local correction algorithm A for C succeeds with probability at least $1 - \frac{1}{6m}$, and so by union bound, with probability at least $5/6$ the values $(v_1, \ldots, v_m)$ will be computed correctly on each good column. Conditioned on this, the local correction algorithm A' for C' computes v correctly with probability at least $5/6$, so the total success probability is $2/3$. □

4.1.2 Tolerant local testing of high-rate tensor codes – proof of Lemma 4.4

The proof of Lemma 4.4 relies on the following *robust* local testing procedure for high-rate tensor codes from [Vid15] which is a local testing procedure with the property that local view on words far from the code is far on average from an accepting view.

Theorem 4.7 (Robust local testing of high-rate tensor codes, [Vid15, Theorem 3.1])**.** *Suppose that $C \subseteq \mathbb{F}^n$ is a linear code of relative distance δ, and $t \geq 3$. Then for any $w \in \mathbb{F}^{n^t}$, the expected relative distance of w from $C^{\otimes 2}$ on a random axis-parallel plane is at least $\delta^{O(t)} \cdot \mathrm{dist}(w, C^{\otimes t})$.*

Proof of Lemma 4.4. Say we are given a string $w \in \mathbb{F}^{n^t}$ and we need to test if it is close to a codeword of $C^{\otimes t}$. Let $\tau \geq \delta^{O(t)}$ be some threshold parameter to be chosen later. The test is to choose a random axis-parallel plane P in $\mathbb{F}^{n^t}$ and find if there is a codeword $c \in C^{\otimes 2}$ which is τ-close to $w|_P$. If yes, then accept, else reject. Clearly this test makes only n^2 queries. Also by Corollary 2.16, when $\tau < (\delta/2)^2$, this can be implemented in $O(T \cdot n)$ time.

To show completeness, let $w \in \mathbb{F}^{n^t}$ be some string which is α-close to a codeword $c \in C^{\otimes t}$ for $\alpha \geq \delta^{O(t)}$ to be chosen later. Since individual points on a random axis-parallel plane are uniform over $\mathbb{F}^{n^t}$, by Markov inequality, the probability that $w|_P$ is τ-far from $c|_P \in C^{\otimes 2}$ is at most α/τ. So the probability that the test rejects w is at most $p_0 := \alpha/\tau$.

To show soundness, let $w \in \mathbb{F}^{n^t}$ be some string which is $(\delta/2)^t$-far from any codeword $c \in C^{\otimes t}$. Then by Theorem 4.7, the expected relative distance of $w|_P$ from $C^{\otimes 2}$ is at least $\delta^{O(t)}$. Thus the probability that the test rejects w is at least $p_1 := \frac{\delta^{O(t)} - \tau}{1-\tau}$.

Next observe that we can choose $\tau \geq \delta^{O(t)}$ and $\alpha \geq \delta^{O(t)}$ sufficiently small so that $p_0 < p_1$. Finally to get the acceptance and rejection probabilities to $2/3$ as in the definition of tolerant locally testable codes, we repeat the above local test $\delta^{-O(t)}$ times and accept a string if it is accepted in at least $\frac{p_0+p_1}{2}$-fraction of the tests. By Chernoff bound, the new test will have the required soundness and completeness. □

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 [HRW17b, 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 4.8. *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 3.10, we first use Lemma 4.1 to obtain *high-rate* locally list recoverable codes, and then use the Alon-Edmonds-Luby (AEL) distance amplification method [AEL95, AL96] to turn these codes into *capacity-achieving* locally list recoverable codes. However, this time we shall use the following version of the AEL method for *local* list recovery from [GKO+18].

Lemma 4.9 (Distance amplification for local list recovery, [GKO+18], Lemma 5.4.)**.** *There exists an absolute constant b_0 such that the following holds for any $\delta, \alpha, \gamma > 0$ and $t \geq (\delta \cdot \alpha \cdot \gamma)^{-b_0}$.*

Suppose that $C \subseteq (\Sigma^{\rho \cdot t})^n$ is an outer code of rate $1-\gamma$ and relative distance δ that is $(Q, \alpha, \varepsilon, \ell, L)$-locally list recoverable, and $C' \subseteq \Sigma^t$ is an inner code of rate ρ and relative distance $1-\rho-\gamma$ that is $(1-\rho-\gamma, \ell', \ell)$-globally list recoverable. Then there exists a code $C_{AEL} \subseteq (\Sigma^t)^n$ of rate $\rho-\gamma$ and relative distance $1-\rho-2\gamma$ that is $(Q \cdot \mathrm{poly}(t), 1-\rho-2\gamma, \varepsilon, \ell', L)$-locally list recoverable.

Moreover,

- *If the local list recovery algorithm for C has preprocessing time T_{pre} and running time T, and C' can be globally list recovered in time T', then the local list recovery algorithm for C_{AEL} has preprocessing time $T_{\mathrm{pre}} + Q \cdot (T' + \mathrm{poly}(t, \log n))$ and running time $T + Q \cdot \mathrm{poly}(t) \cdot (T' + \mathrm{poly}(\log n))$.*
- *If C, C' have encoding times T, T', respectively, then C_{AEL} has encoding time $T + n \cdot (T' + \mathrm{poly}(t, \log n))$.*
- *If C, C' are $\mathbb{F}$-linear then so is C_{AEL}.*

To apply Lemma 4.1, we shall also need a high-rate base code that is both globally list recoverable and locally correctable. We obtain such a code by intersecting the high-rate globally list recoverable codes given by Theorem 2.7 with the high-rate locally correctable codes given by the following lemma.

Lemma 4.10 (High-rate locally correctable codes)**.** *For any $\gamma, \beta > 0$, and integer N where $q := N^\beta$ is a prime power, there exists a code C_N that satisfies the following.*

- *C_N is an $\mathbb{F}_q$-linear code of block length N and alphabet size $N^{(\gamma\beta)^{-O(1/\beta)}}$.*
- *C_N has rate $1-\gamma$ and relative distance $\Omega(\gamma \cdot \beta)$.*
- *C_N is $(N^\beta \cdot (\gamma\beta)^{-O(1/\beta)}, \Omega(\gamma \cdot \beta))$-locally correctable in time $N^\beta \cdot (\gamma\beta)^{-O(1/\beta)}$.*
- *C_N is encodable in time $\mathrm{poly}(N)$.*

We prove the above lemma in Section 4.2.1, based on the high-rate locally correctable codes of [KSY14]. Next we prove Lemma 4.8, based on Lemmas 4.1, 4.9, and 4.10.

Proof of Lemma 4.8. The proof is similar to that of Lemma 3.10, with the main difference being that now we also need to take care that the base code will also be locally correctable. Specifically, we shall first apply Lemma 4.1 on a suitable high-rate base code C that is both globally list recoverable and locally correctable to obtain a high-rate locally list recoverable code C', and then use the transformation given by Lemma 4.9 to obtain a capacity-achieving locally list recoverable code C''.

Base code C: The code C will be the intersection of the efficient high-rate globally list recoverable code given by Theorem 2.7 with the high-rate locally correctable code given by Lemma 4.10, in an appropriate setting of parameters.

Specifically, let $\beta := (\log\log N)^{-o(1)}$ (where the $o(1)$ term in the exponent is an arbitrarily slowly decreasing function of N), and we choose the block length of C to be N^β, and the rate to be $1-\gamma\beta/4$. The code C will be constructed in turn as $D_1 \cap D_2$, where D_1 is the high-rate globally list recoverable code given by Theorem 2.7, and D_2 is obtained using the high-rate locally correctable code given by Lemma 4.10, and both codes D_1, D_2 have block length N^β and rate $1-\gamma\beta/8$. Details follow.

The code D_1: Let $\ell' \geq 1$ be a constant to be chosen later on, and let D_1 be the linear code guaranteed by Theorem 2.7 of block length N^β, rate $1-\gamma\beta/8$, and relative distance $(\log\log N)^{-o(1)}$, that is $((\log\log N)^{-o(1)}, \ell', \exp\exp((\log\log N)^{o(1)}))$-globally list recoverable. Note that such a code exists provided that the alphabet size is sufficiently large even power of a prime $q := \exp((\log\log N)^{o(1)})$.

The code D_2: The code D_2 will be constructed in turn as the concatenation of the high-rate locally correctable code D_2' given by Lemma 4.10 with an efficiently encodable and decodable linear code D_2'' obtained using Fact 2.3. The purpose of the concatenation is to reduce the alphabet size of D_2' to that of D_1, as well as make the code D_2' linear.

We first describe the code D_2'. Suppose that N^{β^2} is a power of q (which holds for infinite number of N's). Lemma 4.10 guarantees the existence of an $\mathbb{F}_q$-linear code D_2' of length $N^\beta \cdot (1-\gamma\beta/16)$, alphabet size q^a for $a = (\log N)^{1+o(1)}$, rate $1-\gamma\beta/16$, and relative distance $(\log\log N)^{-o(1)}$, that is $(N^{O(\beta^2)}, (\log\log N)^{-o(1)})$-locally correctable.

Next we describe the code D_2''. The code D_2'' will be an efficiently encodable and decodable linear code of length $\frac{1}{1-\gamma\beta/16}\cdot a$, alphabet size q, rate $1-\gamma\beta/16$, and relative distance $(\log\log N)^{-o(1)}$. The code D_2'' can be obtained in turn by taking the Reed-Solomon code from Fact 2.3 of length $\frac{1-\gamma\beta/32}{1-\gamma\beta/16}\cdot a$, alphabet size $q^{\log\log N}$ (noting that $q^{\log\log N} > \log^2 N > a$), rate $1-\gamma\beta/32$, and relative distance $(\log\log N)^{-o(1)}$, and concatenating it with another Reed-Solomon code of length $\frac{1}{1-\gamma\beta/32}\cdot\log\log N$, alphabet size q, rate $1-\gamma\beta/32$, and relative distance $(\log\log N)^{-o(1)}$.

Finally, by concatenating D_2' with D_2'' we obtain a linear code D_2 of length N^β, alphabet size $q = \exp((\log\log N)^{o(1)})$, rate $1-\gamma\beta/8$, and relative distance $(\log\log N)^{-o(1)}$, that is $(N^{O(\beta^2)}, (\log\log N)^{-o(1)})$-locally correctable.

We conclude that $C := D_1 \cap D_2$ is a linear code of block length N^β, alphabet size $\exp((\log\log N)^{o(1)})$, rate $1-\gamma\beta/4$, and relative distance $(\log\log N)^{-o(1)}$, that is $((\log\log N)^{-o(1)}, \ell', \exp\exp((\log\log N)^{o(1)}))$-globally list recoverable, and $(N^{O(\beta^2)}, (\log\log N)^{-o(1)})$-locally correctable.

High-rate locally list recoverable code C': Let C' be the code obtained by raising C to a tensor power of $1/\beta = (\log\log N)^{o(1)}$. Then C' has block length N, alphabet size $\exp((\log\log N)^{o(1)})$, rate at least $1-\gamma/4$, and relative distance $\exp(-(\log\log N)^{o(1)})$. Furthermore, by Lemma 4.1, it is $(N^{o(1)}, \exp(-(\log\log N)^{o(1)}), \varepsilon, \ell', N^{o(1)})$-locally list recoverable.

Capacity-achieving locally list recoverable code C'': Let C'' be the code obtained by applying Lemma 4.9 with the outer code being the code C' constructed so far, and the inner code being a capacity-achieving globally list recoverable code D'' of rate $\rho+\gamma/4$ and relative distance at least $1-\rho-\gamma/2$.

Corollaries 2.2 and 2.6 guarantee the existence of a code D'' as above that is $(1-\rho-\gamma/2, \ell, \ell')$-globally list recoverable for some constant ℓ', provided that the alphabet size is a sufficiently large constant prime power, and the block length is sufficiently large. To satisfy the conditions of Lemma 4.9, we further require that the block length of D'' is sufficiently large $\exp((\log\log N)^{o(1)})$, and that the alphabet size of C' is $\exp\exp((\log\log N)^{o(1)})$—the size of D''—which can be achieved by grouping together consecutive symbols of C'.

Lemma 4.9 then implies that C'' is a code of block length N, alphabet size $N^{o(1)}$, rate ρ, and relative distance $1-\rho-\gamma$, that is $(N^{o(1)}, 1-\rho-\gamma, \varepsilon, \ell, N^{o(1)})$-locally list recoverable.

Finally, it can be verified that running times are as claimed (using brute-force encoding and decoding of inner code D''), and that all codes in the process can be taken to be $\mathbb{F}_2$-linear, and all transformations preserve $\mathbb{F}_2$-linearity, so the final code can be guaranteed to be $\mathbb{F}_2$-linear as well. □

4.2.1 High-rate locally correctable codes – proof of Lemma 4.10

Lemma 4.10 is a consequence of the following theorem from [KSY14], summarizing the parameters of multiplicity codes.

Theorem 4.11 (Multiplicity codes, [KSY14], Lemmas 3.5 and 3.6, and [Kop14])**.** *The following holds for any integers s, d, m, and for any prime power $q \geq \max\{10\cdot m, \frac{d+6\cdot s}{s}, 12\cdot(s+1)\}$. There exists an $\mathbb{F}_q$-linear code C of block length q^m, alphabet size $q^{\binom{m+s-1}{m}}$, relative distance at least $\delta := 1-\frac{d}{s\cdot q}$, and rate at least $\left(1-\frac{m^2}{s}\right)\cdot(1-\delta)^m$, that is $(O(s^m\cdot q), \delta/10)$-locally correctable. Moreover, C can be locally corrected in time $O(q/\delta^m)$, and encoded in time* $\mathrm{poly}\left(q^m, \binom{m+s-1}{m}\right)$.

Proof of Lemma 4.10. We set the code C_N to be the code given by Theorem 4.11 with the following parameters. We choose $q := N^\beta$ to be the field size (which exists whenever q is a prime power), and choose $m = 1/\beta$. Note that indeed $q^m = N$. We choose $s = 2m^2/\gamma$, $\delta = \gamma/(2m)$, and $d = s\cdot q\cdot(1-\delta)$.

The alphabet size of the code is

$$q^{\binom{m+s-1}{m}} \leq N^{\beta\cdot(m+s)^m} \leq N^{(\gamma\beta)^{-O(1/\beta)}},$$

the relative distance is at least $\delta \geq \Omega(\gamma\cdot\beta)$, and the rate is at least

$$\left(1-\frac{m^2}{s}\right)\cdot(1-\delta)^m = \left(1-\frac{\gamma}{2}\right)\left(1-\frac{\gamma}{2m}\right)^m \geq 1-\gamma.$$

Furthermore, C_N is locally correctable from $\Omega(\gamma\beta)$-fraction of errors with query complexity

$$O(s^m \cdot q) \leq N^{\beta} \cdot (\gamma\beta)^{-O(1/\beta)}.$$

as required.

Finally, it can be verified that running times are as required. □

4.3 Local correction up to the GV bound

In this section we prove the following lemma which implies Corollary 1.5 from the introduction.

Lemma 4.12. *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 $N^{1+o(1)}$.*

Similarly to Lemma 3.13, the proof of the above lemma relies on the random concatenation Lemma 3.15, as well as the following lemma that is an analogue of Lemma 3.16 for the setting of local list recovery.

Lemma 4.13 (Concatenation for local list recovery)**.** *Suppose that $C \subseteq (\Sigma^{\rho' \cdot t})^n$ is $(Q, \alpha, \varepsilon, \ell, L)$-locally list recoverable, and $C_{\text{con}} \subseteq \Sigma^{tn}$ 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 $T_{\text{pre}} + Q \cdot T'$ and running time $T + Q \cdot T'$.

We prove the above lemma in Section 4.3.1. Finally, we shall also use the following lemma which shows that a locally list decodable code (satisfying the soundness property) is also locally correctable.

Lemma 4.14. *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 $\left(O\big(Q \cdot L \cdot \frac{\log^2 n}{(\delta/2-\alpha)^2}\big), \alpha\right)$-locally correctable.*

Moreover, if the local list decoding algorithm has preprocessing time T_{pre} and running time T, then the local correction algorithm runs in time $T_{\text{pre}} + O\left(T \cdot L \cdot \frac{\log^2 n}{(\delta/2-\alpha)^2}\right)$.

Proof. We first run the local list decoding algorithm, and then choose a local corrector from the output list that is sufficiently close to the received word (which can be checked via sampling).

Specifically, let A be the local list decoding algorithm for C, by Remark 2.11 we may assume that both the completeness and soundness properties of A hold with success probability $1 - \frac{1}{n^{10}}$

instead of $\frac{2}{3}$ at the cost of increasing the query complexity and running time by a multiplicative factor of $O(\log n)$.

On oracle access to $w \in \Sigma^n$ and input coordinate $i \in [n]$, the local correction algorithm A_{corr} for C first runs the local list decoding algorithm A for C, let $A_1, \ldots, A_L$ be the local algorithms in the output list of A. Then for each $j = 1, \ldots, L$, the algorithm A_{corr} runs A_j on a random subset $S_j \subseteq [n]$ of $O\left(\frac{\log n}{(\delta/2-\alpha)^2}\right)$ coordinates, and computes the fraction δ_j of coordinates in S_j on which the decoded values differ from the corresponding values of w. Finally, the algorithm A_{corr} finds some A_j for which $\delta_j \leq \delta/2$ (if such A_j exists), and uses A_j to locally correct the input coordinate i. Clearly, the query complexity and running time of A_{corr} are as claimed. Next we show that A_{corr} satisfies the required local correction guarantee.

Let $c \in C$ be the (unique) codeword which satisfies that $\text{dist}(w, c) \leq \alpha$. We shall show below that with probability $0.9 - o(1)$, there exists some A_j that computes c and satisfies that $\delta_j \leq \delta/2$, and on the other hand, with probability $0.9 - o(1)$, any A_j which does not compute c satisfies that $\delta_j > \delta/2$. This will imply in turn that the algorithm A_{corr} will succeed in decoding the input coordinate correctly with probability $0.8 - o(1) \geq \frac{2}{3}$ as required.

We first show that with probability $0.9 - o(1)$, there exists some A_j that computes c and satisfies that $\delta_j \leq \delta/2$. To see this note that by the completeness property of A, and since $\text{dist}(w, c) \leq \alpha$, with probability at least 0.9 over the randomness of A there exists some A_j that computes c. In this case, by union bound with probability $1 - o(1)$ it holds that each decoded coordinate of A_j in S_j equals to the corresponding coordinate in c. Furthermore, by Chernoff bound with probability $1 - o(1)$ it holds that w and c differ on S_j by at most $\frac{\delta}{2}$-fraction of the coordinates. Consequently, with probability $0.9 - o(1)$ it holds that $\delta_j \leq \delta/2$.

Next we show that with probability $0.9 - o(1)$, any A_j which does not compute c satisfies that $\delta_j > \delta/2$. For this note that by the soundness property of A, with probability at least 0.9 over the randomness of A, any such A_j computes some codeword $c' \in C \setminus \{c\}$. As above, by union bound with probability $1 - o(1)$ it holds that for any such A_j, each decoded coordinate of A_j in S_j equals to the corresponding coordinate in c'. On the other hand, since C has relative distance δ and $\text{dist}(w, c) \leq \alpha$, we have that $\text{dist}(w, c') \geq \delta - \alpha = \delta/2 + (\delta/2 - \alpha)$, and so by Chernoff bound with probability $1 - o(1)$ for any such A_j it holds that w and c' differ on S_j by more than $\frac{\delta}{2}$-fraction of the coordinates. Consequently, with probability $0.9 - o(1)$ it holds that $\delta_j > \delta/2$ for any such A_j.

□

Next we prove Lemma 4.12, based on the above lemma and Lemmas 4.8, 3.15, and 4.13.

Proof of Lemma 4.12. The proof is similar to that of Lemma 3.13. As in Lemma 3.13, we apply random concatenation on the capacity-achieving locally list recoverable code C given by Lemma 4.8. By Lemma 3.15, the resulting code $\tilde{C}$ will approach the Gilbert-Varshmaov bound with high probability, while by Lemma 4.13, the code $\tilde{C}$ will also be locally list recoverable (and in particular, locally list decodable) with high probability. Lemma 4.14 then implies that whenever the list decoding radius exceeds the desired local correction radius, then the code $\tilde{C}$ can also be locally corrected from this radius. Details follow.

The code C: As in Lemma 3.13, let b_0 be the absolute constant guaranteed by Lemma 3.15, and apply Lemma 4.8 with rate $\rho_0 := \frac{\rho}{\theta^{-1}(\rho+\gamma/4)}$, proximity parameter $\gamma_0 := \gamma^2/(4b_0)$, and input

list size $\ell_0 := 2^{1/\gamma}$. Lemma 4.8 then guarantees, for infinite number of N's, the existence of an $\mathbb{F}_2$-linear code C of block length N, alphabet size $N^{o(1)}$, rate ρ_0, and relative distance $1-\rho_0-\gamma_0$, that is $(N^{o(1)}, 1-\rho_0-\gamma_0, 0.1, \ell_0, N^{o(1)})$-locally list recoverable.

The code $\tilde{C}$: Let $\tilde{C} \subseteq \mathbb{F}_2^{tN}$ be a binary linear code obtained from C by applying a random linear code $C^{(i)} \subseteq \mathbb{F}_2^t$ of rate $\rho' := \theta^{-1}(\rho + \gamma/4)$ on each coordinate $i \in [n]$ of C independently. Then the code $\tilde{C}$ has rate ρ, and by Lemma 3.15 it also has relative distance at least $H_2^{-1}(1-\rho) - \gamma/2$ with probability $1 - \exp(-N)$. Moreover, by Theorem 2.4, each $C^{(i)}$ is $(H_2^{-1}(1-\rho'-\gamma), 2^{1/\gamma})$-list decodable with probability $1 - o(1)$, so with probability $1 - \exp(-N)$ this property holds for at least $(1 - \gamma^2/(4b_0))$-fraction of the $C^{(i)}$'s. Lemma 4.13 implies in turn that the code $\tilde{C}$ is $(N^{o(1)}, \tilde{\alpha}, 0.1, N^{o(1)})$-locally list decodable for

$$\tilde{\alpha} = (1 - \rho_0 - \gamma^2/(2b_0)) \cdot H_2^{-1}(1-\rho'-\gamma).$$

Local correction: Next assume that the local list decoding radius $\tilde{\alpha}$ exceeds the desired local correction radius, i.e.,

$$(1 - \rho_0 - \gamma^2/(2b_0)) \cdot H_2^{-1}(1-\rho'-\gamma) \geq \frac{H_2^{-1}(1-\rho) - \gamma}{2}, \tag{4}$$

where $\rho_0 := \frac{\rho}{\theta^{-1}(\rho+\gamma/4)}$ and $\rho' := \theta^{-1}(\rho + \gamma/4)$. It was shown in [Rud07, Section 4.4] that this is indeed the case whenever $\rho \leq 0.02$ and γ is a sufficiently small constant.

Assuming that (4) holds, Lemma 4.14 implies that $\tilde{C}$ is locally correctable from $\frac{H_2^{-1}(1-\rho)-\gamma}{2}$-fraction of errors with query complexity $N^{o(1)}$.

Finally, it can also be verified that running times are as claimed (using brute-force encoding and decoding of inner codes $C^{(i)}$). □

4.3.1 Concatenation for local list recovery – proof of Lemma 4.13

Proof of Lemma 4.13. The local list recovery algorithm $\tilde{A}$ for C_{con} will run the local list recovery algorithm A for C, and answer the queries of A by globally list recovering the $C^{(i)}$'s corresponding to the queries of A.

In more detail, on oracle access to a string of input lists $S \in \binom{\Sigma}{\ell'}^{tn}$, the local list recovery algorithm $\tilde{A}$ for C_{con} runs the local list recovery algorithm A for C, and whenever A asks for some coordinate $i \in [n]$, the algorithm $\tilde{A}$ globally list recovers the i-th block of S of length t from α'-fraction of errors, and feeds the messages corresponding to the first ℓ codewords in the output list as an answer to the query of A. Let $A_1, \ldots, A_L$ be the resulting output local algorithms of A. Then $\tilde{A}$ outputs L local algorithms $\tilde{A}_1, \ldots, \tilde{A}_L$ where each algorithm $\tilde{A}_j$ is defined as follows.

To locally correct the r-th coordinate in the k-th block of C_{con} of length t (that is, a coordinate of the form $(k-1) \cdot t + r \in [tn]$ where $1 \leq k \leq n$ and $1 \leq r \leq t$), the algorithm $\tilde{A}_j$ runs the algorithm A_j on input coordinate k. As above, whenever A_j asks for some coordinate $i \in [n]$, the algorithm $\tilde{A}_j$ globally list recovers the i-th block of S of length t from α'-fraction of errors, and feeds the messages corresponding to the first ℓ codewords in the output list as an answer to the query of A_j. Let $\sigma \in \Sigma^{\rho' \cdot t}$ be the output symbol of A_j. Then the algorithm $\tilde{A}_j$ outputs the r-th symbol of $C^{(k)}(\sigma) \in \Sigma^t$.

Clearly, query complexity, output list size, and running times of $\tilde{A}$ are as claimed. Soundness property also clearly holds. To see that the completeness property holds as well note that if $\mathrm{dist}(\tilde{c}, S) \leq (\alpha - \gamma) \cdot \alpha'$ for some $\tilde{c} \in C_{\mathrm{con}}$, then by Markov's inequality for at most $(\alpha - \gamma)$-fraction of $i \in [n]$ it holds that the i-th block of S of length t is inconsistent with the i-th block of $\tilde{c}$ of length t by more than α'-fraction of the coordinates. Moreover, since at least $(1-\gamma)$-fraction of the codes $C^{(i)}$ are (α', ℓ', ℓ)-list recoverable, list recovery of the $C^{(i)}$'s fails on at most α-fraction of the blocks. Completeness then follows since C is locally list recoverable from α-fraction of errors. □

5 Combinatorial lower bound on output list size

In this section, 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.

Fact 5.1. *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 $\mathbb{F}^{n_1 \times n_2}$.*

5.1 Output list size for list recovering high-rate tensor codes

In this section we prove Theorem 1.6 from the introduction, which we restate here for convenience.

Theorem 1.6 (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 5.2. *Let $C \subseteq \mathbb{F}^n$ be a subspace of dimension k, and let r be a positive integer. Suppose that*

$$\left(1 - \tfrac{1}{r}\right) \cdot n + 1 \leq k \ . \tag{5}$$

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 (5) 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 1.6, based on the above proposition.

Proof of Theorem 1.6. 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 5.2 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.

The set C'. Since C has rate $1-\gamma$, it has dimension $k = (1-\gamma)n$, and so Proposition 5.2 guarantees the existence of a subset $C' \subseteq C$ of $r = 1/\gamma$ nonzero codewords with pairwise disjoint support.

The set C''. 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 $\mathrm{Supp}(c) \cap \mathrm{Supp}(c') = \emptyset$.

The set $\bar{C}$. 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 $\mathbb{F}^{n^t}$,[7] so the set $\bar{C}$ contains ℓ^{r^t} distinct codewords in $C^{\otimes t}$.

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

To this end, we observe that 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 $(i_t,\ldots,i_t)$ coordinate of a codeword in $\bar{C}$. So we can define the corresponding input list $S_{i_1,\ldots,i_t}$ as

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

if such a codeword c exists, and as $S_{i_1,\ldots,i_t} = \{0\}$ otherwise. Note that each set $S_{i_1,\ldots,i_t}$ 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. □

5.2 Concrete lower bound on output list size

In this section, we demonstrate a setting of parameters that yields Corollary 1.7 from the introduction, restated below.

Corollary 1.7. *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 [RW18].

Theorem 5.1 ([RW18], 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 1.7. Let $C \subseteq \mathbb{F}_q^n$ be the linear code given by Theorem 5.1 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 2.2, we may further assume that the code C has relative distance at least δ. Now, by Theorem 1.6 we have that $L' \geq \ell^{(2\delta)^{-t}} = \exp((2\delta)^{-(t-3/2)} \cdot \sqrt{\log L})$. □

[7] This also follows from the fact that all codewords in C'' are linearly independent together with Fact 5.1.

5.3 Lower bound for local list recovering

We now prove Corollary 1.8 from the introduction, restated below.

Corollary 1.8. *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 recall Lemma 4.14 (restated below in a shorter form) 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 5.3. *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 $\left(O\big(Q \cdot L \cdot \frac{\log^2 n}{(\delta/2-\alpha)^2}\big), \alpha\right)$-locally correctable.*

So to prove Corollary 1.8, 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 5.4. *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 5.3.1. To apply this lemma to $C^{\otimes t}$ we further observe that the tensor product preserves the dual distance of the base code.

Lemma 5.5. *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 5.3.2. We now proceed to the proof of Corollary 1.8.

Proof of Corollary 1.8. Let $C \subseteq \mathbb{F}^n$ be a random linear code of rate $1 - 2\delta$. By Corollary 2.2, 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 1.6 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 5.3 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 5.5 we have that $(C^{\otimes t})^\perp$ has (absolute) distance at least $(1 - 3\delta)n$, and consequently Lemma 5.4 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. □

5.3.1 Dual distance is a lower bound on query complexity – proof of Lemma 5.4

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

Fact 5.6 ([ABI86])**.** *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 $a_1, \ldots, a_s \in \mathbb{F}_q$,*

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

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 Fact 5.6 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.

5.3.2 Tensor product preserves dual distance – proof of Lemma 5.5

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). Fact 5.1 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.

Acknowledgements. NR, NRZ, and SS were supported in part by NSF-BSF grant CCF-1814629 and 2017732. SS was supported in part by a Google Fellowship in the School of Engineering at Stanford.

References

[ABI86] 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.

[AEL95] 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.

[AL96] 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.

[BFLS91] 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.

[BS06] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Structures and Algorithms*, 28(4):387–402, 2006.

[BV09] Eli Ben-Sasson and Michael Viderman. Tensor products of weakly smooth codes are robust. *Theory of Computing*, 5(1):239–255, 2009.

[BV15] Eli Ben-Sasson and Michael Viderman. Composition of semi-LTCs by two-wise tensor products. *Computational Complexity*, 24(3):601–643, 2015.

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

[CR05] Don Coppersmith and Atri Rudra. On the robust testability of tensor products of codes. ECCC TR05-104, 2005.

[DSW06] 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.

[FS95] 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.

[GI02] 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.

[GI04] 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.

[Gil52] Edgar N. Gilbert. A comparision of signalling alphabets. *Bell System Technical Journal*, 31:504–522, 1952.

[GK16] Venkatesan Guruswami and Swastik Kopparty. Explicit subspace designs. *Combinatorica*, 36(2):161–185, 2016.

[GKO+18] 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.

[GL89] 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.

[GM12] 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.

[Gol97] Oded Goldreich. A sample of samplers: A computational perspective on sampling. *def*, 1:2n, 1997.

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

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

[GS06] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost linear length. *Journal of ACM*, 53(4):558–655, 2006.

[Gur01] Venkatesan Guruswami. *List decoding of error-correcting codes.* PhD thesis, MIT, 2001.

[GX13] Venkatesan Guruswami and Chaoping Xing. List decoding Reed-Solomon, Algebraic-Geometric, and Gabidulin subcodes up to the Singleton bound. In *Proceedings of the 45th annual ACM symposium on Theory of Computing (STOC)*, pages 843–852. ACM, 2013.

[HRW17a] 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.

[HRW17b] 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.

[HRW17c] 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, 2017.

[KMRS17] 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.

[Kop14] Swastik Kopparty. Some remarks on multiplicity codes. In *Proceedings of the AMS Special Session on Discrete Geometry and Algebraic Combinatorics*, Contemporary Mathematics, 2014.

[KRSW18] 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.

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5):28, 2014.

[KT00] 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.

[Mei09] Or Meir. Combinatorial construction of locally testable codes. *SIAM Journal on Computing*, 39(2):491–544, 2009.

[RS60] 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.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterization of polynomials with applications to program testing. *SIAM Journal of Computing*, 25(2):252–271, 1996.

[Rud07] Atri Rudra. *List Decoding and Property Testing of Error Correcting Codes*. PhD thesis, University of Washington, 2007.

[RW18] 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.

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001.

[Tho83] 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.

[Val05] 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.

[Var57] R. R. Varshamov. Estimate of the number of signals in error correcting codes. *Doklady Akadamii Nauk*, pages 739–741, 1957.

[Vid13] 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.

[Vid15] Michael Viderman. A combination of testability and decodability by tensor products. *Random Structures and Algorithms*, 46(3):572–598, 2015.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il