Electronic Colloquium on Computational Complexity, Report No. 93 (2023)

Relaxed Local Correctability from Local Testing

Vinayak M. Kumar[*] Geoffrey Mon[†]

Abstract

We cement the intuitive connection between relaxed local correctability and local testing by presenting a concrete framework for building a relaxed locally correctable code from any family of linear locally testable codes with sufficiently high rate. When instantiated using the locally testable codes of Dinur et al. (STOC 2022), this framework yields the first asymptotically good relaxed locally correctable and decodable codes with polylogarithmic query complexity, which finally closes the superpolynomial gap between query lower and upper bounds. Our construction combines high-rate locally testable codes of various sizes to produce a code that is locally testable at every scale: we can gradually "zoom in" to any desired codeword index, and a local tester at each step certifies that the next, smaller restriction of the input has low error.

Our codes asymptotically inherit the rate and distance of any locally testable code used in the final step of the construction. Therefore, our technique also yields nonexplicit relaxed locally correctable codes with polylogarithmic query complexity that have rate and distance approaching the Gilbert–Varshamov bound.

1 Introduction

Locally correctable codes (LCCs) and locally decodable codes (LDCs) are error correcting codes that allow any bit of the original codeword or message to be recovered using very few queries to a corrupted form of the codeword. This is a natural and useful property, but unfortunately little is known about the best possible parameter tradeoffs. In particular, the optimal query complexity for locally correctable and decodable codes has been a longstanding mystery. In the asymptotically good (constant rate and distance) regime, existing lower bounds imply that any LDC (and any linear LCC) must make $\Omega(\log n)$ queries [KT00, Woo07]. However, the most query-efficient constant-rate LCCs and LDCs, constructed by Kopparty et al. [KMRS17], require $2^{\tilde{O}(\sqrt{\log n})}$ queries which is subpolynomial but superpolylogarithmic. Whether the true optimal query complexity is polylogarithmic or not is still an open problem. A Reed–Muller code with appropriate parameters brings us tantalizingly close: such a code is locally correctable with polylogarithmic query complexity, but with block length slightly superlinear (see e.g. [Yek12, Section 2.3]).

Ben-Sasson et al. [BGH⁺06] and Gur, Ramnarayan, and Rothblum [GRR20] introduced the notions of *relaxed* locally decodable codes (RLDCs) and *relaxed* locally correctable codes (RLCCs), respectively. These codes admit local decoders or correctors that either return the right answer, or detect corruption in the codeword by returning a rejection symbol $\perp$. For constant-rate RLDCs (and linear RLCCs), the gap between lower and upper bounds is smaller but still significant: the best lower bound is $\tilde{\Omega}(\sqrt{\log n})$ [GL21, DGL21, Gol23b], while the best upper bound, due to Cohen and Yankovitz [CY22], is quasipolylogarithmic: $(\log n)^{O(\log\log\log n)}$.

In this work, we construct RLCCs with constant rate, constant correcting radius, and polylogarithmic query complexity, thereby finally bringing the query upper bound polynomially close to the lower bound.

[*] Department of Computer Science, University of Texas at Austin. vmkumar@cs.utexas.edu. Supported by NSF Grant CCF-2008076 and a Simons Investigator Award (#409864, David Zuckerman).

[†] Department of Computer Science, University of Texas at Austin. gmon@cs.utexas.edu. Supported by NSF Grant CCF-2200956.

ISSN 1433-8092

Theorem 1.1 (informal, see Section 5). *For infinitely many positive n and any constant $R \in (0,1)$, there exist explicit linear RLCCs (and thus RLDCs) of block length n, rate R, constant correcting (or decoding) radius, and query complexity*

$$r = O\big(\log^{69} n\big).$$

We make no effort to optimize the exponent, instead striving for a simpler exposition. The related and well-studied notion of locally testable codes (where errors can be detected with few queries) proves to be key: we are able to build a relaxed local correctable code from any existing linear locally testable code with vanishing loss in rate.

Theorem 1.2 (informal, see Section 4). *Let n be a sufficiently large integer. Let LTC be a linear locally testable code of block length n, rate R_{LTC} and distance δ_{LTC} which has a local tester T with query complexity r_{LTC} that satisfies*

$$\textit{completeness:} \qquad \forall c \in \mathsf{LTC}.\, \Pr[T^c = \bot] = 0, \textit{ and} \tag{1}$$

$$\textit{soundness:} \qquad \forall w \in \{0,1\}^n.\, \Pr[T^w = \bot] \geq \varkappa \cdot \mathrm{dist}(w, \mathsf{LTC}). \tag{2}$$

Then, there exists a linear RLCC with block length n, rate $R_{\mathsf{LTC}} - O(1/\log\log n)$, distance $\geq \delta_{\mathsf{LTC}}$, correcting radius $\delta_{\mathsf{LTC}}/2$, and query complexity $O(r_{\mathsf{LTC}}/\varkappa) \cdot \mathrm{polylog}(n)$.

Then, by leveraging known locally testable codes, we construct explicit RLCCs with high rate (constant arbitrarily close to 1), constant distance, and polylogarithmic query complexity. We also get nonexplicit RLCCs with polylogarithmic query complexity that approach the Gilbert–Varshamov bound, which is the best known general tradeoff between rate and distance for which codes exist. The last known RLCCs to approach the Gilbert–Varshamov bound are the LCCs of Gopi et al. [GKO+18] which require polynomially many (i.e., n^ε) queries.

1.1 Techniques

We first construct RLCCs with high rate and polylogarithmic query complexity, but with subconstant correcting radius. This is the core of the construction. Then, we apply a similar technique to amplify the correcting radius to be constant while still preserving the polylogarithmic query complexity.

Our construction is based on the remarkable power of locally testable codes (LTCs). An LTC is a code which has a testing algorithm T that, given oracle access to a string w, always accepts (returns $\top$) when w is a codeword, but otherwise outputs a rejection symbol $\bot$ with probability proportional, up to a *testability* factor $\varkappa$, to the distance between w and the code (see Equation (2)). Let LTC be a locally testable code with distance δ_{LTC}, and let w be a string. For a threshold $\varepsilon < \delta_{\mathsf{LTC}}/2$, suppose that we would like to test whether $\mathrm{dist}(w, \mathsf{LTC}) \geq \varepsilon$. We can run $(1/\varkappa\varepsilon)\log(1/p)$ independent runs of T so that if $\mathrm{dist}(w, \mathsf{LTC}) \geq \varepsilon$, then with probability $\geq 1 - p$ at least one of these runs will return $\bot$. Notice the number of runs necessary (and hence the query complexity) only depends on $\varkappa$, ε, and p;

Technique	Query complexity	Due to
multiplicity codes	n^ε	[KSY14]
lifted Reed–Solomon codes	n^ε	[GKS13]
expander graphs	n^ε	[HOW15]
distance amplification	$2^{\tilde{O}(\sqrt{\log n})}$	[KMRS17]
repeated tensoring	$(\log n)^{O(\log\log n)}$	[GRR20]
row-evasive partitions	$(\log n)^{O(\log\log\log n)}$	[CY22]
nested LTCs	$\log^{O(1)} n$	Cor. 5.1

Table 1: High-rate RLCCs (and LCCs) with emphasis on query complexity for block length n.

there is no explicit dependence on the block length of the code. Therefore, two different-sized LTCs satisfying the same parameters would have the same query complexity for certifying that an input has Hamming distance $\leq \varepsilon$ from the code.

Local testing techniques have been used to construct RLDCs and RLCCs since their inception, which makes sense because these codes capture intuitive properties of LTCs and probabilistically checkable proofs (PCPs). Relaxed local decodability originates with Ben-Sasson et al. [BGH+06], and their constant-query RLDC constructions make use of PCPs. Gur, Ramnarayan, and Rothblum [GRR20] continued this line of work by introducing relaxed local correctability, and their high-rate RLCC construction works by showing that tensoring, used by Kopparty et al. [KMRS15] to construct LTCs, preserves relaxed local correctability.

In this work, we will make this intuition formal by presenting a black-box transformation that builds an RLCC from *any* family of linear high-rate LTCs—in contrast to prior work, we will assume nothing about the internal structure of the LTCs we use. To do so, we will need to patch a crucial shortcoming of LTCs. What prevents an LTC from being an RLCC? An RLCC is effectively a “targeted LTC” which, when given a noisy codeword w and an index i, needs to determine whether the ith bit of the noisy codeword is uncorrupted. If so, it is safe to return the ith bit as our answer; otherwise if any corruption is detected, we can return $\perp$. However, an LTC is designed to detect corruption uniformly over the entire input. For example, if w_i is the only bit which has been flipped, then $\operatorname{dist}(w, \mathsf{LTC}) = 1/n$. The LTC’s local tester could have $O(1/n)$ probability of returning $\perp$, in which case $\Omega(n)$ independent trials would be required to detect this single bit flip with constant probability (which is as bad as reading the entire input). Essentially, if a few bits of w have been flipped, then the local tester has little chance of even querying any of them, even though they may be critical to achieving relaxed local correctability.

We address this weakness by building a *nested LTC*. This code combines LTCs of various sizes to be locally testable at every scale: we can gradually “zoom in” to any desired index i and run local testers for successively smaller restrictions (or *blocks*) of the input that contain i. If bits have been flipped, then reducing the block length increases the relative proportion of these flipped bits, such that eventually the relative distance of the restricted input from the closest codeword is high enough to be noticeable by the local tester. To construct a nested LTC of block length n, we start with an explicit family of linear LTCs $\{\mathsf{LTC}_1, \dots, \mathsf{LTC}_m\}$ which all have distance δ_{LTC}, testability $\varkappa$, query complexity r, and block lengths $n_1 < \cdots < n_m = n$ such that $\forall j.\, n_{j+1}/n_j \approx Q$ for a small factor Q. Then, we can build a code C such that for every fixed index i of the codeword, there exist a series of nested[1] blocks $B_1 \subsetneq B_2 \subsetneq \cdots \subsetneq B_m = [n]$ such that $i \in B_1$ and $B_j \in \mathsf{LTC}_j$ for all $j \in [m]$.

Let w be an input such that $\operatorname{dist}(w, C) < \delta_{\mathsf{LTC}}/2$. Let c be the unique codeword of C which is closest to w. Since $C \subseteq \mathsf{LTC}_m$ and the minimum distance of LTC_m is $\geq \delta_{\mathsf{LTC}}$, it follows that c is also the unique codeword of LTC_m which is closest to w:

$$\operatorname{dist}(w, C) = \operatorname{dist}(w, c) = \operatorname{dist}(w, \mathsf{LTC}_m).$$

Therefore, we can bound $\operatorname{dist}(w, c)$ by using the local tester T_m for LTC_m. We can run T_m sufficiently many times to detect with probability $\geq 2/3$ whether

$$\operatorname{dist}(w, c) \geq \frac{\delta_{\mathsf{LTC}}}{2Q} \approx \frac{\delta_{\mathsf{LTC}} n_{m-1}}{2n_m}.$$

If this distance is less than $\delta_{\mathsf{LTC}}/2Q$, then the next smaller restriction of w is very close to the corresponding restriction of c. In the worst case, all of the corrupted indices lie in B_{m-1}, so we can compute the absolute number of corrupted bits and renormalize by n_{m-1}.

$$\operatorname{dist}(w|_{B_{m-1}}, c|_{B_{m-1}}) < \frac{\delta_{\mathsf{LTC}}}{2Q} \cdot \frac{n_m}{n_{m-1}} \approx \frac{\delta_{\mathsf{LTC}}}{2}$$

Now we can repeat the same argument; $w|_{B_{m-1}}$ is so close to $c|_{B_{m-1}}$ that

$$\operatorname{dist}(w|_{B_{m-1}}, C|_{B_{m-1}}) = \operatorname{dist}(w|_{B_{m-1}}, c|_{B_{m-1}}) = \operatorname{dist}(w|_{B_{m-1}}, \mathsf{LTC}_{m-1}).$$

[1] Later on we will show that the blocks do not necessarily have to be nested; instead, there will be a constant number of relevant blocks at each level, such that the unions of the blocks at each level are nested.

This allows us to use the next local tester T_{m-1} to test whether $\mathrm{dist}(w|_{B_{m-1}}, c|_{B_{m-1}})$ is high with probability $\geq 2/3$. We can repeat these steps, gradually narrowing down the relevant view of the codeword until we have reached the final restriction $w|_{B_1}$, at which point we can read the entire block and check all of the parity constraints for LTC_1 to determine if there is any corruption present. The probability of a step returning a false negative (that is, failing to detect that the distance of some restricted input is too high) is at most $1/3$, because the first block with distance $\geq \delta_{\mathsf{LTC}}/2Q$ will trigger a $\perp$ with probability $\geq 2/3$. Therefore, with probability $\geq 2/3$, the relaxed local corrector described above can determine whether the ith index of the input is corrupted, and return $\perp$.

The overall query complexity will be

$$\sum_{j=2}^{m} O\left(\frac{Qr}{\kappa\delta_{\mathsf{LTC}}}\right) + n_1 = O\left(\frac{Qrm}{\kappa\delta_{\mathsf{LTC}}}\right) + n_1.$$

We want to emphasize that because all of the LTCs in the family satisfy the same parameters, the query cost is roughly the same for each of the m levels. Also, while this technique is iterative over m many levels, the testing procedure at each level is self-contained, and does not recurse on other levels. Therefore, the query cost of each level is additive, rather than multiplicative as in prior work [GRR20, CY22]. Hence, as long as the number of levels m, the query cost at each level $O(Qr/\kappa\delta_{\mathsf{LTC}})$, and the smallest block length n_1 are all $\mathrm{polylog}(n)$, our total query complexity will be polylogarithmic. In addition, the rate of these LTCs needs to be high enough, i.e., $\geq 1 - O(1/\log n)$, such that the final code still has constant rate. In fact, the recent breakthrough work of Dinur et al. [DEL+22] constructs families of LTCs with rate arbitrarily close to 1 that fulfill all of our requirements,[2] allowing us to instantiate our construction.

1.2 Related work

Prior constructions and lower bounds. The two main parameter regimes for RLDCs and RLCCs are the constant query regime (optimizing block length for k-bit messages and q queries) and the asymptotically good regime (optimizing query complexity for n-bit codewords). In the constant-query regime, the best known block length is $n = O(k^{1+1/q})$ [AS21], following a line of work [BGH+06, GRR20, CGS22]. Interestingly, this asymptotically matches the block length lower bound for full-fledged LDCs [KT00, Woo07].

Table 1 summarizes the historic state of the art for query-efficient high-rate RLCCs. Other prior works [BFLS91, RS96] give constant-rate RLCCs with n^{ε} query complexity, but these codes do not support rate arbitrarily close to 1, and so are not included in the table. In addition, this table does not include Gopi et al. [GKO+18] who construct LCCs with optimized rate approaching the Gilbert–Varshamov bound, but with n^{ε} query complexity.

Recently, Block et al. [BBC+22] prove an exponential block length lower bound for 2-query RLDCs, asymptotically matching the exponential block length lower bound for 2-query LDCs established by Kerenidis and de Wolf [KdW03]. Gur and Lachish [GL21] and Dall'Agnol, Gur, and Lachish [DGL21] establish lower bounds for arbitrary-query RLDCs, while the recent work of Goldreich [Gol23b] provides an alternative and simpler proof, which is also stronger for certain cases.

Alternative error models. LDCs, LCCs, and their relaxed counterparts have been studied in other error models, distinct from the Hamming worst-case bit flip setting that we study in this work. These codes have been studied in the insertion-deletion error model, where a limited number of bits can be added or removed (rather than simply flipped) anywhere in the codeword [OP15, BBG+20, CLZ20, BBC+22]. In addition, both the Hamming and insertion-deletion models have been studied in the computationally bounded setting, where the adversary choosing where to perform bit flips or insertions/deletions has limited resources. Then, cryptographic assumptions can be used to

[2] Similar codes were also independently discovered by Panteleev and Kalachev [PK22] with rate up to 1/2. We require the stronger rate guarantee from Dinur et al. [DEL+22].

construct LDCs and LCCs [OPS07, HO08, HOSW11, BKZ20, BB21, ABB22] as well as their relaxed counterparts [BGGZ21, BB23].

In particular, the latter two works use additional assumptions to construct asymptotically good RLDCs and RLCCs in the Hamming setting with polylogarithmic query complexity. We achieve this unconditionally.

1.3 Organization

The remainder of the paper is organized as follows. Section 2 introduces all preliminary notation, definitions, and theorems necessary for our result. Section 3 demonstrates how to construct RLCCs with $1 - o(1)$ rate and $\mathrm{polylog}(n)$ query complexity, but $o(1)$ correcting radius. Section 4 then shows how amplify the radius of any RLCC using an LTC with good distance. Section 5 finally constructs the final RLCC with constant rate, constant radius, and polylogarithmic query complexity by using the procedure in Section 4 and a suitable LTC to amplify the radius of the RLCC constructed in Section 3.

2 Preliminaries

2.1 General notation

Let $\mathrm{dist}(x, y)$ denote the relative Hamming distance between two binary strings x and y, and let $\mathrm{dist}(x, C)$ denote the relative Hamming distance between a string x and a subset $C \subseteq \{0,1\}^n$. We say that $f(n) \leq \mathrm{poly}(n)$ if there is a fixed polynomial p such that for large enough n, $f(n) \leq p(n)$, and analogously for $\geq$. We say $f(n) = \mathrm{poly}(n)$ if $f(n) \leq \mathrm{poly}(n)$ and $f(n) \geq \mathrm{poly}(n)$. Analogous conventions are used for $\mathrm{polylog}(n)$, which denotes $\mathrm{poly}(\log n)$. The polynomials implicitly defined by poly or polylog are fixed with respect to all parameters involved.

Denote $\mathbb{Z}$ to be the set of integers, $n\mathbb{Z}$ to be the set of multiples of n, $\mathbb{N}$ to be the set of positive integers, and $\mathbb{F}_2$ to be the finite field of 2 elements. For a positive integer x, define $[x] := \{1, 2, \ldots, x\}$. For integers x, y, let $[\![x, y]\!]$ denote the interval $\{x, x+1, \ldots, y-1, y\}$. For a string $x \in \{0,1\}^n$ and an index set $I \subseteq [n]$, let $x|_I$ denote the restriction of x to the indices in I. For a set of strings $S \subseteq \{0,1\}^n$, let $S|_I$ denote the set $\{x|_I : x \in S\}$.

2.2 Error-correcting codes

In this paper, we treat $\{0,1\}$ and $\mathbb{F}_2$ as interchangeable, and all codes will be binary and linear.

Definition 2.1. A *linear code* with block length n, distance δ, and rate R is a subspace $C \subset \mathbb{F}_2^n$ such that $\min_{c \in C} \mathrm{dist}(c, C \setminus \{c\}) \geq \delta$ and $\dim(C) = Rn$. Furthermore, we say a linear code is explicit if its parity-check matrix (and thus its generator matrix) can be computed in time $\mathrm{poly}(n)$.

Definition 2.2. A code C with dimension k is called *systematic* if there exists an index set I such that $C|_I = \{0,1\}^k$.

2.3 Locally correctable and decodable codes

The study of LDCs and LCCs was first formalized by Katz and Trevisan [KT00]; we refer the reader to Yekhanin's comprehensive survey [Yek12] for more context and details.

Definition 2.3. A code $C : \{0,1\}^k \to \{0,1\}^n$ is a *locally correctable code (LCC)* with correcting radius δ and query complexity r if it has a randomized corrector M that makes $\leq r$ queries such that for every $c \in C$ and every $w \in \{0,1\}^n$ with $\mathrm{dist}(w, c) \leq \delta$,

$$\forall i \in [n].\ \Pr[M^w(i) = c_i] \geq \frac{2}{3}.$$

Definition 2.4. A code $C : \{0,1\}^k \to \{0,1\}^n$ is a *locally decodable code (LDC)* with decoding radius δ and query complexity r if it has a randomized decoder M that makes $\leq r$ queries such that for every $m \in \{0,1\}^k$ and every $w \in \{0,1\}^n$ with $\mathrm{dist}(w, C(m)) \leq \delta$,

$$\forall i \in [n].\ \Pr[M^w(i) = m_i] \geq \frac{2}{3}.$$

Note that a systematic LCC implies an LDC with the same radius and query complexity, because every codeword can be uniquely identified by (and associated with) some restriction. Any linear LCC can be made systematic, and hence implies an LDC. In addition, every LCC or LDC with correcting radius δ also has distance $> 2\delta$, or else there exists some w which has Hamming distance δ from two distinct codewords, contradicting the correctness of the local corrector/decoder. Therefore, we often use correcting radius and distance interchangeably.

A *relaxed* locally correctable code is allowed to detect an error instead of successfully correcting a codeword bit. We use the strongest definition of relaxed locally correctable and locally decodable codes, which features perfect completeness. Recent work by Goldberg [Gol23a] shows that for linear relaxed locally correctable codes, this definition is essentially equivalent to allowing imperfect completeness (the corrector/decoder can err even on true codewords) and requiring nonadaptivity (the corrector/decoder's queries do not depend on the outcome of prior queries). All of the codes we construct can be made nonadaptive—see Remark 5.3.

Definition 2.5. A code $C : \{0,1\}^k \to \{0,1\}^n$ is a *relaxed locally correctable code (RLCC)* with correcting radius δ and query complexity r if it has a randomized corrector M that makes $\leq r$ queries such that

1. (Completeness) For every $c \in C$,

$$\forall i \in [n].\ \Pr[M^c(i) = c_i] = 1.$$

2. (Soundness) For every $c \in C$ and every $w \in \{0,1\}^n$ with $\mathrm{dist}(w, c) \leq \delta$,

$$\forall i \in [n].\ \Pr[M^w(i) \in \{c_i, \perp\}] \geq \frac{2}{3}.$$

Relaxed locally decodable codes are analogously defined. A systematic RLCC (and hence any linear RLCC) implies a relaxed locally decodable code with the same parameters in the same way that an LCC implies an LDC.

Definition 2.6. A code $C : \{0,1\}^k \to \{0,1\}^n$ is a *relaxed locally decodable code (RLDC)* with decoding radius δ and query complexity r if it has a randomized corrector M that makes $\leq r$ queries such that

1. (Completeness) For every $m \in \{0,1\}^k$,

$$\forall i \in [k].\ \Pr[M^{C(m)}(i) = m_i] = 1.$$

2. (Soundness) For every $m \in \{0,1\}^k$ and every $w \in \{0,1\}^n$ with $\mathrm{dist}(w, C(m)) \leq \delta$,

$$\forall i \in [n].\ \Pr[M^w(i) \in \{m_i, \perp\}] \geq \frac{2}{3}.$$

Similar to LCCs, an RLCC with correcting radius δ must also have distance $> \delta$, and without making additional restrictions on the query complexity, this is tight: an RLCC with query complexity n could have a relaxed local corrector that reads the entire input and tests whether it is a codeword or not.

The study of RLDCs originates with Ben-Sasson et al. [BGH+06] and is closely related to probabilistically checkable proofs. Gur, Ramnarayan, and Rothblum [GRR20] introduced the notion of RLCCs, and gave the first constructions.

2.4 Locally testable codes

Locally testable codes (LTCs) are codes with testers that are able to locally check for corruption. We will make use of the following (strong) definition of LTCs:

Definition 2.7. A code $C : \{0,1\}^k \to \{0,1\}^n$ is a *locally testable code (LTC)* with distance δ, testability[3] $\varkappa$, and query complexity r if it has a randomized tester M that makes $\leq r$ queries and returns either $\top$ (accept) or $\bot$ (reject), such that

1. (Completeness) For every $c \in C$,
$$\Pr[M^c = \top] = 1.$$

2. (Soundness) For every $w \in \{0,1\}^n$,
$$\Pr[M^w = \bot] \geq \varkappa \cdot \operatorname{dist}(w, C).$$

Recent breakthroughs by Dinur et al. [DEL+22], and by Panteleev and Kalachev [PK22] were able to construct locally testable codes with constant rate, distance, and query complexity (referred to as c^3-LTCs), thereby resolving a longstanding conjecture and capping off decades of work. In particular, Dinur et al. [DEL+22] were able to construct explicit families of linear LTCs with rate arbitrarily close to 1:

Theorem 2.8 ([DEL+22, Theorem 1.1 and Remark 5.3]). *For any rate $R = 1 - \varepsilon \in (0,1)$, there exist $\delta \geq \Omega(\varepsilon^3)$, $\varkappa \geq \Omega(\varepsilon^{15})$, and $r \leq O((1/\varepsilon)^{20})$ such that there is an explicit infinite family of linear LTCs of rate R, minimum distance δ, testability $\varkappa$, and query complexity r.*

In particular, there exists an odd prime power $q = \Theta((1/\varepsilon)^{10})$ such that for all integers $j \in \mathbb{N}$, there exists an LTC LTC_j with rate $\geq R$, minimum distance $\geq \delta$, testability $\geq \varkappa$, query complexity r, and block length $(r/8) \cdot (q^{3j} - q^j)$.

3 Achieving polylogarithmic query complexity

In this section, we will build RLCCs with constant rate and polylogarithmic query complexity, but with subconstant correcting radius. Later on, we will be able to amplify the correcting radius to be constant while still preserving the asymptotic query complexity.

We will build our RLCC using appropriately instantiated LTCs constructed using Theorem 2.8. The details of our parameter choices are left to Appendix A. In summary, we can get a nice family of $o(\log n)$ many LTCs of block lengths ranging from $\operatorname{polylog}(n)$ to n, where each consecutive code increases in block length by a $\operatorname{polylog}(n)$ factor.

Corollary 3.1 (see Appendix A). *For every sufficiently large $\tilde{n} \in \mathbb{N}$, there exists an integer $n \in [\tilde{n}, O(\tilde{n} \log^{30} \tilde{n})]$ and a family of linear LTCs $\{\mathsf{LTC}_1, \ldots, \mathsf{LTC}_m\}$ such that*

(a) Each LTC_j has rate $R \geq 1 - O(1/\log n)$, distance $\delta_{\mathsf{LTC}} \geq \Omega(\log^{-3} n)$, testability $\varkappa \geq \Omega(\log^{-15} n)$, and query complexity $r \leq O(\log^{20} n)$.

(b) If n_j is the block length of LTC_j, then $n_1 \leq O(\log^{50} n)$, $n_m = n$, and $\forall j.\, n_{j+1}/n_j = \Theta(\log^{30} n)$.

(c) The number of codes is $m = O(\log n / \log \log n)$.

Theorem 3.2. *For every sufficiently large $\tilde{n} \in \mathbb{N}$, there is an integer $n \in [\tilde{n}, O(\tilde{n} \log^{30} \tilde{n})]$ such that there exists an explicit linear code C with rate $\geq 1 - O(1/\log \log n)$, which is a relaxed locally correctable code with correcting radius $\Omega(\log^{-3} n)$ and query complexity $\leq O(\log^{69} n / \log \log n)$.*

[3] This parameter has also been referred to as the *detection probability* e.g. [DEL+22].

In our construction, we will require minimal covers of $[n]$ by intervals of size n_j, which we will refer to as *j-blocks*. For each $j \in [m]$, define the family of j-blocks, as well as the family of all blocks:

$$\mathscr{B}_j \coloneqq \left\{ [\![(\ell-1)n_j + 1, \ell n_j]\!] : \ell \in \left[\left\lfloor \frac{n}{n_j} \right\rfloor \right] \right\} \cup \{ [\![n - n_j + 1, n]\!] \}$$

$$\mathscr{B} \coloneqq \bigcup_{j=1}^{m} \mathscr{B}_j$$

Intuitively, for each j, we are simply covering $[\lfloor n/n_j \rfloor n_j]$ by $\lfloor n/n_j \rfloor$ disjoint j-blocks, and then covering the remaining interval of size $n - \lfloor n/n_j \rfloor n_j < n_j$ with a single j-block if needed.

Our RLCC construction proceeds as follows:

1. Construct codes $\{\mathsf{LTC}_1, \dots, \mathsf{LTC}_m\}$ using Corollary 3.1 and $\tilde{n}$ large enough (to be specified in Claim 3.5).
2. Construct m *layers* $L_1, \dots, L_m$ of block length n as follows:

   $$L_j \coloneqq \left\{ x \in \{0,1\}^n : \forall B \in \mathscr{B}_j .\, x|_B \in \mathsf{LTC}_j \right\}$$

   Each layer is a code which enforces that the restriction to each block in $\mathscr{B}_j$ will be a codeword in LTC_j. If n_j divides n, then L_j is the Cartesian product of n/n_j copies of LTC_j.
3. Construct the final code $C \coloneqq \bigcap_{j=1}^{m} L_j$.

3.1 Code parameters

We first verify that the code is explicit and has good rate and distance.

Proposition 3.3. *The code C described above is linear, explicit, and has rate $\geq 1 - O(1/\log\log n)$.*

Proof. C is an intersection of linear codes, so it is linear.

Explicitness. We just need to show that constructing the parity-check matrix of C is efficient. Indeed by Corollary 3.1, the parity checks of the $\{\mathsf{LTC}_j\}$ can be constructed efficiently. Constructing the parity-check matrix of each L_j can be done by concatenating $|\mathscr{B}_j|$ shifts of the parity-check matrix of LTC_j, where the shift is induced by each $B \in \mathscr{B}$. The final parity-check matrix is the concatenation of the m parity-check matrices from each L_j.

Rate. Because our code is linear, we can compute the rate by upper bounding the number of linear constraints. Recall that $R = 1 - O(1/\log n)$ and n_j are the rate and block-length of each LTC_j, respectively. So each LTC_j has $n_j(1-R)$ linear constraints, and each L_j has $\leq n_j(1-R)|\mathscr{B}_j|$ constraints. The final code C then has at most

$$\begin{aligned}
\sum_{j=1}^{m} n_j(1-R)|\mathscr{B}_j| &= (1-R)\sum_{j=1}^{m} n_j \left\lceil \frac{n}{n_j} \right\rceil \\
&\leq (1-R)\sum_{j=1}^{m} (n + n_j) \leq 2(1-R)mn \\
&\leq O\left(\frac{1}{\log n}\right) \cdot O\left(\frac{n \log n}{\log\log n}\right) \leq O\left(\frac{n}{\log\log n}\right)
\end{aligned}$$

linear constraints. Therefore, the rate of this code is $\geq 1 - O(1/\log\log n)$. □

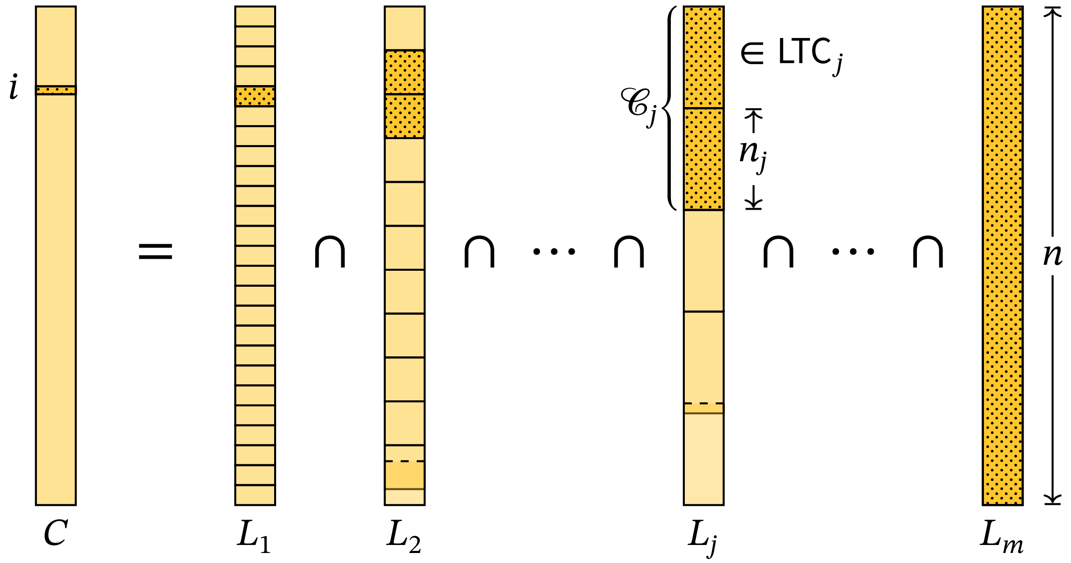

Figure 1: The construction of our RLCC as an intersection of layers. The highlighted (▨) blocks indicate the canary blocks $\mathscr{C}_j$ at each layer L_j for a codeword index i. The dashed lines indicate the overlap between litter blocks and their corresponding runt blocks (see Claim 3.5 for terminology).

3.2 Relaxed local correctability

We now show that C is indeed an RLCC with correcting radius $\delta_{\mathsf{LTC}}/2$ and good query complexity. Let T_j be the tester for LTC_j. Our relaxed corrector M for input w and codeword index i is as follows.

1. Recursively compute the following intervals (which we will refer to as the j-*canary* blocks):
   - Let $\mathscr{C}_1$ be the singleton of an arbitrary 1-block containing i.
   - For $j \geq 2$, $\mathscr{C}_j := \{B \in \mathscr{B}_j : \exists B' \in \mathscr{C}_{j-1} \text{ such that } B \cap B' \neq \varnothing\}$.
2. For descending $j = m, m-1, \ldots, 2$
   - For each canary $B \in \mathscr{C}_j$ repeatedly execute $T_j^{w|_B}$ independently $t_j := 12n_j/\kappa\delta_{\mathsf{LTC}}n_{j-1}$ many times. If $\perp$ is returned by T_j at any point, abort and output $\perp$.
3. For the unique canary $B \in \mathscr{C}_1$, check if $w|_B \in \mathsf{LTC}_1$ by reading all of $w|_B$ and verifying the parity checks of LTC_1. Output w_i if it is, and output $\perp$ otherwise.

Proposition 3.4. *The code C is an explicit RLCC with correcting radius $\delta_{\mathsf{LTC}}/2$ and query complexity $O(\log^{69} n/\log\log n)$.*

Proof.

Explicitness. We first verify M is polytime computable. Step 1 clearly is as this is a matter of dividing integers by n_j. Step 2 runs in polytime since t_j and $j \leq \mathrm{poly}(n)$, and T_j runs in polytime for all j. Step 3 is efficient as LTC_1 is an explicit linear code, so we have access to its parity check matrix.

Query complexity. We now check that the corrector does have polylogarithmic query complexity. Step 1 never queries w, and for Step 3, checking membership in LTC_1 will have query complexity at most the block length n_1. By summing over the iterations of Step 2, the total number of queries is

$$\begin{aligned} n_1 + \sum_{j=2}^{m} \sum_{B \in \mathscr{C}_j} rt_j &\le n_1 + \sum_{j=2}^{m} rt_j \cdot \max_{j \in [m]} |\mathscr{C}_j| \\ &\le n_1 + O\Big(\log^{20} n\Big) \cdot \frac{12}{\kappa \delta_{\mathsf{LTC}}} \cdot \max_{j \in [m]} |\mathscr{C}_j| \cdot \sum_{j=2}^{m} \frac{n_j}{n_{j-1}} \\ &\le O\Big(\log^{50} n\Big) + O\Big(\log^{3+15+20+30} n\Big) \cdot m \cdot \max_{j \in [m]} |\mathscr{C}_j| \\ &\le O\left(\frac{\log^{69} n}{\log\log n}\right) \cdot \max_{j \in [m]} |\mathscr{C}_j|. \end{aligned}$$

We now claim that $\max_j |\mathscr{C}_j| \le 3$, from which the query complexity will follow. To prove this, we will show a stronger claim.

Claim 3.5. For sufficiently large n and all $j \in [m]$, $\mathscr{C}_j$ consists of at most 3 contiguous j-blocks.

Proof. We can set $\tilde{n}$ large enough when constructing the LTC family in Corollary 3.1 so that each $n_{j+1}/n_j > 3$ (which can be done since by construction, each $n_{j+1}/n_j = \omega(1)$). We prove this lemma using induction. The base case is trivial since $\mathscr{C}_1$ is simply one interval.

Now assume $\mathscr{C}_{j-1}$ consists of at most 3 consecutive $(j-1)$-blocks, which implies that the union

$$U := \bigcup_{B \in \mathscr{C}_{j-1}} B$$

is an interval of size $\le 3n_{j-1}$. By construction, $\mathscr{C}_j$ is the set of j-blocks which intersect U. For brevity, call the set of the first $\lfloor n/n_j \rfloor$ contiguous disjoint j-blocks to be the *litter*, and call the remaining single j-block (if it exists) the *runt*. Define the set of points $A = n_j\mathbb{Z} \cap [n]$. Let ℓ be the number of litter j-blocks in $\mathscr{C}_j$. On the one hand, since the right-endpoint of each litter j-block is an element of A, and at most one litter j-block can intersect U but have its right-endpoint outside U, it follows $|U \cap A| \ge \ell - 1$. On the other hand, since $n_j > 3n_{j-1}$ and $|U| \le 3n_{j-1}$, it follows $|U \cap A| \le 1$. Clasping our hands together yields $\ell \le 2$. Accounting for the possibility of a runt j-block gives us the bound of at most 3 consecutive j-blocks in $\mathscr{C}_j$. □

Completeness. We now need to show that for arbitrary $i \in [n]$ and $w \in C$, $M^w(i) = w_i$. This is equivalent to verifying our procedure does not output $\perp$. Indeed by construction of C, we know that for any $w \in C$, $j \in [m]$, and $B \in \mathscr{B}_j$, it must be true that $w|_B \in \mathsf{LTC}_j$. Therefore Step 3 cannot output a $\perp$, and by the completeness of the testers T_j, Step 2 cannot produce a $\perp$. Completeness follows.

Soundness. We are now left with proving perhaps the most nontrivial part of the corrector: its soundness. Let δ_{LTC} be the distance of all the LTC_j. Fix an arbitrary i and assume we have a string $w \in \{0,1\}^n$ such that $w \notin C$ but $\mathrm{dist}(w, C) < \delta_{\mathsf{LTC}}/2$ (which is our correcting radius). Since the distance of C is at least δ_{LTC}, there is a unique codeword $c \in C$ such that $\mathrm{dist}(w, C) = \mathrm{dist}(w, c)$.

We may assume that $w_i \neq c_i$. Notice that M^w either outputs w_i or $\perp$. Therefore, in the case $w_i = c_i$, $M^w(i) \in \{c_i, \perp\}$ surely which satisfies the RLCC definition. It suffices to show that with this assumption, we output $\perp$ with probability $\ge 2/3$. We will utilize two claims. The first tells us how we can use the tester T_j to certify that a j-block in w is close in distance to the corresponding j-block in c.

Claim 3.6. Assume that for some $B \in \mathscr{B}_j$ we have $0 < \varepsilon \le \mathrm{dist}(w|_B, c|_B) < \delta_{\mathsf{LTC}}/2$. Then if T_j is run independently t times on $w|_B$, at least one run will output $\perp$ with probability $\ge 1 - e^{-t\kappa\varepsilon}$.

Proof. Let κ be the testability of all of the $\{\mathsf{LTC}_j\}$. Since $\mathrm{dist}(w|_B, c|_B) < \delta_{\mathsf{LTC}}/2$, we know that $c|_B$ is the closest codeword in LTC_j to $w|_B$. This is because $c|_B \in C|_B \subseteq \mathsf{LTC}_j$ and the distance of LTC_j is

$\geq \delta_{\mathsf{LTC}}$, and so there cannot be a codeword in $\mathsf{LTC}_j \setminus C|_B$ which is closer to $w|_B$ than $c|_B$. Crucially, this allows us to use the local tester T_j to reason about $\mathrm{dist}(w|_B, c|_B)$. Then by the soundness of T_j,

$$\Pr[T_j^{w|_B} = \bot] \geq \kappa \cdot \mathrm{dist}(w|_B, \mathsf{LTC}_j) = \kappa \cdot \mathrm{dist}(w|_B, c|_B) \geq \kappa\varepsilon.$$

Therefore, the probability that t independent runs of T_j never output $\bot$ is at most $(1 - \kappa\varepsilon)^t \leq e^{-t\kappa\varepsilon}$ as desired. □

The second claim shows that low corruption in the j-canaries implies low corruption in the $(j-1)$-canaries.

Claim 3.7. If for all $B \in \mathscr{C}_j$ it is the case $\mathrm{dist}(w|_B, c|_B) < \varepsilon$, then for all $B' \in \mathscr{C}_{j-1}$, $\mathrm{dist}(w|_{B'}, c|_{B'}) < 3\varepsilon n_j / n_{j-1}$.

Proof. From Claim 3.5 we know there are at most 3 blocks in $\mathscr{C}_j$. Therefore if

$$U_j := \bigcup_{B \in \mathscr{C}_j} B,$$

then the number of corrupted bits in $w|_{U_j}$ is

$$\leq \sum_{B \in \mathscr{C}_j} \mathrm{dist}(w|_B, c|_B) \cdot n_j \leq \sum_{B \in \mathscr{C}_j} \varepsilon n_j \leq 3\varepsilon n_j.$$

But by construction of $\mathscr{C}_j$, we know that $\forall B' \in \mathscr{C}_{j-1}. B' \subseteq U_j$. Therefore, the number of corrupted bits in each $w|_{B'}$ must be $\leq 3\varepsilon n_j$ as well, implying that $\mathrm{dist}(w|_{B'}, c|_{B'}) \leq 3\varepsilon n_j / n_{j-1}$ as desired. □

Finally, we are ready to prove soundness. Assume that there is a largest $j \geq 2$ such that there exists a j-canary B with $\mathrm{dist}(w|_B, c|_B) \geq \delta_{\mathsf{LTC}} n_{j-1}/6n_j$. Then, $\mathrm{dist}(w|_B, c|_B) < \delta_{\mathsf{LTC}}/2$:

- If $j = m$ then this follows because $\delta_{\mathsf{LTC}}/2$ is the correcting radius.
- Otherwise, all j'-canaries B' for $j' > j$ satisfy $\mathrm{dist}(w|_{B'}, c|_{B'}) < \delta_{\mathsf{LTC}} n_{j'-1}/6n_{j'}$. Applying Claim 3.7, we get the same upper bound.

This allows us to use Claim 3.6 on B. The probability that the local tester on B returns $\bot$ is

$$\geq 1 - \exp\left(-t_j \kappa \cdot \frac{\delta_{\mathsf{LTC}} n_{j-1}}{6n_j}\right) = 1 - e^{-2} > \frac{2}{3}$$

by Claim 3.6. Let $\mathscr{E}$ be the event that the corrector tests B; that is, if $\neg\mathscr{E}$, then the corrector has terminated early and returned $\bot$ because of a prior iteration. Then,

$$\Pr[M^w(i) = \bot] > \Pr[\neg\mathscr{E}] + (1 - \Pr[\neg\mathscr{E}]) \cdot 2/3 \geq 2/3$$

as needed.

Thus, we may assume that for all $j \geq 2$, every j-canary B satisfies $\mathrm{dist}(w|_B, c|_B) < \delta_{\mathsf{LTC}} n_{j-1}/6n_j$. Then by Claim 3.7, it follows that the unique 1-canary $B_1 \in \mathscr{C}_1$ satisfies

$$\mathrm{dist}(w|_{B_1}, c|_{B_1}) < \frac{3n_2}{n_1} \cdot \frac{\delta_{\mathsf{LTC}} n_1}{6n_2} = \frac{\delta_{\mathsf{LTC}}}{2}.$$

This is less than the minimum distance of LTC_1; because $w_i \neq c_i$, then $w|_{B_1} \notin \mathsf{LTC}_1$ and Step 3 will certainly return $\bot$.

We have shown that for all inputs $w \notin C$ and all $i \in [n]$, the relaxed corrector returns either c_i or $\bot$ with probability $\geq 2/3$, which proves soundness. □

4 Amplifying correcting radius to constant

In this section, we demonstrate how an LTC can be used to amplify the correcting radius of a relaxed locally correctable code. This allows us to strengthen the RLCCs we constructed in the previous section, yielding RLCCs with constant rate, constant correcting radius, and polylogarithmic query complexity.

In the previous section, we constructed RLCCs with all the desired parameters except for the correcting radius, which is $1/\text{polylog}(n)$. In this section, we will boost the radius to a constant by using a similar trick as above: we will take an LTC and nest a layer of our low-radius RLCCs inside it. The LTC allows us to certify that the corruption is low enough that we can reduce to calling the relaxed local corrector on the appropriate restriction of the codeword. In addition, because the local tester is self-contained, the query cost of amplification is additive rather than multiplicative as in prior amplification methods (e.g. [CY22, Claim V.2]).

Theorem 4.1. *Let C be an explicit linear RLCC with block length n, rate R, correcting radius δ, and query complexity r. Suppose that LTC is a linear LTC with block length N, rate R_{LTC}, distance δ_{LTC}, testability $\varkappa$, and query complexity r_{LTC}.*

Then, there exists a linear RLCC C_f with block length N, rate $R_{\mathsf{LTC}} - \Theta(1-R)$, distance $\geq \delta_{\mathsf{LTC}}$, correcting radius $\delta_{\mathsf{LTC}}/2$, and query complexity $r + O(r_{\mathsf{LTC}} N/\varkappa\delta n)$. C_f is explicit whenever LTC is explicit.

Proof. We will again split our N-bit codeword into blocks of size n as follows:

$$\mathscr{B} := \left\{ [\![(\ell-1)n+1, \ell n]\!] \,:\, \ell \in \left[\!\left[\left\lfloor \frac{N}{n} \right\rfloor\right]\!\right] \right\} \cup \{[\![N-n+1, N]\!]\}$$

Build a single layer which enforces that every restriction to a block in B should be a codeword of C:

$$L := \{x \in \{0,1\}^N \,:\, \forall B \in \mathscr{B}.\, x|_B \in C\}$$

Finally, let $C_f := L \cap \mathsf{LTC}$. We can now prove all of the desired properties of C_f. First, we can show the basic properties of C_f as a linear code.

Explicitness. We just need to show that constructing the parity-check matrix of C_f is efficient, when LTC is explicit. Because C is explicit, we can construct the parity-check matrix of L by concatenating $\lceil N/n \rceil$ shifts of the parity-check matrix of C, where the shift is induced by each $B \in \mathscr{B}$. The final parity-check matrix is the concatenation of the matrix of L with the matrix of LTC.

Distance. The distance of C_f is at least δ_{LTC} because $C_f \subseteq \mathsf{LTC}$.

Rate. We proceed by counting the number of linear constraints. Each block in $\mathscr{B}$ contributes $(1-R)n$ linear constraints, and LTC contributes $(1-R_{\mathsf{LTC}})N$ constraints. Then, C_f has at most

$$(1-R_{\mathsf{LTC}})N + \sum_{B \in \mathscr{B}} (1-R)n = (1-R_{\mathsf{LTC}})N + (1-R)n \cdot \left\lceil \frac{N}{n} \right\rceil \leq ((1-R_{\mathsf{LTC}}) + 2(1-R))N$$

linear constraints. It follows that the rate is

$$\geq 1 - ((1-R_{\mathsf{LTC}}) + 2(1-R)) = R_{\mathsf{LTC}} - \Theta(1-R).$$

Next, we can show that C_f is an RLCC with the desired correcting radius and query complexity. Let M be the relaxed local corrector for C, and let T be the local tester for LTC. We define the relaxed local corrector M_f for C_f:

1. Repeatedly execute T^w independently $t := 2N/\varkappa\delta n$ many times. If $\perp$ is returned by T at any point, return $\perp$.
2. Select $B \in \mathscr{B}$ which is an arbitrary block containing i. Execute and return the result of $M^{w|_B}(i)$.

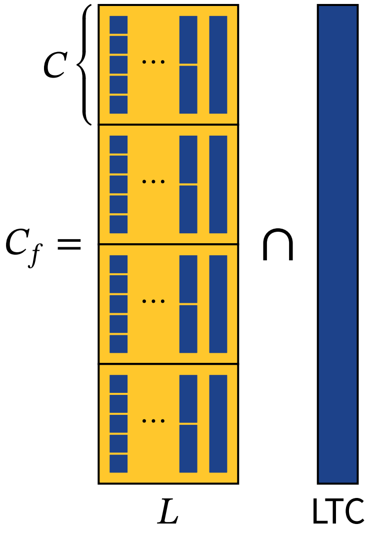

Figure 2: The construction of the final RLCC C_f as an intersection of a layer of smaller RLCCs C with subconstant radius (from Theorem 3.2) with a constant distance LTC.

Explicitness. If both M and T are explicit and efficient, then our corrector M_f is explicit and efficient, since $t \le \mathrm{poly}(N)$.

Query complexity. We make t calls to T and one call to M, so the query complexity is

$$r + r_{\mathsf{LTC}} t = r + O\left(\frac{r_{\mathsf{LTC}} N}{\kappa \delta n}\right).$$

Completeness. M_f always succeeds on every index of an uncorrupted codeword by the completeness of M and T.

Soundness. Let $c \in C_f$ be the closest codeword to an input $w \notin C_f$. By the correcting radius assumption, $\mathrm{dist}(w, c) < \delta_{\mathsf{LTC}}/2$.

- If $\mathrm{dist}(w, c) < \delta n/N$, then for all $B \in \mathscr{B}$, $\mathrm{dist}(w|_B, c|_B) < \delta$. Therefore, either $\perp$ is returned in Step 1, or we return the result of $M^{w|_B}(i)$ in Step 2 for some B that contains i. Because $w|_B$ is within the correcting radius of C, by the soundness of M, the probability that M returns a result which is neither c_i nor $\perp$ is $\le 1/3$. Therefore in either case, with probability $\ge 2/3$, M_f returns c_i or $\perp$.
- Else, $\mathrm{dist}(w, c) \ge \delta n/N$. Then by Claim 3.6, the probability that $\perp$ is returned in Step 1 is

$$\ge 1 - \exp\left(-t\kappa \cdot \frac{\delta n}{N}\right) = 1 - e^{-2} > \frac{2}{3}$$

  Therefore with probability $\ge 2/3$ we return $\perp$ in Step 1.

So in all cases when $0 < \mathrm{dist}(w, C_f) < \delta_{\mathsf{LTC}}/2$, it holds that $\Pr[M_f^w(i) \in \{c_i, \perp\}] \ge 2/3$. □

By instantiating this theorem with our weak RLCC construction from Theorem 3.2, we can get relaxed local correctability from any sufficiently large LTC:

Corollary 4.2. *Let* LTC *be a linear LTC with sufficiently large block length N, and with rate R_{LTC}, distance δ_{LTC}, testability κ, and query complexity r_{LTC}. Then, there exists an RLCC C with block length N,* *rate $R_{\mathsf{LTC}} - O(1/\log\log N)$, distance $\geq \delta_{\mathsf{LTC}}$, correcting radius $\delta_{\mathsf{LTC}}/2$, and query complexity*

$$O\left(\frac{r_{\mathsf{LTC}}}{\kappa} \cdot \log^{33} N + \frac{\log^{69} N}{\log\log N}\right).$$

C is explicit if and only if LTC *is explicit.*

Proof. For any sufficiently large $\tilde{n}$, Corollary 3.1 gives a family of LTCs with maximum block length $n \in [\tilde{n}, O(\tilde{n}\log^{30}\tilde{n})]$, which we use in Theorem 3.2. Hence, pick the largest $\tilde{n}$ such that $n \leq N$. Then,

$$\begin{aligned} \tilde{n} \leq n \leq N \leq (\tilde{n}+1) \cdot O\left(\log^{30}(\tilde{n}+1)\right) \\ \Longrightarrow \frac{N}{n} \leq \frac{N}{\tilde{n}} \leq O\left(\log^{30}\tilde{n}\right) \leq O\left(\log^{30} N\right). \end{aligned}$$

We can plug in this upper bound for N/n in the query complexity, and all of the parameters follow accordingly by substituting the other relevant parameters of our RLCC. □

5 Final construction

We now have all the ingredients to finally construct the codes for the main theorem of this paper.

Corollary 5.1 (explicit RLCCs). *For any rate $R = 1 - \varepsilon \in (0,1)$ and for infinitely many n, there is an explicit RLCC with block length n, rate $R - O(1/\log\log n)$, correcting radius $\Omega(\varepsilon^3)$, and query complexity*

$$O\left((1/\varepsilon)^{35}\log^{33} n + \frac{\log^{69} n}{\log\log n}\right).$$

Proof. We can instantiate Corollary 4.2 using the LTCs of Theorem 2.8. □

Corollary 5.2 (nonexplicit RLCCs approaching Gilbert–Varshamov bound). *Let $H(\cdot)$ be the binary entropy function. For any $R, \delta, \varepsilon \in (0,1)$ such that*

$$R + H(\delta) = 1 - \varepsilon$$

and for infinitely many n, there exists a nonexplicit RLCC with block length n, rate $R - O(1/\log\log n)$ and distance $\geq \delta$, with correcting radius $\delta/2$ and query complexity

$$\mathrm{poly}(1/\varepsilon) \cdot \log^{33} n + O\left(\frac{\log^{69} n}{\log\log n}\right).$$

Proof. Dinur et al. [DEL+22] construct explicit LTCs with rate arbitrarily close to 1, which implies the existence of infinitely many nonexplicit LTCs that approach the Gilbert–Varshamov bound (see [DEL+22, Corollary 1.2]). These LTCs can have any rate R and distance δ such that $R + H(\delta) = 1 - \varepsilon$, in which case the testability is $\kappa \geq \mathrm{poly}(\varepsilon)$ and the query complexity is $r_{\mathsf{LTC}} \leq \mathrm{poly}(1/\varepsilon)$. We can plug these parameters into Corollary 4.2 to yield RLCCs. Because the rate of the RLCC approaches the rate of the LTC, and the distance of the RLCC is at least the distance of the LTC, we can say that the RLCC also approaches the Gilbert–Varshamov bound. □

Remark 5.3. All of the relaxed local correctors we describe can be made nonadaptive (by always performing all of its queries even if $\perp$ is returned) because the local testers of Dinur et al. [DEL+22] are nonadaptive.

Acknowledgments

We would like to thank Dana Moshkovitz for valuable discussions and feedback on a draft of this manuscript. We would also like to thank Siddhartha Jain for valuable advice towards the presentation of this work, Jeffrey Champion for proofreading, and Joshua Cook and Justin Oh for helpful conversations.

References

[ABB22] Mohammad Hassan Ameri, Alexander R. Block, and Jeremiah Blocki. Memory-hard puzzles in the standard model with applications to memory-hard functions and resource-bounded locally decodable codes. In Clemente Galdi and Stanislaw Jarecki, editors, *Security and Cryptography for Networks*, pages 45–68, Cham, 2022. Springer International Publishing. doi:10.1007/978-3-031-14791-3_3. 5

[AS21] Vahid R. Asadi and Igor Shinkar. Relaxed locally correctable codes with improved parameters. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:12, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ICALP.2021.18. 4

[BB21] Alexander R. Block and Jeremiah Blocki. Private and resource-bounded locally decodable codes for insertions and deletions. In *2021 IEEE International Symposium on Information Theory (ISIT)*, pages 1841–1846, 2021. doi:10.1109/ISIT45174.2021.9518249. 5

[BB23] Alexander R. Block and Jeremiah Blocki. Computationally relaxed locally decodable codes, revisited, 2023. arXiv:2305.01083. 5

[BBC$^+$22] Alex Block, Jeremiah Blocki, Kuan Cheng, Elena Grigorescu, Xin Li, Yu Zheng, and Minshen Zhu. On relaxed locally decodable codes for Hamming and insertion-deletion errors, 2022. arXiv:2209.08688. 4

[BBG$^+$20] Alexander R. Block, Jeremiah Blocki, Elena Grigorescu, Shubhang Kulkarni, and Minshen Zhu. Locally decodable/correctable codes for insertions and deletions. In Nitin Saxena and Sunil Simon, editors, *40th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2020)*, volume 182 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.FSTTCS.2020.16. 4

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*, STOC '91, page 21–32, New York, NY, USA, 1991. Association for Computing Machinery. doi:10.1145/103418.103428. 4

[BGGZ21] Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. *IEEE Transactions on Information Theory*, 67(7):4338–4360, 2021. doi:10.1109/TIT.2021.3076396. 5

[BGH$^+$06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM Journal on Computing*, 36(4):889–974, January 2006. Publisher: Society for Industrial and Applied Mathematics. doi:10.1137/S0097539705446810. 1, 3, 4, 6

[BKZ20] Jeremiah Blocki, Shubhang Kulkarni, and Samson Zhou. On locally decodable codes in resource bounded channels. In Yael Tauman Kalai, Adam D. Smith, and Daniel Wichs, editors, *1st Conference on Information-Theoretic Cryptography (ITC 2020)*, volume 163 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITC.2020.16. 5

[CGS22] Alessandro Chiesa, Tom Gur, and Igor Shinkar. Relaxed locally correctable codes with nearly-linear block length and constant query complexity. *SIAM Journal on Computing*, 51(6):1839–1865, 2022. arXiv:https://doi.org/10.1137/20M135515X, doi:10.1137/20M135515X. 4

[CLZ20] Kuan Cheng, Xin Li, and Yu Zheng. Locally decodable codes with randomized encoding, 2020. arXiv:2001.03692. 4

[CY22] Gil Cohen and Tal Yankovitz. Relaxed locally decodable and correctable codes: Beyond tensoring. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 24–35, 2022. doi:10.1109/FOCS54457.2022.00010. 1, 2, 4, 12

[DEL$^+$22] Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 357–374, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3520024. 4, 7, 14, 17, 18

[DGL21] Marcel Dall'Agnol, Tom Gur, and Oded Lachish. A structural theorem for local algorithms with applications to coding, testing, and privacy. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10-13, 2021*, pages 1651–1665. SIAM, 2021. doi:10.1137/1.9781611976465.100. 1, 4

[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. doi:10.1109/TIT.2018.2809788. 2, 4

[GKS13] Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. In *Proceedings of the 4th Conference on Innovations in Theoretical Computer Science*, ITCS '13, page 529–540, New York, NY, USA, 2013. Association for Computing Machinery. doi:10.1145/2422436.2422494. 2

[GL21] Tom Gur and Oded Lachish. On the power of relaxed local decoding algorithms. *SIAM Journal on Computing*, 50(2):788–813, 2021. doi:10.1137/19M1307834. 1, 4

[Gol23a] Guy Goldberg. Linear relaxed locally decodable and correctable codes do not need adaptivity and two-sided error. Technical Report TR23-067, Electronic Colloquium on Computational Complexity (ECCC), May 2023. URL: https://eccc.weizmann.ac.il/report/2023/067/. 6

[Gol23b] Oded Goldreich. On the lower bound on the length of relaxed locally decodable codes. Technical Report TR23-064, Electronic Colloquium on Computational Complexity (ECCC), May 2023. URL: https://eccc.weizmann.ac.il/report/2023/064/. 1, 4

[GRR20] Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Theory of Computing*, 16(18):1–68, 2020. doi:10.4086/toc.2020.v016a018. 1, 2, 3, 4, 6

[HO08] Brett Hemenway and Rafail Ostrovsky. Public-key locally-decodable codes. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 126–143, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. doi:10.1007/978-3-540-85174-5_8. 5

[HOSW11] Brett Hemenway, Rafail Ostrovsky, Martin J. Strauss, and Mary Wootters. Public key locally decodable codes with short keys. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 605–615, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg. doi:10.1007/978-3-642-22935-0_51. 5

[HOW15] Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Information and Computation*, 243:178–190, 2015. 40th International Colloquium on Automata, Languages and Programming (ICALP 2013). doi:10.1016/j.ic.2014.12.013. 2

[KdW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 106–115, New York, NY, USA, 2003. Association for Computing Machinery. doi:10.1145/780542.780560. 4

[KMRS15] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally-testable codes with quasi-polylogarithmic query complexity. Technical Report TR15-110, Electronic Colloquium on Computational Complexity (ECCC), July 2015. URL: https://eccc.weizmann.ac.il/report/2015/110. 3

[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 the ACM*, 64(2):11:1–11:42, 2017. doi:10.1145/3051093. 1, 2

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *Journal of the ACM*, 61(5), September 2014. doi:10.1145/2629416. 2

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, page 80–86, New York, NY, USA, 2000. Association for Computing Machinery. doi:10.1145/335305.335315. 1, 4, 5

[OP15] Rafail Ostrovsky and Anat Paskin-Cherniavsky. Locally decodable codes for edit distance. In Anja Lehmann and Stefan Wolf, editors, *Information Theoretic Security*, pages 236–249, Cham, 2015. Springer International Publishing. doi:10.1007/978-3-319-17470-9_14. 4

[OPS07] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In Lars Arge, Christian Cachin, Tomasz Jurdziński, and Andrzej Tarlecki, editors, *Automata, Languages and Programming*, pages 387–398, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg. doi:10.1007/978-3-540-73420-8_35. 5

[PK22] Pavel Panteleev and Gleb Kalachev. Asymptotically good quantum and locally testable classical LDPC codes. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, page 375–388, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3520017. 4, 7

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996. doi:10.1137/S0097539793255151. 4

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. Technical Report TR07-006, Electronic Colloquium on Computational Complexity (ECCC), January 2007. URL: https://eccc.weizmann.ac.il/report/2007/006. 1, 4

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends in Theoretical Computer Science*, 6(3):139–255, 2012. doi:10.1561/0400000030. 1, 5

A Concrete parameters for locally testable codes

We instantiate the LTC construction from Theorem 2.8 with suitable parameters. In particular, for any $R \in (0,1)$, Dinur et al. [DEL$^+$22] give an explicit construction for a family of LTCs with rate $\geq R$ and distance, testability, and query complexity that are within a polynomial (or inverse polynomial) of $1-R$, such that consecutive codes in the family differ in block size by a factor which is an inverse polynomial of $1-R$. Hence, to avoid a circular dependency in parameters, we can pick an arbitrary sufficiently large $\tilde{n}$ and begin by setting $R = 1 - \Theta(1/\log \tilde{n})$, such that we get a family of LTCs where all of the parameters are polylogarithmic or inversely polylogarithmic in $\tilde{n}$. Then, we show that there is a choice of m such that the mth LTC of the family has block length $n = \mathrm{poly}(\tilde{n})$. This yields a suitable family of m LTCs for an arbitrarily large final block length n, with parameters that are all polylogarithmic or inversely polylogarithmic in n.

Theorem A.1 (computed from [DEL+22, Theorem 1.1, Lemma 5.1, and Remark 5.3]). *For sufficiently large $\tilde{n} \in \mathbb{N}$, there exists an explicit odd prime power $q = \Theta(\log^{10} \tilde{n})$ such that there is an infinite family of explicit linear locally testable codes $\{\mathsf{LTC}_1, \mathsf{LTC}_2, \dots\}$ where every LTC_j satisfies the following parameters:*

(a) block length $n_j = \Theta((q^{3j} - q^j) \cdot \log^{20} \tilde{n})$

(b) rate $R \geq 1 - 1/(100 \log \tilde{n})$

(c) distance $\delta_{\mathsf{LTC}} \geq \Omega(\log^{-3} \tilde{n})$

(d) testability $\varkappa \geq \Omega(\log^{-15} \tilde{n})$

(e) query complexity $r \leq O(\log^{20} \tilde{n})$

Corollary (Corollary 3.1 restated). *For every sufficiently large $\tilde{n} \in \mathbb{N}$, there exists an integer $n \in [\tilde{n}, O(\tilde{n} \log^{30} \tilde{n})]$ and a family of linear LTCs $\{\mathsf{LTC}_1, \dots, \mathsf{LTC}_m\}$ such that*

(a) Each LTC_j has rate $R \geq 1 - O(1/\log n)$, distance $\delta_{\mathsf{LTC}} \geq \Omega(\log^{-3} n)$, testability $\varkappa \geq \Omega(\log^{-15} n)$, and query complexity $r \leq O(\log^{20} n)$.

(b) If n_j is the block length of LTC_j, then $n_1 \leq O(\log^{50} n)$, $n_m = n$, and $\forall j.\, n_{j+1}/n_j = \Theta(\log^{30} n)$.

(c) The number of codes is $m = O(\log n / \log \log n)$.

Proof. Let q and $\{\mathsf{LTC}_1, \mathsf{LTC}_2, \dots\}$ be instantiated with parameter $\tilde{n}$ using Theorem A.1. Let m be the smallest integer such that $n_m \geq \tilde{n}$, and define $n := n_m$. We claim $\{\mathsf{LTC}_1, \dots, \mathsf{LTC}_m\}$ is our desired family of LTCs.

We first show that $\tilde{n}$ and n are asymptotically close so that we can estimate all parameters with respect to n rather than $\tilde{n}$. Since $n_{m-1} < \tilde{n} \leq n_m$ and for all $j \geq 1$,

$$\frac{n_{j+1}}{n_j} = \Theta(1) \cdot \frac{q^{3(j+1)} - q^{j+1}}{q^{3j} - q^j} \leq O(1) \cdot \frac{q^{3(j+1)}}{q^{3j}/2} = O\big(q^3\big),$$

it follows that

$$n = n_m \leq O\big(q^3\big) \cdot n_{m-1} < O\big(q^3\big) \cdot \tilde{n} \leq O\Big(\tilde{n} \log^{30} \tilde{n}\Big).$$

Similarly,

$$\frac{n_{j+1}}{n_j} = \Theta(1) \cdot \frac{q^{3(j+1)} - q^{j+1}}{q^{3j} - q^j} \geq \Omega(1) \cdot \frac{q^{3(j+1)}/2}{q^{3j}} = \Omega\big(q^3\big).$$

This implies that $n_{j+1}/n_j = \Theta(q^3) = \Theta(\log^{30} n)$. In addition, $\tilde{n} \leq n \leq O(q^3 \tilde{n})$, and so $\log n - O(\log \log n) \leq \log \tilde{n} \leq \log n$. Consequently, we can asymptotically express all parameters in terms of the block length of the largest code, n. Now notice

$$n = \Theta\Big((q^{3m} - q^m) \cdot \log^{20} n\Big) \geq \Theta\Big(\log^{20} n\Big) \cdot q^m$$
$$\implies m \leq \log_q\left(\frac{n}{\Theta\big(\log^{20} n\big)}\right) \leq O\left(\frac{\log n}{\log \log n}\right).$$

The desired result follows. □