LCC and LDC: Tailor-Made Distance Amplification and a Refined Separation

Gil Cohen
Department of Computer Science, Tel Aviv University, Israel

Tal Yankovitz
Department of Computer Science, Tel Aviv University, Israel

Abstract

The Alon-Edmonds-Luby distance amplification procedure (FOCS 1995) is an algorithm that transforms a code with vanishing distance to a code with constant distance. AEL was invoked by Kopparty, Meir, Ron-Zewi, and Saraf (J. ACM 2017) for obtaining their state-of-the-art LDC, LCC and LTC. Cohen and Yankovitz (CCC 2021) devised a procedure that can amplify inverse-polynomial distances, exponentially extending the regime of distances that can be amplified by AEL. However, the improved procedure only works for LDC and assuming rate $1 - \frac{1}{\text{poly}\log n}$.

In this work we devise a distance amplification procedure for LCC with inverse-polynomial distances even for vanishing rate $\frac{1}{\text{poly}\log\log n}$. For LDC, we obtain a more modest improvement and require rate $1 - \frac{1}{\text{poly}\log\log n}$. Thus, the tables have turned and it is now LCC that can be better amplified. Our key idea for accomplishing this, deviating from prior work, is to tailor the distance amplification procedure to the code at hand.

Our second result concerns the relation between linear LDC and LCC. We prove the existence of linear LDC that are not LCC, qualitatively extending a separation by Kaufman and Viderman (RANDOM 2010).

2012 ACM Subject Classification Theory of computation → Error-correcting codes

Keywords and phrases Locally Correctable Codes, Locally Decodable Codes, Distance Amplifications

Digital Object Identifier 10.4230/LIPIcs.ICALP.2022.44

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2021/136/` [6]

Funding The research leading to these results has received funding from the ERC starting grant 949499, the Israel Science Foundation grant 1569/18 and from the Azrieli Faculty Fellowship.

1 Introduction

1.1 Distance amplification

It is a recurrent theme in coding theory that the construction of a code is done in two steps. In the first step, a code with weak parameters is constructed, and typically it is the distance of the code that is unsatisfactory. In the second step, one transforms the code obtained in the first step to a code with the desired parameters, where typically, in the process, the other parameters deteriorate only slightly. When the distance is the unsatisfactory parameter, the second step is referred to as a distance amplification step.

Examples that fall into this framework include the breakthrough constructions of near-optimal small-bias sets by Ta-Shma [19], and the state-of-the-art construction of locally decodable codes (LDC), locally correctable codes (LCC), and locally testable codes (LTC) by Kopparty, Meir, Ron-Zewi, and Saraf [17]. A prominent example from the (adjacent) PCP literature is Dinur's celebrated proof of the PCP Theorem by gap amplification [8]. It is interesting to note that in all the above cases the first step is done using algebraic machinery whereas the second step is based on combinatorial arguments.

1.2 LDC and LCC

Informally, a linear (q, δ) *locally decodable code (LDC)* is a code, given by an $\mathbb{F}$-linear encoding function $\mathsf{Enc} : \mathbb{F}^k \to \mathbb{F}^n$, where $\mathbb{F}$ is a finite field, that is also equipped with a "local decoder". The latter is a randomized algorithm, denoted by Dec, with the following guarantee. Given an oracle access to $z \in \mathbb{F}^n$ that is within relative Hamming distance δ from some codeword $\mathsf{Enc}(x)$, and given $i \in [k]$, $\mathsf{Dec}^z(i) = x_i$ with high probability. Moreover, Dec makes at most q queries to z. That is, every message symbol can be decoded, with high probability, by querying only few symbols of a corrupted codeword. A (q, δ) *locally correctable code (LCC)* is the variant in which one wishes to decode (or, more precisely, correct) the codeword symbols rather than the message symbols.

Locally decodable codes were defined by Katz and Trevisan [15] who proved that asymptotically good LDC require $q = \Omega(\log n)$ queries. Whether or not this bound is tight is a major open problem. An intensive research effort is devoted to the study and construction of LDC and LCC. Of particular interest is the study of asymptotically good LDC and LCC [18, 13, 14, 17, 12, 7] where the goal is to minimize the query complexity.

In their seminal work, Kopparty, Meir, Ron-Zewi and Saraf [17] constructed LDC and LCC with sub-polynomial query complexity. For the first step, a code with vanishing distance $\delta = \frac{1}{\mathrm{poly}(\log n)}$ was used [18], having the desired query complexity, namely, $q = 2^{\tilde{O}(\sqrt{\log n})}$. Then, in the second step the authors invoked a distance amplification procedure due to Alon, Edmonds and Luby [2, 1], which was originally introduced in the context of linear-time erasure codes, and observed that it converts an LDC (resp. LCC) with distance δ and query complexity q to an LDC (resp. LCC) with constant distance and query complexity $q_{\mathrm{new}} = q \cdot \mathrm{poly}(\frac{1}{\delta})$.

1.3 Improved distance amplification for LDC

Motivated by the key role that the distance amplification procedure plays in [17], Cohen and Yankovitz [7] asked whether much lower distances can be amplified. Indeed, AEL's procedure is mostly relevant in the regime $\delta = \frac{1}{\mathrm{poly}(\log n)}$. In [7], the authors devised an improved procedure that can amplify distances as low as $\frac{1}{n^\alpha}$ for any constant $\alpha < 1$ with a fairly low cost in query complexity, $q_{\mathrm{new}} = q^{O(\log \log n)}$ [1] (and even for $\alpha = 1 - o(1)$ at a small additional cost in query complexity). However, their improved distance amplification procedure has two drawbacks:

1. Unlike the AEL distance amplification procedure, the improved procedure was only shown to work for LDC (and it may or may not work for LCC).
2. Second, to amplify the distance, the original LDC must have rate close to one, more precisely, rate $1 - \frac{1}{\mathrm{poly}(\log n)}$.

2 Our contribution

We turn to present the two results of this work.

[1] $\mathrm{poly}(\log \log n)$ factors in the exponent of the query complexity can be safely ignored given that, at present, the lowest known query complexity is $2^{\tilde{\Theta}(\sqrt{\log n})}$. Such an overlook will matter only when (and if) the query complexity will go below quasi-poly-logarithmic.

2.1 Tailor-made distance amplification procedure

Our first contribution is a distance amplification procedure for LCC that can amplify distances as low as those handled by [7] (for LDC). Moreover, our procedure works even for vanishing rate LCC.

▶ **Theorem 1** (Distance amplification for LCC; informal). *Let $h \geqslant 1 \geqslant \alpha > 0$ be any constants. There exists a transformation that takes a q-query LCC with distance $\frac{1}{n^{\alpha}}$ and rate $\frac{1}{(\log\log n)^h}$ to an asymptotically good LCC with query complexity*

$$q_{\mathsf{new}} = q^{O((\log\log n)^{2h+2})}.$$

We chose to state our result in a somewhat informal manner. For the formal statement, see Corollary 37.

An example usage of Theorem 1 is given by the next corollary. The corollary shows the implication of a case that an LCC with query complexity meeting the Katz-Trevisan bound is shown to exist - only with a vanishing rate and distance.

▶ **Corollary 2** (Informal). *If there exists a q-query LCC for $q = \log n$, with distance $\frac{1}{\sqrt{n}}$ and rate $\frac{1}{\log\log n}$, then there exists an asymptotically good LCC with query complexity*

$$q_{\mathsf{new}} = (\log n)^{O((\log\log n)^4)}.$$

We now turn to give further details on the result.

Explicitness

In the statement of Theorem 1 we ignore the issue of explicitness. Indeed, understanding LDC and LCC is already interesting in the information-theoretic level. Having said that, our transformation is fairly explicit: It is a zero error randomized transformation that runs in polynomial-time. More precisely, for every "failure" parameter $\varepsilon > 0$, our transformation runs in time $\mathrm{poly}(n) \cdot \log \frac{1}{\varepsilon}$ and produces an LCC with probability at least $1 - \varepsilon$; otherwise, it declares failure. We find this aspect to be a minor issue as, recall, LCC are anyhow randomized in nature. Nonetheless, it will be interesting to obtain a deterministic transformation with matching parameters.

Codes vs. family of codes

A second issue that we chose to sweep under the rug in the statement of Theorem 1 is that the transformation operates on the level of family of codes rather than on the level of individual codes. That is, in order to produce an asymptotically good LCC of a given block-length n, our transformation requires as input a sufficiently dense family of codes. By that we mean that the consecutive block-lengths in the family are not too far apart. The density of the resulted family of codes is the same as that of the original family.

Amplifying lower distances

Like [7], we can even amplify sub-polynomial distances, in particular, distances of the form $1/n^{1-1/g(n)}$ for an increasing function g, and assuming a certain technical relation between g and the rate. In particular, for every constant $m \geqslant 1$ we can handle $g(n) = (\log\log n)^m$, and end up with query complexity

$$q_{\mathsf{new}} = q^{O((\log\log n)^{2h+2m+2})}.$$

We note that constructing a code for $g(n) = \log n$ is trivial.

Amplifying the distance of LDC

We also obtain an improvement for LDC by devising a distance amplification procedure that requires rate $1 - \frac{1}{\text{poly}(\log\log n)}$, modestly improving upon the $1 - \frac{1}{\text{poly}(\log n)}$ rate required by [7]. The reason that we can do much better for LCC is due to the rate amplification procedure of [7] that, informally, can amplify rate ρ LCC with q queries to constant rate LCC with query complexity $q_{\mathsf{new}} = q^{\text{poly}(\frac{1}{\rho})}$. Such a transformation is not known for LDC.

2.1.1 Proof idea

In this section we give a short and informal account on our proof technique, and start by contrasting our technique with prior work. Both the AEL distance amplification procedure, as was used in [17], and the one given by [7] are based on samplers and further involve a "small" code, that is, a code with logarithmic block-length. The latter improves upon the former by using unbalanced samplers (rather than balanced ones, or expander graphs as was used originally [1, 2]) and using a recursive construction. To obtain our result, we deviate from prior work and tailor the distance amplification procedure to the LCC at hand. That is, our procedure is "white box" - it produces a new code with improved distance by first examining the structure of the given code. To tailor the procedure to the LCC at hand, we do not work directly with the definition of LCC as it lacks sufficient structure to work with. Instead, we work with a more combinatorial characterization of LCC as was used in [7]. We turn to elaborate on this.

Let $C \subseteq \mathbb{F}^n$ be a linear (q, δ)-LCC. One can prove the following structural result. With every coordinate $i \in [n]$ one can associate a set, called a *query set*, $A_i = \{Q_1^i, \ldots, Q_m^i\}$ of $m = \delta n/q$ disjoint subsets of $[n]$, each of size at most q, such that the following holds: For every $c \in C$ and $t \in [m]$, c_i can be deduced from $c_{Q_t^i}$. Assume from here on, for simplicity, that $\delta = 1/\sqrt{n}$ and so $m = \sqrt{n}/q$. Denote $\bar{A}_i = \bigcup_{t=1}^m Q_t^i$ and note that $|\bar{A}_i| \leqslant \sqrt{n}$.

For our distance amplification procedure, we make use of a special partition π of $[n]$ into $\sqrt{n}$ parts $P_1, \ldots, P_{\sqrt{n}}$, each of size $\sqrt{n}$. We say that such a partition is a *d-splitter* for C (more precisely, a d-splitter for the query sets $A_1, \ldots, A_n$ obtained from C) if for every $s \in [\sqrt{n}]$ and $i \in [n]$, $|P_s \cap \bar{A}_i| \leqslant d$. We wish to minimize d and thus consider a max load balls into bins like problem: For every $i \in [n]$ we place a ball with color i at each of the coordinates in $\bar{A}_i$. Note that a coordinate $j \in [n]$ may contain many balls of different colors. Indeed, the average number of balls at coordinate $j \in [n]$ is $\sqrt{n}$. Our goal is to choose the partition π in such a way that every part P_t will contain at most d balls of the same color. It is easy to show that a d-splitter for C exists with $d = O(\frac{\log n}{\log\log n})$.

We construct a new code $C' \subseteq \mathbb{F}^n$ as follows. We take C' to be the code $C' \subseteq C$ with the property that for every part P_s of π, when C' is projected to the coordinate set P_s, the obtained vectors consist of codewords of a code $C_{\sqrt{n}}$ having block length $\sqrt{n}$, which is a q'-query LCC. That is to say, we require that for every $c \in C'$ and $s \in [\sqrt{n}]$, $c_{P_s} \in C_{\sqrt{n}}$. Observe that C' can be constructed by adjoining to the parity checks of C, the parity checks of $C_{\sqrt{n}}$ when restricted to each block in π.

We show that if $C_{\sqrt{n}}$ is a smooth LCC, which means that it queries each coordinate with roughly the same probability, then so is C'. Moreover, C' has query complexity qq'. Thus, C can be transformed into a smooth LCC of length n given that a smooth LCC of length $\sqrt{n}$ is at hand. This calls for a recursive construction which results with a smooth LCC with query complexity $q^{O(\log\log n)}$. After obtaining a smooth code, the final step is to invoke the AEL distance amplification to end up with a good LCC. This final step has a minor effect on the query complexity.

The above recursive construction must start with LCC of rate $1 - \frac{1}{\mathrm{poly}(\log\log n)}$. This is due to the rate deterioration throughout the $\log\log n$ recursive calls. For amplifying rate $\frac{1}{\mathrm{poly}(\log\log n)}$ LCC, as stated in Theorem 1, we invoke the rate amplification procedure of [7] before running the recursive construction described above. This has some effect on the density of the LCC family that the recursion has access to which requires some care.

2.2 Refined separation between LDC and LCC

Understanding the relation between LDC and LCC is fundamental. Currently the only regime in which the state of affairs is better understood is the 2-query regime [3, 4, 5]. In the constant-query regime for $q \geqslant 3$, q-LDC with sub-exponential length are known [20, 11, 9] whereas it is not known if this can be matched for q-LCC. Recall that in the constant-rate regime, the state of the art result of [17] achieves sub-polynomial query complexity and holds for LDC and LCC alike.

In the general case, clearly, a systematic LCC is an LDC. As every linear code can be made systematic (by applying Gaussian elimination to its generating matrix), a linear LCC induces a linear LDC with the same parameters. Thus, informally, LCC are stronger than LDC, at least for linear codes.

Are LDC and LCC "equivalent"?

As for the converse, Kaufman and Viderman [16] observed that an LDC is not necessarily an LCC. Their proof starts with an LDC. If it is not an LCC to begin with, we are done. If it is an LCC, the proof goes on by transforming it to a new code by appending to it one additional entry that does not involve low-weight constraints (namely, every vector in the dual code that does not vanish on the new entry is of large weight). In this way, one obtains an LDC with an entry that cannot be corrected with few queries. Such an entry can be shown to exist by a counting argument. This argument can be extended to produce many new bits that cannot be corrected.

While, formally, the argument above establishes the existence of LDC that are not LCC, it has a drawback which makes it somewhat less appealing. In the resulted code, the adjoined bits that cannot be corrected are not needed for decoding the original bits. This means that if one is given a code that is not an LCC because of the above transformation, with the task of taking such a code and "convert" it to an LCC, this could be done easily: simply by removing these coordinates, and this clearly would not harm the code's dimension. This raises the question: Can any linear LDC be so "easily" converted to an LCC of similar dimension and query complexity?

The thought that the answer to this question may turn out to be in the affirmative is not far fetched in the case of linear codes. Indeed, we know that the locality features of linear codes "come from" linear relations between different bits of the codeword and of the message. For example, if a linear code $\mathsf{Enc} : \mathbb{F}^k \to \mathbb{F}^n$ is a q-query LDC, and in particular the i-th bit of each message m can be deduced from a subset $Q \subseteq [n]$ that consists of at most q coordinates of $c = \mathsf{Enc}(m)$, then there exists a linear map $f_{i,Q}$ which satisfies $m_i = f_{i,Q}(c_Q)$ for any m. Likewise, if m_i can as well be deduced from another subset $Q' \subseteq [n]$, $|Q'| \leqslant q$ (as is expected due to the distance guarantee), then there is a linear map $f_{i,Q'}$ satisfying $m_i = f_{i,Q'}(c_{Q'})$ for every m. It follows that in such a case, for every codeword c, $f_{i,Q}(c_Q) = f_{i,Q'}(c_{Q'})$. Since $f_{i,Q}$ and $f_{i,Q'}$ are linear maps (that, we may assume, depend on all their parameters) this means that for every $j \in Q \triangle Q'$, there exists a linear map g_j satisfying $c_j = g_j(c_{Q_j})$ for every codeword c, where $Q_j = (Q \cup Q')\backslash\{j\}$.

Therefore, by the mere fact that $j \in [n]$ is sometimes used in the local decoding process of $i \in [k]$, it is implied that it is possible to "correct" the j-th coordinate by reading only a few locations of the codeword (at most $2q-1$). Thus, the question of whether local decoding implies local correction is in place, in the case of linear codes, and especially so in the setting where k is close to n.

In light of this, the fact that in the separating result of [16] between linear LDC and LCC, the coordinates which are shown to be uncorrectable are not used by the local decoding process, calls for the question of whether there exists a linear LDC with uncorrectable coordinates that are crucial for the decoding process.

Our result

The second contribution of this work is a proof for the existence of an LDC that is not an LCC in the following stronger sense: It contains entries that cannot be corrected which are crucial for the local decoder. This raises the question of what we mean by coordinates that are "crucial". The mere fact that it is possible for a set of coordinates to be queried by the local decoding process should not qualify them as such, as what allows for a code to be locally decodable or locally correctable is that there are many options to decode or correct each symbol. Thus, a more suitable interpretation for a "crucial" set of coordinates $J \subseteq [n]$ is the following: If every coordinate $j \in J$ is "zeroed out" from the code (i.e., for every codeword c we override c_j with zero) then the transformed code is no longer locally decodable. With this we are ready to present our separation.

▶ **Theorem 3** (Separation of LDC and LCC; Informal)**.** *Let $C : \mathbb{F}^k \to \mathbb{F}^n$ for $|\mathbb{F}| > 2$ and $k = \Theta(n)$ be a linear q-query LDC. Then, there exists a linear q^2-query LDC $\widehat{C} : \mathbb{F}^{k^2} \to \mathbb{F}^{n^2}$ with the following property. There exists a subset of coordinates $J \subseteq [n^2]$ in which every coordinate cannot be locally corrected with query complexity $\sqrt{n}$ and correction radius $1/\sqrt{n}$. Moreover, if every coordinate $j \in J$ is zeroed out from the code, then the relative distance of the obtained code is $\widetilde{O}(1/\sqrt{n})$ (and so it is certainly not an LDC).*

For the formal, more general, statement, see Theorem 46. Note that our result does not cover the binary field and it is an interesting question whether it can be extended to include that case.

Proof idea

The underlying idea of the proof of Theorem 3 is an operation on two codes to which we call *weighted tensoring*. The weighted tensoring of codes is similar to the standard tensoring of codes. In the case of standard tensoring, the encoding of the tensor of two codes is done by taking a matrix as input and applying the first code to each column and then applying the second code to each row in the resulted matrix. In the encoding of a weighted tesnor, before the second step, each entry of the matrix is multiplied by a non-zero field element, or weight.

We consider the case of weighted tensoring which is done with *random* weights. We show that while the code resulted from this is an LDC (assuming that the two input codes were so), with high probability there is a set of coordinates in the code that cannot be locally corrected, while being crucial for the decoding. The analysis showing that the set of coordinates cannot be locally corrected is done by considering the affect of the weights on the dual code. A probabilistic argument is then used to show that the argued codes exist.

Discussion

We end this section with a short discussion to clarify a potentially confusing point. While LCC are, in a sense, more powerful than LDC (indeed, our second contribution, Theorem 3, attempts to formalize that better), our first result, given by Theorem 1, transforms a vanishing rate LCC with polynomially-small distance to an asymptotically good LCC–a result that is not known for LDC. So, how can it be that we can do this for LCC and not for the weaker LDC?

Of course, this should cause no confusion as the latter is a *transformation* that works for LCC and not LDC, not a *construction* nor it is even a proof of existence. Put differently, although the transformation generates the stronger object, the transformation is also given it as its input.

3 Preliminaries

3.1 Notations and conventions

Unless stated otherwise, all logarithms are taken to the base 2. For $n \in \mathbb{N}$, we use $[n]$ to denote the set $\{1, \ldots, n\}$. For ease of readability, we sometimes avoid the use of floor and ceiling. This does not affect the stated results. We use $\mathbb{F}$ to denote a field, and any referenced field is assumed to be finite and of a constant size. When n and $\mathbb{F}$ are clear from context, we use $e_i \in \mathbb{F}^n$ to denote the i-th vector of the standard basis. For $q \in \mathbb{N}$, we use H_q to denote the q-ary entropy function, and H to denote the binary entropy function. For a vector $v \in \mathbb{F}^n$, we denote by $|v|$ the *hamming weight* of v, which is the number of its non-zero coordinates $|v| = |\{j \in [n] \mid v_j \neq 0\}|$, and the *support* of v is $\mathsf{supp}(v) = \{j \in [n] \mid v_j \neq 0\}$. For two vectors $u, v \in \mathbb{F}^n$, we denote their (absolute) hamming distance by $\mathsf{dist}(u, v)$. For a linear subspace $L \subseteq \mathbb{F}^n$, we denote by $L^{\leqslant q}$ the set of vectors of weight at most q. For two vector $u, v \in \mathbb{F}^n$, we use $\langle u, v \rangle$ to denote the inner product of u and v, $\sum_{i=1}^{n} u_i v_i \in \mathbb{F}$. For a vector $v \in \mathbb{F}^n$ and a sequence $I = (i_1, \ldots, i_m) \in [n]^m$, we denote by v_I the vector $(v_{i_1}, \ldots, v_{i_m}) \in \mathbb{F}^m$. For a linear subspace $L \subseteq \mathbb{F}^n$ and a sequence $I = (i_1, \ldots, i_m) \in [n]^m$, we denote by L_I the subspace $\{v_I \mid v \in L\}$. Note that L_I is indeed a subspace as it is given by a suitable projection.

A *partition* π of size k of $[n]$ is a set $\{P_1, \ldots, P_k\}$ of disjoint subsets of $[n]$, such that $P_1 \cup \cdots \cup P_k = [n]$. A partition $\{P_1, \ldots, P_k\}$ is *ordered* if each P_i is a sequence rather than a set (and the sequences, when viewed as sets, satisfy the same requirements). Throughout this paper, any partition of $[n]$ will be an ordered partition (though we may not state it explicitly) with the sequences defined by the natural increasing order of $\mathbb{N}$.

3.2 Error correcting codes

We start by recalling the definition of an error correcting code, and of a family of error correcting codes. In this work we only consider linear codes.

▶ **Definition 4.** *For $n \in \mathbb{N}$ and $\mathbb{F}$ a field, a* code *of length n over $\mathbb{F}$ is a linear subspace $C \subseteq \mathbb{F}^n$.*[2] *The dimension of the code, denoted by k, is the dimension of C over $\mathbb{F}$, $\dim_{\mathbb{F}} C$. The (non-local) distance of the code, denoted by d, is $\min_{c \in C, c \neq 0} |c|$. The rate of the code, denoted by ρ, is k/n. The (non-local) relative distance of the code, denoted by Δ, is d/n. The elements of C are called* codewords.

[2] We may omit the phrase "over $\mathbb{F}$" if the underlying field is clear from context.

We will also need to consider encodings of codes.

▶ **Definition 5.** *We call a function* $\mathsf{Enc} : \mathbb{F}^k \to \mathbb{F}^n$ *an* encoding *of a code* C *if it is an injective linear map and* $C = \mathsf{Im}(\mathsf{Enc})$.

▶ **Definition 6.** *For a field* $\mathbb{F}$*, a* code family *over* $\mathbb{F}$ *is a set of codes* $C = \{C^n\}$*, which contains at most one code* C^n *of length* n *over* $\mathbb{F}$*, for every possible length* $n \in \mathbb{N}$*. For every* $n \in \mathbb{N}$*, we denote by* $\lceil n \rceil^C$ *the minimal length of a code in the family* C *of length at least* n*, and by* $\lfloor n \rfloor^C$ *the maximal length of a code in the family of length at most* n*. For constants* $n_0 \in \mathbb{N}$*,* $c \geqslant 1$ *and* $d \leqslant 1$*, we say that the family is* (n_0, c, d)-dense *if for every* $n \geqslant n_0$*,* $\lceil n \rceil^C \leqslant cn$ *and* $\lfloor n \rfloor^C \geqslant dn$*.*

▶ **Definition 7.** *For a field* $\mathbb{F}$*, a* code-encoding family *over* $\mathbb{F}$ *is a set of pairs of codes and corresponding encodings* $C = \{(C^k, \mathsf{Enc}^k)\}$*, which contains at most one code* C^k *of dimension* k *over* $\mathbb{F}$*, for every possible dimension* $k \in \mathbb{N}$*. For every* $k \in \mathbb{N}$*, we denote by* $\lceil k \rceil^C$ *the minimal dimension of a code in the family* C *of dimension at least* k*, and by* $\lfloor k \rfloor^C$ *the maximal dimension of a code in the family of dimension at most* k*. For constants* $k_0 \in \mathbb{N}$*,* $c \geqslant 1$ *and* $d \leqslant 1$*, we say that the family is* (k_0, c, d)-dense *if for every* $k \geqslant k_0$*,* $\lceil k \rceil^C \leqslant ck$ *and* $\lfloor k \rfloor^C \geqslant dk$*.*

▶ **Definition 8.** *Let* C *be a code of length* n *over* $\mathbb{F}$*. The* dual code of C *is defined to be its orthogonal subspace* $C^\perp$*.*

▶ **Definition 9.** *Let* C *be a code of length* n *over* $\mathbb{F}$*, let* $i \in [n]$ *and* $B \subseteq [n]$*. We say that* B determines i in C *if there exists a function* $f : \mathbb{F}^{|B|} \to \mathbb{F}$ *such that for every* $c \in C$*,* $c_i = f(c_B)$*.*

We also need the following property of linear codes.

▶ **Fact 10.** *Let* C *be a code of length* n *over* $\mathbb{F}$*. Further let* $i \in [n]$*,* $Q \subseteq [n]$ *and* $x \in \mathbb{F}^{|Q|}$*. Then, one of the following cases must hold.*

1. *There is at most one* $\alpha \in \mathbb{F}$ *for which there exists some* $c \in C$ *satisfying* $c_Q = x$ *and* $c_i = \alpha$*.*
2. *For every* $\alpha \in \mathbb{F}$ *there is an equal number of* $c \in C$ *for which* $c_i = \alpha$*.*

In particular, either no function (even randomized) of c_Q *can predict* c_i *with probability larger than* $1/|\mathbb{F}|$*, when* $c \in C$ *is randomly chosen uniformly, or* c_Q *determines* c_i *for all* $c \in C$*.*

3.3 Locally decodable codes and locally correctable codes

▶ **Definition 11.** *For* $C \subseteq \mathbb{F}^n$*, we say that a procedure* $f : A \to B$ *is with* oracle access to $c \in C$ *if when* f *is run, it gets besides an input* $a \in A$*, access to* $c \in C$*:* f *can query* c_i *for indices* $i \in [n]$*. To describe a specific run of* f *with input* $a \in A$ *and oracle access to* $c \in C$*, we either say that* $f(a)$ is run with oracle access to c*, or write* $f^c(a)$ *for short. We say that* f *is* non-adaptive *if the queries it makes are independent of* $c \in C$*.*

▶ **Definition 12.** *For a code* C *of length* n *and dimension* k *over* $\mathbb{F}$*, and* Enc *and encoding of it,* (C, Enc) *is called a* (q, δ, ε)-LDC *(locally decodable code, abbreviated) if there exists a randomized procedure* $\mathsf{Dec} : [k] \to \mathbb{F}$ *that is given an oracle access to* $z \in \mathbb{F}^n$*, and has the following guarantee. For every* $i \in [k]$*,* $x \in \mathbb{F}^k$ *and* $z \in \mathbb{F}^n$ *satisfying* $\mathsf{dist}(z, \mathsf{Enc}(x)) \leqslant \delta n$*,* $\mathsf{Dec}^z(i) = x_i$ *with probability at least* $1 - \varepsilon$*. Furthermore,* $\mathsf{Dec}^z(i)$ *always makes at most* q *queries to* z*. We further require that* Dec *is non-adaptive. We call* Dec *a* local decoder *(or* decoder*) for* (C, Enc)*, and the parameter* q *is called the* query complexity *of* (C, Enc)*.*

▶ **Definition 13.** *A code-encoding family $C = \{(C^k, \mathsf{Enc}^k)\}$ of codes over $\mathbb{F}$ is called* a family of good $q(k)$-LDC*, or a* a family of good LDC with query complexity $q(k)$*, if every code C^k in the family is a code with rate at least $\rho(k)$, which is a $(q(k), \delta(k), \varepsilon(k))$-LDC, for $\rho(k) = \Omega(1)$, $\delta(k) = \Omega(1)$, and $\varepsilon(k) \leqslant 1/3$.*

We have the following easy fact.

▶ **Fact 14.** *If C is a code of length n and dimension $k > 0$ over $\mathbb{F}$ and Enc is an encoding of it, and if (C, Enc) is a (q, δ, ε)-LDC, then, provided that $\varepsilon < 1 - 1/|\mathbb{F}|$, the (non-local) relative distance of C, Δ, satisfies $\Delta > \delta$.*[3]

▶ **Definition 15.** *A code C of length n over $\mathbb{F}$ is called a* (q, δ, ε)-LCC *(locally correctable code, abbreviated) if there exists a randomized procedure $\mathsf{Cor} : [n] \to \mathbb{F}$ that is given an oracle access to $z \in \mathbb{F}^n$, and has the following guarantee. For every $i \in [n]$, $y \in C$ and $z \in \mathbb{F}^n$ satisfying $\mathsf{dist}(z, y) \leqslant \delta n$, $\mathsf{Cor}^z(i) = y_i$ with probability at least $1 - \varepsilon$. Furthermore, $\mathsf{Cor}^z(i)$ always makes at most q queries to z. We further require that Cor is non-adaptive and that $\mathsf{Cor}(i)$ never queries i*[4]*. We call Cor a* local corrector *(or* corrector*) for C, and the parameter q is called the* query complexity *of C.*

▶ **Definition 16.** *For a code C of length n over $\mathbb{F}$ (not necessarily a (q, δ, ε)-LCC), and $i \in [n]$, we say that i is a* (δ, q, ε)-correctable coordinate *in C if there exists a procedure $\mathsf{Cor} : [n] \to \mathbb{F}$ such that $\mathsf{Cor}(i)$ satisfies the requirements in Definition 15.*

▶ **Definition 17.** *A family $C = \{C^n\}$ of codes over $\mathbb{F}$ is called* a family of good $q(n)$-LCC*, or a* a family of good LCC with query complexity $q(n)$*, if every code C^n in the family is a code with rate at least $\rho(n)$, which is a $(q(n), \delta(n), \varepsilon(n))$-LCC, for $\rho(n) = \Omega(1)$, $\delta(n) = \Omega(1)$, and $\varepsilon(n) \leqslant 1/3$.*

The following well-known fact is an implication of the fact that every linear code has a systematic encoding[5].

▶ **Fact 18.** *If a code C is a (q, δ, ε)-LCC, then there exists an encoding Enc such that (C, Enc) is a (q, δ, ε)-LDC.*

4 Tailor made distance amplification

4.1 Characterization of LCC

In this section, we will need to use two characterizations of LCC, as was given by Definition 15. The first, given next in Definition 19, is of a (q, τ)-LCC, and resembles the definition of smooth codes given by [15] for LDC. A (q, τ)-LCC differs from a (q, δ, ε)-LCC in that its local correction is only required to succeed if it is given a codeword of the code, rather than a possible corrupted codeword. Accordingly, the correction of a (q, τ)-LCC has no "distance" guarantee, but instead it is required not to query any coordinate with too high probability, i.e., probability larger than τ. When we will construct an LCC, it will be easier to first argue that it is a (q, τ)-LCC and use that to show it can be made into a (q, δ, ε)-LCC for any ε and $\delta = \varepsilon/(\tau n)$.

[3] Note that in the case that $\varepsilon < 1/2$ a stronger bound $\Delta > 2\delta$ holds.

[4] The assumption that $\mathsf{Cor}(i)$ never queries i is only for simplicity. Any LCC which defies this assumption can be easily converted to one which does not, with a negligible effect on δ.

[5] An encoding Enc is a *systematic* encoding if for some $f : [k] \to [n]$, for all $x \in \mathbb{F}^k$ and $i \in [k]$, $\mathsf{Enc}(x)_{f(i)} = x_i$.

The second characterization, which will be given in Definition 23, is of what we call a (q,τ)-*query-set LCC*. Informally, a code is (q,τ)-*query-set LCC* if for every coordinate we have a large enough set of disjoint subsets of $[n]$, from which it can be decoded. The distance amplification procedure that we define utilizes these query sets and so the properties of the input code that we will use are that of its characterization as a (q,τ)-query-set LCC. This is, in a sense, a more "combinatorial" characterization of LCC which can be more conveniently used when a manipulation of these objects is needed.

The three characterizations of LCC all imply each other, but some of the transitions are at some cost to the parameters. Indeed, Claim 20 will show that a (q,τ)-LCC is a (q,δ,ε)-LCC for $\delta = \varepsilon/(\tau n)$, Claim 24 will show that a (q,τ)-query-set is a (q,τ)-LCC, and Claim 25 will complete the cycle and show that a (q,δ,ε)-LCC is a (q,τ)-query-set LCC for $\tau = q/(\delta n)$.

▶ **Definition 19.** *A code C of length n over $\mathbb{F}$ is called a (q,τ)-LCC if there exists a randomized procedure $\mathsf{Cor} : [n] \to \mathbb{F}$ that is given an oracle access to $c \in C$, and has the following guarantee. For every $i \in [n]$ and $c \in C$, $\mathsf{Cor}^c(i) = c_i$, with probability 1. Furthermore, $\mathsf{Cor}^c(i)$ always makes at most q queries to c, and for every $j \in [n]$, the probability that c_j is queried by $\mathsf{Cor}^c(i)$ is at most τ. We further require that Cor is non-adaptive and that $\mathsf{Cor}(i)$ never queries i. We call the parameter q* the query complexity *and the parameter τ the* smoothness *of the LCC.*

▷ Claim 20. Let C be a code of length n which is a (q,τ)-LCC. Then, for any $\varepsilon > 0$, C is a (q,δ,ε)-LCC with $\delta = \varepsilon/(\tau n)$.

Proof. Let $\varepsilon > 0$ and let Cor be a corrector of C. Let $c \in C$ and $z \in \mathbb{F}^n$ such that $\mathsf{dist}(c,z) \leqslant \delta n = \varepsilon/\tau$, and set $B = \{j \in [n] \mid z_j \neq c_j\}$. Fix $i \in [n]$. By the union bound over $j \in B$, except with probability ε, when $\mathsf{Cor}(i)$ is run with oracle access to $c \in C$, it does not make a query to an index in B. If this is the case, then if Cor was given access to z instead of c, it would successfully output c_i, as well. Thus, C is indeed a (q,δ,ε)-LCC as the same corrector Cor can be used with oracle access to strings $z \in \mathbb{F}^n$, and given that $\mathsf{dist}(c,z) \leqslant \delta n$, $\mathsf{Cor}(i)$ is promised to output c_i with probability at least $1-\varepsilon$. ◁

▶ **Definition 21.** *A set $\mathcal{A} = \{A_1, \ldots, A_n\}$ is called an* n-query-set *if for every $i \in [n]$, A_i is a set of disjoint subsets of $[n]\backslash\{i\}$. For every $i \in [n]$ we define $\overline{A_i} = \bigcup_{B \in A_i} B$.*

▶ **Definition 22.** *Let C be a code of length n and let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be an n-query-set. $\mathcal{A}$ is said to be a* query-set for C *if for every $i \in [n]$ and $B \in A_i$, B determines i in C (see Definition 9).*

▶ **Definition 23.** *Let C be a code of length n. C is said to be a* (q,τ)-query-set-LCC *if there exists a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ which is a query-set for C, such that for every $i \in [n]$, $|A_i| \geqslant 1/\tau$ and for every $B \in A_i$, $|B| \leqslant q$.*

▷ Claim 24. Let C be a code of length n over $\mathbb{F}$ which is a (q,τ)-query-set LCC. Then C is a (q,τ)-LCC.

Proof. Let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be a query set that corresponds to C being a (q,τ)-query-set LCC. The following corrector Cor shows that C is a (q,τ)-LCC. Given $i \in [n]$, and oracle access to $c \in C$, $\mathsf{Cor}(i)$ samples uniformly at random some $B \in A_i$ and queries c_B. As B determines i in C, there exists a function f satisfying $f(c_B) = c_i$ for every $c \in C$, and so $\mathsf{Cor}(i)$ uses such a function and outputs its result. Thus, for every $c \in C$, the output of $\mathsf{Cor}(i)$ is always equal to c_i, and note that as any sampled $B \in A_i$ satisfies $|B| \leqslant q$, $\mathsf{Cor}(i)$ always makes at most q queries. Since A_i is of size at least $1/\tau$ and is composed of disjoint subsets of $[n]\backslash\{i\}$, any coordinate is queried by $\mathsf{Cor}(i)$ with probability at most τ, and $\mathsf{Cor}(i)$ never queries i. Thus, C is a (q,τ)-LCC. ◁

▷ Claim 25. Let C be a code of length n over $\mathbb{F}$ which is a (q, δ, ε)-LCC, for $\varepsilon < 1 - 1/|\mathbb{F}|$. Then, C is a (q, τ)-query-set-LCC for $\tau = q/(\delta n)$.

The proof for the claim is similar to the proof in [15] to their Theorem 1 and to the proof in [10] for Theorem 1.1.

Proof for Claim 25. To prove the claim, we need to show that there exists a set $\mathcal{A} = \{A_1, \ldots, A_n\}$ which is a query-set for C, such that for every $i \in [n]$, $|A_i| \geqslant 1/\tau = \delta n/q$ and for every $B \in A_i$, $|B| \leqslant q$. We construct $\mathcal{A}$ with the required properties by constructing each of the subsets separately. Let Cor denote a corrector promised by the fact that C is a (q, δ, ε)-LCC, and let $i \in [n]$. To construct A_i, we construct a sequence of disjoint sets $B^i_1, \ldots, B^i_{m_i} \subseteq [n] \backslash \{i\}$, in an iterative manner. We will eventually set $A_i = \{B^i_1, \ldots, B^i_{m_i}\}$. It will hold that for every j, B^i_j determines i in C, while satisfying $|B^i_j| \leqslant q$, and that $m_i \geqslant \delta n/q$, which will conclude the proof.

The construction of $B^i_1, \ldots, B^i_{m_i} \subseteq [n]$ is done by the following procedure. Start by setting $B^i_0 = \varnothing$. For $j = 1, 2, \ldots$, set $S^i_j = B^i_0 \cup \cdots \cup B^i_{j-1}$. If $|S^i_j| > \delta n$ halt and set $m_i = j - 1$ and $A_i = \{B^i_1, \ldots, B^i_{m_i}\}$. Otherwise, it holds that for every $c \in C$, for every modification of the coordinates in S^i_j to some erroneous values, $\mathsf{Cor}(i)$ correctly outputs c_i with probability at least $1 - \varepsilon$. An equivalent description of this case is the following: for every $c \in C$ and $z : S^i_j \to \mathbb{F}$, define $c^z \in \mathbb{F}^n$ such that for every $r \notin S^i_j$, $c^z_r = c_r$ and for $r \in S^i_j$, $c^z_r = z(r)$. The corrector Cor chooses a set of queries $Q \subseteq [n] \backslash \{i\}$, $|Q| \leqslant q$, according to some distribution[6] and applies some function f_Q on c^z_Q. We know that with probability at least $1 - \varepsilon$, $f_Q(c^z_Q) = c_i$. Since Q is sampled in a manner that is independent of c and z, by an averaging argument, there exists some fixed Q for which when $c \in C$ and $z : S^i_j \to \mathbb{F}$ are chosen randomly in a uniform manner, with probablity at least $1 - \varepsilon$ (this time over the choice of c and z), $f_Q(c^z_Q) = c_i$. Therefore, we can define another function f'_Q that only gets $c_{Q \backslash S^i_j}$, chooses z uniformly at random, and outputs $f_Q(c^z_Q)$. If $c \in C$ is chosen uniformly at random, $f'_Q(c_{Q \backslash S^i_j}) = c_i$ with probability at least $1 - \varepsilon > 1/|\mathbb{F}|$. By Fact 10, this implies that $Q \backslash S^i_j$ determines i in C. We therefore set $B^i_j = Q \backslash S^i_j$[7], and proceed to the next j.

As this process only halts when $|S^i_j| > \delta n$, and for every j, $|S^i_j| \leqslant q(j-1)$, we have that $m_i \geqslant \delta n/q$. Further note that by the choice of each B^i_j, the sets $B^i_1, \ldots, B^i_{m_i}$ are disjoint, and of size at most q, as required. This thus shows how each A_i can be constructed, and the claim follows. ◁

4.2 Splitters for query sets

Splitters for query sets, that are defined as follows, are key ingredients in our distance amplification procedure. Informally, a *c-splitter* for a query set $\mathcal{A} = \{A_1, \ldots, A_n\}$ is partition of $[n]$ which satisfies that for every i, the intersection between $\overline{A_i}$, the union all the sets in A_i that correspond to an index i, and each part of the partition, is not too large, i.e., of size at most c. In the distance amplification procedure, we will describe a corrector which samples a set $B \in A_i$, in some query set $\mathcal{A}$, and then makes queries according to which parts of the c-splitter intersect with B. For the resulted queries to be smooth, we will need the partition to "split" $A_1, \ldots, A_n$, meaning that no part of the partition is too common within any certain A_i.

[6] As the corrector in non-adaptive, $\mathsf{Cor}(i)$ naturally induces a distribution on subsets of $[n]$ which correspond to the possible query sets.

[7] Note that $i \notin B^i_j$, as $i \notin Q$, since $\mathsf{Cor}(i)$ by definition never queries i.

▶ **Definition 26.** *Let $n \in \mathbb{N}$, $\mathcal{A}$ an n-query-set and $c \in \mathbb{N}$. A partition π of $[n]$ is called a* c-splitter of $\mathcal{A}$ *if for every $i \in [n]$ and $P \in \pi$, $|P \cap \overline{A_i}| \leqslant c$.*

The next claim shows that if each A_i is of size at most k, then c-splitters with k parts exist, for c, the bound on the maximal intersection, being equal to roughly the minimal intersection that is possible, up to a constant factor.

▷ Claim 27. Let $n, k, q \in \mathbb{N}$ such that $k/n \leqslant 1$ and $q \geqslant \log n$. Further let $\mathcal{A} = \{A_1, \ldots, A_n\}$ be an n-query-set such that for every $i \in [n]$, $|A_i| \leqslant k$ and for every $B \in A_i$, $|B| \leqslant q$. Then, there exists a partition π of $[n]$ with k parts, each of size n/k, which is a c-splitter of $\mathcal{A}$ for $c = 2eq$.

Proof. The proof is by a probabilistic argument. We randomly choose a partition π with k equally-sized parts in a uniform manner among all such partitions. We bound the probability that π is not a c-splitter for $\mathcal{A}$: this is the case if $|\overline{A_i} \cap P| > c$ for some $i \in [n]$ and P a part of π. Towards this end, we first fix some $i \in [n]$ and $t \in [k]$, and let P_t denote the t-th part of π. We have that for every $j \in \overline{A_i}$ the probability that $j \in P_t$ is $1/k$, and for every fixed subset of $\overline{A_i}$ of size c, the probability that it is contained in P_t is at most $(1/k)^c$ (since for distinct $j, j' \in \overline{A_i}$, the events that $j \in P_t$ and $j' \in P_t$ are negatively correlated). By a union bound over the possible subsets of size c, the probability that $|\overline{A_i} \cap P_t| > c$ is at most

$$\begin{aligned}\binom{|\overline{A_i}|}{c}(1/k)^c &\leqslant \left(\frac{e|\overline{A_i}|}{ck}\right)^c \\ &\leqslant \left(\frac{eq}{c}\right)^c \\ &= \left(\frac{1}{2}\right)^{2eq}.\end{aligned}$$

By taking a union bound over all possible i, t, the probability that there exist $i \in [n]$ and $t \in [k]$ such that $|\overline{A_i} \cap P_t| > c$ is at most $nk\left(\frac{1}{2}\right)^{2eq} \leqslant n^2\left(\frac{1}{2}\right)^{2eq}$, which is less than 1 a $q \geqslant \log n$, and the claim follows. ◁

4.3 The distance amplification procedure

We now turn to define the basic operation behind our distance amplification procedure. This operation "composes"[8] two codes of different lengths, a big code and a small code, in a way that is parameterized by some partition of $[n]$. The result is a code of the same length as the big code, with an improved smoothness (if the partition satisfies certain requirements), as we will have in the claims that follow the definition. The distance amplification procedure (or perhaps, more directly, the smoothness amplification procedure) will be an iterative application of this composition.

▶ **Definition 28.** *Let C_1 be a code of length n_1, C_2 a code of length n_2, π a partition of $[n_1]$ into n_1/n_2 parts of size n_2. We define the* π-composition of C_1 and C_2, *which we denote by $C_1 \odot_\pi C_2$, to be the code $\{c \in C_1 \mid \forall P \in \pi \quad c_P \in C_2\}$.*

A bound on the rate of the composition of two codes is given in the following claim.

[8] Note that the term "composition" here is used in a different sense than the usual composition of two codes in coding theory, which is achieved from the composition of the encoding functions.

▷ Claim 29. If C_1, C_2 are codes with of lengths n_1, n_2 and rates ρ_1, ρ_2 respectively, then $C = C_1 \odot_\pi C_2$ is a code of length n_1 and rate at least $\rho_1 + \rho_2 - 1$.

Proof. That the length of C is n_1 follows from the definition. As for the rate, by inspecting the code dual to C, it can be seen that the dimension of $C^\perp$ is at most

$$d = (1-\rho_1)n_1 + \frac{n_1}{n_2}(1-\rho_2)n_2.$$

From that, the rate of C is at least $1 - d/n_1 = \rho_1 + \rho_2 - 1$. ◁

We now show that if the partition used in the composition is a c-splitter for a query set of the big code, the resulted code has smoothness roughly equal to the product of the two smoothnesses.

▷ Claim 30. Let C_1 be a code of length n_1 and C_2 a code of length n_2 which is a (q_2, τ_2)-LCC. Let $\mathcal{A} = \{A_1, \ldots, A_{n_1}\}$ be a query-set for C_1 such that for every i, $|A_i| \geqslant 1/\tau_1$ and for every $B \in A_i$, $|B| \leqslant q_1$. If π is a c-splitter for $\mathcal{A}$, then $C = C_1 \odot_\pi C_2$ is a (q, τ)-LCC for $q = q_1 q_2$ and $\tau = c\tau_1\tau_2$.

Proof. To show that C is a (q, τ)-LCC we need to show a corrector Cor for it. We first set up some notations. Let Cor_2 be a corrector promised by the fact that C_2 is a (q_2, τ_2)-LCC. For every $j \in [n]$, let P_j denote the part of π that contains j, and let $\bar{j}$ denote the index of j in P_j with respect to the natural order. For $i \in [n]$, and $B \in A_i$, let $f_{i,B} : \mathbb{F}^{|B|} \to \mathbb{F}$ denote a function satisfying $f_{i,B}(c_B) = c_i$ for every $c \in C_1$. Such $f_{i,B}$ is guaranteed to exists as $\mathcal{A}$ is a query-set for C_1.

For $i \in [n]$, $\mathsf{Cor}(i)$ with oracle access to $c \in C$ acts as follows: it first samples $B \in A_i$ uniformally at random. Secondly, for every $j \in B$, the procedure obtains c_j by invoking $\mathsf{Cor}_2(\bar{j})$ with oracle access to c_{P_j}. After obtaining c_j for every $j \in B$, $\mathsf{Cor}(i)$ outputs $f_{i,B}(c_B)$.

That $\mathsf{Cor}(i)$ successfully outputs c_i for every $c \in C$ is immediate, and follows from the fact that for every j, c_{P_j} is a codeword of C_2 and so $\mathsf{Cor}_2(\bar{j})$ with access to c_{P_j} correctly outputs c_j, and from the fact $c \in C_1$ and so $f_{i,B}(c_B) = c_i$. Moreover, $\mathsf{Cor}(i)$ makes at most $q_1 q_2$ queries to c, since $|B| \leqslant q_1$ by assumption, and Cor_2 makes at most q_2 queries.

It remains to bound the probability that a coordinate $r \in [n]$ is queried by $\mathsf{Cor}(i)$ for $i \in [n]$. Let p be the probability that $\mathsf{Cor}(i)$ queries r. Fix $B \in A_i$. Conditioned on the event that B was sampled by $\mathsf{Cor}(i)$ in the first step, r is queried by $\mathsf{Cor}(i)$ if one of the calls to $\mathsf{Cor}_2(\bar{j})$, with oracle access to c_{P_j}, queries c_r for some $j \in B$. That probability is at most $|B \cap P_r|\tau_2$. Indeed, this follows by taking the union bound over the different $j \in B$, noting that if $j \notin P_r$, c_r cannot be queried by $\mathsf{Cor}_2(\bar{j})$, and using that Cor_2 queries any coordinate with probability bounded above by τ_2. Therefore,

$$\begin{aligned}
p &\leqslant \sum_{B \in A_i} \mathbf{Pr}[B \text{ is sampled by } \mathsf{Cor}(i)] \cdot |B \cap P_r|\tau_2 \\
&= \sum_{B \in A_i} \frac{1}{|A_i|} \cdot |B \cap P_r|\tau_2 \\
&\leqslant \sum_{B \in A_i} \tau_1 \cdot |B \cap P_r|\tau_2 \\
&= \tau_1\tau_2 |P_r \cap \overline{A_i}| \\
&\leqslant c\tau_1\tau_2.
\end{aligned}$$

Note that we used the assumptions that $|A_i| \geqslant 1/\tau_1$, and that π is a c-splitter for $\mathcal{A}$. We thus have that $p \leqslant c\tau_1\tau_2$, which concludes the proof. ◁

The following lemma concludes the properties of the code that is achieved by the composition of two codes, when done with the c-splitter that is given by Claim 27.

▶ **Lemma 31.** *Let $n \in \mathbb{N}$. Assume there exists a code C_1 of length n over $\mathbb{F}$, with rate ρ_1, which is a (q_1, τ_1)-query-set-LCC for $q_1 \geqslant \log n$. Further assume that there exists a code C_2 of length $n\tau_1$ over $\mathbb{F}$, with rate ρ_2, which is a (q_2, τ_2)-LCC. Then, there exists a code C of length n, with rate $\rho_1 + \rho_2 - 1$, which is a $(q_1 q_2, 2eq_1\tau_1\tau_2)$-LCC.*

Proof. As C_1 is a (q_1, τ_1)-query-set-LCC, there exists an n-query-set $\mathcal{A} = \{A_1, \ldots, A_n\}$ in which for every i, $|A_i| \geqslant 1/\tau_1$ and for every $B \in A_i$, $|B| \leqslant q_1$. In particular, there exists a query set $\mathcal{A}' = \{A'_1, \ldots, A'_n\}$ in which every A'_i is of size exactly $1/\tau_1$ (which is achieved by, for each A_i, arbitrarily removing sets $B \in A_i$ until it is of size $1/\tau_1$). By Claim 27 invoked with $k = 1/\tau_1$, there exists a partition π of $[n]$, in which every part is of size $\tau_1 n$, which is a c-splitter for $\mathcal{A}'$, with $c = 2eq_1$. We take $C = C_1 \odot_\pi C_2$ to be the code with the claimed properties. Indeed, by Claim 29, C is of length n, and has rate at least $\rho_1 + \rho_2 - 1$. Furthermore, by applying Claim 30, and using that π is a c-splitter for $\mathcal{A}'$, we get that C is a (q, τ)-LCC for $q = q_1 q_2$ and $\tau = 2eq_1\tau_1\tau_2$, and the lemma follows. ◀

The following lemma, or more precisely, its proof, composes the distance amplification procedure. It assumes a family of codes which are LCC, and describes the properties of the code that is obtained by an iterative application of the composition, where at each iteration a code of the family is composed with the "current" code.

▶ **Lemma 32.** *Assume there exists a family of codes $C = \{C^n\}$ over $\mathbb{F}$, in which every code C^n of length n in the family is a code of rate $\rho(n) = 1 - r(n)$, which is a $(q(n), \tau(n))$-query-set-LCC for $q(n) \geqslant \log n$. Then, for every $t \in \mathbb{N}$, there exists a code family $C' = \{(C')^n\}$ over $\mathbb{F}$ which has a code $(C')^n$ of length n for every n which is a code length in C, and $(C')^n$ has the following properties. Define $n_1 = n$ and for $i = 2, \ldots, t+1$ let $n_i = \left\lceil \tau(n_{i-1}) n_{i-1} \right\rceil^C$. Then, $(C')^n$ has rate $\rho'(n) = 1 - \sum_{i=1}^{t} r(n_i)$, and is a $(q'(n), \tau'(n))$-LCC for $q'(n) = \prod_{i=1}^{t} q(n_i)$ and*

$$\tau'(n) = (2e)^{t-1} \frac{n_{t+1}}{n} \prod_{i=1}^{t-1} q(n_i).$$

Proof. To show the existence of a code family with the claimed properties, we describe how for every n that is a length of a code in the family C, a code of the same length, of the family C', can be constructed. Let C^n be a code of length n of the family C. Set $n_1 = n$ and for $i = 2, \ldots, t+1$, $n_i = \left\lceil \tau(n_{i-1}) n_{i-1} \right\rceil^C$, as defined in the claim. We construct a sequence of codes $C'_1, \ldots, C'_t$, where for each $i \in [t]$, C'_i is a code of length n_i and rate ρ'_i, which is a (q'_i, τ'_i)-LCC. We start by setting $C'_t = C^{n_t}$, and for $i = t-1, \ldots, 1$, we take C'_i to be a code which is the result of applying Lemma 31 on C^{n_i} and C'_{i+1}. Note that C^{n_i} is a $(q(n_i), \tau(n_i))$-query-set-LCC and C'_{i+1} is a code of length $n_{i+1} \geqslant \tau(n_i) n_i$, and so in particular C^{n_i} is indeed of smoothness n_{i+1}/n_i, as required for the lemma to be applicable. From Lemma 31 it follows that C'_i is a code of rate

$$\rho'_i = \rho(n_i) + \rho'_{i+1} - 1 = \rho'_{i+1} - r(n_i)$$

which is a (q'_i, τ'_i)-LCC for

$$q'_i = q(n_i) q'_{i+1},$$
$$\tau'_i = 2eq(n_i) \tau'_{i+1} \frac{n_{i+1}}{n_i}.$$

Recall that $C'_t = C^{n_t}$ and so $\rho'_t = 1 - r(n_t)$, $q'_t = q(n_t)$ and $\tau'_t = \tau(n_t)$. It follows inductively that for every $i \in [t]$,

$$\rho'_i = 1 - \sum_{j=i}^{t} r(n_j),$$

$$q'_i = \prod_{j=i}^{t} q(n_j),$$

and

$$\begin{aligned}\tau'_i &= (2e)^{t-i}\left(\prod_{j=i}^{t}\frac{n_{j+1}}{n_j}\right)\left(\prod_{j=i}^{t-1} q(n_j)\right)\\ &= (2e)^{t-i}\frac{n_{t+1}}{n_i}\left(\prod_{j=i}^{t-1} q(n_j)\right).\end{aligned}$$

We set C'_1, which is indeed a code of length n, to be the code $(C')^n$ of C', and from the account given above it follows that its rate, query complexity and smoothness are as stated, i.e., that $q'_1 = q'(n)$, $\rho'_1 = \rho'(n)$ and $\tau'_1 = \tau'(n)$. We thus have that C' is a family of codes with rate at least $\rho(n)$ that are $(q(n), \tau(n))$-LCC, and the lemma follows. ◀

4.4 Corollaries

In this part we present two corollaries of our distance amplification procedure that is given by Lemma 32. As a special case of the first corollary, Corollary 34, we will have that if one has a sufficiently dense code family of $(q(n), \delta(n), \varepsilon(n))$-LCC which is of high rate, meaning that each code has rate $\rho(n)$ that approaches 1 "fast enough", but with $\delta(n)$ that is only polynomially small in n, $\delta(n) = 1/n^{\alpha}$, for some constant $\alpha \in (0, 1)$, then there exists a good family of LCC with query complexity $q(n)^{O(\log\log n)}$. In the general case, a weaker guarantee on $\delta(n)$ can also be handled by Corollary 34, meaning that a sub-polynomial $\delta(n)$ can also be amplified. More precisely, Corollary 34 will state that if $\delta(n) = 1/n^{1-1/g(n)}$ for a (non-decreasing) function $g(n)$, then a family of good LCC can be constructed, with query complexity $q(n)^{O(g(n)\log\log n)}$. The requirement of the rate function $\rho(n)$, which we described as approaching 1 "fast enough", in more detail comes down to the requirement that $\rho(n) \geqslant 1 - 1/(g(n)(\ln\ln n)^2)$.

The second corollary, Corollary 37, addresses the case that the family of $(q(n), \delta(n), \varepsilon(n))$-LCC one starts with is of a much smaller rate, either of a constant rate or of a vanishing rate of $(1/\ln\ln n)^h$ for some constant h. In the case that $\delta(n) = 1/n^{\alpha}$ for some constant $\alpha \in (0, 1)$ and $\rho(n) \geqslant (1/\ln\ln n)^h$, as a special case Corollary 37 we will have that there exists a family of good LCC with query complexity $q(n)^{\mathrm{poly}(\log\log n)}$. Here too, sub-polynomial $\delta(n)$ can also handled by the corollary, as in a more general case, it is shown by Corollary 37 that if $\delta(n) = 1/n^{1-1/g(n)}$ for a non-decreasing $g(n) \leqslant \log n$, and if $\rho(n)$ is at least $(1/\ln\ln n)^h$ for some constant h, then a family of good LCC can be constructed, with query complexity $q(n)^{g(n)\mathrm{poly}(\log\log n)}$. The precise statement Corollary 37 is more generally stated and handles a few more cases that may be of interest.

We remark that while in any case that Corollary 34 can be applied so can Corollary 37 be used, the reason that we state both corollaries is that if one starts with an LCC that satisfies the requirement of Corollary 34 then using it, rather than using Corollary 37, would result in a better bound on the resulted query complexity. We further remark that the proof for

Corollary 37 builds on Corollary 34. Lastly, another reason that Corollary 34 is of interest is that it has an analogous corollary in the case of LDC (see Corollary 35), unlike Corollary 37 (whose proof relies on properties specific to LCC).

The proofs for the corollaries can be found in the full version of the paper (see [6]).

4.4.1 From high rate and low distance LCC to good LCC

The proof for the first corollary relies on the following lemma which states that any family of (q, τ)-LCC with constant rate can be converted to a family of good LCC by paying a multiplicative factor of $\text{poly}(\tau n)$ in query complexity. This lemma follows from the AEL distance amplification procedure [2, 1] and from the adaptation of it by [17] for LDC and LCC. To derive this lemma with certain parameters, some adaptations to these techniques are needed, and so we provide a full proof for Lemma 33 in the appendix of the full version.

▶ **Lemma 33.** *Let $C = \{C^n\}$ be a code family over $\mathbb{F}$ in which every code C^n is a $(q(n), \tau(n))$-LCC with rate $\rho(n) = \Omega(1)$. Then, there exists a code family $C' = \{(C')^n\}$ over $\mathbb{F}$ which has a code $(C')^n$ of length n for every C^n in C, such that $(C')^n$ is a $(q'(n), \delta'(n), \varepsilon)$-LCC for $q'(n) = O(q(n)(n\tau(n))^2)$, $\delta'(n) = \Omega(1)$ and $\varepsilon = 1/3$, with rate $\rho'(n) = \Omega(1)$.*

We now state our first corollary.

▶ **Corollary 34.** *Let $q(n) \geqslant \log n$[9] and $g(n) > 1$ be two non-decreasing functions. Assume there exists a family of codes $C = \{C^n\}$ over $\mathbb{F}$ that is (n_0, c, d)-dense, in which every code C^n of length n has rate*

$$\rho(n) \geqslant 1 - \frac{1}{g(n)(\ln \ln n)^2},$$

and either C^n is a $(q(n), \delta(n), \varepsilon(n))$-LCC, for $\varepsilon(n) < 1 - 1/|\mathbb{F}|$ and $\delta(n) = 1/n^{1-1/g(n)}$, or it is a $(q(n), \tau(n))$-query-set-LCC, for $\tau(n) = q(n)/n^{1/g(n)}$. Then, there exists a family of codes $C' = \{(C')^n\}$ over $\mathbb{F}$ that is (n_0, c, d)-dense, which is a family of good LCC with query complexity $q_{new}(n) = q(n)^{O(g(n) \ln \ln n)}$.

Note that Corollary 34 allows for the code family C in the hypothesis to be one of two types, either a family of (q, δ, ε)-LCC or a family of (q, τ)-query-set-LCC. For the proof, what we actually need is that C is of the second type. However, if one starts with a family C which is known to be of the first (more standard) type, with the specified $\delta(n)$, by Claim 25 it will follow that C is a family of query-set-LCC with the same smoothness $\tau(n)$ that is stated in the corollary in the second case. The corollary explicitly allows both of the types because it is also possible that the base code is already known to be a query-set-LCC, as would be the case in the proof of Corollary 37, which uses Corollary 34. It is preferable to avoid going back and forth between the types, as this has some cost in the resulted parameters.

We further state a corollary analogous to Corollary 34, that holds in the case of LDC. The proof for this corollary is straightforward given the result regarding LCC, and follows the same lines.

[9] We remark that while we assume for simplicity that $q(n) \geqslant \log n$, by the Katz-Trevisan bound (instantiated for the case of rate and distance as specified by the corollary), lifting this assumption would not yield an improvement in the obtained query complexity in any case.

▶ **Corollary 35.** *Let $n(k) > k$, $q(k) \geqslant \log n(k)$ and $g(k) > 1$ be non-decreasing functions. Assume there exists a code-encoding family $C = \{(C^k, \mathsf{Enc}^k)\}$ over $\mathbb{F}$ that is (k_0, c, d)-dense, in which every code C^k of dimension k has rate*

$$\rho(k) \geqslant 1 - \frac{1}{g(k)(\ln \ln k)^2} > \frac{1}{2},$$

and either (C^k, Enc^k) is a $(q(k), \delta(k), \varepsilon(k))$-LDC, for $\varepsilon(k) < 1 - 1/|\mathbb{F}|$ and $\delta(k) = 1/n(k)^{1-1/g(k)}$. Then, there exists a code-encoding family $C' = \{((C')^k, (\mathsf{Enc}')^k)\}$ over $\mathbb{F}$ that is (k_0, c, d)-dense, which is a family of good LDC with query complexity $q_{new}(k) = q(k)^{O(g(k) \ln \ln k)}$.

4.4.2 From low rate and low distance LCC to good LCC

The proof for our second corollary uses the following proposition from [7]. This proposition is basically Proposition 4.14 in [7] but for (q, τ)-query-set-LCC rather than for a different object[10]. That the proposition indeed applies to (q, τ)-query-set-LCC is quite immediate with the account given in [7].

▶ **Proposition 36** (Implicit in [7])**.** *Let C be a code of length n over $\mathbb{F}$ with rate ρ that is a (q, τ)-query-set-LCC. Then, for every $\ell \in \mathbb{N}$, there exists a code C' of length $n' = n^\ell$ with rate $1 - (1 - \rho)^\ell$, which is a (q', τ)-query-set-LCC for $q' = q^\ell$.*

We now state our second corollary.

▶ **Corollary 37.** *Let $h \geqslant 1$ be an arbitrary constant, $q(n) \geqslant \log n$ and $g(n) \in [1, \log n]$ non-decreasing functions, and $\rho(n)$ a non-increasing function, satisfying*

$$\frac{1}{(\ln \ln n)^h} \leqslant \rho(n) \leqslant 1 - \frac{1}{g(n)(\ln \ln n)^2}$$

for every n. Assume further that

$$\frac{1}{\rho(n+1)}(\ln g(n+1) + \ln \ln \ln(n+1)) - \frac{1}{\rho(n)}(\ln g(n) + \ln \ln \ln n) = O\left(\frac{1}{\log n}\right).$$

Assume there exists a family of codes $C = \{C^n\}$ over $\mathbb{F}$ that is $(n_0, 1, 1)$-dense[11], in which every code C^n of length n is a code of rate $\rho(n)$, which is a $(q(n), \delta(n), \varepsilon(n))$-LCC, for $\varepsilon(n) < 1 - 1/|\mathbb{F}|$ and

$$\delta(n) = \frac{1}{n^{1-1/g(n)}}.$$

Then, there exists a family of codes $C' = \{(C')^n\}$ over $\mathbb{F}$, which is a family of good LCC with query complexity $q_{new}(n) = q(n)^{e(n)}$ for

$$e(n) = O\left(\frac{1}{\rho(n)^2}(\ln g(n) + \ln \ln \ln n)^2 g(n) \ln \ln n\right).$$

[10] "dual SLR" in the terminology of [7].

[11] Note that if one starts with a code family C that is (n_0, c, d) for some constants c, d, then it can be easily converted to a $(n_0, 1, 1)$-dense family, with a constant multiplicative cost to the rate and with little affect to the obtained parameters.

5 LDC are not LCC via random weighted tensor codes

In this section we argue that there exist linear codes which are LDC but not LCC, in the following strong sense. Not only are these codes LDC while not being LCC even for a weak requirement of very high query complexity and very low correction radius, moreover, this negative property that local correction with such parameters is impossible is maintained in any puncturing of the code. We will be able to show this to be the case because in the codes that we construct the uncorrectable coordinates are crucial for the distance of the code, and in particular for the LDC feature of the code, thus any attempt to remove them while keeping these properties, fails. In this section here we state the main claims required for proving the result, the proofs for which can be found in the full version of the paper (see [6]).

We start with a few preliminaries for this section. In what follows we will sometimes need to conveniently convert a pair of indices $i_1 \in [m_1]$, $i_2 \in [m_2]$ to an index $i \in [m_1 m_2]$, and so we set the following convention. Where $m_1, m_2 \in \mathbb{N}$ are clear from context and $i_1 \in [m_1]$, $i_2 \in [m_2]$, we denote by $(i_1; i_2)$ the index $(i_2 - 1)m_1 + i_1 \in [m_1 m_2]$.

▶ **Definition 38** (Trivial coordinates). *For a code C of length n over $\mathbb{F}$, we say that a coordinate $j \in [n]$ is* trivial *(in C) if for every $c \in C$, $c_j = 0$.*

▶ **Definition 39** (Puncturing of codes). *Let C be a code of length n and dimension k over $\mathbb{F}$ and let $J \subseteq [n]$. For every codeword $c \in C$, we define the vector $(y_1, \ldots, y_n) \in \mathbb{F}^n$, where $y_j = c_j$ if $j \notin J$ and $y_j = 0$ otherwise, to be the J-*puncturing of c, *and we denote it by $c_{\setminus J}$. We define $\{c_{\setminus J} \mid c \in C\}$ to be the J-*punctured code C, and denote it by $C_{\setminus J}$. *Note that $C_{\setminus J}$ is indeed a code. Furthermore, given an encoding Enc of C, we define $\mathsf{Enc}_{\setminus J} : \mathbb{F}^k \to \mathbb{F}^n$ by $\mathsf{Enc}_{\setminus J}(x) = \mathsf{Enc}(x)_{\setminus J}$ for all $x \in \mathbb{F}^k$.*

5.1 Weighted tensors

We turn to define an operation to which we call the *weighted tensor* of two codes and state several of its properties. The codes of Theorem 3 will be constructed using a weighted tensor. This operation gets two input codes (more precisely, two codes and respective encodings), and a matrix of non-zero entries, and results in a new code. To define the result of the operation, we will define a new encoding function which depends on the encodings of the two input codes and on the weight matrix. We will then take the resulted code to be the image of that encoding. We begin by describing the encoding function of the *weighted tensor*.

Let $\mathsf{Enc}_1 : \mathbb{F}^{k_1} \to \mathbb{F}^{n_1}$ and $\mathsf{Enc}_2 : \mathbb{F}^{k_2} \to \mathbb{F}^{n_2}$ be a linear maps, and let $B \in \mathbb{F}^{n_1 \times k_2}$ be a matrix with non-zero entries. We define the following function $\mathsf{Enc} : \mathbb{F}^{k_1 k_2} \to \mathbb{F}^{n_1 n_2}$ that acts as follows on input $x \in \mathbb{F}^{k_1 k_2}$.

Action of Enc on x

1. Identify x with a matrix $X \in \mathbb{F}^{k_1 \times k_2}$ where for $i_1 \in [k_1]$, $i_2 \in [k_2]$, $X_{i_1, i_2} = x_{(i_1; i_2)}$.
2. Use Enc_1 to encode each column of X and set X' to be the resulted matrix, $X' \in \mathbb{F}^{n_1 \times k_2}$.
3. For each $j_1 \in [n_1], i_2 \in [k_2]$ multiply the element X'_{j_1, i_2} by B_{j_1, i_2} and set X'' to be the resulted matrix.
4. Use Enc_2 to encode each row of X'' and set X''' to be the resulted matrix, $X''' \in \mathbb{F}^{n_1 \times n_2}$.
5. Output $x' \in \mathbb{F}^{n_1 n_2}$ where for $j_1 \in [n_1], j_2 \in [n_2]$, $x'_{(j_1; j_2)} = X'''_{j_1, j_2}$.

▷ Claim 40. If Enc_1 and Enc_2 are injective then so is Enc.

▷ Claim 41. Let $A^1 \in \mathbb{F}^{n_1 \times k_1}$ and $A^2 \in \mathbb{F}^{n_2 \times k_2}$ be the generating matrices of Enc_1 and Enc_2, respectively. Then, for every $x \in \mathbb{F}^{k_1 k_2}$, $\mathsf{Enc}(x) = Ax$, where $A \in \mathbb{F}^{n_1 n_2 \times k_1 k_2}$ is the matrix where for $i_1 \in [k_1], i_2 \in [k_2], j_1 \in [n_1], j_2 \in [n_2]$ we have that $A_{(j_1;j_2),(i_1;i_2)} = A^1_{j_1,i_1} A^2_{j_2,i_2} B_{j_1,i_2}$. In particular, Enc is a linear map.

We can now define the weighted tensor operation.

▶ **Definition 42.** *Let* Enc_1, Enc_2, B *and* Enc *be as above. Let* C_1 *be a code of length* n_1 *and dimension* k_1 *over* $\mathbb{F}$ *such that* Enc_1 *is an encoding of it, and let* C_2 *be a code of length* n_2 *and dimension* k_2 *over* $\mathbb{F}$ *such that* Enc_2 *is an encoding of it. Let* C *be the image of* Enc*. We define the* B-weighted tensor of (C_1, Enc_1) and (C_2, Enc_2) *to be the pair* (C, Enc)*, and denote* $(C, \mathsf{Enc}) = (C_1, \mathsf{Enc}_1) \otimes_B (C_2, \mathsf{Enc}_2)$.

▷ Claim 43. Let $(C, \mathsf{Enc}) = (C_1, \mathsf{Enc}_1) \otimes_B (C_2, \mathsf{Enc}_2)$. Then C is a code of length $n = n_1 n_2$ and dimension $k = k_1 k_2$ over $\mathbb{F}$, and Enc is an encoding of it.

5.2 Local decodability and correctablity of weighted tensors

The weighted tensor of two LDC is an LDC with comparable parameters, regardless of the weight matrix, as we have in the following claim.

▷ Claim 44. Let (C_1, Enc_1) be a $(q_1, \delta_1, \varepsilon_1)$-LDC, where C_1 is a code of length n_1 and dimension k_1 over $\mathbb{F}$. Let (C_2, Enc_2) be a $(q_2, \delta_2, \varepsilon_2)$-LDC, where C_2 is a code of length n_2 and dimension k_2 over $\mathbb{F}$, and let $B \in \mathbb{F}^{n_1 \times k_2}$ be a matrix with no zero entries. Then, $(C, \mathsf{Enc}) = (C_1, \mathsf{Enc}_1) \otimes_B (C_2, \mathsf{Enc}_2)$ is a $(q_1 q_2, \delta_1 \delta_2, 1 - (1 - \varepsilon_1)(1 - \varepsilon_2)^{q_1})$-LDC.

In the next claim we argue that the weighted tensor of two codes, when performed with a randomly chosen weight matrix is, with high probability, not locally correctable. In particular, there exists a subset of the coordinates which cannot be locally corrected even with a small correction radius guarantee, and cannot be removed from the code either if its decodablity is to be preserved.

▷ Claim 45. Let (C_1, Enc_1) be a $(q_1, \delta_1, \varepsilon_1)$-LDC of length n_1 and dimension k_1 over $\mathbb{F}$, and let (C_2, Enc_2) be a $(q_2, \delta_2, \varepsilon_2)$-LDC of length n_2 and dimension k_2 over $\mathbb{F}$. Assume that C_1 and C_2 have no non-trivial coordinates. Let $B \in \mathbb{F}^{n_1 \times k_2}$ be a random matrix of non-zero weights, chosen uniformly and independently, and let $(C, \mathsf{Enc}) = (C_1, \mathsf{Enc}_1) \otimes_B (C_2, \mathsf{Enc}_2)$. For every $t < k_2$ and $\tilde{q}, \tilde{\tilde{q}} \in \mathbb{N}$, $\delta \geqslant \tilde{q}/n_1$, $\delta' \geqslant t/k_2$ and $\varepsilon < 1 - 1/|\mathbb{F}|$, with probability at least $1 - n_1 n_2 \binom{n_1 n_2}{\tilde{q}} |\mathbb{F}|^{\tilde{q}} / (|\mathbb{F}| - 1)^t$ over the choice of B, C satisfies the following. There exists a set $\bar{J} \subseteq [n]$ such that every $j \in \bar{J}$ is not $(\tilde{q}, \delta, \varepsilon)$-locally correctable in C. Further, the relative (non-local) distance of $C_{\setminus \bar{J}}$ is less than t/k_2.

The main theorem of this part is an immediate consequence of Claim 45.

▶ **Theorem 46.** *Let* (C_0, Enc_0) *be a* $(q_0, \delta_0, \varepsilon_0)$*-LDC for a code* C_0 *of dimension* k_0 *and length* n_0 *over* $\mathbb{F}$ *for* $|\mathbb{F}| > 2$*, such that* $\varepsilon_0 < 1 - 1/|\mathbb{F}|$*,* $k_0^{1/2} > 10 \log n_0$*, and assume that* C_0 *has no trivial coordinates. Then, there exists a* $(q_0^2, \delta_0^2, 1 - (1 - \varepsilon_0)^{q_0 + 1})$*-LDC* (C, Enc) *for a code* C *of dimension* $k = k_0^2$ *and length* $n = n_0^2$ *over* $\mathbb{F}$ *satisfying the following property. There exists a set* $J \subseteq [n]$ *of coordinates such that every* $j \in J$*,* j *is not* $(k^{1/4}, k^{1/4}/n^{1/2}, \varepsilon)$*-locally correctable in* C*, for any* $\varepsilon < 1 - 1/|\mathbb{F}|$*. Moreover, the relative distance of* $C_{\setminus J}$ *is less than* $5 \log(n)/k^{1/4}$ *(in particular for any* $\tilde{\tilde{q}} \in \mathbb{N}$ *and* $\varepsilon < 1 - 1/|\mathbb{F}|$*,* $C_{\setminus J}$ *is not a* $(\tilde{\tilde{q}}, 5 \log(n)/k^{1/4}, \varepsilon)$*-LDC).*

References

1 Noga Alon, Jeff Edmonds, and Michael Luby. Linear time erasure codes with nearly optimal recovery. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 512–519. IEEE, 1995.

2 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.

3 Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 519–528, 2011.

4 Arnab Bhattacharyya, Zeev Dvir, Amir Shpilka, and Shubhangi Saraf. Tight lower bounds for 2-query lccs over finite fields. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, pages 638–647. IEEE, 2011.

5 Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query lccs over large alphabet. *arXiv preprint*, 2016. `arXiv:1611.06980`.

6 Gil Cohen and Tal Yankovitz. Lcc and ldc: Tailor-made distance amplification and a refined separation. *Electronic Colloquium on Computational Complexity (ECCC)*, 136, 2021.

7 Gil Cohen and Tal Yankovitz. Rate amplification and query-efficient distance amplification for linear lcc and ldc. In *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

8 Irit Dinur. The PCP theorem by gap amplification. In *Proc. 38th ACM Symp. on Theory of Computing*, pages 241–250, 2006.

9 Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM Journal on Computing*, 40(4):1154–1178, 2011.

10 Zeev Dvir and Kalina Petrova. Lecture 1: Introduction. Lecture notes: `https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC1.pdf`, 2016.

11 Klim Efremenko. 3-query locally decodable codes of subexponential length. *SIAM Journal on Computing*, 41(6):1694–1703, 2012.

12 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.

13 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*, pages 529–540. ACM, 2013.

14 Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Information and Computation*, 243:178–190, 2015.

15 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*, pages 80–86, 2000.

16 Tali Kaufman and Michael Viderman. Locally testable vs. locally decodable codes. In *Approximation, randomization, and combinatorial optimization*, volume 6302 of *Lecture Notes in Comput. Sci.*, pages 670–682. Springer, Berlin, 2010. `doi:10.1007/978-3-642-15369-3_50`.

17 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 (JACM)*, 64(2):11, 2017.

18 Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *Journal of the ACM (JACM)*, 61(5):28, 2014.

19 A. Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 238–251. ACM, 2017.

20 Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008.