Lower Bounds for 2-Query LCCs over Large Alphabet*

Arnab Bhattacharyya[1], Sivakanth Gopi[2], and Avishay Tal[3]

1 **Department of Computer Science and Automation, Indian Institute of Science Bangalore, Bangalore, India**
arnabb@csa.iisc.ernet.in
2 **Department of Computer Science, Princeton University, Princeton, NJ, USA**
sgopi@cs.princeton.edu
3 **School of Mathematics, Institute for Advanced Study, Princeton, NJ, USA**
avishay.tal@gmail.com

Abstract

A locally correctable code (LCC) is an error correcting code that allows correction of any arbitrary coordinate of a corrupted codeword by querying only a few coordinates. We show that any 2-query locally correctable code $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ that can correct a constant fraction of corrupted symbols must have $n \geqslant \exp(k/\log|\Sigma|)$ under the assumption that the LCC is *zero-error*. We say that an LCC is zero-error if there exists a non-adaptive corrector algorithm that succeeds with probability 1 when the input is an uncorrupted codeword. All known constructions of LCCs are zero-error.

Our result is tight upto constant factors in the exponent. The only previous lower bound on the length of 2-query LCCs over large alphabet was $\Omega((k/\log|\Sigma|)^2)$ due to Katz and Trevisan (STOC 2000). Our bound implies that zero-error LCCs cannot yield 2-server private information retrieval (PIR) schemes with sub-polynomial communication. Since there exists a 2-server PIR scheme with sub-polynomial communication (STOC 2015) based on a zero-error 2-query locally decodable code (LDC), we also obtain a separation between LDCs and LCCs over large alphabet.

1998 ACM Subject Classification E.4 Coding and Information Theory

Keywords and phrases Locally correctable code, Private information retrieval, Szemerédi regularity lemma

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2017.30

1 Introduction

In this work, we study error-correcting codes that are equipped with local algorithms. A code is called a locally correctable code (LCC) if there is a randomized algorithm which, given an index i and a received word w close to a codeword c in Hamming distance, outputs c_i by querying only a few positions of w. The maximum number of positions of w queried by the local correction algorithm is called the query complexity of the LCC.

The main problem studied regarding LCCs is the tradeoff between their query complexity and length. Intuitively, these two parameters enforce contrasting properties. Small query

* AB was partially supported by a DST Ramanujan Fellowship. SG was supported by NSF grants CCF-1523816, CCF-1217416 and part of this research was done while the author was at Microsoft Research, Redmond. AT was supported by the Simons Collaboration on Algorithms and Geometry, and by the National Science Foundation grant No. CCF-1412958.

complexity means that individual codeword symbols carry substantial information, while short length along with resilience to corruption means that information is spread out among the codeword symbols. In this paper, we explore one end of the spectrum of tradeoffs by studying 2-query locally correctable codes.

Also called "self-correction", the idea of local correction originated in works by Lipton [22] and by Blum and Kannan [7] on program checkers. In particular, [22, 3] used the fact that the Reed-Muller code is locally correctable to show average-case hardness of the Permanent problem. LCCs are closely related to locally decodable codes (LDCs), where the goal is to recover a symbol of the underlying message when given a corrupted codeword using a small number of queries [18]. LDCs are weaker than LCCs, in the sense that any LCC can be converted into an LDC while preserving relevant parameters (see Appendix A for a formal statement and proof). LDCs and LCCs have found applications in derandomization and hardness results [25, 15, 19]. See [29] for a detailed survey on LDCs and LCCs, as of 2010. In more recent years, the analysis of LDCs and LCCs has led to a greater understanding of basic problems in incidence geometry, the construction of design matrices and the theory of matrix scaling, e.g. [2, 14, 13].

One particularly important feature of LDCs is their tight connection to *information-theoretic private information retrieval (PIR)* schemes. PIR is motivated by the scenario where a user wants to retrieve an item from a database without revealing to the database owner what item he is asking for. Formally, the user wants to retrieve x_i from a k-bit database $\mathbf{x} = (x_1, \ldots, x_k)$. A trivial solution is for the database owner to transmit the entire database no matter what query the user has in mind, but this has a huge communication overhead. Chor et al. [8] observed that while with one database, nothing better than the trivial solution is possible, there are non-trivial PIR schemes if multiple servers can hold replicas of the database. It turns out that t-server PIR schemes with low communication are roughly equivalent to short t-query LDCs. More precisely, a 2-server PIR scheme for k bits of data with s bits of communication translates to a 2-query LDC $\mathcal{C} : \{0,1\}^k \to \Sigma^{2^s}$ where $\Sigma = \{0,1\}^s$. Note that in this translation, $|\Sigma|$ equals the length of the code.

Let $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ be a 2-query LDC/LCC such that the corrector algorithm can tolerate corruptions at δn positions. Katz and Trevisan in their seminal work [18] showed that for 2-query LDCs, $n \geqslant \Omega(\delta(k/\log|\Sigma|)^2)$. (Since LDCs are weaker than LCCs, a lower bound on the length of LDCs also implies a lower bound on the length of LCCs). More than 15 years later, the Katz-Trevisan bound is still the best known for large alphabet Σ. However for small alphabet size, the dependence on k is shown to be exponential. Goldreich et al. [16] showed that $n \geqslant \exp(\delta k/|\Sigma|)$ for linear 2-query LDCs, while Kerenedis and de Wolf [20] (with further improvements in [28]) showed using quantum techniques that $n \geqslant \exp(\delta k/|\Sigma|^2)$ for arbitrary 2-query LDCs. But these lower bounds become trivial when $|\Sigma| = \Omega(n)$. However, the case of large alphabet $|\Sigma| \approx n$ is quite important to understand as this is the regime through which we would be able to prove lower bounds on the communication complexity of PIR schemes.

Given the lack of progress on LDC and PIR lower bounds, it is a natural question to ask whether strong lower bounds are possible for LCCs. In this work, we demonstrate an exponential improvement on the Katz-Trevisan bound for *zero-error LCCs*. We define a zero-error LCC to be an LCC for which the corrector algorithm is non-adaptive and succeeds with probability 1 when the input is an uncorrupted codeword. All current LCC constructions are zero-error, and in fact, any linear LCC can be made zero-error. We state our main theorem below informally, see Theorem 5 for a formal statement.

▶ **Theorem 1** (Informal). *If $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ is a zero-error 2-query LCC that can correct δn corruptions, then $n \geqslant \exp(\mathrm{poly}(\delta) \cdot k/\log|\Sigma|)$.*[1]

1.1 Discussion of Main Result

The lower bound in Theorem 1 is tight in its dependence on k and Σ. Specifically, Yekhanin in the appendix of [4] gives the following elegant construction of a 2-query LCC $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ with $n = 2^{O(k/\log|\Sigma|)}$ for any $\delta \leqslant 1/6, \Sigma$ and k. Assume $|\Sigma| = 2^b$ and $b \mid k$ for simplicity. Write $\mathbf{x} \in \{0,1\}^k$ as $(x_{i,j})_{i\in[b],j\in[k/b]}$. Then, for any $a \in [2^{k/b}]$, let $(\mathcal{C}(\mathbf{x}))_a = (\mathcal{H}(x_{i,1},\ldots,x_{i,k/b})_a : i \in [b]) \in \{0,1\}^b$ where $\mathcal{H}$ is the classical Hadamard encoding $\mathcal{H} : \{0,1\}^r \to \{0,1\}^{2^r}$ defined as $\mathcal{H}(\mathbf{y}) = (\sum_{i=1}^r y_i\xi_i \pmod 2 : \xi_1,\ldots,\xi_r \in \{0,1\})$. It is well-known that $\mathcal{H}$ is a 2-query LCC, and from this, it is easy to check that $\mathcal{C}$ is also. The parameters follow directly from the construction. A simple modification of this construction gives $(2^{O(\delta k/\log|\Sigma|)}/\delta)$-length 2-query LCCs that tolerate δn corruptions. The proof of Theorem 1 shows $n \geqslant \exp(\delta^4 k/\log|\Sigma|)$ which is therefore tight upto $\mathrm{poly}(\delta)$ factors in the exponent.

The 2-query LCC described above is a linear code over $\mathbb{F}_{2^b}$. For linear codes $\mathcal{C} \subseteq \mathbb{F}_q^n$ (i.e., $\mathcal{C}$ is a linear subspace of $\mathbb{F}_q^n$), where $q = p^r$ for a prime p, [4] showed that $n \geqslant \exp(\delta k/r) = \exp(\delta k/\log_p|\Sigma|)$ where $k = \log|\mathcal{C}|$ is the message length and $|\Sigma| = p^r$. Thus, in terms of dependence on k and $|\Sigma|$, we extend the result of [4] from linear codes to all zero-error LCCs. Moreover, this work is much more elementary and simple than [4] which uses non-trivial results from additive combinatorics.

It is important to note that Theorem 1 cannot be true for 2-query LDCs. Such a result would contradict the construction in [12] of a zero-error 2-query LDC with $\log n = \log|\Sigma| = \exp(\sqrt{\log k}) = k^{o(1)}$ and $\delta = \Omega(1)$. So, our result can be interpreted as giving a separation between zero-error LCCs and LDCs over large alphabet. We conjecture that the zero-error restriction in the theorem can be removed, which if true, would yield the first separation between general LCCs and LDCs. It is still quite unclear what the correct lower bound for 2-query LDCs should look like. As mentioned above, Katz and Trevisan [18] show that $n \geqslant \Omega(\delta k^2/\log^2|\Sigma|)$. And the quantum arguments of [20, 28] give the lower bound $n \geqslant \exp(\delta k/|\Sigma|^2)$ which becomes trivial when $|\Sigma| = \Omega(n)$.

1.2 Proof Overview

Like most prior work on 2-query LDCs and LCCs, we view the query distribution of the local correcting algorithm as a graph. However, these previous works did not exploit the structure of the graph much beyond its size and degree, whereas our bound is due to a detailed use of the graph structure.

Let $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ be a 2-query LCC. So, for every $i \in [n]$, there is a corrector algorithm $\mathcal{A}_i$ that when given access to $z \in \Sigma^n$ with Hamming distance at most δn from some codeword y, returns y_i with probability at least $2/3$. Assuming non-adaptivity, the algorithm $\mathcal{A}_i$ chooses its queries from a distribution on $[n]^2$. Katz and Trevisan [18] show how to extract a matching M_i of $\Omega(\delta n)$ disjoint edges on n vertices such that for any edge $e = (j,k)$ in M_i,

$$\Pr_y[\mathcal{A}_i(y) = y_i \mid \mathcal{A} \text{ queries } y \text{ at positions } j \text{ and } k] > \frac{1}{2} + \varepsilon$$

[1] An earlier version [5] of this paper showed that $n \geqslant \exp(c_\delta \cdot k/\log|\Sigma|)$ where c_δ has tower type dependence on δ due to the use of the Szemerédi regularity lemma.

for some constant $\varepsilon > 0$, where the probability is over a uniformly random codeword $y \in \mathcal{C}$. For zero-error LCCs, the situation is simpler in that essentially, for *every* codeword y and edge $e \in M_i$, $\mathcal{A}_i(y)$ returns y_i when it queries the elements of e. This is not exactly correct but let us suppose it's true for the rest of this section.

Let G be the union of $M_1, \ldots, M_n$. So, for every edge (j, k) in G, there is an i such that $(j, k) \in M_i$. Suppose our goal is to guess an unknown codeword c given the values of a small subset of coordinates of c. We assign labels in Σ to vertices of G corresponding to the subset of coordinates of c that we know already. Now, imagine a propagation process where we deduce the labels of unlabeled vertices by using the corrector algorithms. For example, if $(j, k) \in M_i$, j and k are labeled but i is not, we can use $\mathcal{A}_i$ to deduce the label at vertex i. Similarly, if $(x, y) \in M_u$ and $(u, v) \in M_w$, and x, y, v are labeled but u and w are not, we can run $\mathcal{A}_u$ to deduce the label of u and then $\mathcal{A}_w$ to deduce the label of w. The set of labels we infer will be the values of c at the corresponding coordinates. The goal of our analysis is to show that there is a set S of $O_\delta(\log n)$[2] vertices such that if the labels of S are known, then the propagation process can determine the labels of all n vertices. This immediately implies that the total number of codewords, 2^k, is at most $|\Sigma|^{|S|}$ and therefore, $k = O_\delta(\log n \cdot \log |\Sigma|)$. Instead, Katz and Trevisan [18] show that if you know the labels of $\sqrt{n}$ uniformly random coordinates, then you can recover the labels of most of the coordinates which leads to the bound $k = O_\delta(\sqrt{n} \cdot \log |\Sigma|)$. Intuitively, their lower bound is just one step of the propagation process.

The propagation process is perhaps more naturally described on a (directed) 3-uniform hypergraph where there is an edge (i, j, k) if $(j, k) \in M_i$. It "captures" i if (i, j, k) is an edge and j, k are already captured. Coja-Oghlan et al. [9] study exactly this process on random undirected 3-uniform hypergraphs in the context of constraint satisfaction problem solvers. Unfortunately, their techniques are specialized to random hypergraphs. The propagation process is also related to hypergraph peeling [23, 24], but again, most theoretical work is limited to random hypergraphs.

To motivate our approach, suppose $M_1, \ldots, M_n$ are each a perfect matching. For a set $S \subseteq [n]$, let $R(S)$ denote the set of vertices to which we can propagate starting from S. If $R(S) = [n]$, we are done. Otherwise, we show that we can double $|R(S)|$ by adding one more vertex to S. Note that for any $i \notin R(S)$, no edge in M_i can lie entirely inside $R(S)$, for then, i would also have been reached. So, each vertex in $R(S)$ must be incident to one edge in M_i for every $i \notin R(S)$. This makes the total number of edges between $R(S)$ and $[n] \setminus R(S)$ belonging to M_i for some $i \notin R(S)$ equal to $|R(S)| \cdot (n - |R(S)|)$. By averaging, there must be $j \notin R(S)$ that is incident to at least $|R(S)|$ edges, each belonging to some M_i for $i \notin R(S)$. Moreover, all these $|R(S)|$ edges must belong to matchings of different vertices. Hence, adding j to S doubles the size of $R(S)$. Hence, for some S of size $O(\log n)$, $R(S) = [n]$.

In the above special case (where all the matchings were perfect), we used the fact that the size of the cut between $R(S)$ and the rest of the graph is large and that many of these edges belong to M_i for $i \notin R(S)$. We observe that for any graph obtained from an LCC as above, this situation exists whenever $R(S)$ is not too large already and the minimum degree of every vertex in the graph is large (say, $\text{poly}(\delta) \cdot n$). This is because each vertex in $R(S)$ will be incident to many edges in matchings M_i for $i \notin R(S)$ (using the minimum degree requirement and that $|R(S)|$ is small) and such edges cannot have both endpoints inside $R(S)$ (as then $i \in R(S)$). So, indeed, there will be many edges with labels not in $R(S)$

[2] $O_\delta(\cdot)$ means that the involved constant can depend on δ.

crossing the cut, and averaging will yield a vertex whose addition to S will make $R(S)$ grow by a multiplicative factor. Therefore, if the minimum degree requirement is met, we can keep repeating this process until $R(S)$ becomes large, of size $\text{poly}(\delta) \cdot n$. Now, in a key lemma of our proof, we show that for any graph obtained from an LCC as above, we can greedily find a subset of the vertices V' such that the the subgraph induced by the vertices of V' and the edges labeled by V' has large minimum degree. So, we can repeatedly apply the above argument to V' to find a subset S of size $O_\delta(\log n)$ such that $R(S)$ contains $\text{poly}(\delta) \cdot n$ vertices.

Recall that our goal is to find a small set S such that $R(S) = [n]$. So, at this stage, we would ideally like to continue the argument on $V'' = [n] \setminus R(S)$. The only issue we can face is that the graph on V'' restricted to edges labeled by V'' may not have the LCC structure. Indeed, it could be that most edges labeled by V'' are not spanned by vertices in V''. However in this case, there will be a vertex u in V'' incident to many V''-labeled edges that have their other endpoints in $R(S)$, so that we can increase $R(S)$ by adding u to S. Thus, either $R(S)$ may be grown directly or else the rest of the vertices looks approximately like an LCC, so that we can recurse. Modulo some important technical details, our proof is now complete[3].

The zero-error assumption seems necessary to make the propagation process well-defined. Otherwise, for each labeled vertex, there is some probability that the label is incorrect for the codeword in question. But since there may be $\Omega(\log n) = \omega(1)$ steps of propagation, the error probability may blow up by this factor. So, it seems we need different techniques to handle correctors that have constant probability of error when the input is a codeword. One possibility is using information theory to better handle the spread of error[4].

2 Zero-error 2-query LCCs

We begin by formally defining zero-error 2-query LCCs.

▶ **Definition 2.** Let Σ be some finite alphabet and let $\mathcal{C} \subset \Sigma^n$ be a set of codewords. $\mathcal{C}$ is called a $(2, \tau)$-LCC with zero-error if there exists a randomized algorithm $\mathcal{A}$ such that following is true:

1. $\mathcal{A}$ is given oracle access to some $z \in \Sigma^n$ and an input $i \in [n]$. It outputs a symbol in Σ after making at most 2 non-adaptive queries to z.
2. If $z \in \Sigma^n$ is τ-close to some codeword $c \in \mathcal{C}$ in Hamming distance, then for every $i \in [n]$, $\mathbf{Pr}[\mathcal{A}^z(i) = c_i] \geqslant 2/3$.
3. If $c \in \mathcal{C}$, then for every $i \in [n]$, $\mathbf{Pr}[\mathcal{A}^c(i) = c_i] = 1$ i.e. if the received word has no errors, then the local correction algorithm will not make any error.

Note that the above definition differs from the standard notion of non-adaptive 2-query LCCs only in part (3) above. The choice of 2/3 in part (2) of the definition above is somewhat arbitrary. We can make it any constant greater than 1/2. More generally, it is only required

[3] An earlier version [5] of this paper had a different argument for the main theorem, based on a "decomposition theorem" proved using the Szeméredi regularity lemma for directed graphs [26, 1]. The idea was to partition the graph into a constant number of edge expanders. In each such part, the sizes of cuts are large and so the propagation process can be easily analyzed. The proof given here is simpler and yields much better dependence on δ. However, because the decomposition theorem for directed graphs may be of general interest, we have included it in Appendix B of this paper.

[4] This approach is taken in [17] to prove an exponential lower bound for smooth 2-query LDCs over binary alphabet when the decoder has subconstant error probability. Jain's analysis seems to work only for binary codes but is similar in spirit to ours.

that for every $\sigma \neq c_i, \mathbf{Pr}[\mathcal{A}^z(i) = c_i] > \mathbf{Pr}[\mathcal{A}^z(i) = \sigma] + \varepsilon$ for some $\varepsilon > 0$, i.e., c_i should win the plurality vote among all symbols by a constant margin.

We next show that the corrector for any zero-error LCC can be brought into a "normal" form. A similar statement is known for general LDCs and LCCs [18, 29] but we need to be a bit more careful because we want to preserve the zero-error property. Note that the proof overview in Section 1.2 assumed that the set T_1 below is empty.

▶ **Lemma 3.** *Let $\mathcal{C} \subset \Sigma^n$ be a $(2, \tau)$-LCC with zero error. Then, there exists a partition of $[n] = T_1 \cup T_2$ such that:*

1. *For every $i \in T_1$, there exists a distribution $\mathcal{D}_i$ over $[n] \cup \{\phi\}$ and algorithms $\mathcal{R}^i_j$ for every $j \in [n] \cup \{\phi\}$ such that for every codeword $c \in \mathcal{C}$,*

$$\mathop{\mathbf{Pr}}_{j \sim \mathcal{D}_i} \left[\mathcal{R}^i_j(c_j) = c_i\right] \geqslant \frac{2}{3}.^5$$

   Moreover the distribution $\mathcal{D}_i$ is smooth over $[n]$ i.e. for every $j \in [n]$, $\mathbf{Pr}_{\mathcal{D}_i}[j] \leqslant \frac{4}{\tau n}$.
2. *For every $i \in T_2$, there exists a matching $\mathcal{M}_i$ of edges in $[n] \setminus \{i\}$ of size $|\mathcal{M}_i| \geqslant \frac{\tau}{4} n$ such that: For every $c \in \mathcal{C}$, c_i can be recovered from (c_j, c_k) for any $(j, k) \in \mathcal{M}_i$ i.e. there exists algorithms $\mathcal{R}^i_{j,k}$ for every edge $(j, k) \in \mathcal{M}_i$ such that for every $c \in \mathcal{C}$,*

$$\mathcal{R}^i_{j,k}(c_j, c_k) = c_i.$$

Proof. Fix $\varepsilon = \tau/4$. Let $\mathcal{A}$ be the local corrector algorithm for $\mathcal{C}$ and let $\mathcal{Q}_i$ be the distribution over 2-tuples of $[n]$ corresponding to the queries $\mathcal{A}(i)$ makes to correct coordinate i.[6] Let $\mathrm{supp}(\mathcal{Q}_i)$ be the set of edges in the support of $\mathcal{Q}_i$. We have two cases:

Case 1: $\mathrm{supp}(\mathcal{Q}_i)$ contains a matching of size εn.

In this case, we include $i \in T_2$ and define $\mathcal{M}_i$ to be a matching of size εn in $\mathrm{supp}(\mathcal{Q}_i)$. Let $\mathcal{R}^i_{j,k}(z_j, z_k)$ be the output[7] of $\mathcal{A}^z(i)$ when it samples (j, k) from the distribution $\mathcal{Q}_i$. So we have for every $\sigma \in \Sigma$,

$$\mathop{\mathbf{Pr}}_{(j,k) \sim \mathcal{Q}_i} [\mathcal{R}^i_{j,k}(z_j, z_k) = \sigma] = \mathbf{Pr}[\mathcal{A}^z(i) = \sigma].$$

Now since our LCC is zero-error, for every $(j, k) \in \mathrm{supp}(\mathcal{Q}_i)$, we have $\mathcal{R}^i_{j,k}(c_j, c_k) = c_i$. This takes care of part (2).

Case 2: $\mathrm{supp}(\mathcal{Q}_i)$ doesn't contain a matching of size εn.

In this case we include $i \in T_1$. Since $\mathrm{supp}(\mathcal{Q}_i)$ doesn't contain a matching of size εn, there exists a vertex cover of size at most $2\varepsilon n$, say V_i. Also define $B_i \subset [n]$ to be the set of vertices which are queried with high probability by $\mathcal{A}^z(i)$ i.e.

$$B_i = \left\{ j : \mathbf{Pr}[\mathcal{A}^z(i) \text{ queries } j] \geqslant \frac{1}{\varepsilon n} \right\}.$$

Clearly $|B_i| \leqslant 2\varepsilon n$ because $\mathcal{A}^z(i)$ makes at most two queries. We now define a new one-query corrector for i, $\tilde{\mathcal{A}}^z(i)$ as follows: simulate $\mathcal{A}^z(i)$, but whenever $\mathcal{A}^z(i)$ queries z at a coordinate in $V_i \cup B_i$, $\tilde{\mathcal{A}}^z(i)$ doesn't query that coordinate and assumes that the queried coordinate is 0 (or some fixed symbol in Σ). Note that $\tilde{\mathcal{A}}^z(i)$ makes at most one query to z since V_i is a vertex cover for the support of $\mathcal{Q}_i$. Also $\tilde{\mathcal{A}}^c(i)$ behaves exactly

[6] Wlog, we can assume $\mathcal{A}(i)$ always queries two coordinates.

[7] Note that $\mathcal{R}^i_{j,k}$ might use additional randomness.

like $\mathcal{A}^{c'}(i)$ where c' is the word formed by zeroing out the $V_i \cup B_i$ coordinates of c. Since $|V_i \cup B_i| \leqslant 4\varepsilon n \leqslant \tau n$, we have

$$\mathbf{Pr}[\tilde{\mathcal{A}}^c(i) = c_i] = \mathbf{Pr}[\mathcal{A}^{c'}(i) = c_i] \geqslant \frac{2}{3}.$$

Now define the distribution $\mathcal{D}_i$ over $[n] \cup \{\phi\}$ as:

$$\mathop{\mathbf{Pr}}_{\mathcal{D}_i}[j] = \mathbf{Pr}[\tilde{\mathcal{A}}^z(i) \text{ queries } j]$$

for $j \in [n]$ and

$$\mathop{\mathbf{Pr}}_{\mathcal{D}_i}[\phi] = \mathbf{Pr}[\tilde{\mathcal{A}}^z(i) \text{ doesn't make any query}].$$

Since we never query elements of B_i, we have the required smoothness i.e. $\mathbf{Pr}_{\mathcal{D}_i}[j] \leqslant 1/(\varepsilon n)$ for all $j \in [n]$. Also define $\mathcal{R}^i_j(z_j)$ to be the output (can be randomized) of $\tilde{\mathcal{A}}^z(i)$ when it queries $j \in [n]$ and $\mathcal{R}^i_\phi(c_\phi)$ to be the output (can be randomized) of $\tilde{\mathcal{A}}^z(i)$ when it doesn't make any query where c_ϕ is an empty input defined for ease of notation. By definition, we have

$$\mathop{\mathbf{Pr}}_{j \sim \mathcal{D}_i}[\mathcal{R}^i_j(c_j) = c_i] = \mathbf{Pr}[\tilde{\mathcal{A}}^c(i) = c_i] \geqslant \frac{2}{3}.$$

This proves part (1). ◀

3 Proof of lower bound

3.1 An information theoretic lemma

The proof of Theorem 1 works by showing that there is randomized algorithm which can guess an unknown codeword $c \in \mathcal{C} \subset \Sigma^n$ with high probability by making a small number of queries. From this we would like to show that $|\mathcal{C}|$ cannot be large. We will apply Fano's inequality which is a basic information theoretic inequality to achieve this. We will assume familiarity with basic notions in information theory; we refer the reader to [10] for precise definitions and the proofs of the facts we use. Given random variables X, Y, Z, let $H(X)$ be the entropy of X which is the amount of information contained in X. $H(X|Y)$ is the conditional entropy of X given Y which is the amount of information left in X if we know Y. The mutual information $I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$ is the amount of common information between X, Y. If X, Y are independent, then $I(X;Y) = 0$. The conditional mutual information $I(X;Y|Z)$ is the mutual information between X, Y given Z. We have the following chain rule for mutual information:

$$I(X;YZ) = I(X;Z) + I(X;Y|Z).$$

We also need the following basic inequality:

$$I(X;Y|Z) \leqslant H(X|Z) \leqslant \log|\mathcal{X}|$$

where $\mathcal{X}$ is the support of the random variable X. We will now state Fano's inequality which says that if we can predict X very well from Y i.e. there is a predictor $\hat{X}(Y)$ such that $\mathbf{Pr}[\hat{X}(Y) \neq X] \leqslant p_e$ where p_e is small, then $H(X|Y)$ should be small as well (see [10] for a proof). More precisely,

$$H(X|Y) \leqslant h(p_e) + p_e \log(|\mathcal{X}| - 1) \qquad \text{(Fano's inequality)}$$

where $h(x) = -x\log x - (1-x)\log(1-x)$ is the binary entropy function and $\mathcal{X}$ is the support of random variable X.

▶ **Lemma 4.** *Suppose there exists a randomized algorithm $\mathcal{P}$ such that for every $c \in \mathcal{C} \subset \Sigma^n$, given oracle access to c, $\mathcal{P}$ makes at most t queries to c and outputs c with probability $\geqslant 1/2$, then* $\log|\mathcal{C}| \leqslant O(t \log|\Sigma|)$.

Proof. Let X be a random variable which is uniformly distributed over $\mathcal{C}$. Let R be the random variable corresponding to the random string of the algorithm $\mathcal{P}$ and let $S(R)$ be the set of coordinates queried by $\mathcal{P}$ when the random string is R. We can guess the value of X with probability $\geqslant 1/2$ given $X_{S(R)}, R$ where $X_{S(R)}$ is the restriction of X to $S(R)$. By Fano's inequality,

$$H(X \mid X_{S(R)}, R) \leqslant h(1/2) + \frac{1}{2} \cdot \log(|\mathcal{C}| - 1) \leqslant 1 + \frac{1}{2} \log|\mathcal{C}|.$$

We can bound the mutual information between X and $X_{S(R),R}$ as follows:

$$\begin{aligned} I(X; X_{S(R)}, R) &= I(X; R) + I(X; X_{S(R)}|R) && \text{(Chain rule for mutual information.)} \\ &\leqslant 0 + H(X_{S(R)}|R) && \text{(Since } X \text{ and } R \text{ are independent.)} \\ &\leqslant t \log|\Sigma|. \end{aligned}$$

But we also have

$$I(X; X_{S(R)}, R) = H(X) - H(X|X_{S(R)}, R) \geqslant \log|\mathcal{C}| - \frac{1}{2}\log|\mathcal{C}| - 1 \geqslant \frac{1}{2}\log|\mathcal{C}| - 1.$$

Combining the upper and lower bound for $I(X; X_{S(R)}, R)$, we get the required bound. ◀

3.2 Proof of Theorem 1

The following is a restatement of Theorem 1.

▶ **Theorem 5.** *Let $\mathcal{C} \subset \Sigma^n$ be a $(2, \tau)$-LCC which is zero-error, then*

$$|\mathcal{C}| \leqslant \exp\left(O(\tfrac{1}{\tau^4} \cdot \log n \cdot \log|\Sigma|)\right).$$

Proof. We will construct a randomized algorithm $\mathcal{P}$ such that for every $c \in \mathcal{C}$, given oracle access to c, $\mathcal{P}$ makes at most $O(\frac{1}{\tau^4} \cdot \log n)$ queries to c and outputs c with probability $\geqslant 1 - 1/n$. By Lemma 4, we get the required bound.

Let $[n] = T_1 \cup T_2$ be partition of coordinates given by Lemma 3.

▶ **Claim 6.** *Algorithm $\mathcal{P}$ can learn $c|_{T_1}$ with probability $\geqslant 1 - 1/n$ by querying a uniformly random (sampled with repetitions) subset S of size $r = O(\frac{1}{\tau^2} \cdot \log n)$.*

Proof. Let $S = \{Z_1, \cdots, Z_r\}$ where each Z_i is a uniformly random element of $[n]$. By Lemma 3, for every $u \in T_1$, we have a smooth distribution $\mathcal{D}_u$ over $[n]$ and algorithms $\mathcal{R}^u_v$ for every $v \in [n]$. Let's fix $u \in T_1$ and let $p_v = \mathbf{Pr}_{\mathcal{D}_u}[v]$. By smoothness, $p_v \leqslant \frac{4}{\tau n}$ for every $v \in [n]$. The algorithm $\mathcal{P}$ estimates c_u as follows: Define the weight of σ to be

$$W_\sigma = p_\phi \cdot \mathbf{Pr}[\mathcal{R}^u_\phi = \sigma] + \frac{1}{r}\sum_{i=1}^{r} np_{Z_i} \cdot \mathbf{Pr}[\mathcal{R}^u_{Z_i}(c_{Z_i}) = \sigma]$$

and output the symbol with the maximum weight. We will show that

$$\mathbf{Pr}[\mathcal{P} \text{ guesses } c_u \text{ incorrectly}] \leqslant \frac{1}{n^2}.$$

For $\sigma \in \Sigma$ and $v \in [n] \cup \{\phi\}$, let $f_v^\sigma = \mathbf{Pr}[\mathcal{R}_v^u(c_v) = \sigma]$. The weight of σ is given by

$$W_\sigma = p_\phi f_\phi^\sigma + \frac{1}{r}\sum_{i=1}^{r} np_{Z_i} f_{Z_i}^\sigma.$$

We can calculate the expected value of the weight as

$$\begin{aligned}\mathbf{E}[W_\sigma] &= p_\phi f_\phi^\sigma + \mathbf{E}[np_{Z_1} f_{Z_1}^\sigma] \\ &= p_\phi \mathbf{Pr}[\mathcal{R}_\phi^u(c_\phi) = \sigma] + \sum_{v\in[n]} p_v \mathbf{Pr}[\mathcal{R}_v^u(c_v) = \sigma] = \mathop{\mathbf{Pr}}_{v\sim\mathcal{D}_u}[\mathcal{R}_v^u(c_v) = \sigma].\end{aligned}$$

Therefore W_σ is an unbiased estimator for $\mathbf{Pr}_{v\sim\mathcal{D}_u}[\mathcal{R}_v^u(c_v) = \sigma]$. Also $p_{Z_i} \leqslant \frac{4}{\tau n}$ and $f_{Z_i}^\sigma \leqslant 1$, so $np_{Z_i} f_{Z_i}^\sigma \leqslant \frac{4}{\tau}$. Applying Hoeffding's inequality,

$$\mathbf{Pr}\left[|W_\sigma - \mathbf{E}[W_\sigma]| \geqslant \frac{1}{20}\right] \leqslant \exp\left(-\Omega(r\tau^2)\right) \leqslant 1/2n^2$$

when $r \gg \frac{1}{\tau^2}\log n$. By Lemma 3,

$$\mathbf{E}[W_{c_u}] = \mathop{\mathbf{Pr}}_{v\sim\mathcal{D}_u}[\mathcal{R}_v^u(c_v) = c_u] \geqslant \frac{2}{3}.$$

Therefore, $\mathbf{Pr}[W_{c_u} \leqslant \frac{2}{3} - \frac{1}{20}] \leqslant 1/2n^2$. Now we will show that no other symbol can have higher weight than W_{c_u} except with probability $\frac{1}{2n^2}$. For this let us look at

$$\begin{aligned}\sum_{\sigma\in\Sigma} W_\sigma &= \sum_\sigma p_\phi f_\phi^\sigma + \frac{1}{r}\sum_{i=1}^{r} np_{Z_i}\sum_\sigma f_{Z_i}^\sigma \\ &= p_\phi \sum_\sigma \mathbf{Pr}[\mathcal{R}_\phi^u = \sigma] + \frac{1}{r}\sum_{i=1}^{r} np_{Z_i}\sum_\sigma \mathbf{Pr}[\mathcal{R}_{Z_i}^u(c_{Z_i}) = \sigma] \\ &= p_\phi + \frac{1}{r}\sum_{i=1}^{r} np_{Z_i}\end{aligned}$$

So $\mathbf{E}[\sum_{\sigma\in\Sigma} W_\sigma] = p_\phi + \mathbf{E}[np_{Z_1}] = 1$ and $np_{Z_i} \leqslant \frac{4}{\tau}$. Therefore by Hoeffding's inequality applied again, we get

$$\mathbf{Pr}\left[\left|\sum_{\sigma\in\Sigma} W_\sigma - 1\right| \geqslant \frac{1}{20}\right] \leqslant \exp\left(-\Omega(r\tau^2)\right) \leqslant \frac{1}{2n^2}$$

when $r \gg \frac{1}{\tau^2}\log n$. So with probability $\geqslant 1 - \frac{1}{n^2}$, we have $W_{c_u} \geqslant \frac{2}{3} - \frac{1}{20}$ and $\sum_{\sigma\in\Sigma} W_\sigma \leqslant 1 + \frac{1}{20}$. Therefore with probability $\geqslant 1 - \frac{1}{n^2}$, c_u will be the symbol with maximum weight and the algorithm $\mathcal{P}$ will guess c_u correctly with probability $\geqslant 1 - \frac{1}{n^2}$. By union bound, we get that $\mathcal{P}$ can guess c_u correctly for all $u \in T_1$ with probability $\geqslant 1 - \frac{1}{n}$. ◀

We will now show that after learning $c|_{T_1}$, $\mathcal{P}$ can now learn $c|_{T_2}$ by querying a further $O_\tau(\log n)$ coordinates from c and this process will be deterministic i.e. no further randomness is needed. Define $R(S)$ to be the set of coordinates of c that can be recovered correctly given $c|_S$. In Claim 6, we have shown that if S is a randomly chosen subset of size $O_\tau(\log n)$, then $T_1 \subseteq R(S)$ with probability $\geqslant 1 - \frac{1}{n}$. From now on we assume that $\mathcal{P}$ has already recovered coordinates of T_1 correctly i.e. $T_1 \subseteq R(S)$. If $T_2 \subseteq R(S)$ then we are done, the algorithm $\mathcal{P}$ can output the entire c with probability $\geqslant 1 - \frac{1}{n}$. So we can assume that $T_2 \nsubseteq R(S)$. Our goal is to show that we can add a further $O(\mathrm{poly}(1/\tau)\cdot\log n)$ vertices to S and have $R(S) = V = T_1 \cup T_2$. We show that this is indeed the case in the next section by proving the following claim, which completes the proof.

▶ **Claim 7.** *There exists a set S of size $O((1/\tau)^4 \cdot \log n)$ such that $R(S \cup T_1) = V$.* ◀

3.3 Proof of Claim 7

Claim 7 is purely graph theoretical. Let $G = (V, E)$ be the graph with $V = [n] = T_1 \cup T_2$ and $E = \cup_{i \in T_2} \mathcal{M}_i$ where $\mathcal{M}_i$ are partial matchings of size at least $(\tau/4)n$ given by Lemma 3. Let $\delta := \tau/4$. We will label each edge in E with a label in T_2 indicating which matching it belongs to. We can have parallel edges in E, but they will have different labels since they belong to different matchings. Recall that $R(S)$ is the set of coordinates of c that can be inferred from $c|_S$. Lemma 3 implies the following closure property for $R(S)$: if $(i, j) \in \mathcal{M}_k$ and $i, j \in R(S)$ then $k \in R(S)$. Next, we define $R(S)$ formally based on the graph G using this closure property.

▶ **Definition 8.** Let $G = (V, E)$ as above. Let $S \subseteq V$. We define the set $R_G(S) \subseteq V$ to be the smallest set of vertices such that:

1. $S \subseteq R_G(S)$
2. For all $i, j \in R_G(S)$ and $k \in [n]$, if $(i, j) \in \mathcal{M}_k$, then $k \in R_G(S)$. (In words, if there exists an edge (i, j) in the graph G labeled with k and both i and j are in $R_G(S)$, then so is k.)

(When the context is clear, we will use $R(S)$ instead of $R_G(S)$.) Our goal is to show that in any graph G as above, there exists a set $S \subseteq V$ of size $\text{poly}(1/\delta) \cdot \log(n)$ such that $R_G(S \cup T_1) = V$. As a first step, we get rid of the set T_1, by showing that proving the claim in the case $T_1 = \emptyset$ implies Claim 7 for any other set. To see that observe that if we take G' to be the union of G with a collection of partial matching $\{\mathcal{M}_j\}_{j \in T_1}$, then $R_{G'}(S) \subseteq R_G(S \cup T_1)$ for any set $S \subseteq V$. Thus, it suffices to introduce dummy matchings $\{\mathcal{M}_j\}_{j \in T_1}$ for each $\mathcal{M}_j$ of size δn, and prove that there exists a set S of size $\text{poly}(1/\delta) \cdot \log(n)$ such that $R_{G'}(S) = V$.

▶ **Claim 9** (Claim 7, case $T_1 = \emptyset$, restated)**.** *Let $G = (V, E)$ be a graph with $V = [n]$ and $E = \mathcal{M}_1 \cup \cdots \cup \mathcal{M}_n$ where each $\mathcal{M}_i$ is a partial matching of size at least δn. Then, there exists a subset $S \subseteq V$ of size $O((1/\delta)^4 \cdot \log n)$ such that $R_G(S) = V$.*

From here henceforth we assume (without loss of generality) that $T_1 = \emptyset$ and $T_2 = [n]$, and prove Claim 9. The following lemma tells us that we can find a subgraph G' of G such that each vertex in G' has high degree. Note that the lemma finds a subgraph restricted to a set of vertices V', and also restricted to the set of edges labeled with V'.

We shall use this lemma inductively. During induction, we will remove some edges from the matchings. Thus, instead of asserting that all matchings are of size at least $\delta|V|$, we assume that all but $0.1\delta|V|$ of the matchings have at least $0.9\delta|V|$ edges.

▶ **Lemma 10** (Clean-Up Lemma)**.** *Let $G = (V, E)$ be a graph with a finite set of vertices V and $E = \bigcup_{i \in V} \mathcal{M}_i$, where each $\mathcal{M}_i$ is a partial matching on V. Assume all but $0.1\delta|V|$ of the matchings $\mathcal{M}_i$ have size at least $0.9\delta|V|$. Then, there exists a subset $V' \subseteq V$ of size at least $\delta \cdot |V|$ so that the graph $G' = (V', E')$ where $E' = \bigcup_{i \in V'} \mathcal{M}_i \cap (V' \times V')$ has minimal degree at least $(\delta^2/4) \cdot |V|$.*

Proof. We find the set V' greedily. Let $\delta' := \delta^2/4$. Initialize $V' = V$. If the minimum degree in the remaining graph on V' is at least $\delta' \cdot |V|$ then we stop. Otherwise, remove the vertex $i \in V'$ with minimal degree, and remove all edges labeled i. We repeat this process until no vertices of degree smaller than $\delta' \cdot |V|$ exist.

If the process stopped when $|V'| \geqslant \delta|V|$ then we are done. We are left to show that the process cannot proceed past this point. Let's assume by contradiction that we can continue the process after this point. As we decrease the size of V' by one in each iteration, we must reach at a certain point of the process to a set of vertices $V' = V^*$ of size exactly $\delta|V|$.

Denote by

$$E^*(V') := \bigcup_{i \in V^*} \mathcal{M}_i \cap (V' \times V').$$

Next, we upper and lower bound $|E^*(V^*)|$ to derive a contradiction.

The upper bound $|E^*(V^*)| \leqslant |V^*| \cdot |V^*|/2$ follows since the edges $E^*(V^*)$ form a collection of $|V^*|$ partial matchings on V^*. To lower bound $|E^*(V^*)|$ we use the properties of the greedy process. The initial size of the set $E^*(V')$ (when $V' = V$) is at least $0.9\delta|V| \cdot (|V^*| - 0.1\delta|V|) \geqslant 0.9^2\delta^2 \cdot |V|^2$. In every iteration, we remove at most $\delta'|V|$ edges from this set of edges. As there are at most $|V|$ steps, we are left with at least $0.9^2\delta^2|V|^2 - \delta'|V|^2$ edges, i.e., $|E^*(V^*)| \geqslant 0.9^2\delta^2|V|^2 - \delta'|V|^2$. Combining both upper and lower bounds on $|E^*(V^*)|$ gives

$$\frac{1}{2} \cdot \delta^2 \cdot |V|^2 \geqslant |E^*(V^*)| \geqslant (0.9^2\delta^2 - \delta') \cdot |V|^2 = (0.9^2\delta^2 - \delta^2/4) \cdot |V|^2$$

which yields a contradiction since $1/2 < 0.9^2 - 1/4$. ◀

▶ **Lemma 11** (Exponentially growing a set of known coordinates). *Let $G = (V, E)$ be a graph with V and $E = \bigcup_{i \in V} \mathcal{M}_i$ such that each $v \in V$ has degree at least d. Then, there exists a subset $S \subseteq V$ of size at most $O((|V|/d) \cdot \log |V|)$ with $|R(S)| \geqslant d/2$.*

Proof. We pick the set $S \subseteq V$ iteratively, picking one element in each step. We start with $S = \{v\}$ for some arbitrary $v \in V$.

Assume we picked t elements so far for the set S. If $|R(S)| \geqslant d/2$, then we are done. Otherwise, by the definition of $R(S)$, for any $i \in V \setminus R(S)$, none of the edges in the matching $\mathcal{M}_i$ is inside $R(S)$. We wish to show that there exists an $i \in V \setminus R(S)$ with many edges into $R(S)$ marked with labels outside $R(S)$. Then, we will add i to S, which will reveal a lot of new coordinates.

For two disjoint sets of vertices $A, B \subseteq V$ we denote by $E(A, B)$ the set of edges between A and B in the graph G. If A consists of one element, i.e., $A = \{a\}$ we denote $E(a, B) = E(A, B)$. Let $A = R(S)$. Let $B = V \setminus A$. We have

$$\left| E(A, B) \cap \bigcup_{i \in B} \mathcal{M}_i \right| = \sum_{a \in A} \left| E(a, B) \cap \bigcup_{i \in B} \mathcal{M}_i \right| = \sum_{a \in A} \left| E(a, V \setminus \{a\}) \cap \bigcup_{i \in B} \mathcal{M}_i \right| \tag{1}$$

where the last equality follows since there are no edges labeled $i \in B$ between any two vertices in A. For each $a \in A$ there are at least d edges touching a and at most $|A|$ of them appeared in $\bigcup_{i \in A} \mathcal{M}_i$, hence $\left|E(a, V \setminus \{a\}) \cap \bigcup_{i \in B} \mathcal{M}_i\right| \geqslant d - |A| \geqslant d/2$. Plugging this estimate to Eq. (1) gives

$$\left| E(A, B) \cap \bigcup_{i \in B} \mathcal{M}_i \right| \geqslant |A| \cdot d/2 \,.$$

By averaging there exists a vertex $b \in B$ with at least $|A| \cdot \frac{d}{2|V|}$ edges to A labeled with B. So as long as $|A| = |R(S)| \leqslant d/2$ we are extending the set $R(S)$ by at least $|R(S)| \cdot \frac{d}{2|V|}$ elements, i.e. by a multiplicative factor of $(1 + \frac{d}{2|V|})$. Hence, after t iterations, either $|R(S)| \geqslant (1 + \frac{d}{2|V|})^t$ or $|R(S)| \geqslant d/2$. Taking $t = O(\frac{|V|}{d} \cdot \log |V|)$ gives that after at most t iterations $|R(S)| \geqslant d/2$. ◀

▶ **Lemma 12** (Covering $1 - \delta$ fraction of the coordinates implies covering all coordinates). *Let $G = (V, E)$ be a graph with $V = [n]$ and $E = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \ldots \cup \mathcal{M}_n$ and each $\mathcal{M}_i$ is a partial matching of size at least δn. Let $S \subseteq V$. If $|R(S)| > (1 - \delta)n$, then $R(S) = V$.*

Proof. Let $v \in V$. We show that there is an edge inside $R(S)$ marked v. Indeed, there are at least δn edges labeled v and they form a partial matching. If $|V \setminus R(S)| < \delta n$, one of these edges do not touch $(V \setminus R(S))$, i.e., it is an edge connecting two vertices in $R(S)$. ◀

▶ **Lemma 13** (Two Cases). *Let $G = (V, E)$ be a graph with $V = [n]$ and $E = \mathcal{M}_1 \cup \mathcal{M}_2 \cup \ldots \cup \mathcal{M}_n$ where each $\mathcal{M}_i$ is a partial matching of size at least δn. Let $S \subseteq V$. Assume $|R(S)| \leqslant (1-\delta)n$. Then, either*

1. *There exists an $i \in V \setminus R(S)$ such that $|R(S \cup \{i\})| \geqslant |R(S)| + 0.01 \cdot \delta^2 \cdot n$.*
2. *In the graph $G' = (V', E')$ with $V' = V \setminus R(S)$ and $E' = \bigcup_{i \in V'} \mathcal{M}_i \cap (V' \times V')$ all but at most $0.1\delta \cdot |V'|$ of the matchings have at least $0.9\delta \cdot n$ edges.*

Proof. Recall that the labels of edges incident to any vertex i are distinct, since the graph is a union of partial matchings. Denote by $A = R(S)$ and $B = V \setminus R(S)$. Assume for any $i \in B$ there are at most $0.01\delta^2 \cdot n$ edges to A labeled with labels in B. (Otherwise, extend S by i and get $|R(S \cup \{i\})| \geqslant |R(S)| + 0.01\delta^2 \cdot n$.) Then, there are at most $0.01\delta^2 \cdot n \cdot |B|$ edges in the cut (A, B) with labels in B. By definition of $A = R(S)$, there are no edges between A and A labeled with B. Thus, at most $0.01\delta^2 n \cdot |B|$ edges are missing from the matchings labeled by B if we restrict to edges between B and B. Hence, at most $0.1\delta \cdot |B|$ of the matchings may miss more than $0.1\delta \cdot n$ of their edges. ◀

We are now ready to prove Claim 9.

Proof of Claim 9. Initialize $S := \emptyset$. We repeat the following process. While $R(S) \neq V$, check if there exists $i \in V \setminus R(S)$ such that $|R(S \cup \{i\})| \geqslant |R(S)| + 0.01\delta^2 n$. We have two cases:

1. If such an i exists, update $S := S \cup \{i\}$.
2. Else, let $G' = (V', E')$ where $V' = V \setminus R(S)$ and $E' = \bigcup_{i \in V'} \mathcal{M}_i \cap (V' \times V')$. Let $M'_i := \mathcal{M}_i \cap (V' \times V')$. By Lemma 12, $|V'| \geqslant \delta n$. By Lemma 13, all but at most $0.1\delta|V'|$ of the matchings M'_i for $i \in V'$ have at least $0.9\delta n$ edges. Denote by $\delta' = 0.9\delta n/|V'| \geqslant \delta$. We apply Lemma 10 on G' to get a subgraph $G'' = (V'', E'')$ defined by a subset V'' of size $\Omega(\delta'|V'|)$ and $E'' = \bigcup_{i \in V''} \mathcal{M}_i \cap (V'' \times V'')$ with minimal degree $d = \Omega((\delta')^2 \cdot |V'|) \geqslant \Omega(\delta^2 n)$. We apply Lemma 11 on G'' to get a set $S'' \subseteq V''$ of size $O(\log |V''| \cdot (|V''|/d)) = O(\log n \cdot (1/\delta')^2)$ with $|R_{G''}(S'')| \geqslant \Omega(d) \geqslant \Omega(\delta^2 n)$. We update $S := S \cup S''$.

The number of times we apply case 1 or case 2 is at most $O(1/\delta^2)$, since each such step introduces $\Omega(\delta^2 n)$ new vertices to $R(S)$. In each application of case 2, at most $O((1/\delta')^2 \cdot \log n) \leqslant O((1/\delta^2) \cdot \log n)$ elements are added to S. Overall, the size of S at the end of the process will be

$$O\left(\tfrac{1}{\delta^2}\right) + O\left(\tfrac{1}{\delta^2} \cdot \tfrac{1}{\delta^2} \cdot \log n\right) = O\left(\tfrac{1}{\delta^4} \cdot \log n\right) .$$ ◀

References

1 Noga Alon and Asaf Shapira. Testing subgraphs in directed graphs. *Journal of Computer and System Sciences*, 3(69):354–382, 2004.

2 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. ACM, 2011.

3 Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 37–48. Springer, 1990.

4 Arnab Bhattacharyya, Zeev Dvir, Shubhangi Saraf, and Amir Shpilka. Tight lower bounds for linear 2-query LCCs over finite fields. *Combinatorica*, 36(1):1–36, 2016.

5 Arnab Bhattacharyya and Sivakanth Gopi. Lower bounds for 2-query LCCs over large alphabet. *CoRR*, abs/1611.06980v1, 2016. URL: `http://arxiv.org/abs/1611.06980v1`.

6 Arnab Bhattacharyya and Sivakanth Gopi. Lower bounds for constant query affine-invariant LCCs and LTCs. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 12:1–12:17, 2016. `doi:10.4230/LIPIcs.CCC.2016.12`.

7 Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995.

8 Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998.

9 Amin Coja-Oghlan, Mikael Onsjö, and Osamu Watanabe. Propagation connectivity of random hypergraphs. *The Electronic Journal of Combinatorics*, 19(1):P17, 2012.

10 Thomas M. Cover and Joy A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

11 Richard M. Dudley. Central limit theorems for empirical measures. *The Annals of Probability*, pages 899–929, 1978.

12 Zeev Dvir and Sivakanth Gopi. 2-server PIR with sub-polynomial communication. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 577–584. ACM, 2015.

13 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Breaking the quadratic barrier for 3-LCC's over the reals. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, pages 784–793. ACM, 2014.

14 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of Kelly's theorem. In *Forum of Mathematics, Sigma*, volume 2, page e4. Cambridge Univ Press, 2014.

15 Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

16 Oded Goldreich, Howard Karloff, Leonard J. Schulman, and Luca Trevisan. Lower bounds for linear locally decodable codes and private information retrieval. *Computational Complexity*, 15(3):263–296, 2006.

17 Rahul Jain. Towards a classical proof of exponential lower bound for 2-probe smooth codes. *arXiv:cs/0607042*, 2006.

18 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. ACM, 2000.

19 Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 198–207. IEEE, 2009.

20 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*, pages 106–115. ACM, 2003.

21 Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

22 Richard J. Lipton. Efficient checking of computations. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 207–215. Springer, 1990.

23 Michael Mitzenmacher and Justin Thaler. Peeling arguments and double hashing. In *Communication, Control, and Computing (Allerton), 2012 50th Annual Allerton Conference on*, pages 1118–1125. IEEE, 2012.

24 Ryuhei Mori and Osamu Watanabe. Peeling algorithm on random hypergraphs with super-linear number of hyperedges. *arXiv preprint arXiv:1506.00718*, 2015.

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

26 Endre Szemerédi. Regular partitions of graphs. In J. C. Bremond, J. C. Fournier, M. Las Vergnas, and D. Sotteau, editors, *Proc. Colloque Internationaux CNRS 260 – Problèmes Combinatoires et Théorie des Graphes*, pages 399–401, 1978.

27 Amelia Taylor. The regularity method for graphs and digraphs. *arXiv preprint arXiv:1406.6531*, 2014.

28 Stephanie Wehner and Ronald De Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *International Colloquium on Automata, Languages, and Programming*, pages 1424–1436. Springer, 2005.

29 Sergey Yekhanin. Locally decodable codes. In *Computer Science – Theory and Applications*, pages 289–290. Springer, 2011.

A LDCs from LCCs

In this section, we will show that q-query LCCs can be converted into q-query LDCs with only a constant loss in rate and preserving other parameters. Below we define LCCs and LDCs formally.

▶ **Definition 14** (Locally Correctable Code)**.** Let Σ be some finite alphabet and let $\mathcal{C} \subseteq \Sigma^n$ be a set of codewords. $\mathcal{C}$ is called a (q, δ, ε)-LCC if there exists a randomized algorithm $\mathcal{A}$ such that following is true:

1. $\mathcal{A}$ is given oracle access to some $z \in \Sigma^n$ and an input $i \in [n]$. It outputs a symbol in Σ after making at most q queries to z.
2. If $z \in \Sigma^n$ is δ-close to some codeword $c \in \mathcal{C}$ in Hamming distance, then for every $i \in [n]$, $\mathbf{Pr}[\mathcal{A}^z(i) = c_i] \geqslant \frac{1}{2} + \varepsilon$.

It is easy to see that LCCs should have large minimum distance.

▶ **Lemma 15** (Lemma 3.2 in [6])**.** *If $\mathcal{C} \subseteq \Sigma^n$ is a (q, δ, ε)-LCC, then $\mathcal{C}$ has minimum distance 2δ i.e. every two points in $\mathcal{C}$ are 2δ-far in Hamming distance.*

▶ **Definition 16** (Locally Decodable Code)**.** Let Σ be some finite alphabet and let $\mathcal{C} : \{0,1\}^k \to \Sigma^n$. $\mathcal{C}$ is called a (q, δ, ε)-LDC if there exists a randomized algorithm $\mathcal{A}$ such that following is true:

1. $\mathcal{A}$ is given oracle access to some $z \in \Sigma^n$ and an input $i \in [k]$. It outputs a bit after making at most q queries to z.
2. If $z \in \Sigma^n$ is δ-close to a codeword $\mathcal{C}(x)$ in Hamming distance for some $x \in \{0,1\}^k$, then for every $i \in [k]$, $\mathbf{Pr}[\mathcal{A}^z(i) = x_i] \geqslant \frac{1}{2} + \varepsilon$.

We will need the notion of *VC-dimension* for the reduction.

▶ **Definition 17.** Let $A \subseteq \{0,1\}^n$, then the VC-dimension of A, denoted by $\mathrm{vc}(A)$ is the cardinality of the largest set $I \subseteq [n]$ which is shattered by A i.e. the restriction of A to I, $A|_I = \{0,1\}^I$.

The following lemma due to Dudley([11]) says that if a set $A \subseteq \{0,1\}^n$ has points that are far apart from each other, then it has large VC-dimension.

▶ **Lemma 18** (Theorem 14.12 in [21]). *Let $A \subseteq \{0,1\}^n$ such that for every distinct $x, y \in A$, $\|x - y\|_{\ell_2} \geqslant \varepsilon\sqrt{n}$. Then*

$$\mathrm{vc}(A) \geqslant \Omega\left(\frac{\log|A|}{\log(2/\varepsilon)}\right).$$

We are now ready to prove the reduction from LCCs to LDCs.

▶ **Theorem 19.** *Let $\mathcal{C} \subseteq \Sigma^n$ be a (q, δ, ε)-LCC, then there exists a (q, δ, ε)-LDC $\mathcal{C}' : \{0,1\}^k \to \Sigma^n$ with*

$$k = \Omega\left(\frac{\log|\mathcal{C}|}{\log(1/\delta)}\right).$$

Proof. Wlog let us assume $\Sigma = \{0,1\}^s$. Let $\mathcal{C}_0 : \{0,1\}^s \to \{0,1\}^t$ be an error correcting code with distance δ_0 which is some fixed constant. We can extend $\mathcal{C}_0 : \Sigma^n \to \{0,1\}^{nt}$ as

$$\mathcal{C}_0(z_1, \cdots, z_n) = (\mathcal{C}_0(z_1), \cdots, \mathcal{C}_0(z_n)).$$

By Lemma 15, every two points in $\mathcal{C}$ are 2δ-far in Hamming distance, it is easy to see that in the concatenated code $\mathcal{C}_1 = \mathcal{C}_0 \circ \mathcal{C} \subseteq \{0,1\}^{tn}$ every two points are $2\delta \cdot \delta_0$ far apart in Hamming distance. So every two points in $\mathcal{C}_1$ are separated by $\varepsilon\sqrt{nt}$ distance in ℓ_2 norm where $\varepsilon = \sqrt{2\delta\delta_0}$. So by Lemma 18,

$$\mathrm{vc}(\mathcal{C}_1) \geqslant \Omega\left(\frac{\log|\mathcal{C}_1|}{\log(2/\varepsilon)}\right) = \Omega\left(\frac{\log|\mathcal{C}|}{\log(1/\delta)}\right).$$

Therefore there exists a set $I \subseteq [nt]$ of size $k = \mathrm{vc}(\mathcal{C}_1)$ such that $\mathcal{C}_1|_I = \{0,1\}^I$.

Now define $\mathcal{C}' : \{0,1\}^I \to \Sigma^n$ as follows: $\mathcal{C}'(x) = z$ where $z \in \mathcal{C}$ is chosen such that $\mathcal{C}_0(z)|_I = x$ (if there are many such z, you can choose one arbitrarily). So the image $\mathcal{C}'(\{0,1\}^I) \subseteq \mathcal{C}$. Now we claim that $\mathcal{C}'$ is an q-query LDC. Given a word $r \in \Sigma^n$ which is δ-close to $\mathcal{C}'(x)$, say we want to decode the i^{th} message coordinate x_i. Suppose i belongs to the j^{th} block of $(\{0,1\}^t)^n$ for some $j \in [n]$. The local decoder of $\mathcal{C}'$ will run the local corrector of $\mathcal{C}$ to correct the j^{th} coordinate of r and apply $\mathcal{C}_0$ to find the required bit x_i. So the local decoder for $\mathcal{C}'$ makes at most q queries and the probability that it outputs x_i correctly is at least $1/2 + \varepsilon$. ◀

B Decomposition into expanding subgraphs

The goal of this section is to develop a decomposition lemma that approximately partitions any directed graph into a collection of disjoint expanding subgraphs. We use the following notion of edge expansion:

▶ **Definition 20.** A directed graph $G = (V, E)$ is an *α-edge expander* if for every nonempty $S \subset V$,

$$|E(S, V \setminus S)| \geqslant \alpha|S||V \setminus S|.$$

Here, $E(A, B)$ is the set of edges going from A to B.

We will need the following degree form of Szemerédi regularity lemma which can be derived from the usual form of Szemerédi regularity lemma for directed graphs proved in [1].

▶ **Definition 21.** Let $G = (V, E)$ be a directed graph. We denote the indegree of a vertex $v \in V$ by $\deg^-_G(v)$ and the outdegree by $\deg^+_G(v)$. Given disjoint subsets $A, B \subset V$, the density $d(A, B)$ between A, B is defined as

$$d(A, B) = \frac{E(A, B)}{|A||B|}$$

where $E(A, B)$ is the set of edges going from A to B. We say that (A, B) is ε-regular if for every subsets $A' \subset A$ and $B' \subset B$ such that $|A'| \geqslant \varepsilon|A|$ and $|B'| \geqslant \varepsilon|B|$, $|d(A', B') - d(A, B)| \leqslant \varepsilon$.

Note that the order of A, B is important in the definition of an ε-regular pair.

▶ **Lemma 22** (Szemerédi regularity lemma for directed graphs (see Lemma 39 in [27])). *For every $\varepsilon > 0$, there exists an $M(\varepsilon) > 0$ such that the following is true. Let $G = (V, E)$ be any directed graph on $|V| = n$ vertices and let $0 < d < 1$ be any constant. Then there exists a directed subgraph $G' = (V', E')$ of G and an equipartition of V' into k disjoint parts $V_1, \cdots, V_k$ such that*

1. *$k \leqslant M(\varepsilon)$.*
2. *$|V \setminus V'| \leqslant \varepsilon n$.*
3. *All parts $V_1, \cdots, V_k$ have the same size $m \leqslant \varepsilon n$.*
4. *$\deg^+_{G'}(v) \geqslant \deg^+_G(v) - (d + \varepsilon)n$ for every $v \in V'$.*
5. *$\deg^-_{G'}(v) \geqslant \deg^-_G(v) - (d + \varepsilon)n$ for every $v \in V'$.*
6. *G' doesn't contain edges inside the parts V_i i.e. $E'(V_i, V_i) = \emptyset$ for every i.*
7. *All pairs $G'(V_i, V_j)$ with $i \neq j$ are ε-regular, each with density 0 or at least d.*

The regularity lemma above asserts pseudorandomness in the edges going between parts of the partition. For our application and others, it is more natural to require the edges inside each subgraph to display pseudorandomness. As the proof of our Decomposition Lemma shows, we can obtain this from Lemma 22 with some work.

▶ **Lemma 23** (Decomposition Lemma). *Let $G = (V, E)$ be any directed graph on $|V| = n$ vertices. For $0 < d < 1$ and $0 < \varepsilon < d/6$, there exists a directed subgraph $G' = (V', E')$ and a partition of V' into $U_1, U_2, \cdots, U_K$ where $K \leqslant M(\varepsilon)$ depends only on ε such that:*

1. *$|V \setminus V'| \leqslant 3\varepsilon n$.*
2. *$\deg^+_{G'}(v) \geqslant \deg^+_G(v) - (d + 3\varepsilon)n$ for every $v \in V'$.*
3. *$\deg^-_{G'}(v) \geqslant \deg^-_G(v) - (d + 3\varepsilon)n$ for every $v \in V'$.*
4. *There are no edges from U_i to U_j where $i > j$.*
5. *For $1 \leqslant i \leqslant K$, the induced subgraph $G'(U_i)$ is either empty or is a α-edge expander where $\alpha = \alpha(\varepsilon) > 0$.*

Proof. We will first apply Lemma 22 to G to get a directed subgraph $G''(V'', E'')$ along with a partition of $V'' = V_1 \cup \cdots \cup V_k$ as in the lemma where $k \leqslant M(\varepsilon)$. We know that every pair $G''(V_i, V_j)$ is ε-regular with density 0 or at least d. Let us construct a reduced directed graph $R([k], E_R)$ where $(i, j) \in E_R$ iff $G''(V_i, V_j)$ has density at least d. Now R has a partition into strongly connected components say given by $[k] = S_1 \cup \cdots \cup S_K$ where $K \leqslant M(\varepsilon)$ and $S_1, S_2, \cdots, S_K$ are in topological ordering i.e. there are no edges from S_i to S_j when $i > j$. We will find a large subset $V'_j \subset V_j$ for each of the parts such that $|V_j \setminus V'_j| \leqslant 2\varepsilon|V_j|$ and define $U_i = \cup_{j \in S_i} V'_j$. Our final vertex set will be $V' = \cup_{i=1}^K U_i$ and the graph G' will be the subgraph $G''(V')$. We have

$$|V \setminus V'| \leqslant |V \setminus V''| + \sum_{i=1}^{k} |V_i \setminus V'_i| \leqslant 3\varepsilon n.$$

For every $v \in V'$,

$$\deg^-_{G'}(v) \geqslant \deg^-_{G''}(v) - \sum_{i=1}^{k} |V_i \setminus V'_i| \geqslant \deg^-_G(v) - (d+\varepsilon)n - 2\varepsilon n = \deg^-_G(v) - (d+3\varepsilon)n.$$

Similarly $\deg^+_{G'}(v) \geqslant \deg^+_G(v) - (d+3\varepsilon)n$. Because the components $S_1, \cdots, S_k$ are in topological ordering with respect to the reduced graph R, we cannot have any edges between U_i and U_j where $i > j$.

Now we describe how to find these subsets V'_j where $j \in S_i$ for each of the S_i's and also show the required expansion property. If S_i is a singleton set i.e. $S_i = \{j\}$ for some j, then we just define $V'_j = V_j$. In this case, we will have $U_i = V_j$ and the subgraph $G'(U_i)$ will be empty. If $|S_i| > 1$, the subgraph $R(S_i)$ is strongly connected with at least two vertices. So every vertex $j \in S_i$ has at least one outgoing neighbor and one incoming neighbor in $R(S_i)$; choose one outgoing neighbor and call it $N^+(j)$ and choose one incoming neighbor and call it $N^-(j)$. Let $V'_j \subset V_j$ be the subset of vertices with at least $(d-\varepsilon)|V_{N^+(j)}|$ outgoing neighbors in $V_{N^+(j)}$ and at least $(d-\varepsilon)|V_{N^-(j)}|$ incoming neighbors in $V_{N^-(j)}$. We will now show that $|V_j \setminus V'_j| \leqslant 2\varepsilon|V_j|$. Let $B^+_j \subset V_j$ be the set of vertices with less than $(d-\varepsilon)|V_{N^+(j)}|$ neighbors in $V_{N^+(j)}$. Define $B^-_j \subset V_j$ similarly. We have $V'_j = V_j \setminus (B^+_j \cup B^-_j)$. So it is enough to show $|B^+_j| \leqslant \varepsilon|V_j|$ and $|B^-_j| \leqslant \varepsilon|V_j|$.

Consider the ε-regular pair $(V_j, V_{N^+(j)})$ which has density at least d. The density between B^+_j and $V_{N(j)}$ can be bounded as

$$\frac{|E''(B^+_j, V_{N^+(j)})|}{|B^+_j||V_{N^+(j)}|} < d - \varepsilon \leqslant d(V_j, V_{N^+(j)}) - \varepsilon.$$

By ε-regularity of $G''(V_j, V_{N^+(j)})$, we must have $|B^+_j| \leqslant \varepsilon|V_j|$ as required. Similarly we have $|B^-_j| \leqslant \varepsilon|V_j|$.

Now we need to show that $G'(U_i)$ is an α-edge expander. Let $A \subset U_i$. For $j \in S_i$, define $A_j = A \cap V'_j$ and $\bar{A}_j = V'_j \setminus A$ and let $\bar{A} = U_i \setminus A$. We want to show that $E'(A, \bar{A}) \geqslant \alpha|A||\bar{A}|$ for some constant $\alpha(\varepsilon) > 0$. We have three cases:

Case 1: $\exists j, \ell \in S_i$ such that $|A_j| \geqslant 2\varepsilon|V'_j|$ and $|\bar{A}_\ell| \geqslant 2\varepsilon|V'_\ell|$.

Label vertices of $R(S_i)$ with $\mathcal{A}$ if $|A_j| \geqslant 2\varepsilon|V'_j|$ and also with a label $\bar{\mathcal{A}}$ if $|\bar{A}_j| \geqslant 2\varepsilon|V'_j|$.[8] Every vertex should get at least one of the labels and j has label $\mathcal{A}$ and ℓ has label $\bar{\mathcal{A}}$. Since $|S_i| > 1$, we can assume with out loss of generality that $j \neq \ell$. Since the graph $R(S_i)$ is strongly connected, there is a directed path from j to ℓ. On this path, there must exist two adjacent vertices $p, q \in S_i$ such that p has label $\mathcal{A}$, q has label $\bar{\mathcal{A}}$ and there is an edge from p to q in $R(S_i)$. We have

$$|A_p| \geqslant 2\varepsilon|V'_p| \geqslant 2\varepsilon(1-2\varepsilon)|V_p| \geqslant \varepsilon|V_p|$$

and similarly $|\bar{A}_q| \geqslant \varepsilon|V_q|$. By ε-regularity of $G''(V_p, V_q)$, we can lower the bound the number of edges between A and $\bar{A}$ as follows:

$$|E'(A, \bar{A})| \geqslant |E''(A_p, \bar{A}_q)| \geqslant (d-\varepsilon)|A_p||\bar{A}_q| \geqslant \varepsilon^2(d-\varepsilon)n^2/k^2 \geqslant \alpha_0|A||\bar{A}|$$

where $\alpha_0(\varepsilon) = 5\varepsilon^3/M(\varepsilon)^2$ is some constant depending on ε.

[8] Some vertices can get both labels, but every vertex will get at least one label.

Case 2: For every $j \in S_i$, $|A_j| < 2\varepsilon|V'_j|$.

By averaging there exists some $j \in S_i$ such that $|A_j| \geqslant |A|/|S_i| \geqslant |A|/k$. We know that every vertex in V'_j has at least $(d-\varepsilon)|V_{N^+(j)}|$ out neighbors in $V_{N^+(j)}$, out of these at least

$$(d-\varepsilon)|V_{N^+(j)}| - |V_{N^+(j)} \setminus V'_{N^+(j)}| - |A_{N^+(j)}| \geqslant (d-5\varepsilon)|V_{N^+(j)}|$$

should lie in $\bar{A}_{N^+(j)}$. So we can bound the expansion as follows:

$$|E'(A,\bar{A})| \geqslant |E''(A_j, \bar{A}_{N^+(j)})| \geqslant (d-5\varepsilon)|V_{N^+(j)}||A_j| \geqslant (d-5\varepsilon)\frac{n}{k}\frac{|A|}{k} \geqslant \alpha_1|A||\bar{A}|$$

where $\alpha_1 = \varepsilon/M(\varepsilon)^2$ is some constant depending only on ε.

Case 3: For every $j \in S_i$, $|\bar{A}_j| < 2\varepsilon|V'_j|$.

This is very similar to Case 2. By averaging there exists some $j \in S_i$ such that $|\bar{A}_j| \geqslant |\bar{A}|/|S_i| \geqslant |\bar{A}|/k$. Every vertex in V'_j has at least $(d-\varepsilon)|V_{N^-(j)}|$ incoming neighbors in $V_{N^-(j)}$, out of these at least

$$(d-\varepsilon)|V_{N^-(j)}| - |V_{N^-(j)} \setminus V'_{N^-(j)}| - |\bar{A}_{N^-(j)}| \geqslant (d-5\varepsilon)|V_{N-(j)}|$$

should lie in $A_{N^-(j)}$. So,

$$|E'(A,\bar{A})| \geqslant |E''(A_{N^-(j)}, \bar{A}_j)| \geqslant (d-5\varepsilon)|V_{N^-(j)}||\bar{A}_j| \geqslant (d-5\varepsilon)\frac{n}{k}\frac{|\bar{A}|}{k} \geqslant \alpha_1|A||\bar{A}|$$

where $\alpha_1 = \varepsilon/M(\varepsilon)^2$.

Finally we can take $\alpha = \min(\alpha_0, \alpha_1)$, to get the required expansion property. ◀

The decomposition lemma allows to give an alternative proof for Claim 7, with worse dependency on τ. To account for that, we restate Claim 7 and replace $O((1/\tau^4)\cdot\log n)$ with $O_\tau(\log n)$.

▶ **Claim 24.** *Let S be a set of size $O_\tau(\log n)$ such that $R(S) = T_1$. Then, S can be extended by at most $O_\tau(\log n)$ elements, such that $R(S) = V$.*

Proof. Let $\{\mathcal{M}_v : v \in T_2\}$ be the matchings obtained from Lemma 3, we know that $|\mathcal{M}_v| \geqslant \frac{\tau}{4}n$ for each $v \in T_2$. We will construct a directed graph $G(V,E)$ where $V = [n]$ and E is defined as follows. For every $v \in T_2 \setminus R(S)$ and every edge $\{i,j\} \in \mathcal{M}_v$, add directed edges $(i,v),(j,v)$ to E. Thus there is a natural pairing among the directed edges of G, we will call (j,v) the *pairing edge* of (i,v) and vice versa. $\{i,j\}$ is called the *matching edge* corresponding to the pair $(i,v),(j,v)$. Since each matching $\mathcal{M}_v$ has size $\geqslant \tau n/4$, we have $\deg^-_G(v) \geqslant \delta n$ where $\delta := \tau/2$ for every $v \in T_2 \setminus R(S) = V \setminus R(S)$.

We now apply Lemma 23 to get a subgraph $G' = (V', E')$ as described in the lemma where we will choose $\varepsilon = \delta/100$ and $d = \delta/10$. Let $V' = U_1 \cup \cdots \cup U_K$ be the partition of G' as described in the lemma where $K \leqslant M(\delta)$. Let $V_0 = [n] \setminus V'$ be the remaining vertices, we have $|V_0| \leqslant 3\varepsilon n$. Each vertex $v \in V' \cap (T_2 \setminus R(S))$ has $\deg^-_{G'}(v) \geqslant (\delta - d - 3\varepsilon)n$. We also know that each sub-graph $G'(U_i)$ is either empty or is an α-edge expander for some constant $\alpha(\varepsilon) > 0$.

Note that S already has $O_\tau(\log n)$ vertices. We will now grow the set S of coordinates queried by $\mathcal{P}$ iteratively, adding one at a time. Algorithm 1 gives the procedure for growing the set S.

We will finish the analysis in a series of claims. Let us start with a simple claim about properties of $R(S)$.

Algorithm 1 Algorithm for growing S

for $i = 1$ **to** K **do**
  Intialization: Pick one vertex from U_i and add it to S.
  while $U_i \not\subseteq R(S)$ **do**
    Pick any $v \in V \setminus R(S)$ such that adding it to S will add the maximum number of vertices in $U_i \setminus R(S)$ to $R(S)$.
  end while
end for

▶ **Claim 25.** *$R(S)$ has the following properties:*

1. *If $i, j \in R(S)$ and $(i, j) \in \mathcal{M}_k$ then $k \in R(S)$.*
2. *For every edge $(i, k) \in E(R(S), V \setminus R(S))$, there is a unique $j \in V \setminus R(S)$ such that $(i, j) \in \mathcal{M}_k$.*

Proof. (1) We can recover c_i, c_j from $c|_S$ and then use them to recover c_k since by Lemma 3, there exists an algorithm $\mathcal{R}^k_{i,j}$ such that for every $c \in \mathcal{C}$, $\mathcal{R}^k_{i,j}(c_i, c_j) = c_k$.
(2) Let (j, k) be the pairing edge of (i, k) so that $(i, j) \in \mathcal{M}_k$. Now j cannot be in $R(S)$ because of (1). ◀

Algorithm 1 should terminate, since $|U_i \cap R(S)|$ increases by at least one in every iteration of the while loop. At the end of the procedure we clearly have $V' = U_1 \cup \cdots \cup U_K \subset R(S)$. In fact, we can claim that at the end of the procedure $R(S) = V$ i.e. we can recover all the coordinates of c from $c|_S$.

▶ **Claim 26.** *After Algorithm 1 terminates, $R(S) = V = [n]$.*

Proof. After Algorithm 1 terminates, we have $V' \subset R(S)$. Now we are left with $V_0 = V \setminus V'$ where we know that $|V_0| \leqslant 3\varepsilon n$. Now if $w \in V_0 \setminus R(S)$ then $w \in T_2 \setminus R(S)$ since $T_1 \subset R(S)$. Therefore $\deg^-_G(w) \geqslant \delta n$. So there must be $\delta n - |V_0| \geqslant (\delta - 3\varepsilon)n$ incoming edges from V' to w. So two of these incoming edges must from a pair and so we have $w \in R(S)$ by part (1) of Claim 25. Therefore $V_0 \subset R(S)$ as well. ◀

▶ **Claim 27.** *Algorithm 1 terminates after $O_\delta(\log n)$ rounds.*

Proof. We just need to show that the while loop runs for $O_\delta(\log n)$ rounds for each $i \in [K]$ since the outer for loop runs for K times where $K \leqslant M(\delta)$. There are two cases:

Case 1: The subgraph $G'(U_i)$ is empty.

In this case, we will show that U_i must already be contained in $R(S)$. Suppose not, let $w \in U_i \setminus R(S)$, we have $\deg^-_{G'}(w) \geqslant (\delta - d - 3\varepsilon)n$. Moreover, all of these incoming edges come from $U_1, \cdots, U_{i-1}$ (note that this means $i > 1$ for this case to happen). Therefore there must be two incoming edges from $U_1 \cup \cdots \cup U_{i-1}$ which form a pair i.e. there exists $u, v \in U_1 \cup \cdots \cup U_{i-1}$ such that $(u, v) \in \mathcal{M}_w$. So by part (1) of Claim 25, $w \in R(S)$. This is a contradiction.

Case 2: The subgraph $G'(U_i)$ is an α-edge expander.

If $U_i \not\subseteq R(S)$, we will show that after the end of the iteration $t_i := |R(S) \cap U_i|$ increases by a factor of $(1 + \varepsilon\alpha)$. This will prove the required claim because t_i is upper bounded by n.

We first claim that $|U_i \setminus R(S)| \geqslant \varepsilon n$. Suppose this is not true i.e. $|U_i \setminus R(S)| \leqslant \varepsilon n$. Let $w \in U_i \setminus R(S)$. We know that w has $\deg^-_{G'}(w) \geqslant (\delta - d - 3\varepsilon)n$ incoming edges in G'. Since no edges come from U_j for $j > i$, at least $(\delta - d - 3\varepsilon)n - |U_i \setminus R(S)| \geqslant (\delta - d - 4\varepsilon)n$

of them come from $U_1 \cup \cdots \cup U_{i-1} \cup (U_i \cap R(S)) \subset R(S)$. Therefore two of the incoming edges must form a pair and so $w \in R(S)$ which is a contradiction.
Since $G'(U_i)$ is an α-edge expander, we have

$$E(U_i \cap R(S), U_i \setminus R(S)) \geqslant \alpha t_i |U_i \setminus R(S)| \geqslant \alpha\varepsilon t_i n.$$

By part (2) of Claim 25, each edge from $U_i \cap R(S)$ to $U_i \setminus R(S)$ corresponds to a matching edge between $U_i \cap R(S)$ and $V \setminus R(S)$ and it belongs to a matching which corresponds to a vertex in $U_i \setminus R(S)$. Therefore there are at least $\alpha\varepsilon t_i n$ matching edges between $U_i \cap R(S)$ and $V \setminus R(S)$ which belong to $\cup_{w \in U_i \setminus R(S)} \mathcal{M}_w$; by averaging there exists $v \in V \setminus R(S)$ which is incident to $\alpha\varepsilon t_i n / |V \setminus R(S)| \geqslant \alpha\varepsilon t_i$ of these matching edges. So adding this v to S will add $\alpha\varepsilon t_i$ new vertices of $U_i \setminus R(S)$ to $R(S)$, increasing t_i by a factor of $(1 + \alpha\varepsilon)$.

◀
◀