Electronic Colloquium on Computational Complexity, Report No. 189 (2016)

Lower bounds for 2-query LCCs over large alphabet

Arnab Bhattacharyya*
Indian Institute of Science
arnabb@csa.iisc.ernet.in

Sivakanth Gopi†
Princeton University
sgopi@cs.princeton.edu

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 *zero-error* 2-query locally correctable code $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ that can correct a constant fraction of corrupted symbols must have $n \geqslant \exp(k/\log|\Sigma|)$. 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 $\Omega\left((k/\log|\Sigma|)^2\right)$ 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.

For our proof of the result, we need a new decomposition lemma for directed graphs that may be of independent interest. Given a dense directed graph G, our decomposition uses the directed version of Szemerédi regularity lemma due to Alon and Shapira (STOC 2003) to partition almost all of G into a constant number of subgraphs which are either edge-expanding or empty.

*Research partially supported by a DST Ramanujan Fellowship.
†Research partially supported by NSF grants CCF-1523816 and CCF-1217416. Part of this research was done while the author was at Microsoft Research, Redmond.

ISSN 1433-8092

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 complexity means that there are short local dependencies among codeword symbols, while short length along with resilience to corruption means that these dependencies do not impose too many constraints on the code. 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 [Lip90] and by Blum and Kannan [BK95] on program checkers. In particular, [Lip90, BF90] 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 [KT00]. LDCs are weaker than LCCs, in the sense that any LCC can be converted into an LDC while preserving relevant parameters. LDCs and LCCs have found applications in derandomization and hardness results [STV01, DS07, KS09]. See [Yek11] 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 analysis of design matrices and the theory of matrix scaling, e.g. [BDYW11, DSW14b, DSW14a, DGOS16].

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. [CKGS98] 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 [KT00] 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. [GKST06] showed that $n \geqslant \exp(\delta k/|\Sigma|)$ for linear 2-query LDCs, while Kerenedis and de Wolf [KdW03] (with further improvements in [WdW05]) showed using quantum information theory 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 any linear LCC can be made zero-error.

Theorem 1.1 (Informal). *If $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ is a zero-error 2-query LCC with a corrector that can tolerate δn corruptions, then $n \geqslant \exp(c_\delta k / \log |\Sigma|)$ where c_δ is a constant depending only on δ.*

1.1 Discussion of Main Result

The lower bound in Theorem 1.1 is tight in its dependence on k and Σ. Specifically, Yekhanin in the appendix of [BDSS11] 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

$$(\mathcal{C}(\mathbf{x}))_a = \Big(\mathcal{H}\left(x_{i,1}, \ldots, x_{i,k/b}\right)_a : i \in [b]\Big) \in \{0,1\}^b$$

where $\mathcal{H}$ is the classical Hadamard encoding $\mathcal{H} : \{0,1\}^r \to \{0,1\}^{2^r}$ defined as

$$\mathcal{H}(\mathbf{y}) = \left(\sum_{i=1}^{r} y_i \chi_i \pmod 2 : \chi_1, \ldots, \chi_r \in \{0,1\}\right).$$

It is well-known and obvious 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. Note that this LCC is a non-linear code. In fact, it is necessarily so, as in the same paper [BDSS11], the authors show that if the alphabet is a finite field $\mathbb{F}$ and $\mathcal{C} \subseteq \mathbb{F}^n$ is a *linear* 2-query LCC i.e. $\mathcal{C}$ is a subspace of $\mathbb{F}^n$, then $n \geqslant \exp(k)$ where $k = \dim(\mathcal{C})$ is the message length for the code. So, unlike the case of binary 2-query LDCs where the linear Hadamard code has asymptotically optimal parameters, linear codes are necessarily suboptimal here.

The explicit dependence of the lower bound on δ is suppressed in Theorem 1.1. The constant c_δ is actually extremely tiny, due to our use of the Szemerédi regularity lemma in the proof. Getting a better dependence on δ would require different techniques and is an intriguing challenge left open by our work. c_δ may be linear in δ; a simple modification of Yekhanin's construction above gives $(2^{O(\delta k/\log|\Sigma|)}/\delta)$-length 2-query LCCs that tolerate δn corruptions.

It is important to note that Theorem 1.1 cannot be true for 2-query LDCs. Such a result would contradict the construction in [DG15] 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 [KT00] show that $n \geqslant \Omega(\delta k^2/\log^2|\Sigma|)$. And the quantum arguments of [KdW03, WdW05] 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 [KT00] 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,

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

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 $(a,b) \in M_c$ and $(c,d) \in M_e$, and a,b,d are labeled but c and e are not, we can run $\mathcal{A}_c$ to deduce the label of c and then $\mathcal{A}_e$ to deduce the label of e. 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 $O_\delta(\log n)$* 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 [KT00], 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. [COW12] 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 [MT12, MW15], 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

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

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 $O(\log n)$, $R(S) = [n]$.

This simple argument used the fact that the size of the cut between $R(S)$ and the rest of the graph is large. When the matchings are not perfect, this may not happen. (For instance, a codeword of length n could be the concatenation of two LCC codewords of length $n/2$.) It is then natural to try to partition G into a set of expanders, so that we can analyze the propagation for each part separately. The paradigm of showing that a graph is close to a union of disjoint expander (or expander-like) subgraphs has found repeated success in graph theory and algorithms (e.g., [LS93, LR99, GR99, Tre05, PT07, ABS10]; see [MS15] for an overview), and many tools have been found for this purpose. For us, it seems essential that the number of expanders in the decomposition not depend on n. Szemerédi's celebrated regularity lemma [Sze78] provides just such a guarantee.

An added twist in our setup is that in our proof above, we not only wanted the size of the cut between $R(S)$ and the rest of the graph to be large but also, we wanted the edges in this cut to belong to M_i for $i \notin R(S)$. If they all belong to matchings for vertices inside $R(S)$, then adding a single new vertex to S may not increase $R(S)$. Note that if we made the assumption that the LCC is 'undirected', meaning that if $(j, k) \in M_i$, then $(i, j) \in M_k$ and $(i, k) \in M_j$, then all edges in the cut between $R(S)$ and the rest of the graph would be in matchings corresponding to vertices outside $R(S)$, and the situation would be simpler. To get around this assumption, it turns out that a directed version of the regularity lemma is more appropriate.

We consider the directed graph $\overrightarrow{G}$ where for any $(j, k) \in M_i$, there are two directed edges (j, i) and (k, i). We then invoke a regularity lemma for dense directed graphs due to Alon and Shapira [AS04] and reformulate it for graphs with a lower bound on the minimum in-degree. This version of the lemma may be of independent interest. Our lemma yields a collection of vertex-disjoint subgraphs $U_1, U_2, \ldots, U_K$ that include all but a small fraction of vertices and edges; here, K is independent of the size of the graph. Moreover, each U_i is either empty or edge-expanding, and there are no edges from U_i to U_j for $i > j$. Once this decomposition is in place, we first find a set S_1 that propagates to all of U_1, then a set S_2 to propagate to all of U_2, and so on. The edge-expansion inside the U_i's is enough to conclude that each $|S_i| = O(\log n)$, and the proof is complete.

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†. We note that there is a simple information-theoretic proof of the regularity lemma [Tao06], and so perhaps, information theory is the right language to describe the whole argument. However, this appears quite challenging at the moment.

†This approach is taken in [Jai06] 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.

2 Zero-error 2-query LCCs

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

Definition 2.1. *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 an symbol in Σ after making*[‡] *2 non-adaptive queries to z.*

2. *If $z \in \Sigma^n$ is τn-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 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 [KT00, Yek11] 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 2.2. *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 $c \in \mathcal{C}$,*

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

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. 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.

[‡] Without loss of generality, we can assume $\mathcal{A}$ makes exactly 2 queries.
[§] Here c_ϕ is an empty input defined for ease of notation.

In this case, we include $i \in T_2$ and define $\mathcal{M}_i$ to be a matching of size εn in $\mathrm{supp}(\mathcal{Q}_i)$. For any $\alpha, \beta \in \Sigma$, let $\mathcal{R}^i_{j,k}(\alpha, \beta)$ be the output[¶] of $\mathcal{A}^z(i)$ when it samples (j,k) from the distribution $\mathcal{Q}_i$ and $z \in \Sigma^n$ such that $z_j = \alpha$ and $z_k = \beta$. Now since our LCC is zero-error, for every codeword $c \in \mathcal{C}$ and every $(j,k) \in \mathrm{supp}(\mathcal{Q}_i)$, we have $\mathcal{R}^i_{j,k}(c_j, c_k) = c_i$ with probability 1. 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 a coordinate in $V_i \cup B_i$, return 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 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:

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

for $j \in [n]$ and

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

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

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

This proves part (1). □

3 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:

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

Definition 3.1. *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 [AS04].

Definition 3.2. *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 3.3 (Szemerédi regularity lemma for directed graphs (Lemma 39 in [Tay14]))**.** *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 3.3 with some work.

Lemma 3.4 (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 3.3 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 $i > j$. This proves parts (1) to (4).

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 in part (5). 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 $V_{N(j)}$ in G'' 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 $|\bar{A}_j| \geqslant 2\varepsilon|V'_j|$.[‖] 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 without 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:

$$|E'(A,\bar{A})| \geqslant |E''(A_p,\bar{A}_q)| \geqslant (d-\varepsilon)|A_p||\bar{A}_q| \geqslant (d-\varepsilon)\varepsilon^2(1-\varepsilon)^2\frac{n^2}{k^2} \geqslant \alpha_0|A||\bar{A}|$$

where $\alpha_0(\varepsilon) = 5\varepsilon^3(1-\varepsilon)^2/M(\varepsilon)^2$ is some constant depending on ε.
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:

$$|E'(A,\bar{A})| \geqslant |E''(A_j,\bar{A}_{N^+(j)})| \geqslant (d-5\varepsilon)|V_{N^+(j)}||A_j| \geqslant (d-5\varepsilon)(1-\varepsilon)\frac{n}{k}\frac{|A|}{k} \geqslant \alpha_1|A||\bar{A}|$$

where $\alpha_1 = \varepsilon(1-\varepsilon)/M(\varepsilon)^2$ 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,

$$|E'(A,\bar{A})| \geqslant |E''(A_{N^-(j)},\bar{A}_j)| \geqslant (d-5\varepsilon)|V_{N^-(j)}||\bar{A}_j| \geqslant (d-5\varepsilon)(1-\varepsilon)\frac{n}{k}\frac{|\bar{A}|}{k} \geqslant \alpha_1|A||\bar{A}|$$

where $\alpha_1 = \varepsilon(1-\varepsilon)/M(\varepsilon)^2$.

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

4 Proof of lower bound

4.1 An information theoretic lemma

The proof of Theorem 1.1 works by showing that there is a randomized algorithm which can guess an unknown codeword $c \in \mathcal{C} \subset \Sigma^n$ with high probability by querying a small number of coordinates

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

of c. 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 [CT12] for precise definitions and proofs of the facts we use.

Given random variables X, Y, Z with some joint distribution, 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 [CT12] 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 4.1. *Suppose there exists a randomized algorithm $\mathcal{P}$ such that for every $c \in \mathcal{C} \subset \Sigma^n$, given oracle access to c, $\mathcal{P}$ queries at most t coordinates of 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:

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

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. $\square$

4.2 Proof of Theorem 1.1

The following is a restatement of Theorem 1.1.

Theorem 4.2. *Let $\mathcal{C} \subset \Sigma^n$ be a $(2,\tau)$-LCC which is zero-error, then* $|\mathcal{C}| \leqslant \exp\left(O_\tau(\log n \log |\Sigma|)\right)$.

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 $O_\tau(\log n)$ queries to c and outputs c with probability $\geqslant 1 - 1/n$. By Lemma 4.1, we get the required bound.

Let $[n] = T_1 \cup T_2$ be the partition of coordinates given by Lemma 2.2.

Claim 4.3. *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 $r = O_\tau(\log n)$.*

Proof. Let $S = \{Z_1, \cdots, Z_r\}$ where each Z_i is a uniformly random element of $[n]$. By Lemma 2.2, for every $u \in T_1$, we have a smooth distribution $\mathcal{D}_u$ over $[n]$ and algorithms $\mathcal{R}_v^u$ for every $v \in [n] \cup \{\phi\}$. 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 $\sigma \in \Sigma$ to be

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

and output the symbol with the maximum weight. Note that to estimate the weights, $\mathcal{P}$ will only need to query c at the locations $Z_1, \cdots, Z_r$. 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

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

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 2.2,

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

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 4.3, we have shown that if S is a randomly chosen subset of size $O_\tau(\log n)$, then $T_1 \subset R(S)$ 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. $T_1 \subset R(S)$. If $T_2 \subset R(S)$ 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 $T_2 \not\subseteq R(S)$. Our goal is to show that we can add a further $O_\tau(\log n)$ vertices to S and have $R(S) = V = T_1 \cup T_2$.

Let $\{\mathcal{M}_v : v \in T_2\}$ be the matchings obtained from Lemma 2.2, 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 3.4 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 $R(S)$.

Claim 4.4. *$R(S)$ has the following properties:*

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** $U_i \nsubseteq R(S)$ **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**

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 2.2, 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 4.5. *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 i.e. there exists $u, v \in V'$ such that $(u, v) \in M_w$ and so we have $w \in R(S)$ by part (1) of Claim 4.4. Therefore $V_0 \subset R(S)$ as well. □

Claim 4.6. *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 4.4, $w \in R(S)$. This is a contradiction.
Case 2: The subgraph $G'(U_i)$ is an α-edge expander.
If $U_i \nsubseteq R(S)$, 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 4.4, 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 $(1 + \alpha\varepsilon)$. $\square$

Q.E.D.

References

[ABS10] Sanjeev Arora, Boaz Barak, and David Steurer. Subexponential algorithms for unique games and related problems. In *Proc. 51st Annual IEEE Symposium on Foundations of Computer Science*, pages 563–572. IEEE, 2010. 4

[AS04] Noga Alon and Asaf Shapira. Testing subgraphs in directed graphs. *Journal of Computer and System Sciences*, 3(69):354–382, 2004. 4, 7

[BDSS11] Arnab Bhattacharyya, Zeev Dvir, Amir Shpilka, and Shubhangi Saraf. Tight lower bounds for 2-query lccs over finite fields. In *Proc. 52nd Annual IEEE Symposium on Foundations of Computer Science*, pages 638–647. IEEE, 2011. 2

[BDYW11] Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proc. 43rd Annual ACM Symposium on the Theory of Computing*, pages 519–528. ACM, 2011. 1

[BF90] Donald Beaver and Joan Feigenbaum. Hiding instances in multioracle queries. In *Proc. Annual Symposium on Theoretical Aspects of Computer Science*, pages 37–48. Springer, 1990. 1

[BK95] Manuel Blum and Sampath Kannan. Designing programs that check their work. *J. ACM*, 42(1):269–291, 1995. 1

[CKGS98] Benny Chor, Eyal Kushilevitz, Oded Goldreich, and Madhu Sudan. Private information retrieval. *J. ACM*, 45(6):965–981, 1998. 1

[COW12] Amin Coja-Oghlan, Mikael Onsjö, and Osamu Watanabe. Propagation connectivity of random hypergraphs. *The Electronic Journal of Combinatorics*, 19(1):P17, 2012. 3

[CT12] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 10

[DG15] Zeev Dvir and Sivakanth Gopi. 2-server PIR with sub-polynomial communication. In *Proc. 47th Annual ACM Symposium on the Theory of Computing*, pages 577–584. ACM, 2015. 2

[DGOS16] Zeev Dvir, Ankit Garg, Rafael Oliveira, and József Solymosi. Rank bounds for design matrices with block entries and geometric applications. *Preprint arXiv:1610.08923*, 2016. 1

[DS07] 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. 1

[DSW14a] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Breaking the quadratic barrier for 3-LCC's over the reals. In *Proc. 46th Annual ACM Symposium on the Theory of Computing*, pages 784–793. ACM, 2014. 1

[DSW14b] 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. 1

[GKST06] 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. 1

[GR99] Oded Goldreich and Dana Ron. A sublinear bipartiteness tester for bounded degree graphs. *Combinatorica*, 19(3):335–373, 1999. 4

[Jai06] Rahul Jain. Towards a classical proof of exponential lower bound for 2-probe smooth codes. *arXiv:cs/0607042*, 2006. 4

[KdW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proc. 35th Annual ACM Symposium on the Theory of Computing*, pages 106–115. ACM, 2003. 1, 2

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Proc. 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 198–207. IEEE, 2009. 1

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proc. 32nd Annual ACM Symposium on the Theory of Computing*, pages 80–86. ACM, 2000. 1, 2, 3, 5

[Lip90] Richard J Lipton. Efficient checking of computations. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 207–215. Springer, 1990. 1

[LR99] Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *J. ACM*, 46(6):787–832, 1999. 4

[LS93] Nathan Linial and Michael Saks. Low diameter graph decompositions. *Combinatorica*, 13(4):441–454, 1993. 4

[MS15] Guy Moshkovitz and Asaf Shapira. Decomposing a graph into expanding subgraphs. In *Proc. 26th ACM-SIAM Symposium on Discrete Algorithms*, pages 1283–1295. SIAM, 2015. 4

[MT12] Michael Mitzenmacher and Justin Thaler. Peeling arguments and double hashing. In *Proc. 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 1118–1125. IEEE, 2012. 3

[MW15] Ryuhei Mori and Osamu Watanabe. Peeling algorithm on random hypergraphs with superlinear number of hyperedges. *arXiv preprint arXiv:1506.00718*, 2015. 3

[PT07] Mihai Pătraşcu and Mikkel Thorup. Planning for fast connectivity updates. In *Proc. 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 263–271. IEEE, 2007. 4

[STV01] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Journal of Computer and System Sciences*, 62(2):236–266, 2001. 1

[Sze78] 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. 4

[Tao06] Terence Tao. Szemerédi's regularity lemma revisited. *Contributions to Discrete Mathematics*, 1(1), 2006. 4

[Tay14] Amelia Taylor. The regularity method for graphs and digraphs. *arXiv preprint arXiv:1406.6531*, 2014. 7

[Tre05] Luca Trevisan. Approximation algorithms for unique games. In *Proc. 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 197–205. IEEE, 2005. 4

[WdW05] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *Annual International Conference on Automata, Languages, and Programming*, pages 1424–1436. Springer, 2005. 1, 2

[Yek11] Sergey Yekhanin. Locally decodable codes. In *Computer Science–Theory and Applications*, pages 289–290. Springer, 2011. 1, 5

ECCC ISSN 1433-8092

http://eccc.hpi-web.de