Locality in Coding Theory

Sivakanth Gopi

A Dissertation
Presented to the Faculty
of Princeton University
in Candidacy for the Degree
of Doctor of Philosophy

Recommended for Acceptance
by the Department of
Computer Science
Adviser: Professor Zeev Dvir

September 2018

Abstract

Error correcting codes have been extremely successful in practice to build storage and communication systems which are resilient to noise and corruptions. They also found several theoretical applications in complexity theory, pseudorandomness, probabilistically checkable proofs and cryptography. Each application requires codes with specific properties. One such property desirable in many applications is 'locality'. Locality refers to the ability to perform operations like decoding/correction/testing in sublinear or sometimes constant time. For example, a constant query locally decodable code (LDC) allows decoding of any message bit in constant time given a corrupted encoding of the message.

Much of the work in this thesis is to understand the power and limitations of codes with locality. We show that one can get non-trivial locality and still match the best known rate-distance tradeoffs of traditional error correcting codes (Gilbert-Varshamov bound). We prove several conditional lower bounds on codes with locality and give new directions for constructing such codes by showing an analytic characterization of LDCs.

We also explore applications of such codes to additive combinatorics, information privacy and data storage. We show how to use ideas from existing constructions of LDCs to design 2-server private information retrieval schemes where a user can efficiently and privately query a database replicated among two (non-communicating) servers without revealing any information about their query to either server. We also show limits and improved constructions of maximally recoverable local reconstruction codes which are locally correctable codes designed specifically for distributed data storage applications.

Acknowledgements

This section was the hardest for me to write. I am indebted to a great many people, and these few pages wouldn't be sufficient to express my gratitude to all.

I was extremely fortunate to have Zeev as my advisor and his guidance was valuable throughout my PhD. He introduced me to some beautiful questions in coding theory and gave me lots of freedom to work on whatever interested me. He encouraged me to be brave and work on important problems, notwithstanding their difficulty. I was always amazed by the novelty of his ideas. It was an absolute pleasure working with him. This work is supported by NSF CAREER award 1451191 and NSF grant CCF-1523816.

I would like to thank Swastik Kopparty, Ran Raz, Avi Wigderson and Mark Zhandry for agreeing to be on my committee despite their busy schedules. I would be remiss if I did not thank my collaborators Arnab Bhattacharyya, Jop Briët, Venkatesan Guruswami, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, Shubhangi Saraf, Avishay Tal and Sergey Yekhanin from whom I have learnt a lot and who contributed to the results in this thesis.

I want to convey my special thanks to Sergey for being an excellent mentor during my internship at Microsoft Research Redmond in the Fall of 2016 and again for hosting me in the summer of 2017 for a few weeks. During this time, I had the wonderful opportunity to collaborate with Venkatesan Guruswami and I am grateful for the guidance and support they gave me.

The theory community at Princeton has been a great source of inspiration for me. The courses I attended here by Sanjeev Arora, Manjul Bhargava, Mark Braverman, Assaf Naor and Ran Raz have influenced my thinking greatly. Avi Wigderson and the CSDM seminar at IAS have also been instrumental in creating a great atmosphere at Princeton for theoretical computer science.

During my the course of my PhD at Princeton, I have had several friends and colleagues who have been a constant support and made my life here so much more enjoyable. This is a shout out to all of them, particularly Eshan Chattopadhyay, Ankit Garg, Sumegha Garg, Suryateja Gavva, Sravya Jangareddy, Siddhartha Jayanti, Young Kun Ko, Shujing Liu, Jieming Mao, Yonatan Naamad, Huy Le Nguyen, Andrej Risteski and Omri Weinstein. I want to specially thank my roommates Naman Agarwal, Rafael Oliveira and Jon Schneider for making my stay at Princeton so much more fun.

I thank all my teachers from school and college who were instrumental in shaping my interests and thinking. I would like to thank the professors of my undergraduate college IIT Bombay and specifically my undergraduate thesis advisors Prahladh Harsha and Srikanth Srinivasan who inspired me to work in theoretical computer science. I owe a lot to my school in Kakinada where I started to develop an interest in mathematics and sciences; the time I spent solving challenging high school math problems was as exciting as any research problem that I later worked on. I want to thank a few teachers from my schooling specifically who have greatly contributed to my learning: Ashwath Rao, Babji, Ramakrishna, Shivakumar, Solomon, Subramanian GN and Suman. The friends I made over all these years have influenced me in many ways and I am thankful for all of them. My family has been a constant source of love and support for me, and my greatest strength; the time I spend with them is one of the happiest times for me. My parents always encouraged scientific curiosity ever since we were kids. I want to specially thank my elder brother Srikanth who has always been there for me ever since I was a kid, I still remember him trying hard to teach me Pythagoras theorem when I was 12. I am extremely fortunate to have a great family, teachers and friends who made me what I am today. It is to them that I dedicate my thesis.

To my family, teachers and friends

असतो मा सद्गमय ।

Asatō mā sadgamaya

तमसो मा ज्योतिर्गमय ।

tamasō mā jyōtirgamaya

- Br̥hadāraṇyaka Upaniṣad (900 - 600 BCE)

From untruth lead me to truth

From darkness lead me to light

Contents

Chapter 1

Introduction

Coding theory is the study of error correcting codes(ECCs) which help us build systems which are resilient to noise and corruptions. They are the reason why we can communicate through noisy channels, why we can store data reliably in faulty storage systems and why quantum computation is even feasible. ECCs were formally introduced in the pioneering work of Shannon [Sha48], who showed the existence of optimal (capacity-achieving) ECCs using the probabilistic method. The problem of constructing explicit and efficient codes has fueled the development of coding theory ever since.

An error correcting code encodes messages into longer strings called codewords so that given a corrupted codeword one can correct the corruptions and decode the encoded message. In many applications, it is enough to correct only one symbol of the corrupted codeword or to decode only one symbol of the message (or a tiny part). In such applications, it is desirable that these tasks are done extremely fast, sublinear or perhaps even in constant time; for example in distributed storage, to recover a single crashed server, we shouldn't be reading all the rest of the servers. This means that we are only allowed to read a tiny part of the corrupted codeword, this is referred to as "locality". The number of coordinates we are allowed to read[1] is

[1] They can be chosen using a randomized procedure

called query complexity. Studying codes with such local procedures has turned out to be an extremely useful and productive area of coding theory. They have applications in distributed storage, probabilistically checkable proofs (PCPs), program checking, private information retrieval, hardness amplification, cryptography etc.. The main question for local codes is to understand the best possible rate (i.e. what's smallest codeword length for a given message length) for a given query complexity. But unfortunately, we are still very far from understanding such codes. They will the be main focus of this thesis.

1.1 Codes with locality

A locally decodable code (LDC) is an error correcting code which allows for ultra fast decoding of a single message symbol by reading only a tiny part of the corrupted codeword. A locally correctable code (LCC) has a stronger property, it allows for ultra fast correction of any codeword symbol by reading only a tiny part of the corrupted codeword. The local decoding and local corrections algorithms work well only when there are not too many corruptions, otherwise they behave unpredictably. So sometimes we want to test quickly if a given corrupted codeword has too many corruptions before we can use the local decoding or correction algorithms. A code which supports a fast testing algorithm which read only a tiny part a given corrupted codeword is called a locally testable code (LTC). We will now define them a bit more formally.

An error correcting code is a map $C : \{0,1\}^k \to \Sigma^n$ which encodes k-bit message into codewords of length n over some finite alphabet Σ. Given some string z which is close enough to some codeword $C(x)$ in Hamming distance (some small constant fraction of errors), we want to recover the codeword $C(x)$ and thus the message x which is encoded. There are three natural algorithmic tasks here:

- Correction: Correct the errors in z to get $C(x)$

- Decoding: Decode z to get x

- Testing: Test whether z is actually close to some codeword.

Locally Decodable Code: C is called a q-query LDC if there is a randomized local decoding algorithm which given some $i \in [k]$, reads at most q locations of z (which can be chosen randomly) and outputs x_i with good probability.

Locally Correctable Code: C is called a q-query LCC if there is a randomized local correction algorithm which given some $i \in [n]$, reads at most q locations of z and outputs $C(x)_i$ with good probability.

Locally Testable Code: C is called a q-query LTC if there is a randomized local testing algorithm which reads at most q locations of z and accepts if there are no corruptions, and rejects with good probability if there are too many corruptions.

LDCs are weaker than LCCs, in the sense that any LCC can be converted into an LDC while preserving relevant parameters (see Section 2.4.1 for a formal statement and proof).

An example - Hadamard Code To get familiar with the definitions, let us look at the example of Hadamard code which is simultaneously a 2-query LDC, 2-query LCC and 3-query LTC!

The Hadamard code is a exponential length linear code, $H : \mathbb{F}_2^k \to \mathbb{F}_2^n$ where $n = 2^k$. The codeword coordinates are indexed by $y \in \mathbb{F}_2^k$ and for a message $x \in \mathbb{F}_2^k$, the encoding is given by $H(x)_y = \langle x, y \rangle$ i.e. the codewords are just evaluations of linear functions on $\mathbb{F}_2^k$. This is a 2-query LCC. Suppose we are given a corrupted version of a codeword $H(x)$, say $\widetilde{H(x)}$. To correct the symbol at $y \in \mathbb{F}_2^n$, the local corrector queries $\widetilde{H(x)}$ at $z, z + y$ for a uniformly random $z \in \mathbb{F}_2^k$ and computes the parity of the two bits. If the number of corruptions are small, with good probability both the queries land in the uncorrupted part of $\widetilde{H(x)}$, and if that's the case,

$$\widetilde{H(x)}_z + \widetilde{H(x)}_{z+y} = \langle x, z\rangle + \langle x, z+y\rangle = \langle x, y\rangle$$

which is the correct symbol. Since the message symbols are part of the codeword ($H(x)_{e_i} = \langle x, e_i\rangle = x_i$), it is also a 2-query LDC.

To test if some given word is close to some codeword is equivalent to testing if a function $f : \mathbb{F}_2^k \to \mathbb{F}_2$ is linear. To test this, a local tester can sample $z, y \in \mathbb{F}_2^k$ and query f at $z, y, z+y$ and accept if

$$f(z+y) = f(z) + f(y).$$

This is the famous linearity test of Blum, Luby and Rubinfeld [BLR93]. It clearly accepts a linear function and it will reject a function which is far from linear with good probability. Thus H is a 3-query LTC.

1.1.1 History and applications of local codes

The notion of locality has a long history in computer science by now. 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. LDCs were first formally defined in the context of channel coding in [KT00], although they (and LCCs) implicitly appeared in several previous works in other settings (in particular, local codes based on multivariate polynomials), such as probabilistically checkable proofs (PCPs) [AS98, ALM$^+$98], private information retrieval (PIR) schemes [CGKS98] and proof checking [BFLS91]. Since then, they found many more applications, here is a short list of applications of LCCs/LDCs where are useful or appear implicitly.

- Probabilistically Checkable Proofs (PCPs) [AS98, ALM$^+$98]
- Private information retrieval (PIR) schemes [CGKS98, GKST06, BIW07]
- Hardness amplification [STV01, Vad12]
- Multiparty secure computation [IK04]
- Polynomial identity testing (PIT) [DS07]
- Matrix rigidity [Dvi11]
- Time-space tradeoffs for Nearest Neighbor search [ALRW17]
- Secret sharing [LVW17, LVW18]
- Banach space geometry [BNR12]
- Additive combinatorics [BDG17, BG17b]
- Quantum complexity theory [Aar18]

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. [BDYW11, DSW14b, DSW14a]. LDC-inspired objects called local reconstruction codes found applications in fault tolerant distributed storage systems [GHSY12]. See [Yek12] for a survey on LDCs, LCCs and some of their applications. Also see the survey [Vad12] for applications of coding theory and locality to pseudorandomness.

Research on LTCs implicitly started with Blum, Luby, and Rubinfeld's seminal discovery [BLR93] that the Hadamard code is an LTC with query complexity 3 and there was lot of work on testing low degree multivariate polynomials [AS03]. LTCs were first formally defined by Goldreich and Sudan in [GS06a]. They have been used (implicitly and explicitly) in many contexts, most notably in the construction of PCPs [AS98, ALM$^+$98, Din07].

The idea of local decoding has also been extremely useful in the context of distributed storage. But in practice, the codes deployed need to be very efficient (rate close to one) and need local decoding with a constant number of queries, which is impossible [KT00]. Thus the notion of LDCs is relaxed to the setting where the number of errors are constant. These codes are called Local Reconstruction Codes (LRCs) and introduced by Gopalan, Huang, Simitci and Yekhanin [GHSY12]. They are already being deployed in practice, outperforming traditional codes like Reed-Solomon codes. Maximally recoverable LRCs achieve the maximal reliability for a given rate and locality. The main questions about maximally recoverable LRCs is the field size required to construct them. The performance of encoding and decoding algorithms in practice is extremely sensitive to the field size.

1.1.2 What is the cost of locality?

Despite their many applications, our knowledge of LDCs/LCCs is very limited; the best-known constructions are far from what is currently known about their limits. A random binary linear code achieves the Gilbert-Varshamov bound, the best rate-distance trade-offs known for binary codes. For many pseudorandom objects like expanders, extractors etc.., random constructions achieve optimal parameters. But the situation is very different for local codes. Although standard random (linear) ECCs do allow for some weak local-decodability, they are outperformed by even the earliest explicit constructions [KS07]. All the known constructions of LDCs/LCCs were obtained by explicitly designing such codes using some algebraic objects like low-degree polynomials or matching vectors [Yek12].

2-query LDCs over the binary alphabet need exponential length [KW04, GKST06] and the Hadamard code achieves this i.e. they should encode k bits to $n = \exp(\Omega(k))$

bits. But there is a huge gap between upper and lower bounds for q-query LDCs for $q \geq 3$. The best lower bounds are polynomial [KT00, KW04] ($n = k^{\Omega_q(1)}$) and the best upper bounds are sub-exponential [Yek08, Efr09] ($n = \exp(k^{o(1)})$). Even for 2-query LDCs over large alphabet, there are huge gaps between the best upper and lower bounds, this is closely related to the communication complexity of private information retrieval schemes.

Surprisingly, lower bounds on LDCs have found several applications in areas like Banach space theory [BNR12],time-space tradeoffs for data structures [ALRW17], additive combinatorics and probability theory [BDG17, BG17b]. This shows that studying the limitations of locally decodable codes is a very fruitful area of research and a central problem with connections to several areas of mathematics. Indeed there are several works which study lower bounds for constant query LDCs/LDCs [KT00, GKST06, DS07, KW04, BDYW11, BDSS11, Woo12, DSW14a] Yet, the upper and lower bounds are far apart.

The situation is much better for LTCs. We know the existence of 3-query LTCs which encode k bits to $n = k \cdot \text{polylog}(k)$ bits [BS08, Din07, Vid15]. The main open question about LTCs is whether there are constant query LTCs with constant rate and distance. Understanding the cost of locality is one of the central problems in coding theory.

1.2 Summary of contributions

The following is a brief summary of the main contributions of this thesis. A more detailed description of results in each chapter and the organization of the thesis is given in Section 2.6.

Locally decodable codes (LDCs) and Locally correctable codes (LCCs) In Chapter 5, we show an equivalence between LDCs and outlaw distributions[2] over smooth functions showing an interesting connection to probability theory. For this, we also develop an average-case to worst-case reduction for LDCs (see Section 2.3.2) i.e. we can convert an LDC which can decode a random coordinate of a random message into an LDC which has worst-case guarantees.

For constant q, all the known constructions for q-query LCCs come from a class of codes called affine-invariant codes[3] which have a rich set of symmetries that makes them amenable to local correction. In Chapter 6, using tools from higher order Fourier analysis (specifically the inverse Gowers theorem [TZ12]), we show that Reed-Muller codes are optimal q-query LCCs among the class of affine-invariant codes, extending [BSS11] who showed it for linear affine-invariant codes.

2-query LCCs of small length over a small (but growing) alphabet can also be used to construct 2-server PIR protocols. In Chapter 7, we show a tight lower bound on the length of 2-query LCCs[4] over growing alphabet, improving [BDSS16]. Our bounds imply that 2-query LCCs cannot be used to improve current PIR protocols.

Gilbert-Varshamov (GV) bound is the best known rate-distance tradeoff known for binary error correcting codes. In Chapter 4, we show the existence of codes which admit local correction and codes with local testing which almost lie on the GV rate-distance curve. Thus we can match the best rate-distance tradeoffs known for binary codes and still have non-trivial (sub-linear) locality!

Private information retrieval (PIR) Information privacy is a problem of great importance as our lives are increasingly getting connected to Internet. Search engines

[2] These are distributions on functions over a small domain such that, to get close to the true mean of the distribution in L_∞-norm, we need to take the empirical mean of a large number of samples. The term outlaw is a reference to the Law of Large Numbers.

[3] These are codes whose coordinates are indexed by a vector space and the set of codewords is invariant under affine linear transformations of the coordinate space.

[4] Our bound only holds if the corrector doesn't make any errors when the codeword is not corrupted.

can know more things about you from your search queries than your friends. Private information retrieval allows a user to retrieve an entry from a remote database of k entries while not revealing any information about which entry the user wants. To get information theoretic privacy, the user cannot do better than asking for the entire database which requires k bits of communication. Surprisingly, one can do much better when the database is replicated among two non-communicating servers (2-server PIR). In Chapter 3, we construct a 2-server PIR scheme with $k^{o(1)}$ bits of communication improving the $O(k^{1/3})$-scheme of [CGKS98] for the first time and overcoming a barrier result due to Razborov and Yekhanin [RY06]. It is known that good q-query LDCs imply good q-server PIR schemes in a blackbox way [KT00]. There are no good 2-query LDCs unfortunately, but we show a way to convert subexponential q-query LDCs from [Yek08, Efr09] for $q \geq 3$ in a non-blackbox way into good $\lceil q/2 \rceil$-server PIR protocols to achieve our result. The best known lower bound on the communication cost is only $5 \log k$ [WdW05a], so we are still very far! In subsequent work, our new 2-server PIR scheme has found applications in cryptography and lead to improved schemes for conditional disclosure of secrets and secret sharing [LVW17, LVW18].

Applications to additive combinatorics A subset $D \subset \mathbb{N}$ is ℓ-intersective if every dense subset of $\mathbb{N}$ contains a non-trivial arithmetic progression of length ℓ. Szemerédi proved that $\mathbb{N}$ is ℓ-intersective for every ℓ. But much smaller sets can be ℓ-intersective; for example perfect powers $\{1^t, 2^t, 3^t, \cdots\}$, shifted primes $\{p + 1 : p \text{ is prime}\}$. It is natural to ask at what density random sets become ℓ-intersective. In Chapter 8, we use techniques used to prove LDC lower bounds to improve the bounds on the density of random sets which are ℓ-intersective and some large deviation estimates on the number of arithmetic progressions in a random set.

Codes for distributed storage In massive data centers, data is stored among different servers which often crash or fail to respond. To deal with this,, data is encoded using codes which are resilient to erasures. When one (or a few) server crashes, we want a fast local correction procedure which reads only a few other servers (like LCCs), but with rates matching the best traditional codes. Maximally recoverable local reconstruction codes (MR LRCs) are such a class of codes, which have good local correction in the typical scenario when a few servers crash, but still protect the data from a large number of crashes. They have already been deployed by the Windows Server system with significantly better space and time performance over the traditional distributed storage systems like RAID which uses Reed-Solomon codes. It is easy to show the existence of MR LRCs over fields of size exponential in n, the length of the code. But working over large fields makes encoding and decoding extremely slow in practice, ideally we need the field size to be linear in n and to have characteristic 2. Constructing MR LRCs over fields of small size has been a challenging open problem.

In Chapter 9, we proved the first polynomially growing (in n) lower bounds on the field size using vertex expansion in point-hyperplane incidence graph. Prior to our work, no super linear lower bounds were known. We also showed good constructions in some practically relevant parameter range and some connections to elliptic curves and arithmetic progression free subsets of integers.

1.3 Future directions

We are still very far from understanding locality in codes. I believe that the techniques developed to study locality in codes will be useful in a wider range of problems which involve locality like sublinear algorithms, sketching, dynamic data structures, property testing, PCPs etc.. Local codes also have some surprising connections to Approximate nearest neighbor search [ALRW17], Banach space theory [BNR12], additive combinatorics and probability theory [BDG17, BG17b]. I intend to continue thinking about these questions.

In the heart of our 2-server PIR scheme in Chapter 3 are objects called matching vector families (MVFs) introduced by Grolmusz [Gro99] for the purpose of constructing explicit pseudorandom objects called Ramsey graphs. Grolmusz's construction of MVFs is based on the representation of the OR function by a $O(\sqrt{k})$-degree polynomial over $\mathbb{Z}/6\mathbb{Z}$. Lowering this degree will lead to better MVFs and thus better PIR schemes, but currently the best lower bound we know on the degree of such a polynomial is only $\Omega(\log k)$ [TB98]. There wasn't much progress on this problem for a long time much like circuit lower bounds for AC0 circuits with mod 6 gates. There also seems to be an interesting connection of low-degree representations of OR function to submodular optimization with modular constraints [NSZ18].

Though exponential lower bounds are known for 2-query LDCs over constant alphabet, we only know extremely weak lower bounds when the alphabet is growing. This is closely related to communication cost of 2-server PIR schemes. It has been shown recently that LDC lower bounds over large alphabet would also lead to progress on time-space tradeoffs for approximate nearest neighbor search in the cell-probe model [ALRW17]. This raises an intriguing question whether LDCs can be used to design non-trivial data structures for approximate nearest neighbor search.

There is an interesting connection between LDCs and type (cotype) constants of certain Banach spaces formed as injective (projective) tensor products of ℓ_p spaces[5]. These constants are related to understanding the norm of a randomly signed sum of k tensors i.e. $\mathbb{E}_{\varepsilon\in\{-1,1\}^k}||\sum_i \varepsilon_i X_i||$. Improving these constants would yield tensor concentration inequalities analogous to matrix concentration inequalities and would have an amazing range of applications in LDC lower bounds, additive combinatorics, probability theory and machine learning. Conversely, the techniques used to prove

[5] For example, in [BNR12], the existence of subexponential 3-query LDCs was shown to imply the failure of cotype for projective tensor products.

LDC lower bounds are often useful to prove such tensor concentration inequalities. See Section 2.3.6 and Section 8.1.4 for more details.

Coding theory has been great source of problems which are both mathematically beautiful and have immediate practical applications. Local reconstruction codes (LRCs) [GHSY12] for distributed storage are a good example. As alternative models of storage and computation like DNA storage [OAC$^+$17] and quantum computation are invented, we would need to design codes which would make these systems robust to noise. I believe coding theory would continue to be a rich source of problems with both theoretical and practical significance.

Chapter 2

Preliminaries

2.1 Notation

We will write $A \lesssim B$ or $A = O(B)$ to denote that $A \leq cB$ for some absolute constant $c > 0$ independent of all the parameters involved, $A \gtrsim B$ or $A = \Omega(B)$ is similarly defined. A subscript containing some parameters is used when the constants depend only on those parameters and independent of other parameters. For example, $A \lesssim_\delta n$ or $A = O_\delta(B)$ is used to denote that $A \leq c(\delta)B$ for some constant $c(\delta) > 0$ that depends only on δ but independent of other parameters.

2.2 Error Correcting Codes

Let Σ be some finite alphabet. The relative distance (Hamming distance) between two strings $x, y \in \Sigma^n$ is defined as:

$$\mathrm{dist}_H(x, y) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x_i \neq y_i)$$

which is the fraction of coordinates where x, y differ. For $S \subset \Sigma^n$, $\mathrm{dist}_H(x, S)$ is defined as $\min_{y \in C} \mathrm{dist}_H(x, y)$.

A subset $C \subset \Sigma^n$ is called an error correcting code with the following associated parameters:

- Minimum distance: Largest δ such that for every two distinct $x, y \in C$, $\mathrm{dist}_H(x, y) \geq \delta$
- Rate: $\frac{\log |C|}{n \log |\Sigma|}$
- Alphabet size: $|\Sigma|$

The elements of C are called codewords. The rate is the amount of information that a codeword contains per bit. The error-correction property of codes comes from the simple but important observation that there is at most one codeword within distance $\delta/2$ of any given word $w \in \Sigma^n$; finding this codeword given w is the problem of decoding. Let Γ be some finite alphabet. An injective map $C : \Gamma^k \to \Sigma^n$ is referred to as an encoding map and the image of the map, $C(\Gamma^k)$, is the associated error correcting code. A string $x \in \Gamma^k$ is called a message and $C(x)$ is called its encoding. The distance of a string $z \in \Sigma^n$ to the code C is denoted by $\mathrm{dist}_H(z, C)$ which is equal to $\mathrm{dist}_H(z, C(\Gamma^k))$.

Linear Codes If $\Sigma = \mathbb{F}$ for some finite field $\mathbb{F}$ and C is a linear subspace of $\mathbb{F}^n$, then C is called a linear code. Note that in a linear code, the minimum distance is the smallest weight of a non-zero codeword normalized by n. A linear code can be represented by an injective linear map $C : \mathbb{F}^k \to \mathbb{F}^n$ such that the code is the image of the map. A linear code can always be made systematic by a linear change of basis and permuting the coordinates i.e. the first k coordinates of the encoding can be made equal to the message.

2.3 Locally Decodable Codes (LDCs)

Locally Decodable Codes allow decoding of a individual message coordinates by reading only a small number of coordinates of a corrupted codeword. We will define them formally here:

Definition 2.3.1 (Locally decodable code (LDC))**.** *Let Σ be some finite alphabet. For positive integers k, n, q and parameters $\eta, \delta > 0$, a map $C : \{0,1\}^k \to \Sigma^n$ is a (q, δ, η)-locally decodable code if, for every $i \in [k]$, there exists a randomized decoder (a probabilistic algorithm) $\mathcal{A}_i$ such that:*

- *For every message $x \in \{0,1\}^k$ and $y \in \Sigma^n$ such that $\mathrm{dist}_H(C(x), y) \leq \delta$,*

$$\Pr[\mathcal{A}_i(y) = x_i] \geq \Pr[\mathcal{A}_i(y) \neq x_i] + \eta. \tag{2.1}$$

- *The decoder $\mathcal{A}_i$ queries non-adaptively at most q coordinates of y.*

When the parameter η is not mentioned, usually it is assumed that it is some fixed constant. We can assume that on input $y \in \Sigma^n$, the decoder $\mathcal{A}_i$ first samples a q-tuple of coordinates $(j_1, \ldots, j_q)$ from $[n]$ according to a probability distribution $\mathcal{D}^i$ depending on i alone and then returns a random bit depending only on i, $(j_1, \ldots, j_q)$ and the values of $y_{j_1}, \ldots, y_{j_q}$. When the message bits are represented by $\{-1, 1\}$ instead of $\{0, 1\}$, the decoding condition can be expressed alternatively as follows:

$$\mathbb{E}_{(j_1,\ldots,j_q)\sim\mathcal{D}^i}[x_i D^i_{j_1,\ldots,j_q}\left(y_{j_1}, \ldots, y_{j_q}\right)] \geq \eta$$

where $D^i_{j_1,\ldots,j_q} : \Sigma^q \to [-1, 1]$ are some decoding functions.

We could also define LDCs which encode messages over a larger alphabet, say Γ. A map $C : \Gamma^k \to \Sigma^n$ is a (q, δ, η)-*locally decodable code* if the randomized decoder $\mathcal{A}_i$ makes at most q queries and satisfies the following: For every message $x \in \Gamma^k$ and string $y \in \Sigma^n$ that differs from the codeword $C(x)$ in at most δn coordinates,

$$\Pr[\mathcal{A}_i(y) = x_i] \geq \Pr[\mathcal{A}_i(y) = \gamma] + \eta, \tag{2.2}$$

for any $\gamma \in \Gamma$ such that $\gamma \neq x_i$. Basically this condition enables one to amplify the successful decoding probability by repeating the decoder and taking plurality vote.

Though we didn't specify that an LDC should have large minimum distance, it can be inferred from the local decoding condition.

Lemma 2.3.2. *Let $\mathcal{C} : \{0,1\}^k \to \Sigma^n$ be an (q, δ, η) LDC, then the minimum distance of $\mathcal{C}$ is at least 2δ.*

Proof. Let $x, y \in \{0,1\}^k$ be two distinct messages such that there corresponding codewords are less than 2δ apart in relative distance i.e. $\text{dist}_H(C(x), C(y)) < 2\delta$. Let $z \in \Sigma^n$ be the (approximate) midpoint of $C(x)$ and $C(y)$, i.e. z is δ-close to both $C(x)$ and $C(y)$. Let $i \in [k]$ be such that $x_i \neq y_i$. By the LDC property,

$$\Pr[x_i = \mathcal{A}_i(z)] \geq \frac{1}{2} + \frac{\eta}{2},$$

$$\Pr[y_i = \mathcal{A}_i(z)] \geq \frac{1}{2} + \frac{\eta}{2}.$$

This is a contradiction since $x_i \neq y_i$. Therefore every two codewords must be at least 2δ apart. □

Adaptive vs Non-adaptive

One can also define adaptive q-query LDCs where the local decoder can make adaptive queries i.e. it can query a new location based on what it has seen in previous locations. But throughout this thesis, we will only consider non-adaptive LDCs. Most constructions we know are non-adaptive and moreover, any q-query adaptive LDC is also a $\big((|\Sigma|^q - 1)/(|\Sigma| - 1)\big)$-query non-adaptive LDC where the non-adaptive decoder just queries all the possible queries that the adaptive decoder can make, see [KT00] for more discussion on this.

2.3.1 Smoothness

Katz and Trevisan [KT00] observed that LDC decoders must have the property that they select their queries according to distributions that do not favor any particular coordinate. The intuition for this is that if they did favor a certain coordinate, then corrupting that coordinate would cause the decoder to err with too high a probability. If instead, queries are sampled according to a "smooth" distribution, they will all fall on uncorrupted coordinates with good probability provided the fraction of corrupted coordinates δ and query complexity q aren't too large. Note that, we can always assume that the marginal distribution of each query is identical. This is because the decoder can always uniformly permute the queries before making them. The following definitions allows us to make this intuition precise.

Definition 2.3.3 (Smooth distribution)**.** *A distribution $\mathcal{D}$ over $[n]$ is called c-smooth if for every $i \in [n]$, $\Pr_{\mathcal{D}}[i] \le \frac{c}{n}$.*

Definition 2.3.4 (Smooth LDC)**.** *Let Σ be some finite alphabet. For positive integers k, n, q and parameters $\eta, c > 0$, a map $C : \{0,1\}^k \to \Sigma^n$ is a (q, c, η)-smooth code if, for every $i \in [k]$, there exists a randomized decoder $\mathcal{A}_i$ such that*

1. *For every $x \in \{0,1\}^k$,*

$$\Pr\left[x_i = \mathcal{A}_i\Big(C(x)\Big)\right] \ge \Pr\left[x_i \neq \mathcal{A}_i\Big(C(x)\Big)\right] + \eta. \tag{2.3}$$

2. *The decoder $\mathcal{A}_i$ (non-adaptively) queries at most q coordinates of $C(x)$.*

3. *The distribution of each query that $\mathcal{A}_i$ makes is c-smooth (as defined in Definition 2.3.3).*

When the parameter η is not explicity mentioned, usually it is assumed to be some fixed absolute constant. A $(q, 1, \eta)$-smooth LDC is called a perfectly smooth LDC. In a perfectly smooth LDC, the marginal distribution of each query that the decoder makes is uniform over all the coordinates. The following lemma from [KT00] shows that LDCs and smooth LDCs are closely related.

Proposition 2.3.5 ([KT00])**.** *If $C : \{0,1\}^k \to \Sigma^n$ is a (q, δ, η)-LDC, then C is also a $(q, 1/\delta, \eta)$-smooth LDC. Conversely, if $C : \{0,1\}^k \to \Sigma^n$ is a (q, c, η)-smooth code, then C is also a $(q, \delta, \eta - 2qc\delta)$-LDC.*

2.3.2 An average-case to worst-case reduction

In this section, we will prove an average-case to worst-case reduction for smooth LDCs i.e. smooth LDCs that are only required to work *on average* (for a random message, to decode a random bit) can be turned into smooth LDCs that can decode every bit of every message, losing only a constant factor in the rate and success probability.

Definition 2.3.6 (Average-case smooth code)**.** *A code as in Definition 2.3.4 is a (q, c, η)-*average-case *smooth code if instead of the first item,* (2.3) *is required to hold only on average over uniformly distributed $x \in \{0,1\}^k$ and uniformly distributed $i \in [k]$, which is to say that*

$$\Pr\left[x_i = \mathcal{A}_i\big(C(x)\big)\right] \geq \Pr\left[x_i \neq \mathcal{A}_i\big(C(x)\big)\right] + \eta,$$

where the probability is taken over x, i and the randomness used by $\mathcal{A}_i$.

The following theorem is from [BDG17] where we used it construct LDCs from outlaw distributions.

Theorem 2.3.7 ([BDG17])**.** *Let $C : \{0,1\}^k \to \{0,1\}^n$ be a (q, c, η)-average-case smooth code. Then, there exists an $(q, c, \Omega(\eta))$-smooth LDC sending $\{0,1\}^\ell$ to $\{0,1\}^n$*

where

$$\ell \gtrsim \eta^2 k / \log(1/\eta).$$

The idea behind the proof of Theorem 2.3.7 is as follows. We first switch the message and codeword alphabets to $\{-1,1\}$ and let $f_i : \{-1,1\}^k \to [-1,1]$ be the expected decoding function $f_i(z) = \mathbb{E}[\mathcal{A}_i(z)]$. The properties of C then easily imply that the set $T \subseteq [-1,1]^k$ given by $T = \{(f_1(z), \ldots, f_k(z)) : z \in \{-1,1\}^k\}$ has large *Gaussian width*, in particular it holds that for a standard k-dimensional Gaussian vector g, we have $\mathbb{E}[\sup_{t\in T}\langle g, t\rangle] \gtrsim \varepsilon k$. Next, we employ a powerful result of [MV03] showing that T contains an l-dimensional hypercube-like structure with edge length some absolute constant $c \in (0,1]$, for $l \gtrsim k$. Roughly speaking, this implies that C is a smooth code on $\{-1,1\}^l$ whose decoding probability depends on ε and c. Note that we can convert the smooth LDC obtained into an LDC using Proposition 2.3.5.

Proof of Theorem 2.3.7

To prove Theorem 2.3.7, we need the notion of the Vapnik–Chervonenkis dimension (VC-dimension).

Definition 2.3.8 (VC-dimension)**.** *For $T \subset [-1,1]^k$ and $w > 0$, $\mathrm{vc}(T, w)$ is defined as the size of the largest subset $\sigma \subset [k]$ such that there exists a shift $s \in [-1,1]^k$ satisfying the following: for every $x \in \{-1,1\}^\sigma$, there exists $t \in T$ such that for every $i \in \sigma$, $(t_i - s_i)x_i \geq w/2$.*

Observe that if T is convex, then $\mathrm{vc}(T, w)$ is the maximum dimension of a shifted hypercube with edge lengths at least w contained in T .

Definition 2.3.9 (Gaussian width)**.** *Let g be a k-dimensional standard Gaussian vector, with independent standard normal distributed entries. The* Gaussian width *of a set $T \subseteq \mathbb{R}^k$ is defined as*

$$E(T) = \mathbb{E}_g[\sup_{t\in T}\langle g, t\rangle].$$

It is easy to see that a large VC-dimension implies a large Gaussian width. The following theorem shows the converse: containing a hypercube-like structure is the only way to have large Gaussian width.

Theorem 2.3.10 ([MV03]). *Let $T \subset [-1,1]^k$. Then, the Gaussian width of T is bounded as*

$$E(T) \lesssim \sqrt{k} \int_{\alpha E(T)/k}^{1} \sqrt{\mathrm{vc}(T,w)\log(1/w)}dw$$

for some absolute constant $\alpha > 0$.

Finally, we use that fact that, as for LDCs, we can assume that on input $y \in \{0,1\}^n$, the decoder $\mathcal{A}_i$ of a smooth code first samples a set $S \subseteq [n]$ of at most q coordinates according to a probability distribution that depends on i only and then returns a random sign depending only on i, S and the values of y at S.

Proof of Theorem 2.3.7. The proof works by showing that the average-case smooth code property implies that the image of the (average) decoding functions should have large Gaussian width. We then use Theorem 2.3.10 to find a hypercube like structure inside the image, which we use to construct a smooth code.

Recall the switch of the message and codeword alphabets to $\{-1,1\}$. For each $i \in [k]$, let $f_i : \{-1,1\}^n \to [-1,1]$ be the expected decoding function $f_i(z) = \mathbb{E}[\mathcal{A}_i(z)]$. Let g be a standard k-dimensional Gaussian vector and $T = \{(f_1(z), \dots, f_k(z)) : z \in \{-1,1\}^n\}$. By the definition of average-case smooth code we have

$$\eta k \leq \mathbb{E}_{x\in\{-1,1\}^k}\left[\sum_{i=1}^{k} x_i f_i(C(x))\right] \leq \mathbb{E}_{x\in\{-1,1\}^k}\left[\sup_{t\in T}\langle x,t\rangle\right] \lesssim \mathbb{E}_g\left[\sup_{t\in T}\langle g,t\rangle\right].$$

(See for instance [Tal14a, Lemma 3.2.10] for the last inequality.) By Theorem 2.3.10, for some constant $\alpha > 0$, we have

$$\eta k \lesssim \sqrt{k}\int_{\alpha\eta}^{1}\sqrt{\mathrm{vc}(T,w)\log(1/w)}dt \leq \sqrt{k}\cdot\sqrt{\mathrm{vc}(T,\alpha\eta)\log(1/\alpha\eta)}$$

where we used the fact that $\mathrm{vc}(T, w)$ is decreasing in w. So for $\tau = \alpha\eta$, we have $\mathrm{vc}(T, \tau) \gtrsim \eta^2 k/\log(1/\eta)$. By the definition of VC-dimension, there exists a subset $\sigma \subset [k]$ of size $|\sigma| \geq \mathrm{vc}(T, \tau)$ and a shift $s \in [-1, 1]^k$ such that for every $x \in \{-1, 1\}^\sigma$ there exists $t \in T$ such that $(t_i - s_i)x_i \geq \tau/2$ for every $i \in \sigma$.

Now we will define the code $C' : \{-1, 1\}^\sigma \to \{-1, 1\}^n$. Given $x \in \{-1, 1\}^\sigma$, there exists $t(x) \in T$ such that $(t(x)_i - s_i)x_i \geq \tau/2$ for every $i \in \sigma$. Define $C'(x) \in \{-1, 1\}^n$ to be one of the preimages of $t(x)$ under f, that is,

$$(f_1(C'(x)), \ldots, f_k(C'(x))) = t(x).$$

Let W_p denote a $\{-1, 1\}$-valued random variable with mean p. The decoding algorithms $\mathcal{A}'_i(y)$ run $\mathcal{A}_i(y)$ internally and give their output as follows:

$$\mathcal{A}'_i(y) = \begin{cases} \text{Output } W_{(1-s_i)/2} & \text{if } \mathcal{A}_i(y) \text{ returns } 1 \\ \text{Output } -W_{(1+s_i)/2} & \text{if } \mathcal{A}_i(y) \text{ returns } -1 \end{cases}$$

Therefore, for every $x \in \{-1, 1\}^\sigma$ and for every $i \in \sigma$,

$$\begin{aligned} x_i \mathbb{E}[\mathcal{A}'_i(C'(x))] &= x_i \mathbb{E}\left[\frac{(1 + \mathcal{A}_i(C'(x)))}{2} W_{(1-s_i)/2} - \frac{(1 - \mathcal{A}_i(C'(x)))}{2} W_{(1+s_i)/2}\right] \\ &= \frac{x_i}{2} \mathbb{E}\left[\mathcal{A}_i(C'(x)) - s_i\right] \\ &= \frac{x_i}{2}(f_i(C'(x)) - s_i) \\ &= \frac{x_i}{2}(t(x)_i - s_i) \\ &\geq \frac{\tau}{4} \gtrsim \eta. \end{aligned}$$

Since the probability that $\mathcal{A}'_i(C'(x))$ queries any particular location of $C'(x)$ is still at most c/n, it follows that C' is a $(q, c, \Omega(\eta))$-smooth code. □

2.3.3 Constructions for LDCs

The earliest constructions of LDCs were the Hadamard code and its higher degree generalization, Reed-Muller codes. These are also locally correctable codes which is a stronger notion. We will define these in Section 2.4.2.

Definition 2.3.11 (Hadamard Code)**.** *The Hadamard code is the map* $H : \mathbb{F}_2^k \to \mathbb{F}_2^{\mathbb{F}_2^k}$ *defined as* $H(x)_y = \langle x, y \rangle$.

The Hadamard code is a $(2, 1, 1)$-perfectly smooth LDC of length $n = 2^k$. To decode x_i, the local decoder queries $H(x)$ at $z, z + e_i$ for a uniformly random $z \in \mathbb{F}_2^k$ and computes the parity of the two bits. Since

$$H(x)_z + H(x)_{z+e_i} = \langle x, z \rangle + \langle x, z + e_i \rangle = x_i,$$

and the marginal distribution of each query is uniform over $\mathbb{F}_2^k$, this is a perfectly smooth 2-query decoder for the Hadamard code.

When $q = 3$, the best constructions are from a family of codes called matching vector codes (MVCs).

Theorem 2.3.12 ([Yek08, Efr09])**.** *There exists a* $(3, 1, 1)$*-perfectly smooth LDC* $C : \{0, 1\}^k \to \{0, 1\}^n$ *of length*

$$n \leq \exp\left(\exp\left(O(\sqrt{\log n \log \log n})\right)\right).$$

Matching vector codes are the best known constructions for any constant q.

Theorem 2.3.13 ([Yek08, Efr09])**.** *There exists a* $(2^r, 1, 1)$*-perfectly smooth LDC* $C : \{0, 1\}^k \to \{0, 1\}^n$ *of length*

$$n \leq \exp\left(\exp\left(O_r((\log n \log \log n)^{1-1/r})\right)\right).$$

When q becomes logarithmic in n, Reed-Muller codes attain polynomial length i.e. there exists $(O(\log n), 1, 1)$-perfectly smooth LDCs of length $n = k^{O(1)}$. Increasing the queries further, there are $n^{o(1)}$-query LDCs with constant rate i.e. $n = O(k)$ [KSY14, KMRS17].

2.3.4 Lower bounds for constant query LDCs

In the paper where LDCs were first define [KT00], Katz and Trevisan also showed that constant query LDCs should stretch the message to super linear length.

Theorem 2.3.14 ([KT00])**.** *Let $C : \{0,1\}^k \to \Sigma^n$ be a (q, δ, η)-LDC. Then*

$$n \gtrsim_{\eta,\delta,q} \left(\frac{k}{|\Sigma|}\right)^{\frac{q}{q-1}}.$$

A similar lower bound was shown for *adaptive* q-query LDCs in [DJK+02]. The Katz-Trevisan lower bound was signficantly improved for 2-query LDCs where exponential lower bounds were shown [GKST06, KW04] i.e. $n \geq \exp(\Omega_{\delta,\eta}(n))$. By using a reduction to these exponential lower bounds, the lower bounds for q-query LDCs for $q \geq 3$ were improved as well.

Theorem 2.3.15 ([KW04])**.** *Let $C : \{0,1\}^k \to \{0,1\}^n$ be a (q, δ, η)-LDC for $q \geq 3$. Then*

$$n \gtrsim_{\eta,\delta,q} \left(\frac{k}{\log k}\right)^{1+\frac{1}{\lceil q/2\rceil - 1}}.$$

2.3.5 Exponential lower bound for two query LDCs

Exponential lower bounds are known for the length of 2-query LDCs over finite alphabet. Let $C : \{0,1\}^k \to \Sigma^n$ be a $(2, \delta, \eta)$-LDC. It was showin in [GKST06] using isoperimetric inequalities on the hypercube that if C is a linear code then

$$n \geq \exp\left(\Omega(\delta\eta k/|\Sigma|)\right).$$

This was extended to arbitrary 2-query LDCs by Kerenidis and de Wolf [KW04] with further improvements in [WdW05a] by using quantum information theory, they proved that

$$n \geq \exp\left(\Omega(\delta\eta^2 k/|\Sigma|^2)\right).$$

Suppose $\ell = \log|\Sigma|$ is the number of bits in each symbol of the alphabet, if the decoder only uses b bits of each queried position then an improved lower bound is obtained in [WdW05a],

$$n \geq \exp\left(\Omega\left(\frac{\delta\eta^2 k}{2^b \sum_{i=0}^{b}\binom{\ell}{b}}\right)\right).$$

When $\Sigma = \{0,1\}$, a exponential lower bound using matrix hypercontractive inequalities in [BARDW08] avoids the use of quantum information theory. But their proof gives a worse dependence on δ, η, they prove $n \geq \exp(\Omega(\delta^2\eta^4 n))$. A more direct proof using matrix concentraiton inequality (Proposition 2.3.18) is found by Pisier [Pis12]. A proof along these lines can be found in [Bri16] where they prove $n \geq \exp(\Omega(\delta^2\eta^2 n))$. Here we present a proof which follows the same strategy, but by a more careful reduction we get the same dependence on δ, η as obtained by the quantum information theory proof in [KW04]. The proof we present here is from [BG18]. The exponential lower bound in [KW04] was recently used in proving data structure trade-offs in the cell-probe model for two cell probes in [ALRW17]. The linear dependence on δ in the exponent is crucial for that application.

For the purpose of the lower bound, it is convenient to represent the binary alphabet by $\{-1,1\}$ instead of $\{0,1\}$.

Theorem 2.3.16. *Let $C : \{-1,1\}^k \to \{-1,1\}^n$ be a $(2,\delta,\eta)$-LDC, then*

$$n \geq \exp\left(\Omega(\delta\eta^2 k)\right).$$

Dependence on δ, η The optimal dependence on δ, η for linear LDCs is shown in [Oba02] and given by

$$n = \exp\left(\Theta(\delta k/(1-\eta))\right).$$

But for general LDCs, the optimal dependence on δ, η is not known. $(2, \delta, \eta)$-LDC of length

$$n = \text{poly}\left(\frac{1}{\delta\eta}\right) \exp\left(O(\max(\delta, \eta)\delta n)\right)$$

are constructed in [Woo08] by modifying Hadamard codes. A matching lower bound is shown in [Woo08] if the decoder has a particular structure called a 'matching sum decoder' in that paper. The proof of Theorem 2.3.16 only uses a weaker property of the decoders, that they work well on a random $x \in \{-1,1\}^k$. Codes with length $n = \exp\left(O(\delta\eta^2 k)\right)$, which satisfy this weaker property, can be constructed as follows. Partition the message of length k into groups of size $1/\eta^2$ and replace them with the majority of those bits to get $\eta^2 k$ bits. Now break this $\eta^2 k$ bits into $1/\delta$ parts and encode each part using a Hadamard code.

We will need the following lemma which shows that to work on an average codeword, the decoders can just sample their queries from a large matching. We will give a proof of this lemma later.

Lemma 2.3.17. *Let $C : \{-1,1\}^k \to \{-1,1\}^n$ be a $(2, \delta, \eta)$-LDC, then there exists local decoders for C as follows:*

1. *For some partial matchings $M_1, \dots, M_k$ on n vertices of size at least $\lfloor \delta n \rfloor$, the local decoder for decoding the i^{th} bit samples a random edge from M_i and queries the vertices of that edge.*

2. *The decoders can predict x_i with η advantage for a uniformly random message $x \in \{-1,1\}^k$, i.e.*

$$\mathbb{E}_x \mathbb{E}_{(j,k)\in M_i}\left[x_i D^i_{jk}(C(x)_j, C(x)_k)\right] \geq \eta$$

where $D^i_{jk} : \{-1,1\}^2 \to [-1,1]$ are some fixed decoding functions.

We will also need the notion of spectral norm of a matrix. Let A be an $n \times n$ matrix over the reals. The spectral norm of A denoted by $\|A\|_{S_\infty}$ is defined as:

$$\|A\|_{S_\infty} = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_{\ell_2}} = \sup_{x,y \neq 0} \frac{y^T A x}{\|y\|_{\ell_2} \|x\|_{\ell_2}}.$$

The spectral norm is also the largest singular value of the matrix A. Let $a_1, \dots, a_k \in \mathbb{R}$ and let $x \in \{-1,1\}^k$ be uniformly random, then

$$\mathbb{E}_x \left[\left| \sum_{i=1}^k x_i a_i \right| \right] \leq \sqrt{\sum_{i=1}^k a_i^2}.$$

The following proposition is the analogue of this fact for matrices first proved in [TJ74].

Proposition 2.3.18 (Tomczak-Jaegermann)**.** *Let $A_1, \cdots, A_k$ be $n \times n$ matrices over the reals, then*

$$\mathbb{E}_{x \in \{-1,1\}^k} \left[\left\| \sum_{i=1}^k x_i A_i \right\|_{S_\infty} \right] \lesssim \sqrt{\log n} \left(\sum_{i=1}^k \|A_i\|_{S_\infty}^2 \right)^{1/2}$$

where the expectation is over a uniformly random $x \in \{-1,1\}^k$.

See [Tro15, Theorem 4.1.1] for the statement above and [Tro15] for more on such matrix concentration inequalities.

Proof of Theorem 2.3.16. By applying Lemma 2.3.17, we get partial matchings $M_1, \dots, M_k$ on $[n]$ vertices of size $\lfloor \delta n \rfloor$ and decoding functions $D^i_{jk} : \{-1,1\}^2 \to [-1,1]$ such that:

$$\mathbb{E}_x \mathbb{E}_{(j,k) \in M_i} \left[x_i D^i_{jk}(C(x)_j, C(x)_k) \right] \geq \eta.$$

By incurring a constant factor loss in η, we can assume that the decoders just output the parity of the two bits they have queried or the negation (see [BARDW08] for a details) i.e.

$$D^i_{jk}(a,b) = s^i_{jk}ab$$

where $s^i_{jk} \in \{-1,1\}$ is such that $s^i_{jk} = \mathrm{sgn}(\mathbb{E}_x[x_i C(x)_j C(x)_y])$. Therefore we have:

$$\mathbb{E}_{(j,k)\in M_i}\left[s^i_{jk}\mathbb{E}_x[x_i C(x)_j C(x)_k]\right] \gtrsim \eta.$$

Let $t = 1/\delta$, we define $[n]^t \times [n]^t$ matrices $A_1, \ldots, A_k$ as follows: Let $(j_1, \ldots, j_t)$ be some row of A_i. Let $\ell \in [t]$ be the smallest such that j_ℓ participates in matching M_i. If there is no such ℓ, then that row is set to zero. Let $k \in [n]$ be the such that $(j_\ell, k) \in M_i$. Then the row has a single non-zero entry given by:

$$A_i\left((j_1,\ldots,j_t),(k_1,\ldots,k_t)\right) = s^i_{j_\ell,k_\ell} \text{ where } k_\ell = k \text{ and } k_{\ell'} = j_{\ell'} \text{ for } \ell' \neq \ell.$$

By symmetry each edge of M_i contributes equal number of times to A_i. Since $t = 1/\delta$ and $|M_i| = \lfloor \delta n \rfloor$, a random subset of $[n]$ of size t hits the matching M_i with constant probability. Therefore a constant fraction of rows of A_i are non-zero. Therefore:

$$\mathbb{E}_x[x_i \left\langle A_i, C(x)^{\otimes t} \otimes C(x)^{\otimes t}\right\rangle] \gtrsim n^t \mathbb{E}_{(j,k)\in M_i}\left[s^i_{j,k}\mathbb{E}_x[x_i C(x)_j C(x)_k]\right] \gtrsim \eta n^t.$$

Adding the above inequality for each $i \in [k]$, we get

$$\mathbb{E}_x\left[\left\langle \left(\sum_{i=1}^k x_i A_i\right), C(x)^{\otimes t} \otimes C(x)^{\otimes t}\right\rangle\right] \gtrsim \eta k n^t.$$

We can upper bound the LHS of the above inequality as

$$\begin{aligned} E_x\left[(C(x)^{\otimes t})^T \left(\sum_{i=1}^k x_i A_i\right) C(x)^{\otimes t}\right] &\leq \mathbb{E}_x\left[\left\|\sum_{i=1}^k x_i A_i\right\|_{S_\infty} \cdot \left\|C(x)^{\otimes t}\right\|_{\ell_2}^2\right] \\ &= n^t \cdot \mathbb{E}_x\left[\left\|\sum_{i=1}^n x_i A_i\right\|_{S_\infty}\right]. \end{aligned}$$

Since the matrix A_i is equivalent to a diagonal matrix with $\{-1,0,1\}$ entries after permuting rows and columns, it is easy to see that $\|A_i\|_{S_\infty} \leq 1$. So Lemma 2.3.18 implies that

$$\mathbb{E}_x \left[\left\| \sum_{i=1}^{n} x_i A_i \right\|_{S_\infty} \right] \lesssim \sqrt{\log(n^t)}\sqrt{k} = \sqrt{tk \log n}.$$

Combining the above inequalities, we get

$$n^t \sqrt{tk \log n} \gtrsim \eta k n^t \Rightarrow n \geq \exp(\Omega(\eta^2 k/t)) = \exp(\Omega(\delta \eta^2 k)).$$

□

Proof of Lemma 2.3.17

To prove the lemma, we need the following proposition which is a generalization of the Birkhoff-Von Neumann theorem for partial matchings.

Proposition 2.3.19 (Theorem 4.1 in [CC18]). *Let $s \leq n$ be some positive integers. Let $\mathcal{P}_{n,s}$ be the polytope of* doubly substochastic *$n \times n$ matrices (i.e. matrices with non-negative entries and whose row and column sums are at most 1), with the sum of all entries equal to s. The extreme points of $\mathcal{P}_{n,s}$ are incidence matrices of partial matchings on $[n]$ vertices of size s.*

Proof of Lemma 2.3.17. By Proposition 2.3.5, C is also a $(2, 1/\delta, \eta)$-smooth LDC. Say $\mathcal{A}_1, \ldots, \mathcal{A}_k$ be smooth decoders for C i.e. the distribution of their queries are $(1/\delta)$-smooth. For each $i \in [k]$, define the $n \times n$ matrix A_i as follows:

$$A_i(j,k) = \Pr[\mathcal{A}_i \text{ queries } j, k].$$

A_i is a non-negative matrix with total sum 1, we can also assume that the diagonal of A_i is zero. Moreover, because the marginal distributions of the queries that $\mathcal{A}_i$ makes are $(1/\delta)$-smooth, each row and column sum of A_i is at most $1/\delta n$. If we denote the expected output of $\mathcal{A}_i$ after querying $C(x)$ at $j, k \in [n]$ as $D^i_{jk}(C(x)_j, C(x)_k)$, then the decoding condition can be written as:

$$\forall\; x \in \{-1,1\}^k, \quad \sum_{j,k\in[n]} A_i(j,k) \cdot \Big(x_i D^i_{jk}(C(x)_j, C(x)_k)\Big) \geq \eta.$$

By taking average over a uniformly random $x \in \{-1,1\}^k$, we get:

$$\sum_{j,k\in[n]} A_i(j,k) \cdot \mathbb{E}_x \Big[x_i D^i_{jk}(C(x)_j, C(x)_k)\Big] \geq \eta.$$

Since $\lfloor \delta n \rfloor A_i$ is a doubly substochastic matrix with total sum $\lfloor \delta n \rfloor$, by Proposition 2.3.19, $\lfloor \delta n \rfloor A_i$ can be written as the convex combination of partial matchings of size $\lfloor \delta n \rfloor$ on n vertices i.e.

$$\lfloor \delta n \rfloor A_i = \sum_{|M|=\lfloor \delta n \rfloor} \lambda_{i,M} M.$$

Therefore there exists a partial matching M_i of size $\lfloor \delta n \rfloor$ such that

$$\sum_{j,k\in[n]} \frac{1}{\lfloor \delta n \rfloor} M_i(j,k) \cdot \mathbb{E}_x \Big[x_i D^i_{jk}(C(x)_j, C(x)_k)\Big] \geq \eta.$$

□

2.3.6 Lower bounds for q-query LDCs?

Here is one way to generalize the approach in Section 2.3.5, to give lower bounds for q-query LDCs for $q \geq 3$. For this, we need the following lemma similar to Lemma 2.3.17 for $q \geq 3$, but only gives matchings of size $\Omega_q(\eta\delta n)$.

Lemma 2.3.20 (See [BARDW08]). *Let $\mathcal{C} : \{-1,1\}^k \to \{-1,1\}^n$ be a (q, δ, η)-LDC. For each $i \in [k]$, there exists a set $\mathcal{M}_i$ of at least $\delta\eta n/q^2$ disjoint tuples, each of at* *most q elements from $[n]$, and a sign $a_{i,Q} \in \{-1,1\}$ for each $Q \in \mathcal{M}_i$, such that*

$$\mathbb{E}_{x \in \{-1,1\}^k}\left[a_{i,Q} x_i \prod_{j \in Q} \mathcal{C}(x)_j\right] \geq \frac{\eta}{2^q}$$

where the expectation is over a uniformly random $x \in \{-1,1\}^k$.

Given a q-multilinear form Λ, we define its norm as:

$$\|\Lambda\| = \sup\left\{\Lambda(x_1, \cdots, x_q) : \|x_1\|_{\ell_q} \leq 1, \cdots, \|x_q\|_{\ell_q} \leq 1\right\}. \tag{2.4}$$

Let $\mathcal{C} : \{-1,1\}^k \to \{-1,1\}^n$ be a q-query LDC and let $\mathcal{M}_i$ be the matchings obtained by applying Lemma 2.3.20. For each matching $\mathcal{M}_i$, we can define a q-multilinear form Λ_i as:

$$\Lambda(x_1, \cdots, x_q) = \sum_{(j_1, \cdots, j_q) \in \mathcal{M}_i} s_{i, j_1, \cdots, j_q} \prod_{i=1}^{q} (x_i)_{j_i}$$

where $s_{i,j_1,\cdots,j_q} \in \{-1,1\}$ are the signs obtained in Lemma 2.3.20. So for every $x \in \{-1,1\}^k$,

$$x_i \Lambda_i(C(x), \cdots, C(x)) \gtrsim_{\varepsilon,\delta,q} n.$$

Summing over $i \in [k]$ and taking expectation over a uniformly random $x \in \{-1,1\}^k$, we have

$$\mathbb{E}_x\left[\left(\sum_{i=1}^{k} x_i \Lambda_i\right)(C(x), \cdots, C(x))\right] \gtrsim_{\varepsilon,\delta,q} nk.$$

We can upper bound the LHS as:

$$\mathbb{E}_x\left[\left(\sum_{i=1}^{k} x_i \Lambda_i\right)(C(x), \cdots, C(x))\right] \leq \mathbb{E}_x\left[\left\|\sum_{i=1}^{k} x_i \Lambda_i\right\| \|C(x)\|_{\ell_q}^q\right] = n \cdot \mathbb{E}_x\left[\left\|\sum_{i=1}^{k} x_i \Lambda_i\right\|\right].$$

Therefore we have

$$\mathbb{E}_x\left[\left\|\sum_{i=1}^{k} x_i \Lambda_i\right\|\right] \gtrsim_{\varepsilon,\delta,q} k.$$

By applying Hölder's inequality, we can show that each of the Λ_i which arise from matchings have $\|\Lambda_i\| \leq 1$. So if we have a statement analogous to Proposition 2.3.18, which gives a good upper bound on $\mathbb{E}_x\left[\left\|\sum_{i=1}^k x_i \Lambda_i\right\|\right]$, we get good q-query LDC lower bounds. It can be proved that $\mathbb{E}_x\left[\left\|\sum_{i=1}^k x_i \Lambda_i\right\|\right] \leq f_q(n)\sqrt{k}$, which implies that $k \leq f_q(n)^2$. Proposition 2.3.18 implies that $f_2(n) \lesssim \sqrt{\log n}$. The existence of subexponential 3-query LDCs [Efr09] implies that $f_3(n) \geq \exp(\sqrt{\log n})$. Showing that $f_3(n) \leq n^{1/4-\alpha}$ for some $\alpha > 0$ implies super quadratic lower bounds for 3-query LDCs which is currently not known. $f_q(n)$ is related to the type constants of the Banach space on q-multilinear forms with norm as defined in Equation 2.4. See Section 8.1.4 for more information on type-constants.

2.4 Locally Correctable Codes (LCCs)

Locally Correctable Codes (LCCs) are strengthening of LDCs where coordinates of a corrupted codeword can be corrected locally. Intuitively, a code is said to be locally correctable [BFLS91, STV01, KT00] if, given a codeword $x \in C$ that has been corrupted by some errors, it is possible to decode any coordinate of x by reading only a small part of the corrupted version of x. Formally, it is defined as follows.

Definition 2.4.1 (Locally correctable code (LCC)). *Let Σ be some finite alphabet. For positive integers n, q and parameters $\eta, \delta > 0$, a subset $C \subset \Sigma^n$ is a (q, δ, η)-* locally correctable code *if, for every $i \in [n]$, there exists a randomized* corrector *(a probabilistic algorithm) $\mathcal{A}_i$ such that:*

- *For every codeword $x \in C$ and $y \in \Sigma^n$ such that $\mathrm{dist}_H(x, y) \leq \delta$,*

$$\Pr[\mathcal{A}_i(y) = x_i] \geq \Pr[\mathcal{A}_i(y) = \sigma] + \eta, \tag{2.5}$$

  for any $\sigma \in \Sigma$ such that $\sigma \neq x_i$.

- *The decoder $\mathcal{A}_i$ queries non-adaptively at most q coordinates of y.*

When the parameter η is not mentioned, usually it is assumed to be some fixed absolute constant. Sometimes LCCs are defined with Equation 2.5 replaced with $\Pr[\mathcal{A}_i(y) = x_i] \geq 2/3$ which is a stronger definition. We can assume that on input $y \in \{0,1\}^n$, the corrector $\mathcal{A}_i$ first samples a set $S \subseteq [n]$ of at most q coordinates according to a probability distribution depending only on i and then returns a random bit depending only on i, S and the values of y at S.

We can define (q, c, η)-smooth LCCs and perfectly smooth LCCs in a similar way as we defined for LDCs. In a (q, c, η)-smooth LCC, the marginal distribution of each query is c-smooth and in a perfectly smooth LCC, the marginal distribution of each query is uniform over all coordinates.

Similar to LDCs, LCCs should have large minimum distance as well.

Lemma 2.4.2. *Let $C \subset \Sigma^n$ be an (q, δ, η) LCC, then the minimum distance of $\mathcal{C}$ is at least 2δ.*

Proof. Let $x, y \in C$ be two distinct codewords such that $\mathrm{dist}_H(x, y) < 2\delta$. Let z be the midpoint of x and y, i.e. z is δ-close to both x and y. Let $i \in [n]$ be such that $x_i \neq y_i$. Since $x_i \neq y_i$, by the LCC property,

$$\Pr[x_i = \mathcal{A}_i(z)] \geq \Pr[y_i = \mathcal{A}_i(z)] + \eta,$$

$$\Pr[y_i = \mathcal{A}_i(z)] \geq \Pr[x_i = \mathcal{A}_i(z)] + \eta,$$

which is a contradiction. Therefore every two codewords must be at least 2δ apart. □

2.4.1 LDCs from LCCs

Locally correctability is a stronger notion than local decodability. For example if we have a linear LCC, by change of basis and permuting the coordinates one can make the code systematic and thus we get local decodability of message coordinates. So every linear LCC is also a linear LDC with the same parameters. We will show that any (possibly non-linear) q-query LCCs can be converted into q-query LDCs with only a constant loss in rate and preserving other parameters.

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

Definition 2.4.3. *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([Dud78]) 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 2.4.4 (Theorem 14.12 in [LT13])**.** *Let $A \subseteq \{0,1\}^n$ such that for every distinct $x, y \in A$, $\|x - y\|_{\ell_2} \geq \varepsilon\sqrt{n}$. Then*

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

We are now ready to prove the reduction from LCCs to LDCs. The following theorem is from [BGT17].

Theorem 2.4.5. *Let $\mathcal{C} \subseteq \Sigma^n$ be a (q, δ, η)-LCC, then there exists a (q, δ, η)-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 2.4.2, 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 2.4.4,

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

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 $1/2 + \eta$. □

2.4.2 Constructions for LCCs

Reed-Muller Codes

The best known constructions of constant-query LCCs over constant size alphabet are Reed-Muller codes.

Definition 2.4.6. *(Reed-Muller Codes) Let q be a prime power and $1 \leq d \leq q$. The Reed-Muller code of degree d over $\mathbb{F}_q$ is the subspace $C \subset \mathbb{F}_q^{\mathbb{F}_q^m}$ given by evaluations of degree $\leq d$ polynomials in $\mathbb{F}_q[x_1, \ldots, x_m]$ over all points of $\mathbb{F}_q^m$.*

Reed-Muller codes of degree d are perfectly smooth $(d+1)$-query LCCs of dimension

$$k = \binom{n+d}{d} = \Omega_d(m^d).$$

A smooth decoder for a degree d Reed-Muller code, to decode the value of a degree $\leq d$ polynomial at $z \in \mathbb{F}_q^m$, queries the values of the polynomial at $d+1$ points of a random line through z. Since the restriction of a degree $\leq d$ polynomial to a line is a univariate polynomial of degree $\leq d$, one can recover the value at z from the values at $d+1$ points on the line. See the survey [Yek12] for more discussion on local correction of Reed-Muller codes. Here we reproduce a table from [Yek12] which shows the length n of q-query LCCs obtainable from Reed-Muller codes in terms of the dimension k.

Table 2.1: Local correctability of Reed-Muller codes

q	n
$q = O(1)$	$\exp\left(O_q(k^{1/(q-1)})\right)$
$O(\log k \log\log k)$	$k^{O(\log\log k)}$
$(\log k)^t, t > 1$	$k^{1+1/t+o(1)}$
$O(k^{1/t}\log k), t \geq 1$	$t^{t+o(t)} \cdot k$

Constant rate LCCs

As shown in Table 2.4.2, Reed-Muller codes with appropriate parameters give LCCs with rate $\varepsilon^{1/\varepsilon}$ which are locally correctable from a constant fraction of errors with $O_\varepsilon(n^\varepsilon \log n)$ queries. There is a different family of codes called multiplicity codes, introduced in [KSY14] which do much better. Multiplicity codes are generalizations of Reed-Muller codes, where the evaluations also include all partial derivatives upto a certain order. They achieve any rate $r \in (0,1)$ and locally correctable from a constant fraction of errors (depending on ε, r) with n^ε queries. Lifted codes from [GKS13] and codes based on expander graphs from [HOW15] also achieve similar parameters. The best known family of LCCs with constant rate and low query complexity are from [KMRS17] which are obtained by starting with multiplicity codes and amplifying their distance.

Theorem 2.4.7 ([KMRS17]). *Let $r \in (0,1)$ be some fixed constant. There exists an infinite family of linear codes $\{C_n\}_n$ such that $C_n \subset \mathbb{F}_2^n$ is a linear $(q(n), \Omega_r(1), 2/3)$-LCC of rate r where*

$$q(n) = \exp\left(O\left(\sqrt{\log n \log\log n}\right)\right).$$

2.4.3 Lower bounds for LCCs

Since LCCs are a stronger notion than LDCs, any lower bounds for LDCs also apply to LCCs. But stronger lower bounds for LCCs are not known in general. In Chapter 7, we show a much stronger lower bound for zero-error[1] 2-query LCCs over large alphabet than possible for 2-query LDCs. The best known lower bounds for 3-query LDCs is quadratic; in [DSW14a], a super-quadratic lower bound is shown for 3-query LCCs defined over real numbers.

2.5 Locally Testable Codes (LTCs)

Intuitively, a code is said to be locally testable [FS95, RS96, GS06b] if, given a string $y \in \Sigma^n$, it is possible to determine whether y is a codeword of C, or rather far from C, by reading only a small part of y. There are two variants of LTCs in the literature, "weak" LTCs and "strong" LTCs, where the only difference is that weak LTCs are required to reject only words which are of sufficiently large constant relative distance from C, while strong LTCs are required to reject any word y not in C with probability proportional to the relative distance of y from C. We will define the strong LTCs here and we always mean strong LTCs when we refer to LTCs unless explicitly stated otherwise.

[1] Zero-error refers to the assumption that the local corrector will succeed with probability one if it is given a codeword with no corruptions. This is true for linear LCCs and almost all known constructions.

Definition 2.5.1 (Locally testable code (LTC))**.** *Let Σ be some finite alphabet. For positive integers k, n, q and $\delta, \rho > 0$, a map $C : \{0,1\}^k \to \Sigma^n$ is a (q, δ, ρ)-**locally testable code** if, there exists a randomized tester (a probabilistic algorithm) $\mathcal{T}$ such that:*

- *The minimum distance of the code is at least δ.*
- *For every message $x \in \{0,1\}^k$,*

$$\Pr[\mathcal{T}(C(x)) \text{ accepts}] = 1. \tag{2.6}$$

- *For every $y \in \Sigma^n$,*

$$\Pr[\mathcal{T}(y) \text{ rejects}] \geq \rho \cdot \mathrm{dist}_H(y, C). \tag{2.7}$$

- *The tester $\mathcal{T}$ queries non-adaptively at most q coordinates of its input.*

We can assume that on input $y \in \{0,1\}^n$, the tester $\mathcal{T}$ first samples a set $S \subseteq [n]$ of at most q coordinates according to some fixed probability distribution $\mathcal{D}$ and then accepts or rejects depending only on S and the values of y at S. Given an LTC with $\rho < \frac{1}{4}$, it is possible to amplify ρ up to $\frac{1}{4}$ at the cost of increasing the query complexity by a multiplicative factor of $1/\rho$ [KMRS17].

2.5.1 Constructions and lower bounds for LTCs

The best known constructions of constant query LTCs have $n = k \cdot \mathrm{polylog}(k)$.

Theorem 2.5.2 ([BS08, Din07, Vid15])**.** *There exists some constants $q, \delta, \rho > 0$ and some constant size field $\mathbb{F}$ such that for infinitely many $n \in \mathbb{N}$, there exists a linear code $C_n : \mathbb{F}^k \to \mathbb{F}^n$ which is a (q, δ, ρ)-LTC with $n = k \cdot \mathrm{polylog}(k)$.*

If we want the code to have constant rate, then there are LTCs which require only $(\log n)^{O(\log\log n)}$ queries.

Theorem 2.5.3 ([KMRS17])**.** *Let $r \in (0,1)$ be some fixed constant. There exists an infinite family of linear codes $\{C_n\}_n$ such that $C_n \subset \mathbb{F}_2^n$ is a linear $(q(n), \Omega_r(1), 1/4)$-LTC of rate r where*

$$q(n) = (\log n)^{O(\log\log n)}.$$

It is not known if there are LTCs with constant rate, constant distance and testable with constant number of queries.

Question 2.5.4. *Are there constant rate (q, δ, ρ)-LCCs with $q = O(1)$, $\delta = \Omega(1)$ and $\rho = \Omega(1)$ i.e. with constant distance, constant rate and testable with constant number of queries?*

2.6 Results and structure of this thesis

The chapters in this thesis are mostly self-contained, all the required prerequisites are contained in the preliminaries chapter (Chapter 2). This section contains a short description of the main results of each chapter. For a brief summary of the contributions of this thesis, see Section 1.2.

Chapter 3 - Private Information Retrieval

A 2-server Private Information Retrieval (PIR) scheme allows a user to retrieve the ith bit of an n-bit database replicated among two non-communicating servers, while not revealing any information about i to either server. In this chapter we construct a 2-server PIR scheme with total communication cost $n^{O\left(\sqrt{\frac{\log\log n}{\log n}}\right)}$. This improves over previously known 2-server protocols which all require $\Omega(n^{1/3})$ communication. Our construction circumvents the $n^{1/3}$ barrier of [RY06] which holds for the restricted model of bilinear group-based schemes (covering all previous 2-server schemes). The improvement comes from reducing the number of servers in existing protocols, based on Matching Vector Codes, from 3 or 4 servers to 2. This is achieved by viewing these protocols in an algebraic way (using polynomial interpolation) and extending them using partial derivatives. The results of this chapter are from [DG16].

Chapter 4 - Locality near Gilbert-Varshamov bound

One of the most important open problems in the theory of error-correcting codes is to determine the tradeoff between the rate R and minimum distance δ of a binary code. The best known tradeoff is the Gilbert-Varshamov bound, and says that for every $\delta \in (0, 1/2)$, there are codes with minimum distance δ and rate $R = R_{\mathsf{GV}}(\delta) > 0$ (for a certain simple function $R_{\mathsf{GV}}(\cdot)$). In this chapter we show that the Gilbert-Varshamov bound can be achieved by codes which support *local* error-detection and error-correction algorithms.

Specifically, we show the following results.

1. **Local Testing:** For all $\delta \in (0, 1/2)$ and all $R < R_{\mathsf{GV}}(\delta)$, there exist codes with length n, rate R and minimum distance δ that are locally testable with $\mathsf{quasipolylog}(n)$ query complexity.

2. **Local Correction:** For all $\varepsilon > 0$, for all $\delta < 1/2$ sufficiently large, and all $R < (1 - \varepsilon)R_{\mathsf{GV}}(\delta)$, there exist codes with length n, rate R and minimum distance δ that are locally correctable from $\frac{\delta}{2} - o(1)$ fraction errors with $O(n^{\varepsilon})$ query complexity.

Furthermore, these codes have an efficient randomized construction, and the local testing and local correction algorithms can be made to run in time polynomial in the query complexity. Our results on locally correctable codes also immediately give locally decodable codes with the same parameters.

Our local testing result is obtained by combining Thommesen's random concatenation technique and the best known locally testable codes from [KMRS17]. Our local correction result, which is significantly more involved, also uses random concatenation, along with a number of further ideas: the Guruswami-Sudan-Indyk list decoding strategy for concatenated codes, Alon-Edmonds-Luby distance amplification, and the local list-decodability, local list-recoverability and local testability of Reed-Muller codes. Curiously, our final local correction algorithms go via local list-decoding and local testing algorithms; this seems to be the first time local testability is used in the construction of a locally correctable code. The results of this chapter are from [GKdO$^+$17].

Chapter 5 - LDCs from Outlaw distributions

Locally decodable codes (LDCs) are error correcting codes that allow for decoding of a single message bit using a small number of queries to a corrupted encoding. Despite decades of study, the optimal trade-off between query complexity and codeword length is far from understood. In this chapter, we give a new characterization of LDCs using distributions over Boolean functions whose expectation is hard to approximate (in L_∞ norm) with a small number of samples. We coin the term 'outlaw distributions' for such distributions since they 'defy' the Law of Large Numbers. We show that the existence of outlaw distributions over sufficiently 'smooth' functions implies the existence of constant query LDCs and vice versa. We give several candidates for outlaw distributions over smooth functions coming from finite field incidence geometry, additive combinatorics and from hypergraph (non)expanders. The results of this chapter are from [BDG17].

Chapter 6 - Lower bounds for affine invariant local codes

Affine-invariant codes are codes whose coordinates form a vector space over a finite field and which are invariant under affine transformations of the coordinate space. They form a natural, well-studied class of codes; they include popular codes such as Reed-Muller and Reed-Solomon. A particularly appealing feature of affine-invariant codes is that they seem well-suited to admit local correctors and testers.

In this chapter, we give lower bounds on the length of locally correctable and locally testable affine-invariant codes with constant query complexity. We show that if a code $\mathcal{C} \subset \Sigma^{\mathbb{K}^n}$ is an r-query affine invariant locally correctable code (LCC), where $\mathbb{K}$ is a finite field and Σ is a finite alphabet, then the number of codewords in $\mathcal{C}$ is at most $\exp(O_{\mathbb{K},r,|\Sigma|}(n^{r-1}))$. Also, we show that if $\mathcal{C} \subset \Sigma^{\mathbb{K}^n}$ is an r-query affine invariant locally testable code (LTC), then the number of codewords in $\mathcal{C}$ is at most $\exp(O_{\mathbb{K},r,|\Sigma|}(n^{r-2}))$. The dependence on n in these bounds is tight for constant-query LCCs/LTCs, since Guo, Kopparty and Sudan [GKS13] construct affine-invariant codes via lifting that have the same asymptotic tradeoffs. Note that our result holds for non-linear codes, whereas previously, Ben-Sasson and Sudan [BSS11] assumed linearity to derive similar results.

Our analysis uses higher-order Fourier analysis. In particular, we show that the codewords corresponding to an affine-invariant LCC/LTC must be far from each other with respect to Gowers norm of an appropriate order. This then allows us to bound the number of codewords, using known decomposition theorems which approximate any bounded function in terms of a finite number of low-degree non-classical polynomials, up to a small error in the Gowers norm. The results of this paper are from [BG17a].

Chapter 7 - Lower bounds for 2-query LCCs

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. In this chapter, 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 $n \geq \exp(k/\log|\Sigma|)$ 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 [KT00]. Our bound implies that zero-error LCCs cannot yield 2-server private information retrieval (PIR) schemes with sub-polynomial communication. Since our results from Chapter 3 construct a 2-server PIR scheme with sub-polynomial communication based on a zero-error 2-query locally decodable code (LDC), we also obtain a separation between LDCs and LCCs over large alphabet. The results of this chapter are from [BGT17].

Chapter 8 - Applications to additive combinatorics

In this chapter, we give a few applications of the theory of LDCs to additive combinatorics. Specifically, we show how techniques used to prove LDC lower bounds can be used to prove upper bounds on the Gaussian width of special point sets in $\mathbb{R}^k$. The point sets are formed by the image of the n-dimensional Boolean hypercube under a mapping $\psi : \mathbb{R}^n \to \mathbb{R}^k$, where each coordinate is a constant-degree multilinear polynomial with 0-1 coefficients. We show the following applications of our bounds. Let $[\mathbb{Z}/N\mathbb{Z}]_p$ be the random subset of $\mathbb{Z}/N\mathbb{Z}$ containing each element independently with probability p.

- A set $D \subseteq \mathbb{Z}/N\mathbb{Z}$ is *ℓ-intersective* if any dense subset of $\mathbb{Z}/N\mathbb{Z}$ contains a proper $(\ell+1)$-term arithmetic progression with common difference in D. Our main result implies that $[\mathbb{Z}/N\mathbb{Z}]_p$ is ℓ-intersective with probability $1 - o(1)$

provided $p \geq \omega(N^{-\beta_\ell} \log N)$ for $\beta_\ell = (\lceil(\ell+1)/2\rceil)^{-1}$. This gives a polynomial improvement for all $\ell \geq 2$ of a previous bound due to Frantzikinakis, Lesigne and Wierdl. This reproves more directly the same improvement which can also be deduced from the results of Chapter 5 and the lower bounds from [KW04].

- Let X_k be the number of k-term arithmetic progressions in $[\mathbb{Z}/N\mathbb{Z}]_p$ and consider the large deviation rate $\rho_k(\delta) = \log \Pr[X_k \geq (1+\delta)\mathbb{E}X_k]$. We give quadratic improvements of the best-known range of p for which a highly precise estimate of $\rho_k(\delta)$ due to Bhattacharya, Ganguly, Shao and Zhao is valid for all odd $k \geq 5$. In particular, the estimate holds if $p \geq \omega(N^{-c_k} \log N)$ for $c_k = (6k\lceil(k-1)/2\rceil)^{-1}$.

The results of this chapter are from [BG17b].

Chapter 9 - Local codes for distributed storage

The explosion in the volumes of data being stored online has resulted in distributed storage systems transitioning to erasure coding based schemes. In this chapter, we will explore codes with local correctors which are tailored for distributed storage applications called Local Reconstructions Codes (LRCs). The main difference from LCCs is that these codes only need to locally correct in the presence of a *constant number* of errors (instead of constant fraction) which is the typical scenario in practice.

An (n, r, h, a, q)-LRC is a linear code over $\mathbb{F}_q$ of length n, whose codeword symbols are partitioned into $g = n/r$ local groups each of size r. Each local group has a local parity checks that allow recovery of up to a erasures within the group by reading the unerased symbols in the group. There are a further h "heavy" parity checks to provide fault tolerance from more global erasure patterns. Such an LRC is Maximally Recoverable (MR), if it corrects all erasure patterns which are information-theoretically correctable under the stipulated structure of local and global parity checks, namely patterns with up to a erasures in each local group and an additional h (or fewer) erasures anywhere in the codeword.

The efficiency of the encoding and decoding procedures is extremely sensitive to the field size and thus obtaining MR LRCs over finite fields of minimal size is crucial in practice and has been the goal of a line of work in coding theory. The existing constructions require fields of size $n^{\Omega(h)}$ while no superlinear lower bounds were known for any setting of parameters. Is it possible to get linear field size similar to the related MDS codes (e.g. Reed-Solomon codes)? In this chapter, we answer this question by showing superlinear lower bounds on the field size of MR LRCs. In particular, we show that when a and h are constant and r may grow with n, for every MR LRC with $g = n/r$ local groups,

$$q \geq \Omega_{a,h}\left(n \cdot r^{\alpha}\right) \text{ where } \alpha = \frac{\min\left\{a, h - 2\lceil h/g \rceil\right\}}{\lceil h/g \rceil}.$$

MR LRCs deployed in practice have a small number of global parities, typically $h = 2, 3$ [HSX^{+}12]. We complement our lower bounds by giving constructions with small field size for $h \leq 3$. When $h = 2$, we give a linear field size construction, whereas previous constructions required quadratic field size in some parameter ranges. Note that our lower bound is superlinear only if $h \geq 3$. When $h = 3$, we give a construction with $O(n^3)$ field size, whereas previous constructions needed $n^{\Theta(a)}$ field size. Our construction for $h = 2$ makes the choices $r = 3, a = 1, h = 3$ the next smallest setting to investigate regarding the existence of MR LRCs over fields of near-linear size. We answer this question in the positive via a novel approach based on elliptic curves and arithmetic progression free sets. The results of this chapter are from [GGY17].

Chapter 3

Private Information Retrieval

3.1 Introduction

Private Information Retrieval (PIR) was first introduced by Chor, Goldreich, Kushilevitz and Sudan [CGKS98]. In a k-server PIR scheme, a user can retrieve the ith bit a_i of an n-bit database $\mathbf{a} = (a_1, \cdots, a_n) \in \{0,1\}^n$ replicated among k servers (which do not communicate) while giving no information about i to any server. The goal is to design PIR schemes that minimize the *communication cost* defined as the worst case number of bits transferred between the user and the servers in the protocol. The trivial solution which works even with one server is to make a server send the entire database $\mathbf{a}$ to the user, which has communication cost n.

When $k = 1$ the trivial solution cannot be improved [CGKS98]. But when $k \geq 2$, the communication cost can be brought down significantly. In [CGKS98], a 2-server PIR scheme with communication cost $O(n^{1/3})$ and a k-server PIR scheme with cost $O\left(k^2 \log(k) \cdot n^{1/k}\right)$ were presented. The k-server PIR schemes were improved further in subsequent papers [Amb97, BI01, BIKR02]. In [BIKR02], a k-server PIR scheme with cost $n^{O\left(\frac{\log\log k}{k \log k}\right)}$ was obtained. Then, in a breakthrough result of Yekhanin [Yek08], the first 3-server scheme with sub-polynomial communication was given (assuming a number theoretic conjecture). Yekhanin's construction was cast in a nice framework using homomorphisms in [Rag07] which was used by Efremenko [Efr09] to give an unconditional k-server PIR scheme with sub-polynomial cost for $k \geq 3$. These were slightly improved in [IS10, CFL$^+$13]. These new PIR schemes follow from the constructions of constant query smooth Locally Decodable Codes (LDCs) of sub-exponential length called Matching Vector Codes (MVCs). A k-query LDC [KT00] is an error correcting code which allows the receiver of a corrupted encoding of a message to recover the ith bit of the message using only k (random) queries. In a *smooth* LDC, each query of the reconstruction algorithm is uniformly distributed among the code word symbols. Given a k-query smooth LDC, one can construct a k-server PIR scheme by letting each server simulate one of the queries. For more information on the relation between PIR and LDC we refer to the survey [Yek12].

Despite the advances in 3-server PIR schemes, the 2-server PIR case remained stuck at $O(n^{1/3})$ communication. An explanation to the apparent $n^{1/3}$ barrier for 2-server PIR was given by [RY06] who proved an $\Omega(n^{1/3})$ lower bound for a restricted model of 2-server PIR called *bilinear group based* PIR which contains all the previously known constructions. This is in stark contrast to the best known $5 \log n$ lower bound for general PIR schemes [WdW05a]. We elaborate more on the relation between this model and our construction after we present our results below.

PIR is extensively studied and there are several variants of PIR in literature. The most important variant with cryptographic applications is called Computationally Private Information Retrieval (CPIR). In CPIR, the privacy guarantee is based on computational hardness of certain functions i.e. a computationally bounded server cannot gain any information about the user's query. In this case, non-trivial schemes exist even in the case of one server under some cryptographic hardness assumptions. For more information on these variants of PIR see [Gas04, OSI07]. In this paper, we are only concerned with information theoretic privacy i.e. even a computationally unbounded server cannot gain any information about the user's query which is the strongest form of privacy.

3.1.1 Main Results

We start with a formal definition of a 2-server PIR scheme. A 2-server PIR scheme involves two servers $\mathcal{S}_1$ and $\mathcal{S}_2$ and a user $\mathcal{U}$. A database $\mathbf{a} = (a_1, \cdots, a_n) \in \{0,1\}^n$ is replicated between the servers $\mathcal{S}_1$ and $\mathcal{S}_2$. We assume that the servers cannot communicate with each other. The user $\mathcal{U}$ wants to retrieve the ith bit of the database a_i without revealing any information about i to either server. The following definition is from [CGKS98]:

Definition 3.1.1. *A 2-server PIR protocol is a triplet of algorithms $\mathcal{P} = (\mathcal{Q}, \mathcal{A}, \mathcal{R})$. At the beginning of the protocol, the user $\mathcal{U}$ obtains a uniformly random string $\mathbf{r}$. Next, $\mathcal{U}$ invokes $\mathcal{Q}(i, \mathbf{r})$ to generate a pair of queries $(\mathbf{q}_1, \mathbf{q}_2)$. $\mathcal{U}$ sends $\mathbf{q}_1$ to $\mathcal{S}_1$ and $\mathbf{q}_2$ to $\mathcal{S}_2$. Each server S_j responds with an answer $\mathbf{ans}_j = \mathcal{A}(j, \mathbf{a}, \mathbf{q}_j)$. Finally, $\mathcal{U}$ computes its output by applying the recovery algorithm $\mathcal{R}(\mathbf{ans}_1, \mathbf{ans}_2, i, \mathbf{r})$. The protocol should satisfy the following conditions:*

- *Correctness: For any n, $\mathbf{a} \in \{0,1\}^n$ and $i \in [n]$, the user outputs the correct value of a_i with probability 1 (where the probability is over random strings $\mathbf{r}$) i.e. $\mathcal{R}(\mathbf{ans}_1, \mathbf{ans}_2, i, \mathbf{r}) = a_i$.*

- *Privacy: Each server learns no information about i. That is, for any fixed database $\mathbf{a}$ and for $j = 1, 2$, the distributions of $\mathbf{q}_j(i_1, \mathbf{r})$ and $\mathbf{q}_j(i_2, \mathbf{r})$ are identical for all $i_1, i_2 \in [n]$ when $\mathbf{r}$ is randomly chosen.*

The communication cost of the protocol is the total number of bits exchanged between the user and the servers in the worst case.

k-server PIR is similarly defined, with the database replicated among k servers which cannot communicate between themselves. We only defined 1-round PIR i.e. there is only one round of interaction between the user and the servers. All known constructions of PIR schemes are 1-round and it is an interesting open problem to find if interaction helps. We now state our main theorem:

Theorem 3.1.2. *There exists a 2-server PIR scheme with communication cost* $n^{O\left(\sqrt{\frac{\log\log n}{\log n}}\right)}$.

In [Efr09] a 2^r-server PIR scheme was given with $n^{O\left((\log\log n/\log n)^{1-1/r}\right)}$ communication cost for any $r \geq 2$. Using our techniques, we can reduce the number of servers in this scheme by a factor of two. That is, we prove the following stronger form of Theorem 3.1.2.

Theorem 3.1.3. *For any* $r \geq 2$*, there exists a* 2^{r-1}*-server PIR scheme with communication cost* $n^{O\left((\log\log n/\log n)^{1-1/r}\right)}$.

Other than the dramatic improvement for the 2-server case, Theorem 3.1.3 also gives a more modest improvement over known results in some range of the parameters. The 2^r query complexity of Matching Vector Codes in [Efr09] was reduced to $9 \cdot 2^{r-4}$ for $r \geq 6$ in [IS10] while keeping the encoding length the same. This was improved in [CFL$^+$13] to $3^{\lceil r/2 \rceil}$ for $2 \leq r \leq 103$ and $(\frac{3}{4})^{51} \cdot 2^r$ for $r \geq 104$. Using these LDCs directly to get a PIR scheme is better than our scheme when the number of servers is more than 26, whereas our scheme is better than these when the number of servers are less than 9.

3.1.2 Proof Overview

On a very high level, the new protocol combines the existing 2-server scheme of [WY05], which uses polynomial interpolation using derivatives, with Matching Vector Codes (MV Codes) [Yek08, Efr09]. In particular, we make use of the view of MV codes as polynomial codes, developed in [DGY10]. This short overview is meant as a guide to the ideas in the construction (a detailed description will follow in the next sections). The 2-server scheme of [WY05] works by embedding the database $\mathbf{a} = (a_1, \ldots, a_n)$ as evaluations of a degree 3 polynomial $F(x_1, \ldots, x_k)$ at n points $P_1, \ldots, P_n \in \mathbb{F}_q^k$, with $k \sim n^{1/3}$ and $\mathbb{F}_q$ a finite field. To recover the value $a_i = F(P_i)$ the user passes a random line through the point P_i, picks two random points Q_1, Q_2 on that line and sends the point Q_j to the jth server. Each server responds with the value of F at Q_j and the values of all partial derivatives $\partial F/\partial x_\ell, \ell = 1, \ldots, k$ at that point. The restriction of F to the line is a univariate degree 3 polynomial and the user can recover the values of this polynomial at two points as well as the value of its derivative at these points. These four values (two evaluations plus two derivatives) are enough to recover the polynomial and so its value at P_i. The user can compute the derivatives of the restricted polynomial from the partial derivatives of F (knowing the line equation) using the chain rule. The protocol is private since each query Q_j is uniformly distributed in $\mathbb{F}_q^k$ and so independent of i.

We now describe the PIR schemes of [Yek08, Efr09] which are based on MV families. An MV family is a pair of lists $\mathcal{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$, $\mathcal{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_n)$ with each list element $\mathbf{u}_i$ and $\mathbf{v}_j$ belonging to $\mathbb{Z}_m^k$ and m is a small integer. These lists must satisfy the condition that $\langle \mathbf{u}_i, \mathbf{v}_j \rangle$ (taken mod m) is zero iff $i = j$. When m is a composite, say $m = 6$, one can construct such families of vectors of size $n = k^{\omega(1)}$ [Gro99] (this is impossible if m is prime). From such a family we can construct an m-server PIR scheme as follows: given a message $\mathbf{a} = (a_1, \ldots, a_n) \in \{0, 1\}^n$ define the polynomial $F(x_1, \ldots, x_k) = \sum_{i=1}^n a_i \mathbf{x}^{\mathbf{u}_i}$ (we denote $\mathbf{x}^{\mathbf{c}} = x_1^{c_1} \ldots x_k^{c_k}$). We think of F as a polynomial with coefficients in some finite field $\mathbb{F}_q$ containing an element $\gamma \in \mathbb{F}_q$ of order m.

To recover a_i the user picks a random $\mathbf{z} \in \mathbb{Z}_m^k$ and considers the restriction of F to the 'multiplicative line' given by $L = \{\gamma^{\mathbf{z}+t\mathbf{v}_i} \,|\, t \in \mathbb{Z}_m\}$, where $\gamma^{\mathbf{c}} = (\gamma^{c_1}, \ldots, \gamma^{c_k})$

for all $\mathbf{c} \in \mathbb{Z}_m^k$. That is, we denote $G(t) = F(\gamma^{\mathbf{z}+t\mathbf{v}_i})$. In [DGY10] it was observed that this restriction can be seen as a polynomial $g(T)$ of degree at most $m-1$ in the new 'variable' $T = \gamma^t$ and so can be reconstructed from the m values on the line $g(\gamma^t) = G(t), t = 0, 1, \dots, m-1$. The final observation is that $g(0)$ is a nonzero multiple of a_i (since the only contribution to the free coefficient comes from the monomial $a_i\mathbf{x}^{\mathbf{u}_i}$) and so we can recover it if we know $g(T)$. Hence, the user can recover a_i by asking the t'th ($t = 0, 1, \dots, m-1$) server for the value $G(t) = F(\gamma^{\mathbf{z}+t\mathbf{v}_i})$, which requires sending the uniformly random point $\mathbf{z}+t\mathbf{v}_i$ to the server. The communication cost is $O(k) = n^{o(1)}$ due to the super polynomial size of the MV family.

Our protocol extends the MV based protocol by asking each server for the evaluations of F at a point, as well as the values of a certain differential operator (similar to first order derivatives). For this to work we need two ingredients. The first is to replace the field $\mathbb{F}_q$ with a certain ring which has characteristic m and an element of order m (we only use $m = 6$ and can take the polynomial ring $\mathbb{Z}_m[\gamma]/(\gamma^6 - 1)$). The second is an observation that, in known MV families constructions [Gro99], the inner products $\langle \mathbf{u}_i, \mathbf{v}_j \rangle$ that are nonzero (that is, when $i \neq j$) can be made to fall in a small set. More precisely, over $\mathbb{Z}_6$, the inner products are either zero or in the set $\{1, 3, 4\}$. This means that the restricted polynomial only has nonzero coefficients corresponding to powers of T coming from the set $\{0, 1, 3, 4\}$. Such a polynomial has four degrees of freedom and can be recovered from two evaluations and two derivatives (of order one). We are also able to work with arbitrary MV families by using derivatives up to second order at two points (which are sufficient to recover a degree 5 polynomial)(see Appendix 3.7).

3.1.3 Organization

In Section 3.3 we give some preliminary definitions and notations. In Section 3.4, we review the construction of a 2-server PIR scheme with $O(n^{1/3})$ communication cost which is based on polynomial interpolation with partial derivatives [WY05]. In Section 3.5, we present our new 2-server scheme and prove Theorem 3.1.2. The proof of Theorem 3.1.3 is given in Section 3.7. We conclude in Section 3.8 with some remarks on future directions.

3.2 LDCs and PIR

There is a very close connection between k-server PIR protocols and k-query LDCs as observed in [KT00]. Given a $(k, 1, 1)$-perfectly smooth LDC $C : \{0, 1\}^n \to \Sigma^N$ (see Section 2.3.1 for definition), one can obtain a k-server PIR protocol with query size $\log N$ and answer size $\log |\Sigma|$. Each server stores the database $\mathbf{a}$ as $C(\mathbf{a})$. To decode a_i, the user uses the perfectly smooth decoder $\mathcal{A}_i$ to obtain k queries where the marginal distribution of each query is uniform over $[N]$ and sends a query each to the k servers. This implies the required privacy. The servers respond with the value of $C(\mathbf{a})$ at the queried location. Then the user can decode a_i with probability 1 from the values of $C(\mathbf{a})$ at the k queried locations.

Conversely, given a k-server PIR protocol over a database of n bits with query length t and answer length s, one can obtain a $(k, 1, \Omega(1))$-perfectly smooth LDC $C : \{0, 1\}^n \to \Sigma^N$ where $|\Sigma| = 2^s$ and $N = O(k2^t)$. Thus k-server PIR protocols are essentially k-query LDCs where the alphabet size is comparable to the length of the code.

Because of this connection, our main theorem (Theorem 3.1.2) can be rephrased as a construction of a two query LDC as follows:

Theorem 3.2.1. *There exists an explicit perfectly smooth two query LDC* $C : \{0, 1\}^n \to \Sigma^N$ *where*

$$N = |\Sigma| = \exp\left(\exp\left(O(\sqrt{\log n \log \log n})\right)\right).$$

3.2.1 Lower bounds for PIR

Because of the close connection between PIR and LDCs, lower bounds for PIR can be obtained from 2 query LDC lower bounds in Section 2.3.5. The best lower bounds on a two query perfectly smooth LDC $C : \{0,1\}^n \to \Sigma^N$ are from [KT00, KW04]:

$$N \geq \left(\frac{n}{|\Sigma|}\right)^2 \text{ and } N \geq \exp\left(\Omega(n/|\Sigma|^2)\right).$$

But since in the PIR setting, $N \approx |\Sigma|$, the lower bounds are ineffective. The best lower bound on the communication cost of PIR protocols is $5 \log n$ from [WDW05b].

Lower bounds for bilinear group-based PIR

In [RY06], an $\Omega(n^{1/3})$ lower bound was shown for a restricted model of 2-server PIR schemes. This lower bound holds for schemes that are both *bilinear* and *group-based.* Our scheme can be made into a bilinear scheme (see Section 3.5.1) over the field $\mathbb{F}_3$ of three elements (Our scheme can in fact be made linear and using a simple transformation given in [RY06], any linear scheme can be converted to a bilinear scheme). However, it does not satisfy the property of being group-based as defined in [RY06]. Our scheme does satisfy a weaker notion of *employing a group-based secret sharing scheme* (another technical term defined in [RY06]). The difference between these two notions (of being group-based as opposed to employing a group-based secret sharing scheme) is akin to the difference between LCCs and LDCs (LCCs being the stronger notion). In group-based PIR, the database is represented by the values of a function over a subset of a group but the user should be able to recover the value of that function at *every* group element. Our scheme encodes the database as a function over a group and the user will only be able to recover the bits of the database from the function.

3.3 Preliminaries

Notation

We will use bold letters like $\mathbf{x}, \mathbf{u}, \mathbf{v}, \mathbf{z}$ etc. to denote vectors. The inner product between two vectors $\mathbf{u} = (u_1, \cdots, u_k), \mathbf{v} = (v_1, \cdots, v_k)$ is denoted by $\langle \mathbf{u}, \mathbf{v} \rangle = \sum_{i=1}^{k} u_i v_i$. For a commutative ring, $\mathcal{R}$ we will denote by $\mathcal{R}[x_1, \cdots, x_k]$ the ring of polynomials in formal variables $x_1, \ldots, x_k$ with coefficients in $\mathcal{R}$. We will use the notation $\mathbf{x}^{\mathbf{z}}$ with $\mathbf{x} = (x_1, \cdots, x_k)$, $\mathbf{z} = (z_1, \cdots, z_k) \in \mathbb{Z}^k$ to denote the monomial $\prod_{i=1}^{k} x_i^{z_i}$. So any polynomial $F(\mathbf{x}) \in \mathcal{R}[x_1, \cdots, x_k]$ can be written as $F(\mathbf{x}) = \sum_{\mathbf{z}} c_{\mathbf{z}} \mathbf{x}^{\mathbf{z}}$. $\mathbb{Z}_m = \mathbb{Z}/m\mathbb{Z}$ is the ring of integers modulo m. When $\mathbf{u} \in \mathbb{Z}_m^k$, $\mathbf{x}^{\mathbf{u}}$ denotes $\mathbf{x}^{\tilde{\mathbf{u}}}$ where $\tilde{\mathbf{u}} \in \{0, 1, \cdots, m-1\}^k$ is the unique vector such that $\mathbf{u} \equiv \tilde{\mathbf{u}} \mod m$. $\mathbb{F}_q$ denotes the finite field of size q.

3.3.1 The rings $\mathcal{R}_{m,r}$

For our construction it will be convenient (although not absolutely necessary, see Section 3.5.1) to work over a ring which has characteristic 6 and contains an element of order 6. We now discuss how to construct such a ring in general.

Let $m > 1$ be an integer and let γ be a formal variable. We denote by

$$\mathcal{R}_{m,r} = \mathbb{Z}_m[\gamma]/(\gamma^r - 1)$$

the ring of univariate polynomials $\mathbb{Z}_m[\gamma]$ in γ with coefficients in $\mathbb{Z}_m$ modulo the identity $\gamma^r = 1$. More formally, each element $f \in \mathcal{R}_{m,r}$ is represented by a degree $\leq r-1$ polynomial $f(\gamma) = \sum_{\ell=0}^{r-1} c_\ell \gamma^\ell$ with coefficients $c_\ell \in \mathbb{Z}_m$. Addition is done as in $\mathbb{Z}_m[\gamma]$ (coordinate wise modulo m) and multiplication is done over $\mathbb{Z}_m[\gamma]$ but using the identity $\gamma^r = 1$ to reduce higher order monomials to degree $\leq r-1$. It is easy to see that this reduction is uniquely defined: to obtain the coefficient of γ^ℓ we sum all the coefficients of powers of γ that are of the form $\ell + kr$ for some integer $k \geq 0$. This implies the following lemma.

Lemma 3.3.1. *Let $f = \sum_{\ell=0}^{r-1} c_\ell \gamma^\ell$ be an element in $\mathcal{R}_{m,r}$. Then, $f = 0$ in the ring $\mathcal{R}_{m,r}$ iff $c_i = 0$ (in $\mathbb{Z}_m$) for all $0 \leq i \leq r-1$.*

Remark 3.3.2. *For any $t \in \{0, 1, \cdots, r-1\}$, γ^t is not a zero divisor in the ring $\mathcal{R}_{m,r}$. This holds since the coefficients of $\gamma^t \cdot f(\gamma)$ are the same as those of $f(\gamma)$ (shifted cyclically t positions).*

The rings $\mathcal{R}_{m,r}$ are sometimes denoted by $\mathbb{Z}_m[C_r]$ and referred to as the *group ring* of the cyclic group C_r with coefficients in $\mathbb{Z}_m$. See e.g., [KKS13, HH11] for some recent applications of these rings in cryptography.

3.3.2 Matrices over Commutative Rings

Let $\mathcal{R}$ be a commutative ring (with unity). Let $M \in \mathcal{R}^{n \times n}$ be an $n \times n$ matrix with entries from $\mathcal{R}$. Most of the classical theory of determinants can be derived in this setting in exactly the same way as over fields. One particularly useful piece of this theory is the *adjugate* (or *classical adjoint*) matrix. For an $n \times n$ matrix $M \in \mathcal{R}^{n \times n}$ the adjugate matrix is denoted by $\mathrm{adj}(M) \in \mathcal{R}^{n \times n}$ and has the (j,i)-cofactor of A as its (i,j)th entry (recall that the (i,j)-cofactor is the determinant of the matrix obtained from M after removing the ith row and jth column multiplied by $(-1)^{i+j}$). A basic fact in matrix theory is the following identity.

Lemma 3.3.3 (Theorem 1.7 from [McD84])**.** *Let $M \in \mathcal{R}^{n \times n}$ with $\mathcal{R}$ a commutative ring with identity. Then $M \cdot \mathrm{adj}(M) = \mathrm{adj}(M) \cdot M = \det(M) \cdot I_n$ where I_n is the $n \times n$ identity matrix.*

The way we will use this fact is as follows:

Remark 3.3.4. *Suppose* $M \in \mathcal{R}^{n\times n}$ *has nonzero determinant and let* $\mathbf{a} = (a_1, \ldots, a_n)^t \in \mathcal{R}^n$ *be some column vector where* $a_1 = 0$ *or* $a_1 = c$ *and* c *is not a zero-divisor. Then we can determine the value of* a_1 *(i.e., tell whether its* 0 *or* c*) from the product* $M \cdot \mathbf{a}$*. The way to do it is to multiply* $M \cdot \mathbf{a}$ *from the left by* $\operatorname{adj}(M)$ *and to look at the first entry. This will give us* $\det(M) \cdot a_1$ *which is zero iff* a_1 *is (since* $\det(M) \cdot c$ *is always nonzero).*

3.3.3 Matching Vector Families

Definition 3.3.5 (Matching Vector Family)**.** *Let* $S \subset \mathbb{Z}_m \setminus \{0\}$ *and let* $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ *where* $\mathcal{U} = (\mathbf{u}_1, \cdots, \mathbf{u}_n), \mathcal{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_n)$ *and* $\forall i \; \mathbf{u}_i, \mathbf{v}_i \in \mathbb{Z}_m^k$*. Then* $\mathcal{F}$ *is called an* S*-matching vector family over* $\mathbb{Z}_m$ *of size* n *and dimension* k *if* $\forall \; i, j$,

$$\langle \mathbf{u}_i, \mathbf{v}_j \rangle \begin{cases} = 0 & \textit{if } i = j \\ \in S & \textit{if } i \neq j \end{cases}$$

If S *is omitted, it implies that* $S = \mathbb{Z}_m \setminus \{0\}$.

Theorem 3.3.6 (Theorem 1.4 in [Gro99])**.** *Let* $m = p_1 p_2 \cdots p_r$ *where* $p_1, p_2 \cdots, p_r$ *are distinct primes with* $r \geq 2$*, then there exists an explicitly constructible* S*-matching vector family* $\mathcal{F}$ *in* $\mathbb{Z}_m^k$ *of size* $n \geq \exp\left(\Omega\left(\frac{(\log k)^r}{(\log\log k)^{r-1}}\right)\right)$ *where* $S = \{a \in \mathbb{Z}_m : a \bmod p_i \in \{0, 1\} \; \forall \; i \in [r]\} \setminus \{0\}$.

Remark 3.3.7. *The size of* S *in the above theorem is* $2^r - 1$ *by the Chinese Remainder Theorem. Thus, there are matching vector families of size super-polynomial in the dimension of the space with inner products restricted to a set of size* $2^r = |S \cup \{0\}|$*.*

In the special case when $p_1 = 2, p_2 = 3$, we have $m = 6$ and the following corollary:

Corollary 3.3.8. *There is an explicitly constructible* S*-matching vector family* $\mathcal{F}$ *in* $\mathbb{Z}_6^k$ *of size* $n \geq \exp\left(\Omega\left(\frac{(\log k)^2}{\log\log k}\right)\right)$ *where* $S = \{1, 3, 4\} \subset \mathbb{Z}_6$

3.4 Review of $O(n^{1/3})$ cost 2-server PIR

There are several known constructions of 2-server PIR with $O(n^{1/3})$ communication cost. We will recall here in detail a particular construction due to [WY05] which uses polynomial interpolation using derivatives (over a field). In the next section we will replace the field with a ring and see how to use matching vector families to reduce the communication cost.

Let $\mathbf{a} = (a_1, \cdots, a_n)$ be the database, choose k to be smallest integer such that $n \leq \binom{k}{3}$. Let $\mathbb{F}_q$ be a finite field with $q > 3$ elements. Let $\phi : [n] \mapsto \{0,1\}^k \subset \mathbb{F}_q^k$ be an embedding of the n coordinates into points in $\{0,1\}^k$ of Hamming weight 3. Such an embedding exists since $n \leq \binom{k}{3}$.

Define $F(x_1, \cdots, x_k) = F(\mathbf{x}) \in \mathbb{F}_q[x_1, \cdots, x_k]$ as

$$F(\mathbf{x}) = \sum_{i=1}^{n} a_i \left(\prod_{j:\phi(i)_j = 1} x_j \right)$$

Note that $F(\mathbf{x})$ is a degree 3 polynomial satisfying $F(\phi(i)) = a_i \ \forall \ i \in [n]$. Fix any two nonzero field elements $t_1 \neq t_2 \in \mathbb{F}_q \setminus \{0\}$.

Suppose the user $\mathcal{U}$ wants to recover the bit a_τ. The protocol is as follows: The user picks a uniformly random element $\mathbf{z} \in \mathbb{F}_q^k$ and sends $\phi(\tau) + t_1\mathbf{z}$ to $\mathcal{S}_1$ and $\phi(\tau) + t_2\mathbf{z}$ to $\mathcal{S}_2$. Each server S_i then replies with the value of F at the point received $F(\phi(\tau) + t_i\mathbf{z})$ as well as the values of the k partial derivatives of F at the same point

$$\nabla F(\phi(\tau) + t_i\mathbf{z}) = \left(\frac{\partial F}{\partial x_1}(\phi(\tau) + t_i\mathbf{z}), \cdots, \frac{\partial F}{\partial x_k}(\phi(\tau) + t_i\mathbf{z}) \right)$$

The partial derivatives here are defined in the same way as for polynomials over the real numbers.

$$\boxed{\begin{array}{l} \qquad\quad \mathcal{U} : \text{Picks a uniformly random } \mathbf{z} \in \mathbb{F}_q^k \\ \mathcal{U} \to \mathcal{S}_i : \phi(\tau) + t_i\mathbf{z} \\ \mathcal{S}_i \to \mathcal{U} : F(\phi(\tau) + t_i\mathbf{z}), \nabla F(\phi(\tau) + t_i\mathbf{z}) \end{array}}$$

The protocol is private since $\phi(\tau) + t\mathbf{z}$ is uniformly distributed in $\mathbb{F}_q^k$ for any τ and $t \neq 0$. Consider the univariate polynomial

$$g(t) = F(\phi(\tau) + t\mathbf{z}).$$

Observe that, by the chain rule,

$$g'(t) = \langle \nabla F(\phi(\tau) + t\mathbf{z}), \mathbf{z} \rangle .$$

Thus the user can recover the values $g(t), g'(t)$ for $t = t_1, t_2$ from the server's responses. From this information the user needs to find $g(0) = F(\phi(\tau)) = a_\tau$. Since F is a degree 3 polynomial, $g(t)$ is a univariate degree 3 polynomial, let $g(t) = \sum_{\ell=0}^{3} c_\ell t^\ell$. Therefore we have the following matrix equation:

$$\begin{bmatrix} g(t_1) \\ g'(t_1) \\ g(t_2) \\ g'(t_2) \end{bmatrix} = \begin{bmatrix} 1 & t_1 & t_1^2 & t_1^3 \\ 0 & 1 & 2t_1 & 3t_1^2 \\ 1 & t_2 & t_2^2 & t_2^3 \\ 0 & 1 & 2t_2 & 3t_2^2 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} = M \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

The matrix M has determinant $det(M) = (t_2 - t_1)^4$ and so M is invertible as long as $t_1 \neq t_2$. Thus the user can find $c_0 = g(0) = F(\phi(\tau)) = a_\tau$ by multiplying by the inverse of M.

The communication cost of this protocol is $O(k) = O(n^{1/3})$ since the user sends a vector in $\mathbb{F}_q^k$ to each server and each server sends an element in $\mathbb{F}_q$ and a vector in $\mathbb{F}_q^k$ to the user.

3.5 The new 2-server scheme: Proof of Theorem 3.1.2

In this section we describe our main construction which proves Theorem 3.1.2. Before describing the construction we set up some of the required ingredients and notations. The first ingredient is a matching vector family over $\mathbb{Z}_6$ as in Corollary 3.3.8. That is, we construct an $S = \{1,3,4\}$- matching vector family $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ where $\mathcal{U} = (\mathbf{u}_1, \cdots, \mathbf{u}_n), \mathcal{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_n)$ have elements in $\mathbb{Z}_6^k$. Corollary 3.3.8 tells us that this can be done with $n = \exp(\Omega(\log^2 k / \log\log k))$ or $k = \exp(O\left(\sqrt{\log n \ \log\log n}\right))$.

We will work with polynomials over the ring

$$\mathcal{R} = \mathcal{R}_{6,6} = \mathbb{Z}_6[\gamma]/(\gamma^6 - 1)$$

(see Section 3.3). We will denote the vector $(\gamma^{z_1}, \gamma^{z_2}, \cdots, \gamma^{z_k})$ by $\gamma^{\mathbf{z}}$ where $\mathbf{z} = (z_1, \cdots, z_k) \in \mathbb{Z}_6^k$. We will need to extend the notion of partial derivatives to polynomials in $\mathcal{R}[x_1, \ldots, x_k]$. This will be a non-standard definition, but it will satisfy all the properties we will need. Instead of defining each partial derivative separately, we define one operator that will include all of them.

Definition 3.5.1. *Let $\mathcal{R}$ be a commutative ring and let $F(\mathbf{x}) = \sum c_{\mathbf{z}} \mathbf{x}^{\mathbf{z}} \in \mathcal{R}[x_1, \ldots, x_k]$. We define*
$F^{(1)} \in (\mathcal{R}^k)[x_1, \ldots, x_k]$ to be

$$F^{(1)}(\mathbf{x}) := \sum (c_{\mathbf{z}} \cdot \mathbf{z}) \mathbf{x}^{\mathbf{z}}$$

For example, when $F(x_1, x_2) = x_1^2x_2 + 4x_1x_2 + 3x_2^2$ (with integer coefficients),

$$F^{(1)}(x_1, x_2) = \begin{bmatrix}2\\ \\1\end{bmatrix} x_1^2x_2 + 4\begin{bmatrix}1\\ \\1\end{bmatrix} x_1x_2 + 3\begin{bmatrix}0\\ \\2\end{bmatrix} x_2^2 = \begin{bmatrix}2\\ \\1\end{bmatrix} x_1^2x_2 + \begin{bmatrix}4\\ \\4\end{bmatrix} x_1x_2 + \begin{bmatrix}0\\ \\6\end{bmatrix} x_2^2.$$

One can think of $F^{(1)}$ both as a polynomial with coefficients in $\mathcal{R}^k$ as well as a k-tuple of polynomials in $\mathcal{R}[x_1, \ldots, x_k]$. This will not matter much since the only operation we will perform on $F^{(1)}$ is to evaluate it at a point in $\mathcal{R}^k$.

The Protocol

Let $\mathbf{a} = (a_1, a_2 \cdots, a_n) \in \{0, 1\}^n$ be an n-bit database shared by two servers $\mathcal{S}_1$ and $\mathcal{S}_2$. The user $\mathcal{U}$ wants to find the bit a_τ without revealing any information about τ to either server. For the rest of this section, $\mathcal{R} = \mathcal{R}_{6,6} = \mathbb{Z}_6[\gamma]/(\gamma^6 - 1)$. The servers represent the database as a polynomial $F(\mathbf{x}) \in \mathcal{R}[\mathbf{x}] = \mathcal{R}[x_1, \cdots, x_k]$ given by

$$F(\mathbf{x}) = F(x_1, \cdots, x_k) = \sum_{i=1}^{n} a_i \mathbf{x}^{\mathbf{u}_i},$$

where $\mathcal{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ are given by the matching vector family $\mathcal{F} = (\mathcal{U}, \mathcal{V})$.

The user samples a uniformly random $\mathbf{z} \in \mathbb{Z}_6^k$ and then sends $\mathbf{z} + t_1\mathbf{v}_\tau$ to $\mathcal{S}_1$ and $\mathbf{z} + t_2\mathbf{v}_\tau$ to $\mathcal{S}_2$ where we fix $t_1 = 0$ and $t_2 = 1$ (other choices of values would also work). $\mathcal{S}_i$ then responds with the value of F at the point $\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau}$, that is with $F(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$ and the value of the 'first order derivative' at the same point $F^{(1)}(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$. Notice that the protocol is private since $\mathbf{z} + t\mathbf{v}_\tau$ is uniformly distributed over $\mathbb{Z}_6^k$ for any fixed τ and t.

$\mathcal{U}$: Picks a uniformly random $\mathbf{z} \in \mathbb{Z}_6^k$

$\mathcal{U} \to \mathcal{S}_i : \mathbf{z} + t_i\mathbf{v}_\tau$

$\mathcal{S}_i \to \mathcal{U} : F(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau}), F^{(1)}(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$

Recovery

Define

$$G(t) := F(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}) = \sum_{i=1}^{n} a_i \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle + t\langle \mathbf{v}_\tau,\mathbf{u}_i\rangle}$$

Using the fact that $\gamma^6 = 1$, we can rewrite $G(t)$ as:

$$G(t) = \sum_{\ell=0}^{5} c_\ell \cdot \gamma^{t\ell},$$

with each $c_\ell \in \mathcal{R}$ given by

$$c_\ell = \sum_{i:\langle \mathbf{u}_i,\mathbf{v}_\tau\rangle = \ell \mod 6} a_i \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle}.$$

Since

$$\langle \mathbf{u}_i, \mathbf{v}_\tau\rangle \mod 6 \begin{cases} = 0 & \text{if } i = \tau \\ \in S = \{1,3,4\} & \text{if } i \neq \tau \end{cases}$$

we can conclude that $c_0 = a_\tau \gamma^{\langle \mathbf{u}_\tau, \mathbf{z}\rangle}$ and $c_2 = c_5 = 0$. Therefore

$$G(t) = c_0 + c_1\gamma^t + c_3\gamma^{3t} + c_4\gamma^{4t}.$$

Next, consider the polynomial

$$g(T) = c_0 + c_1 T + c_3 T^3 + c_4 T^4 \in \mathcal{R}[T].$$

By definition we have

$$\begin{aligned} g(\gamma^t) &= G(t) = F(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}) \\ g^{(1)}(\gamma^t) &= \sum_{\ell=0}^{5} \ell c_\ell \gamma^{t\ell} = \left\langle F^{(1)}(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}), \mathbf{v}_\tau \right\rangle, \end{aligned}$$

where the last equality holds since

$$\left\langle F^{(1)}(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}), \mathbf{v}_\tau \right\rangle = \left\langle \sum_{i=1}^{n} a_i \mathbf{u}_i \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle + t\langle \mathbf{v}_\tau, \mathbf{u}_i\rangle}, \mathbf{v}_\tau \right\rangle = \sum_{i=1}^{n} a_i \langle \mathbf{u}_i, \mathbf{v}_\tau\rangle \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle + t\langle \mathbf{v}_\tau, \mathbf{u}_i\rangle}$$
$$= \sum_{\ell=0}^{5} \ell \left(\sum_{i:\langle \mathbf{u}_i, \mathbf{v}_\tau\rangle = \ell \mod 6} a_i \gamma^{\langle \mathbf{z},\mathbf{u}_i\rangle} \right) \gamma^{t\ell} = \sum_{\ell=0}^{5} \ell c_\ell \gamma^{t\ell}$$

So the user can find the values of $g(\gamma^t), g^{(1)}(\gamma^t)$ for $t = t_1, t_2$. Since $t_1 = 0, t_2 = 1$, we obtain the following matrix equation:

$$\begin{bmatrix} g(1) \\ g^{(1)}(1) \\ g(\gamma) \\ g^{(1)}(\gamma) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 4 \\ 1 & \gamma & \gamma^3 & \gamma^4 \\ 0 & \gamma & 3\gamma^3 & 4\gamma^4 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ c_4 \end{bmatrix} = M \begin{bmatrix} c_0 \\ c_1 \\ c_3 \\ c_4 \end{bmatrix} \tag{3.1}$$

The determinant (over $\mathcal{R}$) of the matrix M is

$$\det(M) = \gamma(\gamma-1)^4(\gamma^2+4\gamma+1) = 3\gamma^5 + 4\gamma^4 + 3\gamma^3 + 2\gamma \tag{3.2}$$

and so, by Lemma 3.3.1, is a nonzero element of the ring $\mathcal{R}$. Since $c_0 = a_\tau \gamma^{\langle \mathbf{u}_\tau, \mathbf{z}\rangle}$, either $c_0 = 0$ or $c_0 = \gamma^{\langle \mathbf{u}_\tau, \mathbf{z}\rangle}$ which is not a zero-divisor by Remark 3.3.2. Hence, by Remark 3.3.4, the user can find whether $c_0 = 0$ from the vector $[g(1), g^{(1)}(1), g(\gamma), g^{(1)}(\gamma)]^t$ by multiplying it from the left by $\mathrm{adj}(M)$. Since $c_0 = a_\tau \gamma^{\langle \mathbf{u}_\tau, \mathbf{z}\rangle}$, a_τ will be zero iff c_0 is and so the user can recover $a_\tau \in \{0, 1\}$.

Communication Cost

The user sends a vector in $\mathbb{Z}_6^k$ to each server. Each server sends a element of $\mathcal{R}$ and a vector in $\mathcal{R}^k$ to the user. Since elements of $\mathcal{R}$ have constant size description, the total communication cost is $O(k) = n^{O\left(\sqrt{\frac{\log\log n}{\log n}}\right)} = n^{o(1)}$.

3.5.1 Working over $\mathbb{Z}_6$ or $\mathbb{F}_3$

Using the ring $\mathcal{R}_{6,6} = \mathbb{Z}_6[\gamma]/(\gamma^6 - 1)$ in the above construction makes the presentation clearer but is not absolutely necessary. Observing the proof, we see that one can replace it with any ring $\mathcal{R}$ as long as there is a homomorphism from $\mathcal{R}_{6,6}$ to $\mathcal{R}$ such that the determinant of the matrix M (Eq. 3.2) doesn't vanish under this homomorphism.

For example, we can work over the ring $\mathbb{Z}_6$ and use the element -1 as a substitute for γ. Since $(-1)^6 = 1$ all of the calculations we did with γ carry through. In addition, the resulting determinant of M is non zero when setting $\gamma = -1$ and so we can complete the recovery process. More formally, define the homomorphism $\tau : \mathbb{Z}_6[\gamma]/(\gamma^6 - 1) \mapsto \mathbb{Z}_6$ by extending the identity homomorphism on $\mathbb{Z}_6$ using $\tau(\gamma) = -1$. Observe that the determinant of the matrix M in Eq. 3.2 doesn't vanish under this homomorphism, $\tau(\det(M)) = -4 = 2$.

A more interesting example is the ring of integers modulo 3, which we denote by $\mathbb{F}_3$ to highlight that it is also a field. We can use the homomorphism $\phi : \mathbb{Z}_6[\gamma]/(\gamma^6 - 1) \mapsto \mathbb{F}_3$ by extending the natural homomorphism from $\mathbb{Z}_6$ to $\mathbb{F}_3$ (given by reducing each element modulo 3) using $\phi(\gamma) = -1$. Again the determinant in Eq. 3.2 doesn't vanish. This also shows that our scheme can be made to be *bilinear*, as defined in [RY06], since the answers of each server become linear combinations of database entries over a field and the recovered bit is also a linear combination of the answers of each server.

3.6 An Alternative Construction

In the construction of Section 3.5, we used the special properties of Grolmusz's construction, namely that the nonzero inner products are in the special set $S = \{1, 3, 4\}$. Here we show how to make the construction work with any matching vector family (over $\mathbb{Z}_6$). This construction also introduces higher order differential operators, which could be of use if one is to generalize this work further.

Suppose we run our protocol (with $\mathcal{R} = \mathcal{R}_{6,6}$) using a matching vector family with $S = \mathbb{Z}_6 \setminus \{0\}$. Then, we cannot claim that $c_2 = c_5 = 0$, but we still have $c_0 = a_\tau \gamma^{\langle \mathbf{u}_\tau, \mathbf{z} \rangle}$. We can proceed by asking for the 'second order' derivative of $F(\mathbf{x}) = \sum_{i=0}^{n} a_i \mathbf{x}^{\mathbf{u}_i}$ which we define as

$$F^{(2)}(\mathbf{x}) := \sum c_{\mathbf{z}} \; (\mathbf{z} \otimes \mathbf{z}) \; \mathbf{x}^{\mathbf{z}}$$

where $\mathbf{z} \otimes \mathbf{z}$ is the $k \times k$ matrix defined by $(\mathbf{z} \otimes \mathbf{z})_{ij} = z_i z_j$. For example, when $P(x_1, x_2) = x_1^2 x_2 + 4x_1 x_2 + 3x_2^2$,

$$\begin{aligned} P^{(2)}(x_1, x_2) &= \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} x_1^2 x_2 + 4 \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} x_1 x_2 + 3 \begin{bmatrix} 0 & 0 \\ 0 & 4 \end{bmatrix} x_2^2 \\ &= \begin{bmatrix} 4 & 2 \\ 2 & 1 \end{bmatrix} x_1^2 x_2 + \begin{bmatrix} 4 & 4 \\ 4 & 4 \end{bmatrix} x_1 x_2 + \begin{bmatrix} 0 & 0 \\ 0 & 12 \end{bmatrix} x_2^2. \end{aligned}$$

The final protocol is:

$\mathcal{U}$: Picks a uniformly random $\mathbf{z} \in \mathbb{Z}_m^k$

$\mathcal{U} \to \mathcal{S}_i : \mathbf{z} + t_i \mathbf{v}_\tau$

$\mathcal{S}_i \to \mathcal{U} : F(\gamma^{\mathbf{z} + t_i \mathbf{v}_\tau}), F^{(1)}(\gamma^{\mathbf{z} + t_i \mathbf{v}_\tau}), F^{(2)}(\gamma^{\mathbf{z} + t_i \mathbf{v}_\tau})$

Notice that privacy is maintained and the communication is $O(k^2) = n^{o(1)}$ as before. For recovery, define $g(T) \in \mathcal{R}[T]$ as before and notice that, in addition to the identities

$$\begin{aligned} g(\gamma^t) &= \sum_{\ell=0}^{5} c_\ell \gamma^{t\ell} = F(\gamma^{\mathbf{z} + t\mathbf{v}_\tau}) \\ g^{(1)}(\gamma^t) &= \sum_{\ell=0}^{5} \ell c_\ell \gamma^{t\ell} = \left\langle F^{(1)}(\gamma^{\mathbf{z} + t\mathbf{v}_\tau}), \mathbf{v}_\tau \right\rangle, \end{aligned}$$

we also get the second order derivative of g from

$$g^{(2)}(\gamma^t) = \sum_{\ell=0}^{5} \ell^2 c_\ell \gamma^{t\ell} = \left\langle F^{(2)}(\gamma^{\mathbf{z} + t\mathbf{v}_\tau}), \mathbf{v}_\tau \otimes \mathbf{v}_\tau \right\rangle,$$

where the inner product of matrices is taken entry-wise and using the identity $\langle \mathbf{u}\otimes\mathbf{u},\mathbf{v}\otimes\mathbf{v}\rangle = \langle \mathbf{u},\mathbf{v}\rangle^2$. By choosing $t_1 = 0, t_2 = 1$, we have the following matrix equation:

$$\begin{bmatrix} g(1) \\ g^{(1)}(1) \\ g^{(2)}(1) \\ g(\gamma) \\ g^{(1)}(\gamma) \\ g^{(2)}(\gamma) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 \\ 0 & 1 & 4 & 9 & 16 & 25 \\ 1 & \gamma & \gamma^2 & \gamma^3 & \gamma^4 & \gamma^5 \\ 0 & \gamma & 2\gamma^2 & 3\gamma^3 & 4\gamma^4 & 5\gamma^5 \\ 0 & \gamma & 4\gamma^2 & 9\gamma^3 & 16\gamma^4 & 25\gamma^5 \end{bmatrix} \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix} = M \begin{bmatrix} c_0 \\ c_1 \\ c_2 \\ c_3 \\ c_4 \\ c_5 \end{bmatrix}.$$

$\det(M) = 4\gamma^3(\gamma-1)^9 = 4 + 2\gamma^3 \neq 0$ and so we can use recover a_τ as before.

3.7 Generalization to more servers: Proof of Theorem 3.1.3

In this section we will prove Theorem 3.1.3. We will allow the database symbols to belong to a slightly larger alphabet $\mathbb{Z}_m$. Let $q = 2^{r-1}$ denote the number of servers $\mathcal{S}_1, \cdots, \mathcal{S}_q$ for some $r \geq 2$. Let $m = p_1 p_2 \cdots p_r$ where $p_1, p_2, \cdots, p_r$ are distinct primes. By Theorem 3.3.6, there is an explicit S-matching vector family $\mathcal{F} = (\mathcal{U}, \mathcal{V})$ of size n and dimension $k = n^{O\left((\log\log n/\log n)^{1-1/r}\right)}$ where $S = \{a \in \mathbb{Z}_m : a \mod p_i \in \{0,1\}\ \forall\ i \in [r]\} \setminus \{0\}$. By Remark 3.3.7, $|S \cup \{0\}| = 2^r = 2q$.

The Protocol

We will work over the ring $\mathcal{R} = \mathcal{R}_{m,m} = \mathbb{Z}_m[\gamma]/(\gamma^m - 1)$. The servers represent the database $\mathbf{a} = (a_1, \cdots, a_n) \in \mathbb{Z}_m^n$ as a polynomial $F(\mathbf{x}) \in \mathcal{R}[\mathbf{x}] = \mathcal{R}[x_1, \cdots, x_k]$ given by

$$F(\mathbf{x}) = F(x_1, \cdots, x_k) = \sum_{i=1}^{n} a_i \mathbf{x}^{\mathbf{u}_i},$$

where $\mathcal{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_n)$ are given by the matching vector family $\mathcal{F} = (\mathcal{U}, \mathcal{V})$.

The user samples a uniformly random $\mathbf{z} \in \mathbb{Z}_m^k$ and then sends $\mathbf{z} + t_i \mathbf{v}_\tau$ to $\mathcal{S}_i$ for $i \in [q]$ where $t_i = i - 1$. $\mathcal{S}_i$ then responds with the value of F at the point $\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau}$, that is with $F(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$ and the value of the 'first order derivative' at the same point $F^{(1)}(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$. Notice that the protocol is private since $\mathbf{z}+t\mathbf{v}_\tau$ is uniformly distributed over $\mathbb{Z}_m^k$ for any fixed τ and t.

$\mathcal{U}$: Picks a uniformly random $\mathbf{z} \in \mathbb{Z}_m^k$

$\mathcal{U} \to \mathcal{S}_i : \mathbf{z} + t_i \mathbf{v}_\tau$

$\mathcal{S}_i \to \mathcal{U} : F(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau}), F^{(1)}(\gamma^{\mathbf{z}+t_i\mathbf{v}_\tau})$

Recovery

Similar to the 2-server recovery, we define

$$G(t) := F(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}) = \sum_{i=1}^{n} a_i \gamma^{\langle \mathbf{z}, \mathbf{u}_i \rangle + t \langle \mathbf{v}_\tau, \mathbf{u}_i \rangle} = c_0 + \sum_{\ell \in S} c_\ell \gamma^{t\ell},$$

and

$$g(T) = c_0 + \sum_{\ell \in S} c_\ell T^\ell \in \mathcal{R}[T],$$

so that $c_0 = a_\tau \gamma^{\langle \mathbf{u}_\tau, \mathbf{z} \rangle}$ and

$$g(\gamma^t) = G(t) = F(\gamma^{\mathbf{z}+t\mathbf{v}_\tau})$$

$$g^{(1)}(\gamma^t) = \sum_{\ell=0}^{m-1} \ell c_\ell \gamma^{t\ell} = \left\langle F^{(1)}(\gamma^{\mathbf{z}+t\mathbf{v}_\tau}), \mathbf{v}_\tau \right\rangle,$$

Hence, the user can calculate the values of $g(\gamma^t), g^{(1)}(\gamma^t)$ for $t = t_1, \cdots, t_q$ and we end up with the following (square) system of equations:

$$\begin{bmatrix} g(\gamma^{t_1}) \\ g^{(1)}(\gamma^{t_1}) \\ \vdots \\ g(\gamma^{t_q}) \\ g^{(1)}(\gamma^{t_q}) \end{bmatrix} = \begin{bmatrix} 1 & \cdots & \gamma^{t_1\ell} & \cdots & \\ 0 & \cdots & \ell\gamma^{t_1\ell} & \cdots & \\ \vdots & & \vdots & & \vdots \\ 1 & \cdots & \gamma^{t_q\ell} & \cdots & \\ 0 & \cdots & \ell\gamma^{t_q\ell} & \cdots & \end{bmatrix} \begin{bmatrix} c_0 \\ \vdots \\ c_\ell \\ \vdots \end{bmatrix} = M \begin{bmatrix} c_0 \\ \vdots \\ c_\ell \\ \vdots \end{bmatrix}$$

where the $2^r = 2q$ columns are indexed by $\ell \in \{0\} \cup S$. Instead of computing the determinant (and the adjugate matrix), we will use the following lemma (proven below).

Lemma 3.7.1. *There exists a row vector*

$$\boldsymbol{\lambda} = [\alpha_1, \beta_1, \cdots, \alpha_q, \beta_q] \in \mathcal{R}^{2q}$$

such that $\boldsymbol{\lambda} M = [\mu, 0, \cdots, 0]$ *for some* $\mu \in \mathcal{R}$ *where* $\mu \mod p_i \neq 0 \ \forall i \in [r]$.

Using this lemma, the user can recover a_τ as follows. We have

$$\nu := \boldsymbol{\lambda} \begin{bmatrix} g(\gamma^{t_1}) \\ g^{(1)}(\gamma^{t_1}) \\ \vdots \\ g(\gamma^{t_q}) \\ g^{(1)}(\gamma^{t_q}) \end{bmatrix} = \boldsymbol{\lambda} M \begin{bmatrix} c_0 \\ \vdots \\ c_\ell \\ \vdots \end{bmatrix} = [\mu, 0, \cdots, 0] \begin{bmatrix} c_0 \\ \vdots \\ c_\ell \\ \vdots \end{bmatrix} = \mu c_0$$

Taking this equation modulo p_i we get,

$$(\nu \mod p_i) = (\mu c_0 \mod p_i) = (\mu \mod p_i)(a_\tau \mod p_i)\gamma^{\langle \mathbf{u}_\tau, \mathbf{z} \rangle}.$$

Let $\mu = \sum_{j=0}^{m-1} \mu_j \gamma^j$ and $\nu = \sum_{j=0}^{m-1} \nu_j \gamma^j$. Since $\mu \mod p_i \neq 0$, there exists j such that $\mu_j \mod p_i \neq 0$. So $(a_\tau \mod p_i) = (\mu_j \mod p_i)^{-1}(\nu_{j+\langle \mathbf{u}_\tau, \mathbf{z}\rangle} \mod p_i)$. So we can find $a_\tau \mod p_i$ for each $i \in [r]$. Finally we use Chinese Remainder Theorem to find $a_\tau \in \mathbb{Z}_m$.

To prove Lemma 3.7.1, we will need the following simple number-theoretic lemma. Recall that the *order* of an element a in a finite multiplicative group G is the smallest integer $w \geq 1$ so that $a^w = 1$.

Lemma 3.7.2. *Let $\mathbb{F}_p$ be a field of prime order p and let $k \geq 1$ be an integer co-prime to p. Then, the algebraic closure of $\mathbb{F}_p$ contains an element ζ of order k.*

Proof. Since k, p are co-prime, $p \in \mathbb{Z}_k^*$ which is the multiplicative group of invertible elements in $\mathbb{Z}_k$. Let $w \geq 1$ be the order of p in the group $\mathbb{Z}_k^*$, so k divides $p^w - 1$. Consider the extension field $\mathbb{F}_{p^w}$, which is a sub field of the algebraic closure of $\mathbb{F}_p$. The multiplicative group $\mathbb{F}_{p^w}^*$ of this field is a cyclic group of size $p^w - 1$. Since k divides this size, there must be an element in $\mathbb{F}_{p^w}$ of order k. □

3.7.1 Proof of Lemma 3.7.1

For any $\boldsymbol{\lambda} = [\alpha_1, \beta_1, \cdots, \alpha_q, \beta_q] \in \mathcal{R}^{2q}$ we can define a function $h : S \cup \{0\} \mapsto \mathcal{R}$ as:

$$h(\ell) = (\boldsymbol{\lambda} M)_\ell = \left(\sum_{i=1}^{q} \alpha_i \gamma^{t_i \ell} \right) + \ell \left(\sum_{i=1}^{q} \beta_i \gamma^{t_i \ell} \right).$$

Our goal is then to construct an h of this form such that

$$h(\ell) \begin{cases} = 0 & \text{if } \ell \in S \\ = \mu & \text{if } \ell = 0 \end{cases}$$

where $(\mu \mod p_i) \neq 0 \; \forall i \in [r]$. Notice that, by Chinese Remaindering,

$$\mathcal{R} = \mathcal{R}_{m,m} \cong \mathcal{R}_{p_1,m} \times \ldots \times \mathcal{R}_{p_r,m}, \tag{3.3}$$

where we recall that $\mathcal{R}_{p_i,m} = \mathbb{Z}_{p_i}[\gamma]/(\gamma^m - 1)$. Therefore, we also get that, for a formal variable x, the rings of univariate polynomials also satisfy

$$\mathcal{R}[x] \cong \mathcal{R}_{p_1,m}[x] \times \ldots \times \mathcal{R}_{p_r,m}[x].$$

In other words, any family of polynomials $f_i \in \mathcal{R}_{p_i,m}[x]$, $i \in [r]$ can be 'lifted' to a single polynomial $f \in \mathcal{R}[x]$ so that $(f \mod p_i) = f_i$ for all i (reducing f mod p_i is done coordinate-wise). Moreover, since this lift is done coefficient-wise (using Eq. 3.3), we get that the degree of f is equal to the maximum of the degrees of the f_i's. We begin by constructing, for each $i \in [r]$ the following polynomial $f_i(x) \in \mathcal{R}_{p_i,m}[x]$:

$$f_i(x) = \prod_{\ell \in S,\; \ell = 0 \mod p_i} (x - \gamma^\ell)$$

The degree of f_i is $2^{r-1} - 1 = q - 1$ so, by the above comment, we can find a polynomial $f(x) \in \mathcal{R}[x]$ of degree $q - 1$ such that $f(x) \equiv f_i(x) \mod p_i$ for all $i \in [r]$. Define $\alpha_i, i \in [q]$ to be the coefficients of the polynomial f so that $f(x) = \sum_{i=1}^q \alpha_i x^{i-1}$. Since we defined $t_i = i - 1$, we have $f(x) = \sum_{i=1}^q \alpha_i x^{t_i}$. Define $\beta_i = -\alpha_i$ for all $i \in [q]$. Our final construction of h is thus

$$h(\ell) = f(\gamma^\ell) - \ell f(\gamma^\ell).$$

Claim 3.7.3. $h(\ell) = 0 \; \forall \ell \in S$

Proof. Since $0 \notin S$, $\ell \neq 0$. We will look at $h(\ell)$ modulo each of the primes.

$$\begin{aligned} h(\ell) \mod p_i &= f_i(\gamma^\ell) - (\ell \mod p_i) f_i(\gamma^\ell) \\ &= \begin{cases} f_i(\gamma^\ell) = 0 & \text{if } \ell = 0 \mod p_i \\ f_i(\gamma^\ell) - f_i(\gamma^\ell) = 0 & \text{if } \ell = 1 \mod p_i \end{cases} \end{aligned}$$

Therefore, using Chinese Remaindering, $h(\ell) = 0 \ \forall \ell \in S$. □

Claim 3.7.4. *$(h(0) \mod p_j) \neq 0$ for all $j \in [r]$*

Proof. Suppose in contradiction that $(h(0) \mod p_j) = 0$, then

$$h(0) \mod p_j = f_j(1) = \prod_{\ell \in S,\ \ell = 0 \mod p_j} (1 - \gamma^\ell) = 0.$$

The above equation holds in the ring $\left(\mathbb{Z}_{p_j}[\gamma]/(\gamma^m - 1)\right)$. Therefore, if we consider what happens in the ring $\mathbb{Z}_{p_i}[\gamma] \cong \mathbb{F}_{p_i}[x]$ (we replace the formal variable γ with x to highlight the fact that x does not satisfy any relation) we get that

$$\prod_{\ell \in S,\ \ell = 0 \mod p_j} (1 - x^\ell) = (x^m - 1)\theta(x) \tag{3.4}$$

for some polynomial $\theta(x) \in \mathbb{F}_{p_j}[x]$. The above equation is an identity in the ring $\mathbb{F}_{p_j}[x]$. So we can check its validity by substituting values for x from the algebraic closure of $\mathbb{F}_{p_j}$. Let $m' = m/p_j$ and let ζ be an element in the algebraic closure of $\mathbb{F}_{p_j}$ of order m' (so $\zeta^\ell = 1$ iff m' divides ℓ). Since m' and p_j are co-prime, such an element exists by Lemma 3.7.2. If we substitute ζ into Eq. 3.4, the RHS is zero (since m' divides m). However, each term in the LHS product is nonzero, since if $\ell = 0 \mod p_j$ and m' divides ℓ then $\ell = 0 \mod m$ but we know that $0 \notin S$. Since we are working over the algebraic closure of $\mathbb{F}_{p_j}$ which is a field, the product of nonzero elements is nonzero. This is a contradiction, and so Eq. 3.4 does not hold. □

3.8 Concluding remarks

In this work we presented the first 2-server PIR scheme with sub-polynomial cost. It is unclear what is the optimal communication cost of 2-server schemes and we conjecture that our protocol is far from optimal. Clearly, a construction of MV families in $\mathbb{Z}_6^k$ of larger size will immediately give better 2-server PIR schemes. There is very little known about the limitations of this approach and current upper bounds on the size of MV families are nearly exponential [DGY10, BDL13, DH13]. In [BDL13], for constant m, an upper bound of $\exp\left(c(m)k/\log k\right)$ was obtained on the the size of MV families over $\mathbb{Z}_m^k$ where $c(m)$ depends only on m, assuming a well-known conjecture in additive combinatorics called the polynomial Freiman-Ruzsa conjecture over $\mathbb{Z}_m$. If this bound is tight, it would give a $O(\log n \log\log n)$ communication constant server PIR. When m is large, an upper bound of $q^{O(q\log q)}m^{k/2}$ was shown on the size of S-matching vector families over $\mathbb{Z}_m^k$ where $q = |S| + 1$. Thus all the known upper bounds are consistent with obtaining poly-logarithmic communication cost for 2-server protocols using our approach.

Another approach to decrease the communication cost is to take m to be a product of $r > 2$ prime factors in Theorem 3.3.6 to get a larger S-matching vector family where $S = \{a \in \mathbb{Z}_m : a \mod p_i \in \{0,1\}\ \forall\ i \in [r]\} \setminus \{0\}$ which is of size $2^r - 1$. So we need 2^{r-1} independent equations from each server to find c_0. We can ask the servers for derivatives of F at $\gamma^{\mathbf{z}+t\mathbf{v}_\tau}$ up to order $2^{r-1} - 1$. If these equations are 'independent' i.e. the determinant of the coefficient matrix doesn't vanish then we can find c_0. If we can do this, we can decrease the cost to $n^{O\left(2^r(\log\log n/\log n)^{1-1/r}\right)}$. But observe that for each $l \in S$, $l^2 = l \mod m$ since $l \mod p_i \in \{0,1\}\ \forall i \in [r]$. So higher order derivatives of g are equal to the first order derivative and we get repeated rows in the coefficient matrix M (Eq. 3.1).

All the known PIR schemes are single round and it is an interesting open problem to see if interaction can decrease the cost.

3.9 Subsequent work

The results of this chapter has led to new developments in (information theoretic) cryptography. The 2-server PIR schemes from this chapter have been used to get improved schemes for conditional disclosure of secrets and secret sharing [LVW17, LVW18].

Chapter 4

Locality near Gilbert-Varshamov bound

4.1 Introduction

In this paper, we show the existence of binary locally testable codes and locally correctable codes with rate-distance tradeoff matching what is known for general error-correcting codes.

One of the main combinatorial problems of coding theory is to determine the best tradeoff between the rate and the minimum distance for binary error-correcting codes. The best tradeoff known today is known as the Gilbert-Varshamov (GV) bound, and states that for every $\delta \in (0, 1/2)$, there exist codes of arbitrarily large length with minimum distance δ and rate $R = R_{\mathsf{GV}}(\delta)$. Here R_{GV} is the function:

$$R_{\mathsf{GV}}(\delta) = 1 - H(\delta),$$

where H is the binary entropy function. There are many known families of codes, including random codes, that achieve the GV bound, and it has been often conjectured that the GV bound is tight.

On the algorithmic side, it is not known how to deterministically construct codes that achieve the GV bound in polynomial time. Nevertheless, efficient deterministic constructions of codes with quite good rate-distance tradeoff are known, and furthermore these codes come equipped with efficient error-detection and correction algorithms. An alternate research direction, which is most relevant for us, has been to show existence of *highly structured* codes achieving the GV bound. Here we mention the beautiful results of Thommesen [Tho83], who gave a randomized construction of codes closely related to Reed-Solomon codes that meet the GV bound, and of Guruswami-Indyk [GI04], who gave a polynomial time algorithm for decoding Thommesen's codes from $\delta/2$-fraction errors (for sufficiently large $\delta < 1/2$). This latter work uses deep results of Guruswami-Sudan [GS99, GS00, GS02] on list-decoding Reed-Solomon codes and concatenated codes.

Our main result is that there are codes that approach the GV bound, that can be locally tested and locally corrected from $(\delta/2 - o(1))$-fraction errors with sublinear (even polynomially small) query complexity. For binary codes, it was previously known how to do this for codes approaching the Zyablov bound [KMRS17], with the added advantage that the code also had an efficient deterministic construction.
We give the formal statements of our main results next.

Theorem 4.1.1 (Locally testable codes approaching the GV bound)**.** *Let* $\delta \in (0, 1/2)$ *and* $R < R_{\mathsf{GV}}(\delta)$*. For every large enough* $n \in \mathbb{N}$*, there exists a binary linear code* $C_n \subset \{0,1\}^n$ *such that:*

- *rate of* C_n *is at least* R*,*
- C_n *is a* $(q, \delta, 1/4)$*-LTC i.e.* C_n *is locally testable with* $q = (\log n)^{O(\log \log n)}$ *queries.*

Moreover, there exists a randomized algorithm which, on input $n \in \mathbb{N}$*, runs in time* $\mathrm{poly}(n)$ *and outputs with high probability a generating matrix for a code* $C_n \subseteq \{0,1\}^n$

with the properties above and a local tester for C_n. The local testing algorithm runs in time $(\log n)^{O(\log \log n)}$. [1]

Theorem 4.1.2 (Locally correctable codes approaching the GV bound)**.** *Let $\varepsilon > 0$, and let $\xi > 0$ be sufficiently small (depending on ε). Additionally, let $\delta = \frac{1}{2} - \xi$ and $R < (1 - \varepsilon) \cdot R_{\mathsf{GV}}(\delta)$. For every large enough $n \in \mathbb{N}$, there exists a binary linear code $C_n \subset \{0, 1\}^n$ such that:*

- *the minimum distance of C_n is at least δ,*
- *the rate of C_n is at least R,*
- *C_n is a $(q, (\frac{\delta}{2} - o(1)), 1/3)$-LCC with $q = O(n^{\varepsilon})$ i.e. C_n is locally correctable from $(\frac{\delta}{2} - o(1))$-fraction errors with $O(n^{\varepsilon})$ queries.*

Furthermore, there exists a randomized algorithm which, on input $n \in \mathbb{N}$, runs in time $\mathrm{poly}(n)$ *and outputs with high probability the generating matrix of a code $C_n \subseteq \{0, 1\}^n$ with the properties above and a local correction algorithm. The local correction algorithm runs in time $n^{O(\varepsilon)}$.* [2]

Remark 4.1.3. *To simplify presentation, throughout this paper, we will assume that the LTCs involved have the robustness parameter from Definition 2.5.1 fixed to $\rho = 1/4$. Similarly we will assume that all the LCCs in this paper have the correction probability to be $\geq 2/3$ i.e. Equation 2.5 in Definition 2.4.1 is replaced by the stronger condition that the probability of correcting any given coordinate is $\geq 2/3$. Since our results are about constructions, these assumptions only make our results stronger.*

Note that our result about local testability allows for codes with rate and distance arbitrarily close to the GV bound for any distance $\delta \in (0, 1/2)$, but our result about

[1] The randomized algorithm can output different codes and testers (correctors) under different random choices, we are only guaranteed that the output code and the corresponding tester (corrector) have the required local testing (correcting) properties with high probability.

[2] See Footnote 1.

local correctability only achieves this for distances sufficiently close to $1/2$ depending on ε where $O(n^{\varepsilon})$ is the query complexity, and with a further $(1-\varepsilon)$-factor loss in the rate. These results are the first to show that codes with distance $\delta = 1/2 - \xi$ and rate $\Omega(\xi^2)$ can be locally tested / locally corrected from $(\delta/2 - o(1))$-fraction errors.

We remark that analogous results over large alphabets were only recently obtained [KMRS17]. In this setting, the best tradeoff between R and δ for general codes is completely known. Every code must satisfy $R \leq 1 - \delta$; this bound is known as the Singleton bound. Furthermore, Reed-Solomon codes achieve $R = 1 - \delta$, and they can be decoded from a $\delta/2$-fraction of errors in polynomial time. In [KMRS17], it was shown that there exist explicit locally testable codes and locally correctable codes which satisfy $R = 1 - \delta - \varepsilon$ (for all $\varepsilon > 0$), and which can further be locally tested and locally corrected from $(\delta/2 - o(1))$-fraction errors in sublinear (and even subpolynomial) time.

4.1.1 Methods

The starting point for our constructions is the random concatenation technique of Thommesen [Tho83], which he used to show that codes of a particular simple form can achieve the GV bound. Specifically, he showed that if one takes a Reed-Solomon code over a large alphabet as the outer code, and concatenate it with binary linear inner codes chosen uniformly at random and independently for each outer coordinate, then the resulting code C lies on the GV bound with high probability. In fact, the only property of Reed-Solomon codes that is used in this result is that the rate and distance of Reed-Solomon codes lie on the Singleton bound.

Our construction of locally testable codes approaching the GV bound then follows from the result of [KMRS17], which gave constructions of locally testable codes with rate and distance approaching the Singleton bound. We start with such a locally testable code from [KMRS17] as the outer code, and then concatenate it with uniformly random binary linear inner codes (independently for each coordinate of the outer code). The required rate-distance tradeoff of the concatenated code follows from Thommesen's arguments, and the local testability follows easily from the local testability of the outer code.

It is also known how to construct locally correctable codes with rate and distance approaching the Singleton bound [KMRS17]. If we use these codes along with the random concatenation idea, we get locally correctable codes approaching the GV bound. *But Theorem 4.1.2 requires the fraction of errors correctable by the local correction algorithm to approach* $\delta/2$. The natural local correction algorithm for concatenated codes (using the local correction algorithm of the outer code, and decoding inner codes by brute-force whenever an outer coordinate needs to be accessed) turns out to only decode to a much smaller radius (namely half the Zyablov bound); see [KMRS17] for details.

Our proof of Theorem 4.1.2 uses several more ideas. The next ingredient we use is an insight of Guruswami and Indyk [GI04]. They noted that the code C constructed by Thommesen could be decoded from $\delta/2$ fraction errors in polynomial time, provided the distance δ of C is sufficiently large (equivalently, provided the rate R of C is sufficiently small). The main idea is to use the *list-decoding* algorithms for concatenated codes developed by Guruswami and Sudan [GS00, GS02], which for binary codes of distance nearly 1/2, can list decode from a fraction of errors that is also nearly 1/2 – in particular, the fraction of errors correctable is far more than half the minimum distance (which is around 1/4). One first list-decodes each of the inner concatenated codes (by brute-force) to get a list of candidate symbols for each coordinate of the Reed-Solomon code, and then one applies the *list-recovery*[3] algorithm (of Guruswami and Sudan [GS99]) for the outer Reed-Solomon code to

[3] A list-recovery algorithm is a generalization of a list-decoding algorithm. Here one is given a small list of candidate symbols S_i for each coordinate i of the code, and the goal is to find all codewords $c \in C$ which have the property that for at most α fraction of the coordinates i, the ith coordinate of c does not lie in S_i.

get a list of candidate codewords. Finally, by computing the distance between each of these candidate codewords and the given received word, one can identify the one codeword (if any) that lies within distance $\delta/2$ of the received word.

Our local correction algorithm will try to implement this high-level strategy in the local setting. We will choose an outer code C over a large alphabet with suitable properties, and concatenate it with independently chosen random binary linear inner codes. We describe the properties required of C next:

- First of all, our choice of C should be such that the concatenated code approaches the GV bound with high probability. We will achieve this by ensuring that C has a sufficiently good rate-distance tradeoff[4].

- Next, we would like C to be *locally* list-recoverable. A local list-recovery algorithm is an algorithm that solves the list-recovery problem (in an implicit manner) using few queries. Instead of outputting all nearby codewords (which is impossible using few queries), the local list-recovery algorithm outputs implicit description of words $w_1, \dots, w_L$ which is guaranteed to contain all nearby codewords.

- Finally, we would like C to be *locally testable.* This is in order to identify which of the words w_i are actually codewords. Having done this, we can easily identify, by estimating distances via sampling, the one codeword w_i that is $\delta/2$-close to our original received word.

To encapsulate the requirements on C, we define the stronger notion of "sound" local list-recoverability (see Definition 4.2.6), which requires that each w_i in the output list of C describes some codeword. This is why we need local testability in addition to the standard list-recoverability: to weed out w_j which do not describe any codeword.

[4] The rate-distance tradeoff will be quite close to, but not approaching, the Singleton bound. This is the reason for our final locally correctable codes of Theorem 4.1.2 achieving rate that is smaller than $R_{\mathsf{GV}}(\delta)$ by a factor $(1-\varepsilon)$.

Summarizing, we want C to be locally list-recoverable, locally testable and have a decent rate-distance tradeoff. One might have hoped that the recently constructed codes of [KMRS17], which achieve local testability and local correctability with optimal rate-distance tradeoff (on the Singleton bound) would be good candidates for C. Unfortunately, none of the codes from [KMRS17] are known to achieve local list-recoverability.

Instead, we go further back in time to the mother of all local codes, Reed-Muller codes. they do not have good rate-distance tradeoff. This brings us to our final ingredient: Alon-Edmonds-Luby (AEL) distance amplification [AEL95]. This distance amplification method improves the rate-distance tradeoff for codes. Furthermore, it was shown in [KMRS17] that this method preserves local testability and local correctability. Here we observe that this distance amplification method also preserves local list-recoverability.[5] Thus, applying AEL distance amplification to Reed-Muller codes gives us a code that is locally list-recoverable, locally testable, and also has a decent rate-distance tradeoff (which turns out to be good enough for our purposes)[6]. This gives us the code C, and completes the high-level description of our constructions.

4.1.2 Further remarks

LTCs approaching the GV bound with constant query complexity? Our construction of LTCs approaching the GV bound is based on two ingredients: an LTC approaching the Singleton bound [KMRS17], and Thommesen's random concatenation technique. The result of [KMRS17] is in fact quite general: given any LTC family which can achieve rate arbitrarily close to 1, one can construct another

[5] In [], it was observed that AEL distance amplification preserves list-recoverability (without the locality requirement).

[6] This description suffices for the existence part of Theorem 4.1.2. However, to achieve *sublinear time* decoding, we will need one further trick: to concatenate the Reed-Muller codes down to a smaller alphabet before applying the AEL transformation - this smaller alphabet size is needed to let us perform the brute force list decoding of the random inner codes step quickly.

LTC family which approaches the Singleton bound with only a constant factor blowup in the query complexity. Putting everything together: if there exist LTCs with rate arbitrarily close to 1 with query complexity q, then there are LTCs approaching the GV bound with query complexity $O(q)$. It has often been lamented (at least once in print [BSGK+10], see page 2) by researchers in the area that we do not know any lower bounds on the rate-distance tradeoff of LTCs that distinguish them from general codes, and that for all we know, there could be constant query LTCs on the GV bound. Our result shows that such a lower bound would imply something much more qualitative - that there do not exist constant query LTCs with rate arbitrarily close to 1.

Locally decodable codes. Any linear LCC can be converted into a linear LDC by a simple basis change (see Section 2.4.1). Since the LCCs we construct in Theorem 4.1.2 are linear, it also implies a corresponding result for LDCs.

4.1.3 Organization of this paper

This paper is structured as follows: in Section 4.2 we provide some background on error correcting codes and set up the notation that will be used throughout the paper. In Section 4.3 we show the existence of locally testable codes approaching the GV bound. In Section 4.4 we show how to convert any code on a large alphabet with (somewhat) good rate and distance into a binary code nearly approaching the GV bound. In Section 4.5 we show the existence of locally correctable codes approaching the GV bound using the latter transformation. In Sections 4.6 and 4.7 we provide further ingredients needed for the construction of our locally correctable codes, namely local list recovery algorithm for Reed-Muller codes (in Section 4.6), and distance amplification procedure for local list recovery (in Section 4.7).

4.2 Preliminaries

We denote by $\mathbb{F}_q$ the finite field of q elements. For any finite alphabet Σ and for any string $x \in \Sigma^n$ the relative weight $\mathrm{wt}(x)$ of x is the fraction of non-zero coordinates of x, that is, $\mathrm{wt}(x) := |\{i \in [n] : x_i \neq 0\}|/n$. For any pair of strings $x, y \in \Sigma^n$, the relative distance between x and y is the fraction of coordinates on which x and y differ, and is denoted by $\mathrm{dist}_H(x, y) := |\{i \in [n] : x_i \neq y_i\}|/n$. For a positive integer ℓ we denote by $\binom{\Sigma}{\ell}$ the set containing all subsets of Σ of size ℓ, and for any $x \in \Sigma^n$ and $S \in \binom{\Sigma}{\ell}^n$ we denote by $\mathrm{dist}_H(x, S)$ the fraction of coordinates $i \in [n]$ for which $x_i \notin S_i$, that is, $\mathrm{dist}_H(x, S) := |\{i \in [n] : x_i \notin S_i\}|/n$. Throughout the paper, we use $\exp(n)$ to denote $2^{\Theta(n)}$. Whenever we use log, it is to the base 2.

4.2.1 Error-correcting codes

Let Σ be an alphabet and let n be a positive integer (the block length). A code is simply a subset $C \subseteq \Sigma^n$. If $\mathbb{F}$ is a finite field and Σ is a vector space over $\mathbb{F}$, we say that a code $C \subseteq \Sigma^n$ is $\mathbb{F}$-linear if it is an $\mathbb{F}$-linear subspace of the $\mathbb{F}$-vector space Σ^n. If $\Sigma = \mathbb{F}$, we simply say that C is linear. The rate of a code is the ratio $\frac{\log |C|}{\log(|\Sigma|^n)}$, which for $\mathbb{F}$-linear codes equals $\frac{\dim_{\mathbb{F}}(C)}{n \cdot \dim_{\mathbb{F}}(\Sigma)}$.

The elements of a code C are called codewords. The relative distance $\mathrm{dist}_H(C)$ of C is the minimum $\delta > 0$ such that for every pair of distinct codewords $c_1, c_2 \in C$ it holds that $\mathrm{dist}_H(c_1, c_2) \geq \delta$, which for $\mathbb{F}$-linear codes equals the minimum $\delta > 0$ such that $\mathrm{wt}(c) \geq \delta$ for every $c \in C$. We will use the notation $\mathrm{dist}_H(w, C)$ to denote the relative distance of a string $w \in \Sigma^n$ from C, and say that w is ε-close (respectively, ε-far) to C if $\mathrm{dist}_H(w, C) < \varepsilon$ (respectively, if $\mathrm{dist}_H(w, C) \geq \varepsilon$).

An encoding map for C is a bijection $E_C : \Sigma^k \to C$, where $|\Sigma|^k = |C|$. For a code $C \subseteq \Sigma^n$ of relative distance δ, a given parameter $\alpha < \delta/2$, and a string $w \in \Sigma^n$, the problem of decoding from α fraction of errors is the task of finding the unique $c \in C$ (if any) which satisfies $\text{dist}_H(c, w) \leq \alpha$.

List decodable and list recoverable codes. List decoding is a paradigm that allows one to correct more than $\delta/2$ fraction of errors by returning a small list of close-by codewords. More formally, $\alpha \in [0, 1]$ and an integer L we say that a code $C \subseteq \Sigma^n$ is (α, L)-list decodable if for any $w \in \Sigma^n$ there are at most L different codewords $c \in C$ which satisfy that $\text{dist}_H(c, w) \leq \alpha$. The Johnson bound (see e.g., Corollary 3.2. in [Gur06]) states that any code $C \subseteq \Sigma^n$ of relative distance at least $(1 - \frac{1}{|\Sigma|})\delta$ is (α, L)-list decodable for $\alpha \approx (1 - \frac{1}{|\Sigma|})(1 - \sqrt{1-\delta})$ and constant L (independent of n).

Theorem 4.2.1 (Johnson bound)**.** *Let $C \subseteq \Sigma^n$ be a code of relative distance at least $(1 - \frac{1}{|\Sigma|})\delta$. Then C is $\Big((1 - \frac{1}{|\Sigma|})\alpha, L\Big)$-list decodable for any $\alpha < 1 - \sqrt{1-\delta}$ with $L = \frac{1}{(1-\alpha)^2-(1-\delta)}$.*

For decoding concatenated codes it is often useful to consider the notion of list recovery where one is given as input a small list of candidate symbols for each of the coordinates and is required to output a list of codewords that are consistent with many of the input lists. More specifically, we say that a code $C \subseteq \Sigma^n$ is (α, ℓ, L)-list recoverable if for any $S \in \binom{\Sigma}{\ell}^n$ there are at most L different codewords $c \in C$ which satisfy that $\text{dist}_H(c, S) \leq \alpha$.

Some useful codes. In what follows we mention several families of codes that will be used in our construction.

Let q be a prime power, let d, n be positive integers such that $d \leq n \leq q$, and let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be n distinct points in $\mathbb{F}_q$. The Reed-Solomon code $RS_n(d, q)$ is the subset of $\mathbb{F}_q^n$ containing all words of the form $(p(\alpha_1), p(\alpha_2), \ldots, p(\alpha_n))$ where $p \in \mathbb{F}_q[x]$ is a univariate polynomial of degree less than d over $\mathbb{F}_q$. It can be verified that $RS_n(d,q)$ has rate d/n and it is well-known that it has relative distance at least $1-d/n$. In [Sud97, GS99] it was shown that the Reed-Solomon codes can be efficiently list decoded up to the Johnson bound. We state here a stronger form that applies also to list recovery (see e.g., Theorem 4.11 in [Gur06]).

Theorem 4.2.2 (List recovery of Reed-Solomon codes)**.** *The following holds for any prime power q, and integers d, n, ℓ which satisfy that $\ell d \leq n \leq q$. There exists a deterministic algorithm which given an input $S \in \binom{\mathbb{F}_q}{\ell}^n$, outputs all codewords $c \in RS_n(d,q)$ such that $\mathrm{dist}_H(c,S) < 1 - \sqrt{\ell \cdot \frac{d}{n}}$. The running time of the algorithm is* $\mathrm{poly}(q)$.

For a prime power q and integers $d < q$ and m the Reed-Muller code $RM(m,d,q)$ is the subset of $\mathbb{F}_q^{q^m}$ containing all words of the form $(p(\alpha))_{\alpha \in \mathbb{F}_q^m}$ where $p \in \mathbb{F}_q[x_1, \ldots, x_m]$ is a polynomial of (total) degree less than d in m variables over $\mathbb{F}_q$. Note that $RS_q(d,q) = RM(1,d,q)$ for every d, q. It can also be verified that $RM(m,d,q)$ has rate

$$\frac{\binom{m+d}{m}}{q^m} \geq \left(\frac{d}{mq}\right)^m$$

and relative distance at least $1 - d/q$.

We shall also use the following fact which says that a random binary linear code achieves the Gilbert-Varshamov bound [Gil52, Var57] with high probability. For $x \in [0,1]$ let $H(x) = x \log x + (1-x)\log(1-x)$ denote the binary entropy function.

Fact 4.2.3 (Gilbert-Varshamov (GV) codes)**.** *For any $\delta \in [0, 1/2)$ and $R \in (0, 1 - H(\delta))$, for sufficiently large n, a random binary linear code of block length n and rate R has relative distance at least δ with probability at least $1 - \exp(-n)$.*

4.2.2 Locally list decodable and list recoverable codes.

The following definition generalizes the notion of locally correctable codes to the setting of list decoding. In this setting the corrector algorithm is required to find all the nearby codewords in an implicit sense. Note that our definition below also includes a nonstandard soundness property.

Definition 4.2.4 (Locally list decodable code)**.** *We say that a code $C \subseteq \Sigma^n$ is $(q, \alpha, \varepsilon, L)$-locally list decodable if there exists a randomized algorithm A that satisfies the following requirements:*

- ***Input:*** *A gets oracle access to a string $w \in \Sigma^n$.*

- ***Query complexity:*** *A makes at most q queries to the oracle w.*

- ***Output:*** *A outputs L randomized algorithms $A_1, \ldots, A_L$. When algorithm A_j is given as input a coordinate $i \in [n]$, it makes at most q queries to the oracle w and outputs a symbol in Σ.*

- ***Completeness:***[7] *For every codeword $c \in C$ that is α-close to w, with probability at least $1 - \varepsilon$ over the randomness of A the following event happens: there exists some $j \in [L]$ such that for all $i \in [n]$,*

$$\Pr[A_j(i) = c_i] \geq \frac{2}{3},$$

  where the probability is over the internal randomness of A_j.

- ***Soundness:*** *With probability at least $1 - \varepsilon$ over the randomness of A, the following event happens: for every $j \in [L]$, there exists some $c \in C$ such that for all $i \in [n]$,*

$$\Pr[A_j(i) = c_i] \geq \frac{2}{3},$$

  where the probability is over the internal randomness of A_j.

[7] We can in fact satisfy the following, and more natural definition of completeness: *with high probability, all codewords in the ball appear in the output of A.* The reason is that we can amplify the tester so it will reject close-by codewords with probability at most $1/n$ by paying a multiplicative factor of $O(\log n)$ in number of queries. Since number of close-by codewords is $O(n^\beta)$ one can then apply a union bound over all close-by codewords to show that with probability at least 0.99, say, all of them will be accepted.

We say that A has running time T if A outputs the description of the algorithms $A_1, \dots, A_L$ in time at most T and each A_j has running time at most T.

Remark 4.2.5. (On the soundness property) *Typically locally list decodable codes are defined without the soundness property. For us, the soundness property is important to allow us to identify the unique closest codeword to the given received word.*

In a later section, we will first construct a locally list decodable code without the soundness property, and then we will achieve soundness via local testing.

The definition of locally list decodable codes can also be extended to the setting of list recovery, the only difference is that the input is a tuple $S \in \binom{\Sigma}{\ell}^n$ instead of a string $w \in \Sigma^n$. The same remarks about the soundness property apply to this case also.

Definition 4.2.6 (Locally list recoverable code)**.** *We say that a code $C \subseteq \Sigma^n$ is $(q, \alpha, \varepsilon, \ell, L)$-locally list recoverable if there exists a randomized algorithm A that satisfies the following requirements:*

- ***Input:*** *A gets oracle access to an $S \in \binom{\Sigma}{\ell}^n$.*
- ***Query complexity:*** *A makes at most q queries to the oracle S.*
- ***Output:*** *A outputs L randomized algorithms $A_1, \dots, A_L$, where each A_j takes as input a coordinate $i \in [n]$, makes at most q queries to the oracle S and outputs a symbol in Σ.*
- ***Completeness:*** *For every codeword $c \in C$ for which $\mathrm{dist}_H(c, S) \le \alpha$, with probability at least $1 - \varepsilon$ over the randomness of A, the following event happens: there exists some $j \in [L]$ such that for all $i \in [n]$,*

  $$\Pr[A_j(i) = c_i] \ge \frac{2}{3},$$

  where the probability is over the internal randomness of A_j.

- ***Soundness:*** *With probability at least $1-\varepsilon$ over the randomness of A, the following event happens: for every $j \in [L]$, there exists some $c \in C$ such that for all $i \in [n]$,*
$$\Pr[A_j(i) = c_i] \geq \frac{2}{3},$$
where the probability is over the internal randomness of A_j.

As above, we say that A has running time T if A outputs the description of the algorithms $A_1, \ldots, A_L$ in time at most T and each A_j has running time at most T. Note that a code is $(q, \alpha, \varepsilon, L)$-locally list decodable if and only if it is $(q, \alpha, \varepsilon, 1, L)$-locally list recoverable.

4.3 LTCs approaching the GV bound

In this section we prove the following theorem which implies Theorem 4.1.1 from the introduction.

Theorem 4.3.1 (LTCs approaching the GV bound)**.** *For any $\delta \in [0, \frac{1}{2})$ and $0 < \gamma < 1 - H(\delta)$, there exists an infinite family $\{C'_n\}_n$ of binary linear codes which satisfy the following. The code C'_n has block length n, rate at least $1 - H(\delta) - \gamma$, relative distance at least δ, and is $(\log n)^{O(\log\log n)}$-locally testable.*

Moreover, there is a randomized polynomial time algorithm which, given n, with probability $1 - \exp(-n)$, outputs a code of length n and a tester with the above properties,[8] and the local testing algorithm runs in time polynomial in the number of queries i.e. $(\log n)^{O(\log\log n)}$.

To prove the above theorem we first show in Section 4.3.1 a transformation which turns any large alphabet code approaching the Singleton bound into a binary code

[8] It might return different codes and testers under different random choices, we are only guaranteed that the output code and tester will have the required properties with high probability.

approaching the GV bound. In Section 4.3.2 we will then use this transformation to obtain LTCs approaching the GV bound.

4.3.1 Approaching the GV bound via random concatenation

Thommesen [Tho83] used the operation of random concatenation to construct a binary code lying on the GV bound out of a large alphabet code lying on the Singleton bound. In actuality, Thommesen's proof is more general than that: it shows that one can transform, via random concatenation, any linear code of rate R and large enough distance δ over a large alphabet (2^t) into a binary code with the same rate R and distance δ' which is only slightly smaller than $\delta/2$. The following lemma shows an approximate version of Thommesen's argument. With this approximate version, an important corollary is that we can replace "lying on the GV bound" with "close to the GV bound" in Thommesen's original statement, which we will use in this paper. The proof is identical to Thommesen's.

The intuition is based on the following observations: since the codes are linear, in order to preserve high distance it is enough to preserve the Hamming weights of the codewords. Suppose the large alphabet is $\mathbb{F}_{2^t}$. A random invertible mapping from $\mathbb{F}_{2^t} \to \mathbb{F}_2^t$ maps the nonzero elements of $\mathbb{F}_{2^t}$ uniformly over $\mathbb{F}_2^t \setminus \{0\}$. Thus, if each coordinate of the large alphabet is mapped to an element of $\mathbb{F}_2^t$ by a random invertible mapping (each mapping being chosen independently for each coordinate), one expects the weight of the image of each nonzero coordinate to be its typical value. Hence, there will be a small probability that the weight of the new codeword is small (over the choice of random mappings). To bound this probability, we crucially use (for the union bound over all codewords of the original code) the fact that the relative distance of the original code is sufficiently large.

Lemma 4.3.2. *Let $t \in \mathbb{N}$ be a large enough integer. The following holds for any $\delta \in (2/t, 1/2)$, and sufficiently large n. Let $C \subseteq \mathbb{F}_{2^t}^n$ be a linear code of rate R and* *relative distance δ. Let $C' \subseteq \mathbb{F}_2^{t\cdot n}$ be a code obtained from C by applying a (uniformly) random invertible $\mathbb{F}_2$-linear transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^t$ on each coordinate $i \in [n]$ of C independently. Then C' has rate R and relative distance at least $\delta' = H^{-1}(\delta - 2/t)$ with probability at least $1 - \exp(-n)$.*

Proof. The proof follows the arguments of [Tho83].

Fix a codeword $c \in C$ with $\mathrm{wt}(c) = \alpha \geq \delta$ and let $c' \in \mathbb{F}_2^{tn}$ be a word obtained from c by applying a uniformly random invertible $\mathbb{F}_2$-linear transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^t$ on each coordinate $i \in [n]$ of c independently. Then for each non-zero coordinate i of c it holds that the i-th block of c' of length t is distributed uniformly over $\mathbb{F}_2^t \setminus \{0\}$. Let c'' be a random word obtained by assigning uniformly random values in $\mathbb{F}_2^t$ to non-zero coordinates of c. We thus have that

$$\begin{aligned} \Pr[\mathrm{wt}(c') < \delta'] \ &\leq \Pr[\mathrm{wt}(c'') < \delta'] \ \leq \binom{\alpha tn}{\leq \delta' tn} 2^{-\alpha tn} \\ &\leq 2^{H(\delta'/\alpha)\alpha tn} \cdot 2^{-\alpha tn}, \end{aligned}$$

where the last inequality follows from the well known fact $\binom{m}{\leq \beta m} \leq 2^{H(\beta)\cdot m}$ for $\beta \leq 1/2$. (Here we use the fact that $\delta' < \delta/2 \leq \alpha/2$, given our choice of δ'.)

Next we apply a union bound over all codewords $c \in C$. For this fix $\alpha > 0$ such that $\alpha \geq \delta$ and $\alpha n \in \mathbb{N}$. The number of codewords in C of relative weight α is at most

$$\binom{n}{\alpha n} \cdot \left(2^t\right)^{\alpha n - \delta n} \leq 2^n \cdot 2^{(\alpha-\delta)tn},$$

where the above bound follows since there are at most $\binom{n}{\alpha n}$ choices for the location of the non-zero coordinates, and for any such choice fixing the value of the first $\alpha n - \delta n$ non-zero coordinates determines the value of the rest of the non-zero coordinates (since two different codewords cannot differ on less than δn coordinates).

Consequently, we have that

$$
\begin{aligned}
&\Pr[\mathrm{dist}_H(C') < \delta'] \leq && (4.1)\\
&\sum_{\substack{\delta \leq \alpha \leq 1,\\ \alpha n \in \mathbb{N}}} 2^n \cdot 2^{(\alpha-\delta)tn} \cdot 2^{H(\delta'/\alpha)\alpha tn} \cdot 2^{-\alpha tn} = \\
&\sum_{\substack{\delta \leq \alpha \leq 1,\\ \alpha n \in \mathbb{N}}} \exp\left[-tn\left(\delta - \alpha \cdot H\left(\frac{\delta'}{\alpha}\right) - \frac{1}{t}\right)\right] = \\
&\sum_{\substack{\delta \leq \alpha \leq 1,\\ \alpha n \in \mathbb{N}}} \exp\left[-tn\left((\delta - 2/t) - \alpha \cdot H\left(\frac{\delta'}{\alpha}\right) + \frac{1}{t}\right)\right].
\end{aligned}
$$

So for $\mathrm{dist}_H(C') \geq \delta'$ to hold with probability at least $1 - \exp(-n)$ it suffices to show that for any $\delta \leq \alpha \leq 1$,

$$\delta - 2/t \geq \alpha \cdot H\left(\frac{\delta'}{\alpha}\right),$$

or equivalently,

$$\alpha \cdot H^{-1}\left(\frac{\delta - 2/t}{\alpha}\right) \geq \delta'.$$

To proceed, we recall an elementary inequality implicit in [Tho83] (see Lemma 3 in [GR10] for an explicit form). Let $H^{-1} : [0,1] \to [0, \frac{1}{2}]$ be the inverse of the binary entropy function H in the domain $[0, \frac{1}{2}]$.

Fact 4.3.3. *For any $0 \leq x \leq y \leq 1$ it holds that $\frac{H^{-1}(x)}{x} \leq \frac{H^{-1}(y)}{y}$.*

We now complete the proof of the lemma.

$$
\begin{aligned}
\alpha \cdot H^{-1}\left(\frac{\delta - 2/t}{\alpha}\right) &= (\delta - 2/t) \cdot \frac{H^{-1}((\delta - 2/t)/\alpha)}{(\delta - 2/t)/\alpha} \\
&\geq (\delta - 2/t) \cdot \frac{H^{-1}(\delta - 2/t)}{\delta - 2/t} \\
&= H^{-1}(\delta - 2/t) \\
&= \delta',
\end{aligned}
$$

where the second inequality follows from Fact 4.3.3. □

Plugging in the parameters $R = 1 - H(\delta) - \gamma$ and $\delta = 1 - R - \gamma/2$, for any $\gamma \geq 4/t$, we obtain the approximate version of Thommesen's result for the GV bound. We state this formally in the following corollary.

Corollary 4.3.4. *The following holds for any $\delta \in [0, 1/2)$, $0 < \gamma < 1 - H(\delta)$, $t \geq \frac{4}{\gamma}$, and sufficiently large n. Let $C \subseteq \mathbb{F}_{2^t}^n$ be a linear code of rate $R = 1 - H(\delta) - \gamma$ and relative distance at least $1 - R - \frac{\gamma}{2}$. Let $C' \subseteq \mathbb{F}_2^{t \cdot n}$ be a code obtained from C by applying a (uniformly) random invertible $\mathbb{F}_2$-linear transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^t$ on each coordinate $i \in [n]$ of C independently. Then C' has rate R and relative distance at least δ with probability at least $1 - \exp(-n)$.*

4.3.2 LTCs approaching the GV bound

We now use Corollary 4.3.4 to show the existence of LTCs approaching the GV bound. To this end we first prove the following lemma which says that if C is locally testable then so is the code C' obtained in Corollary 4.3.4.

Lemma 4.3.5. *Let $C \subseteq \mathbb{F}_{2^t}^n$ be a code and let $C' \subseteq \mathbb{F}_2^{t \cdot n}$ be a code obtained from C by applying an invertible transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^t$ on each coordinate $i \in [n]$ of C. Suppose furthermore that C is q-locally testable in time T. Then C' is $O(q \cdot t^2)$-locally testable in time $O(T \cdot \mathrm{poly}(t))$.*[9]

Proof. Let A be the local tester for C. We will first define a local tester A'' for C' which will have small soundness. We will amplify the soundness by repeating A'' $O(t)$ times and accepting only if all the tests are accepted to get the final local tester A'. Given oracle access to $w' \in \mathbb{F}_2^{t \cdot n}$ the local tester A'' for C' runs A and answers each query i of A by inverting T_i on the i-th block of w' of length t.

[9] The tester for C' should know the transformations T_i's.

The completeness property clearly holds. To show that the soundness property holds as well suppose that $w' \notin C'$ and let $w \in \mathbb{F}_{2^t}^n$ be the word obtained from w' by inverting all the transformations T_i. As each corrupted symbol of w corresponds to at most t corrupted symbols in w', we have $\mathrm{dist}_H(w', C') \leq t \cdot \mathrm{dist}_H(w, C)$. Then A'' rejects w' with probability at least $\frac{1}{4} \cdot \mathrm{dist}_H(w, C) \geq \frac{1}{4} \cdot \mathrm{dist}_H(w', C')/t$. By running A'' $O(t)$ times, the rejection probability gets amplified to $\frac{1}{4} \cdot \mathrm{dist}_H(w', C')$. Finally, note that the overall query complexity is $O(q \cdot t^2)$ and overall running time is $O(T \cdot \mathrm{poly}(t))$. □

To obtain the final LTCs we shall also use the following theorem from [KMRS17, Theorem 1.2] which states the existence of LTCs approaching the Singleton bound.

Theorem 4.3.6 (LTCs approaching the Singleton bound)**.** *For any $\gamma > 0$, $0 < R \leq 1 - \gamma$, and $t \geq \mathrm{poly}(1/\gamma)$ there exists an infinite family $\{C_n\}_n$ of linear codes where each $C_n \subseteq \mathbb{F}_{2^t}^n$ has rate R, relative distance at least $1 - R - \gamma$, and is $(\log n)^{O(\log \log n)}$-locally testable.*

Moreover, the code C_n can be constructed by a deterministic polynomial time algorithm, and the local testing algorithm can be implemented to run in time $(\log n)^{O(\log \log n)}$.

Proof of Theorem 4.3.1. Let $\delta \in [0, \frac{1}{2})$, $0 < \gamma < 1 - H(\delta)$ and $t \in \mathbb{N}$ such that $t \geq \mathrm{poly}(4/\gamma)$ be fixed constants. By Theorem 4.3.6, there exists a family of codes $\{C_n\}_n$ with rate $R < 1 - H(\delta) - \gamma$ and relative distance at least $1 - R - \gamma/2 = H(\delta) + \frac{\gamma}{2}$ which is $(\log n)^{O(\log \log n)}$-locally testable.

Corollary 4.3.4 implies that each member of the family of codes $\{C'_n\}_n$ has relative distance at least δ with probability at least $1 - \exp(-n)$. Moreover, Lemma 4.3.5 implies that each C'_n is $O(t^2 \cdot (\log n)^{O(\log \log n)}) = (\log n)^{O(\log \log n)}$-locally testable in time $(\log n)^{O(\log \log n)}$, as we wanted. □

4.4 Approaching the GV bound via random concatenation, again

In this section, we revisit the random concatenation operation and show that it can be used to get codes approaching the GV bound with weaker hypotheses on the outer code. In Section 4.3.1 we showed that given a large alphabet code close to the Singleton bound, we can get a binary code close to the GV bound. We now show that even if we are given a large alphabet code lying slightly far from the Singleton bound (but with some decent rate-distance tradeoff and sufficiently large distance), the random concatenation operation still gives a binary code which is sufficiently close to the GV bound. In Section 4.5 we will use this to obtain LCCs nearly-approaching the GV bound with some small (but constant) rate.

Our parameters we aim to construct a code C' approaching the GV bound having relative distance at least $\frac{1}{2}-\xi$ and rate at least $\left(1 - H\left(\frac{1}{2} - \xi\right)\right)\cdot(1-\gamma)$. Thus, we have two parameters of choice, namely ξ and γ, where the first accounts for the relative distance and the latter accounts for the *multiplicative* factor approximation that we will lose on the rate of the new code. Although Lemma 4.4.1 is valid for any choices of $0 < \xi \leq \gamma/4 < 1/4$, in Section 4.5 we will set γ to be any universal constant less than 1, whereas ξ will be a small constant depending on γ and on another universal constant to be defined in Theorem 4.5.1. The parameters ξ and γ also provide constraints on the large alphabet codes which we can use in this construction, as we will see in the next paragraph.

We now give some intuition about the next lemma. Given any large alphabet code (of alphabet size 2^t) of rate R and relative distance $\delta = 1 - O(\gamma\xi)$ for $0 < \xi \leq \gamma/4 < 1/4$, the following lemma shows that we can construct a binary code with rate only a $(1-\gamma)$ factor away from the GV bound. Our approach is to once again apply Thommesen's concatenation directly to the large alphabet code. For simplicity, suppose for now that the rate of the inner codes is chosen to be $r = 1$, so the rate of the final code is R (For our local decoding algorithm we shall in fact require that r is a sufficiently small constant, so that each inner code will be decodable from a sufficiently large constant fraction of errors with high probability).

Now, if the distance δ of the outer code is very close to one, and we are interested only in multiplicative approximation to the GV bound, then the upper bound given in (4.1) on the probability that distance of final code is less than δ' becomes roughly

$$2^{(1-\delta)tn} \cdot 2^{-(1-H(\delta'))tn}, \tag{4.2}$$

where the term $2^{(1-\delta)tn}$ comes from a union bound over all codewords, and the term $2^{-(1-H(\delta'))tn}$ is an upper bound on the probability that any particular codeword has weight less than δ'. For (4.2) to be smaller than 1 we need to choose $H(\delta') \approx \delta$, and so to get $R \approx 1 - H(\delta')$ we need that $\delta \approx 1 - R$, i.e., that outer code almost satisfies the Singleton bound. In the lemma below the outer code is not sufficiently close to the Singleton bound so the above union bound fails. For this we observe that one can replace the term of $2^{(1-\delta)tn}$ in (4.2) by the smaller upper bound of 2^{Rtn} on number of codewords. Using this bound, we only need that $R \approx 1 - H(\delta')$ for (4.2) to be smaller than 1, and so we obtain the GV bound (up to multiplicative factor).

Lemma 4.4.1. *The following holds for any $0 < \gamma < 1$, $0 < \xi \leq \gamma/4$, integer t, and sufficiently large n. Let $C \subseteq \mathbb{F}_{2^t}^n$ be a linear code of rate R and relative distance at least $\delta = 1 - \frac{\gamma}{6} \cdot \xi$. Let $0 < r < 1$ be such that $C' \subseteq \mathbb{F}_2^{t \cdot n/r}$ is a code obtained from C by applying a random $\mathbb{F}_2$-linear transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^{t/r}$ on each coordinate $i \in [n]$ of C independently.*

Suppose furthermore that

$$r \cdot R \le \left(1 - H\left(\frac{1}{2} - \xi\right)\right)(1 - \gamma).$$

Then C' has relative distance at least $\frac{1}{2} - \xi$ with probability at least $1 - \exp(-n)$.

For the proof of the above lemma we shall use the Taylor expansion of the binary entropy function at half. The formula follows easily from the Taylor expansion of $\log(1 + x)$ at zero.

Fact 4.4.2. *For any $|x| \le 1$ it holds that*

$$H\left(\frac{1+x}{2}\right) = 1 - \sum_{k=1}^{\infty} \frac{x^{2k}}{(2k-1) \cdot 2k \cdot \ln 2}.$$

Additionally, we will use the following easy claim about the series above:

Claim 4.4.3. *The following holds for any $0 < x, \beta < 1$:*

$$1 - H\left(\frac{1 + x(1-\beta)}{2}\right) > (1-x)(1-\beta^2)\left[1 - H\left(\frac{1+x}{2}\right)\right]$$

Proof of Claim 4.4.3. The following holds:

$$\begin{aligned}
&\sum_{k=1}^{\infty} \frac{(1-x)(1-\beta)^2 \cdot x^{2k}}{(2k-1) \cdot 2k \cdot \ln 2} < \frac{(1-x)(1-\beta)^2}{2\ln 2} \cdot \sum_{k=1}^{\infty} x^{2k} \\
&= \frac{x^2(1-\beta)^2}{2\ln 2} \cdot \frac{1-x}{1-x^2} < \frac{x^2(1-\beta)^2}{2\ln 2} \\
&< \sum_{k=1}^{\infty} \frac{[x(1-\beta)]^{2k}}{(2k-1) \cdot 2k \cdot \ln 2}.
\end{aligned}$$

□

Proof of Lemma 4.4.1. Let $N = tn/r$. Fix a codeword $c \in C$, and note that $\mathrm{wt}(c) \ge \delta$. Let $c' \in \mathbb{F}_2^N$ be a word obtained from c by applying a random linear transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^{t/r}$ on each coordinate $i \in [n]$ of c independently. Then for each non-zero coordinate i of c it holds that the i-th block of c' of length t/r is distributed uniformly over $\mathbb{F}_2^{t/r}$, and so at least δN coordinates of c' are uniformly distributed.

Consequently it holds that

$$\begin{aligned}\Pr\left[\mathrm{wt}(c') < \frac{1}{2} - \xi\right] &\leq \sum_{i=0}^{(1/2-\xi)N} \binom{\delta N}{i} 2^{-\delta N} \\ &\leq 2^{-\left(1-H\left(\frac{1/2-\xi}{\delta}\right)\right)\cdot \delta N}.\end{aligned} \tag{4.3}$$

By union bound over all codewords $c \in C$, recalling that $|C| = 2^{tRn} = 2^{RrN}$, the above implies in turn that

$$\begin{aligned}&\Pr\left[\mathrm{dist}_H(C') < \frac{1}{2} - \xi\right] \leq 2^{RrN} \cdot 2^{-\left(1-H\left(\frac{1/2-\xi}{\delta}\right)\right)\cdot \delta N} \\ &\exp\left\{-N\left[\delta \cdot \left(1 - H\left(\frac{1/2-\xi}{\delta}\right)\right) - r \cdot R\right]\right\}.\end{aligned} \tag{4.4}$$

So for $\mathrm{dist}_H(C') \geq \frac{1}{2} - \xi$ to hold with probability at least $1 - \exp(-n)$ it suffices to show that

$$\left(1 - \frac{\xi\gamma}{6}\right) \cdot \left(1 - H\left(\frac{1/2-\xi}{1-\xi\gamma/6}\right)\right) > r \cdot R.$$

For this we use Claim 4.4.3 to compute

$$\begin{aligned}1 - H\left(\frac{1/2-\xi}{1-\xi\gamma/6}\right) &\geq 1 - H\Big((1/2-\xi)\cdot(1+\xi\gamma/3)\Big) \\ &\geq 1 - H\left(\frac{1}{2} - \xi\left(1-\gamma/6\right)\right) \\ &> (1-2\xi)\left(1-\frac{\gamma}{6}\right)^2 \cdot \left(1 - H\left(\frac{1}{2}-\xi\right)\right).\end{aligned}$$

Consequently we have that

$$\begin{aligned}
&\left(1-\frac{\xi\gamma}{6}\right)\cdot\left(1-H\left(\frac{1/2-\xi}{1-\xi\gamma/6}\right)\right)\\
&\geq\left(1-\frac{\xi\gamma}{6}\right)\cdot\left(1-\frac{\gamma}{6}\right)^2\cdot(1-2\xi)\cdot\left(1-H\left(\frac{1}{2}-\xi\right)\right)\\
&>\left(1-\frac{\gamma}{6}-\frac{\gamma}{3}-2\xi\right)\cdot\left(1-H\left(\frac{1}{2}-\xi\right)\right)\\
&\geq(1-\gamma)\cdot\left(1-H\left(\frac{1}{2}-\xi\right)\right)\\
&\geq r\cdot R,
\end{aligned}$$

where the third inequality follows by choice of $\xi \leq \gamma/4$ and the fourth inequality follows by choice of $r \cdot R \leq \left(1-H\left(\frac{1}{2}-\xi\right)\right)(1-\gamma)$. □

4.5 LCCs approaching the GV bound

In this section we prove the following theorem which implies Theorem 4.1.2 from the introduction.

Theorem 4.5.1 (LCCs approaching the GV bound)**.** *For any constants $\beta, \gamma > 0$ there exists a constant $\xi_0 = \xi_0(\beta, \gamma)$, such that for any constant $\xi > 0$ which satisfy that $\xi \leq \xi_0$ there exists an infinite family $\{C'_n\}_n$ of binary linear codes which satisfy the following. The code C'_n has block length at least n, rate $\left(1-H\left(\frac{1}{2}-\xi\right)\right)(1-\gamma)$, relative distance at least $\frac{1}{2}-\xi$, and is $\left(n^\beta \cdot \mathrm{poly}(1/\gamma'), \frac{1}{2}\cdot(\frac{1}{2}-\xi)-\gamma'\right)$-locally correctable for any $\gamma' > 0$.*

Moreover, there is a randomized polynomial time algorithm which, given n, with probability $1-\exp(-n)$, outputs a code of length n and a corrector with the above properties,[10] and the local testing algorithm runs in time $\mathrm{poly}(n^\beta, 1/\gamma')$.

[10] It might return different codes and correctors under different random choices, we are only guaranteed that the output code and corrector will have the required properties with high probability.

4.5.1 Proof overview and main ingredients

For the proof of the above theorem we shall use Lemma 4.4.1 as well as the following three lemmas.

The first lemma establishes (an alphabet independent) Johnson bound for list recovery. For a similar (alphabet dependent) statement and a proof sketch, we refer the reader to Theorem 5 in [GS01]. For completeness we provide a simple combinatorial proof of this lemma in Appendix 4.8, based on the proof of the Johnson bound for list decoding given in [Gur06].

Lemma 4.5.2 (Johnson bound for list recovery)**.** *Let $C \subseteq \Sigma^n$ be a code of relative distance at least δ. Then C is (α, ℓ, L)-list recoverable for any $\alpha < 1 - \sqrt{\ell \cdot (1-\delta)}$ with $L = \frac{\delta\ell}{(1-\alpha)^2 - \ell(1-\delta)}$.*

The second lemma gives a local list recovery algorithm for Reed-Muller codes. The algorithm is similar to the local list decoding algorithm for Reed-Muller codes from [STV01], with an additional local testing procedure that guarantees the soundness requirement in our definition of locally list recoverable codes, and is given in Section 4.6.

Lemma 4.5.3 (Local list recovery of Reed-Muller codes)**.** *There exists an absolute constant c' such that for any $\alpha, \varepsilon > 0$ and integers m, d, q, ℓ which satisfy $\alpha < 1 - c' \cdot \sqrt{\frac{\ell d}{q}}$ the Reed-Muller code $RM(m, d, q)$ is $\left(\tilde{q}, \alpha, \varepsilon, \ell, \tilde{L}\right)$-locally list recoverable with $\tilde{q} = O(q^2 \cdot \log(q/\varepsilon))$ and $\tilde{L} = O(q \log(1/\varepsilon))$. Moreover, the local list recovery algorithm can be implemented to run in* $\mathrm{poly}(m, q, \log(1/\varepsilon))$ *time.*

Finally, we shall use the following lemma which gives a distance amplification procedure for local list recovery. This procedure is similar to the distance amplification procedure for locally correctable codes from [KMRS17], and is given in Section 4.7.

On a high level, given a code C_{out} which is locally list-recoverable from a small $\alpha_{out} \ll 1$ fraction of errors, and a small code C_{in} which is (globally) list-recoverable from a large fraction of errors α_{in}, they can be combined using AEL transformation to get a new code C which is locally list-recoverable from almost α_{in} fraction of errors but doesn't use many more queries than C_{out}, and other code parameters are not significantly affected. Thus this procedure amplifies the distance from which we can list-recover without significantly worsening other parameters.

Lemma 4.5.4 (Distance amplification for local list recovery)**.** *Suppose the codes C_{out} and C_{in} exist with the following parameters:*

- *C_{out} is an $\mathbb{F}$-linear code of block length n_{out}, alphabet size σ_{out}, rate r_{out}, and relative distance δ_{out} that is $(q, \alpha_{out}, \varepsilon, \ell_{out}, L_{out})$-locally list recoverable.*
- *C_{in} is an $\mathbb{F}$-linear code of block length n_{in}, alphabet size σ_{in}, rate r_{in}, and relative distance δ_{in} that is $(\alpha_{in}, \ell_{in}, L_{in})$-(globally) list recoverable.*

There exists a $d = d(\delta_{out}, \alpha_{out}, \gamma) = (1/\delta_{out} + 1/\alpha_{out} + 1/\gamma)^{O(1)}$ such that if the parameters of C_{out} and C_{in} satisfy $n_{in} \geq d$, $\sigma_{out} = \sigma_{in}^{r_{in} \cdot n_{in}}$ and $L_{in} \leq \ell_{out}$, then there exists an $\mathbb{F}$-linear code C of block length n_{out}, alphabet size $\sigma_{in}^{n_{in}}$, rate $r_{in} \cdot r_{out}$ and relative distance at least $\delta_{in} - 2\gamma$ that is $(O(q \cdot n_{in}^2 \cdot \log(n_{in})), \alpha_{in} - \gamma, \varepsilon, \ell_{in}, L_{out})$-locally list recoverable.

Moreover,

- *If the running time of the local list recovery algorithm for C_{out} is T_{out} and the running time of the global list recovery algorithm for C_{in} is T_{in} then the running time of the local list recovery algorithm for C is*

$$O(T_{out}) + O(q \cdot T_{in}) + \text{poly}(q, n_{in}, \ell_{in}).$$

- *If the encoding times of C_{out}, C_{in} are $\hat{T}_{out}, \hat{T}_{in}$, respectively, then the encoding time of C is*

$$O(\hat{T}_{out} + n_{out} \cdot \hat{T}_{in}) + n_{out} \cdot \mathrm{poly}(n_{in}, \log(n_{out})).$$

We will prove Lemma 4.5.4 in Section 4.7. We now provide a high-level overview of the construction of the code $C' := C'_n$ of Theorem 4.5.1.

Recall that our goal is to construct a code C' of relative distance $1/2 - \xi$ that approximately satisfies the GV bound (up to multiplicative factor of $1 - \gamma$), and which additionally can be locally corrected from half its minimum distance. Via the Guruswami-Indyk list decoding approach, the latter requirement is satisfied if C' is locally list decodable from above half the minimum distance, say from $1/4$ fraction of errors. Indeed, if this is the case then one can locally list decode C' to obtain a short list of candidate codewords, then estimate the distance of each of these codewords from the received word using a few more queries, and finally output the local corrector that corresponds to the unique closest codeword. So it suffices to construct a code C' of relative distance $1/2 - \xi$ that approximately satisfies the GV bound, and additionally is locally list decodable from $1/4$ fraction of errors.

We shall construct the code C' by applying random concatenation on a carefully designed outer code C. To this end, first observe that by Lemma 4.4.1 the code C' will approximately satisfy the GV bound if the code C has reasonable rate vs. distance tradeoff. Specifically, assuming that rate of inner codes is constant, we need C to have relative distance $1 - O(\gamma\xi)$ and rate $\theta(\xi^2)$.

Next observe that in order for the concatenated code C' to be locally list decodable from $1/4$ fraction of errors it suffices that (1) Most of the inner random codes are globally list decodable from slightly more than $1/4$ fraction of errors; and (2) The outer code C is locally list recoverable from input lists of size equal to that of output lists of inner codes (where a small constant fraction of these input lists may be erroneous). Indeed, if this is the case then one can locally list decode C' by applying the local algorithm that locally list recovers the outer code C, and answering each of its queries by globally list decoding the corresponding inner code. On the other hand, we do not care too much about the rates of outer and inner codes, as long as they are constant, since we only want C' to have a small constant rate.

Now, by the Johnson bound for list recovery (Lemma 4.5.2), most of the inner codes are list decodable from more than $1/4$ fraction of errors with constant size output lists if the rate of inner codes is a sufficiently small constant. So we are left with the task of obtaining a code C of relative distance $1 - O(\gamma\xi)$ and rate $\theta(\xi^2)$ that is locally list recoverable from constant size input lists and a small constant fraction of errors.

One attempt to obtain such a code C may be to use a Reed-Muller code of relative distance $1 - O(\gamma\xi)$ that is locally list recoverable using n^β queries. The required distance can be guaranteed by picking the degree d to be $O(\gamma\xi) \cdot |\mathbb{F}|$, while the required query complexity can be ensured by picking field size $|\mathbb{F}|$ and number of variables m to be roughly n^β and $1/\beta$, respectively (so we can locally list recover by decoding on lines of size roughly n^β). However, such a code would have terrible rate of the form $(d/|\mathbb{F}|)^m \approx (\gamma\xi)^{1/\beta} \ll \theta(\xi^2)$.

To remedy the above situation we use a Reed-Muller code of higher rate and lower distance and then amplify its distance using the AEL transformation (Lemma 4.5.4). Specifically, we apply this transformation with the outer code C_{out} being a Reed-Muller code of constant relative distance that is locally list recoverable using n^β queries (so $d = O(|\mathbb{F}|)$, $|\mathbb{F}| \approx n^\beta$, $m \approx 1/\beta$)[11]. Thus C_{out} has rate $1/c_\beta$ for some constant depending only on β. The inner code C_{in} on the other hand will be

[11] To obtain sublinear time decoding, in addition to sublinear query complexity, we will in fact concatenate the Reed-Muller code with another Reed-Solomon code with appropriate parameters to slightly reduce the alphabet size and enable fast brute force decoding of inner codes; See Section 4.5.2 for more details.

a Reed-Solomon code of rate $c_\beta \cdot \theta(\xi^2)$ and relative distance $1 - c_\beta \cdot \theta(\xi^2)$. Lemma 4.5.4 then implies that the rate of C is the product of rates of C_{out} and C_{in} which is $\theta(\xi^2)$, and most importantly the code C inherits the relative distance of C_{in} which is $1 - c_\beta \cdot \theta(\xi^2) \geq 1 - O(\gamma\xi)$ (where the latter inequality holds if ξ is chosen to be sufficiently small compared to β and γ).

For completeness, we provide full details of the construction in Section 4.5.2. In Section 4.5.3, we analyze the rate and relative distance of C'. In Section 4.5.4, we show that the code C' is locally list decodable from 1/4 fraction of errors. We then use this property in Section 4.5.5 to show that C' is locally correctable from half the GV bound. This shows that C' satisfies the local correction requirement.

4.5.2 Construction of C'

In what follows we present the construction of the code $C' := C'_n$. To this end we first set some parameters and then describe the construction of the codes C_{in},C_{out}, C and C'. For better readability, in what follows we will denote each variable v that is set to some absolute constant (independent of $\beta, \gamma, \xi, \gamma'$ and n) by $\hat{v}$. In what follows we will assume that $n^{\beta/4}$ is a sufficiently large power of 2.

Parameter setting. Let $\frac{1}{4} < \hat{\alpha}_0 < \frac{1}{2}$ be an arbitrary constant, and note that by the Johnson bound for list decoding (Theorem 4.2.1) there exist a constant $\hat{\delta}_0 \in [0, 1/2)$ and an integer $\hat{L}_0$ such that any binary code of relative distance at least $\hat{\delta}_0$ is $(\hat{\alpha}_0, \hat{L}_0)$-list decodable. We will choose the parameters below so that the random binary codes encoded by the T_i's of Lemma 4.4.1 will have relative distance at least $\hat{\delta}_0$ (with high probability), and the code C will be locally list recoverable from input lists of size $\hat{L}_0$ and sufficiently large constant fraction of errors. It will then follow that the final code C' is locally list decodable from $\frac{1}{4}$ fraction of errors and consequently locally correctable from half the GV bound.

The code C_{in}. Choose an arbitrary constant $0 < \hat{\alpha}_{in} < 1$ such that $\hat{\alpha}_{in} \cdot \hat{\alpha}_0 > \frac{1}{4}$ (such $\hat{\alpha}_{in}$ exists since $\hat{\alpha}_0 > \frac{1}{4}$), and note that by the Johnson bound for list recovery (Lemma 4.5.2) there exist a constant $\hat{\delta}_{in} \in (0,1)$ and an integer $\hat{L}_{in}$ such that any code of relative distance at least $\hat{\delta}_{in}$ is $(\hat{\alpha}_{in}, \hat{L}_0, \hat{L}_{in})$-list recoverable.

Let $\sigma_{in} \geq n_{in}$ be growing functions of n such that $\sigma_{in}^{n_{in}} = n^{\beta/4}$ and σ_{in} is a power of 2, and let $\delta_{in} = \delta_{in}(\beta, \gamma, \xi) < 1$ be a constant to be determined later on which satisfies that $\delta_{in} \geq \hat{\delta}_{in}$. Let C_{in} be a Reed-Solomon code of block length n_{in}, alphabet size σ_{in}, relative distance δ_{in}, and rate $r_{in} := 1 - \delta_{in}$. Then by the above discussion the code C_{in} is $(\hat{\alpha}_{in}, \hat{L}_0, \hat{L}_{in})$-(globally) list recoverable in time $\mathrm{poly}(n^{\beta})$ (via brute force).[12]

The code C_{out}. The code C_{out} will be a concatenation of an outer Reed-Muller code C'_{out} with an inner Reed-Solomon code C''_{out}. The purpose of the concatenation step is to reduce the alphabet size of the code C_{out} from $n^{\beta/4}$ to $n^{r_{in} \cdot \beta/4}$, so that the code C obtained by applying Lemma 4.5.4 on C_{out} and C_{in} (see Section 4.5.2 below) would have alphabet size $n^{\beta/4} = o(n)$ (as opposed to $n^{\beta/(4r_{in})} = \omega(n)$). This latter property is needed in turn to ensure that the random inner codes used for the construction of C' (see Section 4.5.2 below) will have $n^{\beta/4}$ different codewords each, and consequently can be brute force list decoded in sublinear time.

We start by defining the inner Reed-Solomon code C''_{out}. Choose an arbitrary constant $0 < \hat{\alpha}''_{out} < 1$, and note that by the Johnson bound for list recovery (Lemma 4.5.2) there exist a constant $\hat{\delta}''_{out} \in (0,1)$ and an integer $\hat{L}''_{out}$ such that any code of relative distance at least $\hat{\delta}''_{out}$ is $(\hat{\alpha}''_{out}, \hat{L}_{in}, \hat{L}''_{out})$-list recoverable. Let C''_{out} be a Reed-Solomon code of relative distance $\hat{\delta}''_{out}$, rate $\hat{r}''_{out} := 1 - \hat{\delta}''_{out}$, block length $1/(r_{in}\hat{r}''_{out})$ and alphabet size $n^{r_{in} \cdot \beta/4}$. Then by the above the code C''_{out} is $(\hat{\alpha}''_{out}, \hat{L}_{in}, \hat{L}''_{out})$-(globally) list recoverable in time $\mathrm{poly}(n^{\beta})$ (via brute force).

[12] We do not need to use Guruswami-Sudan's list recovery algorithm for Reed-Solomon codes because the running time of our final code will already have a $\mathrm{poly}(n^{\beta})$ dependence due to the increase in query complexity. See the running time of Lemma 4.5.4.

Next we define the outer Reed-Muller code C'_{out}. Choose an arbitrary constant $0 < \hat{\alpha}'_{out} < 1$, and note that by Lemma 4.5.3 there exists a constant $\hat{\delta}'_{out} \in (0,1)$ such that a Reed-Muller code of relative distance $\hat{\delta}'_{out}$, block length $n \cdot r_{in}\hat{r}''_{out}$ and alphabet size $n^{\beta/4}$ is $(q'_{out}, \hat{\alpha}'_{out}, \varepsilon'_{out}, \ell'_{out}, L'_{out})$-locally list recoverable for $q'_{out} = n^{\beta/2}\text{polylog}n$, $\varepsilon'_{out} = 1/n$, $\ell'_{out} = \hat{L}''_{out}$, and $L'_{out} = n^{\beta/4}\text{polylog}n$ in time poly(n^{β}). Let C'_{out} be a Reed-Muller code of block length $n \cdot r_{in} \cdot \hat{r}''_{out}$, alphabet size $n^{\beta/4}$, relative distance $\hat{\delta}'_{out}$, and rate $r'_{out} = r'_{out}(\beta)$. Then the code C'_{out} is $(q'_{out}, \hat{\alpha}'_{out}, \varepsilon'_{out}, \ell'_{out}, L'_{out})$-locally list recoverable in time poly(n^{β}).

Finally, let C_{out} be the code obtained by concatenating the outer Reed-Muller code C'_{out} with the inner Reed-Solomon code C''_{out}. Then it can be verified that C_{out} is an $\mathbb{F}_2$-linear code of block length n, alphabet size $n^{r_{in}\cdot\beta/4}$, relative distance $\hat{\delta}'_{out} \cdot \hat{\delta}''_{out}$, rate $r'_{out} \cdot \hat{r}''_{out}$, and in addition it is $(q_{out}, \alpha_{out}, \varepsilon_{out}, \ell_{out}, L_{out})$-locally list recoverable with $q_{out} = n^{\beta/2}\text{polylog}n$, $\alpha_{out} = \hat{\alpha}'_{out} \cdot \hat{\alpha}''_{out}$, $\varepsilon_{out} = \frac{1}{n}$, $\ell_{out} = \hat{L}_{in}$ and $L_{out} = n^{\beta/4}\text{polylog}n$ in time poly(n^{β}) by emulating the local list recovery algorithm of C'_{out} in the natural way.

The code C. Let C be the code guaranteed by invoking Lemma 4.5.4 with $\xi = \frac{\gamma}{24}\cdot\xi$ for the codes C_{out}, C_{in} and $d = d\left(\hat{\delta}'_{out} \cdot \hat{\delta}''_{out}, \hat{\alpha}'_{out} \cdot \hat{\alpha}''_{out}, \frac{\gamma}{24} \cdot \xi\right)$ (note that $n_{in} \geq d$ when n is large enough, $\sigma_{out} = \sigma_{in}^{r_{in}\cdot n_{in}}$, and $L_{in} = \hat{L}_{in} = \ell_{out}$). Then C is an $\mathbb{F}_2$-linear code of block length n, alphabet size $n^{\beta/4}$, relative distance at least $\delta_{in} - \frac{\gamma}{12}\cdot\xi$, rate $r_{in}\cdot r'_{out}\cdot\hat{r}''_{out}$, and is $(q, \alpha, \varepsilon, \ell, L)$-locally list recoverable with $q = n^{\beta/2}\text{polylog}n$, $\alpha = \hat{\alpha}_{in} - \frac{\gamma}{12} \cdot \xi$, $\varepsilon = \frac{1}{n}$, $\ell = \hat{L}_0$ and $L = n^{\beta/4}\text{polylog}n$ in time poly(n^{β}).

The code C'. Let $t = \log(n^{\beta/4})$ and let $\hat{r}_0$ be a constant such that a random binary linear code of rate $\hat{r}_0$ and block length t has relative distance at least $\hat{\delta}_0$ with probability at least $1 - \exp(-t)$ (such $\hat{r}_0$ exists by Fact 4.2.3). Considering each symbol of the code C as an element of $\mathbb{F}_{2^t}$, let $C' \subseteq \mathbb{F}_2^{t\cdot n/\hat{r}_0}$ be a code obtained from

C by applying a random $\mathbb{F}_2$-linear transformation $T_i : \mathbb{F}_{2^t} \to \mathbb{F}_2^{t/\hat{r}_0}$ on each coordinate $i \in [n]$ of C independently.

Choice of δ_{in} and ξ_0. Finally, set

$$\delta_{in} = \delta_{in}(\beta, \gamma, \xi) = 1 - \frac{\left(1 - H\left(\frac{1}{2} - \xi\right)\right)(1-\gamma)}{r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0},$$

and note that δ_{in} can be chosen this way because C_{in} has super-constant block length and alphabet size, and hence can attain any constant relative distance that is desired.

Also, set

$$\xi_0 := \min \begin{cases} \frac{1-\hat{\delta}_{in}}{5} \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0, \\ \frac{\gamma}{60} \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0, \\ \frac{\gamma}{4}, \\ \hat{\alpha}_{in} - \frac{1}{4\hat{\alpha}_0}, \end{cases} \tag{4.5}$$

and note that ξ_0 depends only on β and γ (recalling that r'_{out} depends only on β).

Moreover, the choice of $\xi_0 \leq \frac{1-\hat{\delta}_{in}}{5} \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0$ guarantees that $\delta_{in} \geq \hat{\delta}_{in}$ whenever $\xi \leq \xi_0$, as required in Section 4.5.2. We additionally require that $\xi_0 \leq \frac{\gamma}{60} \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0$ and $\xi_0 \leq \gamma/4$ so the code C will satisfy the requirements of our random concatenation Lemma 4.4.1 (see Section 4.5.3), and that $\xi_0 \leq \alpha_{in} - \frac{1}{4\hat{\alpha}_0}$ for the code C to be list recoverable from sufficiently large fraction of errors (see Section 4.5.4).

4.5.3 Rate and relative distance of C'

We first show that C' has the desired rate and distance.

Claim 4.5.5. *The code C' is a binary linear code of block length at least n and rate $\left(1 - H\left(\frac{1}{2} - \xi\right)\right)(1-\gamma)$. Moreover, C' has relative distance at least $\frac{1}{2} - \xi$ with probability $1 - \exp(-n)$ over the choice of the T_i's.*

Proof. By construction C' is a binary linear code of block length $tn/\hat{r}_0 \geq n$ and rate

$$\begin{aligned} r_{in} \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0 &= (1-\delta_{in}) \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0 \\ &= \left(1 - H\left(\frac{1}{2} - \xi\right)\right)(1-\gamma). \end{aligned}$$

To show that C' has the required distance we use Lemma 4.4.1. To see that the conditions of this lemma hold note first that the relative distance of C is at least

$$\begin{aligned} \delta_{in} - \frac{\gamma}{12} \cdot \xi &= 1 - \frac{\left(1 - H\left(\frac{1}{2} - \xi\right)\right)(1-\gamma)}{r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0} - \frac{\gamma}{12} \cdot \xi \\ &\geq 1 - \frac{\gamma}{6} \cdot \xi, \end{aligned}$$

where the inequality is by our choice of $\xi_0 \leq \frac{\gamma}{60} \cdot r'_{out} \cdot \hat{r}''_{out} \cdot \hat{r}_0$ in (4.5). Moreover, by choice of $\xi_0 \leq \gamma/4$ in (4.5) we have that $\xi \leq \gamma/4$. Thus Lemma 4.4.1 implies that C' has relative distance at least $\frac{1}{2} - \xi$ with probability at least $1 - \exp(-n)$. □

4.5.4 Local list decoding of C'

Next we show that C' can be locally list decoded from 1/4 fraction of errors.

Claim 4.5.6. *The code C' is $(q', \alpha', \varepsilon', L')$-locally list decodable for $q' = n^{\beta/2} \cdot \mathrm{polylog} n$, $\alpha' = \frac{1}{4}$, $\varepsilon' = \frac{1}{n}$, and $L' = n^{\beta/4} \cdot \mathrm{polylog} n$ with probability $1 - \exp(-n)$ over the choice of the T_i's. Moreover, the local list decoder of C' can be implemented to run in time $\mathrm{poly}(n^{\beta})$.*

Proof. By construction the code C is $(q, \alpha, \varepsilon, \ell, L)$-locally list recoverable with $q = n^{\beta/2}\mathrm{polylog} n$, $\alpha = \hat{\alpha}_{in} - \frac{\gamma}{12} \cdot \xi$, $\varepsilon = \frac{1}{n}$, $\ell = \hat{L}_0$ and $L = n^{\beta/4}\mathrm{polylog} n$ in time $\mathrm{poly}(n^{\beta})$, where $\hat{\alpha}_{in} > \frac{1}{4\hat{\alpha}_0}$, and each T_i is $(\hat{\alpha}_0, \hat{L}_0)$-list decodable with probability at least $1 - \exp(-t) = 1 - o_n(1)$. The local list decoding algorithm A' for C' will run the local list recovery algorithm A for C and answer the queries of A by list decoding the T_i's corresponding to the queries of A. Details follow.

Let A be the algorithm that local list recovers C. On oracle access to $w \in \mathbb{F}_2^{tn/\hat{r}_0}$ the algorithm A' that local list decodes C' runs A and whenever A asks for some coordinate $i \in [n]$, the algorithm A' list decodes the i-th block of w of length $t/\hat{r}_0$ from $\hat{\alpha}_0$ fraction of errors, and feeds the messages corresponding to the $\hat{L}_0$ codewords in the output list as an answer to the query of A. Let $A_1, \ldots, A_L$ be the resulting output algorithms of A for $L = n^{\beta/4} \cdot \text{polylog} n$. Then A' outputs L algorithms $A'_1, \ldots, A'_L$ where each algorithm A'_j is defined as follows

To decode the k'-th coordinate in the k-th block of C' of length $t/\hat{r}_0$, the algorithm A'_j runs the algorithm A_j on input coordinate k. As above, whenever A_j asks for some coordinate $i \in [n]$, the algorithm A'_j list decodes the i-th block of w of length $t/\hat{r}_0$ from $\hat{\alpha}_0$ fraction of errors, and feeds the messages corresponding to the $\hat{L}_0$ codewords in the output list as an answer to the query of A_j. Let $\sigma \in \mathbb{F}_{2^t}$ be the output symbol of A_j. Then the algorithm A'_j outputs the k'-th bit of $T_k(\sigma) \in \mathbb{F}_2^{t/\hat{r}_0}$.

The query complexity of A' is at most $n^{\beta/2} \cdot \text{polylog} n \cdot t = n^{\beta/2} \cdot \text{polylog} n$, and the output list size of A' is $n^{\beta/4}\text{polylog} n$. Each block of w of length $t/\hat{r}_0$ can be brute-force list decoded in time $2^{t/\hat{r}_0} = \text{poly}(n^\beta)$ and so the overall running time is $\text{poly}(n^\beta)$. The soundness property clearly holds.

To see that the completeness property holds as well note that if $\text{dist}_H(w, c') \leq \frac{1}{4}$ for some $c' \in C'$, then by Markov's inequality for at most $1/(4\hat{\alpha}_0)$ fraction of $i \in [n]$ it holds that the i-th block of w of length $t/\hat{r}_0$ differs from the i-th block of c' of length $t/\hat{r}_0$ by more than $\hat{\alpha}_0$ fraction of the coordinates. Moreover, since each T_i is $(\hat{\alpha}_0, \hat{L}_0)$-list decodable with probability at least $1 - o_n(1)$, with probability at least $1 - \exp(-n)$ it holds that at most $\xi/2$ fraction of the T_i's do not have this property. This implies in turn that the list decoding of the T_i's fails on at most $\xi/2 + 1/(4\hat{\alpha}_0)$ fraction of the blocks. The completeness then follows since C is locally list recoverable from $\hat{\alpha}_{in} - \frac{\gamma}{12} \cdot \xi > \xi/2 + 1/(4\hat{\alpha}_0)$ fraction of errors (where the inequality holds by choice of $\xi_0 \leq \hat{\alpha}_{in} - 1/(4\hat{\alpha}_0)$ in (4.5)) and input list size $\hat{L}_0$. □

4.5.5 Local correction of C'

Finally, we show that the code C' is locally correctable from half the GV bound.

Claim 4.5.7. *For any $\gamma' > 0$ the code C' is $\left(n^\beta \cdot \text{poly}(1/\gamma'), \frac{1}{2} \cdot (\frac{1}{2} - \xi) - \gamma'\right)$-locally correctable with probability $1 - \exp(-n)$ over the choice of the T_i's. Moreover, the local corrector of C' can be implemented to run in time $\text{poly}(n^\beta, 1/\gamma')$.*

Proof. By claims 4.5.5 and 4.5.6 we have that C' has relative distance at least $\frac{1}{2} - \xi$ and in addition it is $(q', \alpha', \varepsilon', L')$-locally list decodable for $q' = n^{\beta/2} \cdot \text{polylog} n$, $\alpha' = \frac{1}{4}$, $\varepsilon' = \frac{1}{n}$, and $L' = n^{\beta/4} \cdot \text{polylog} n$ in time $\text{poly}(n^\beta)$ with probability at least $1 - \exp(-n)$ over the choice of the T_i's. In what follows assume that these two properties hold, we will show that in this case C' is also $\left(n^\beta \cdot \text{poly}(1/\gamma'), \frac{1}{2} \cdot (\frac{1}{2} - \xi) - \gamma'\right)$-locally correctable in time $\text{poly}(n^\beta, 1/\gamma')$ for any $\gamma' > 0$.

Let A' be the algorithm that local list decodes C'. By increasing the query complexity of A' by a multiplicative factor of $\text{polylog} n$ we may assume that both the completeness and soundness properties of A' hold with success probability $1 - \frac{1}{n^2}$ instead of $\frac{2}{3}$. On oracle access to $w \in \mathbb{F}_2^{tn/\hat{r}_0}$ and input coordinate $i \in [tn/\hat{r}_0]$, the algorithm $\tilde{A}$ that local corrects C' first runs A' with oracle access to w, let $A'_1, \ldots, A'_L$ be the output algorithms for $L = n^{\beta/4} \text{polylog} n$. The algorithm $\tilde{A}$ then runs each of the A'_j's on a random subset $S_j \subseteq [tn/\hat{r}_0]$ of coordinates of size $O(\log n/(\gamma')^2)$, and computes the fraction of coordinates δ_j in S_j on which the decoded values differ from the values of w. Finally, the algorithm $\tilde{A}$ finds some A'_j for which $\delta_j \leq \frac{1}{2} \cdot (\frac{1}{2} - \xi)$ (if such A'_j exists), and uses A'_j to decode the input coordinate i.

The query complexity of $\tilde{A}$ is

$$n^{\beta/4} \cdot \text{polylog} n \cdot O(\log n/(\gamma')^2) \cdot n^{\beta/2} \cdot \text{polylog} n \cdot O(\log n),$$

where the first factor of $n^{\beta/4} \text{polylog} n$ comes from the number of codewords in the list, which is at most $n^\beta \cdot \text{poly}(1/\gamma')$ for sufficiently large n, and the running time of $\tilde{A}$ is $\text{poly}(n^\beta, 1/\gamma')$. Next we show that $\tilde{A}$ satisfies the required local correction guarantees.

Let $c' \in C'$ be the (unique) codeword which satisfies that $\text{dist}_H(w, c') \leq \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right) - \gamma'$. We shall show below that with probability $1 - o(1)$, there exists some A'_j that computes c' and satisfies that $\delta_j \leq \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$, and on the other hand, any A'_j which does not compute c' satisfies that $\delta_j > \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$. This will imply in turn that the algorithm $\tilde{A}$ will succeed in decoding the input coordinate with probability $1 - o(1) \geq \frac{2}{3}$ as required.

We first show that with probability at least $1 - \frac{3}{n}$ there exists some A'_j that computes c' and satisfies that $\delta_j \leq \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$. To see this note that by the completeness property of A' and since $\text{dist}_H(w, c') \leq \frac{1}{4}$, with probability at least $1 - \frac{1}{n}$ over the randomness of A' there exists some A'_j that computes c'. In this case, by union bound with probability at least $1 - \frac{1}{n}$ it holds that each decoded coordinate of A'_j in S_j equals to the corresponding coordinate in c'. Furthermore, by Chernoff bound with probability at least $1 - \frac{1}{n}$ it holds that w and c' differ on S_j by at most $\frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$ fraction of the coordinates. Consequently, with probability at least $1 - \frac{3}{n}$ it holds that $\delta_j \leq \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$.

Next we show that with probability at least $1 - \frac{3}{n}$, any A'_j which does not compute c' satisfies that $\delta_j > \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$. For this note that by the soundness property of A', with probability at least $1 - \frac{1}{n}$ over the randomness of A', for every such A'_j there exists a codeword $\tilde{c} \in C' \setminus \{c'\}$ such that A'_j computes $\tilde{c}$. As above, by union bound this implies in turn that with probability at least $1 - \frac{1}{n}$ it holds that each decoded coordinate of A'_j in S_j equals to the corresponding coordinate on $\tilde{c}$. On the other hand, since C' has relative distance at least $\frac{1}{2} - \xi$ and $\text{dist}_H(w, c') \leq \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right) - \gamma'$ we have that $\text{dist}_H(w, \tilde{c}) > \frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right) + \gamma'$, and so by Chernoff bound with probability at least $1 - \frac{1}{n}$ it holds that w and $\tilde{c}$ differ on S_j by more than $\frac{1}{2} \cdot \left(\frac{1}{2} - \xi\right)$ fraction of the coordinates. Consequently, with probability at least $1-\frac{3}{n}$ it holds that $\delta_j > \frac{1}{2}\cdot\left(\frac{1}{2}-\xi\right)$ for any such A'_j which completes the proof of the claim.

□

4.6 Local list recovery of Reed-Muller codes

In this section we prove Lemma 4.5.3 which we recall here.

Lemma 4.5.3 (Local list recovery of Reed-Muller codes)**.** *There exists an absolute constant c' such that for any $\alpha, \varepsilon > 0$ and integers m, d, q, ℓ which satisfy $\alpha < 1-c'\cdot\sqrt{\frac{\ell d}{q}}$ the Reed-Muller code $RM(m,d,q)$ is $\left(\tilde{q}, \alpha, \varepsilon, \ell, \tilde{L}\right)$-locally list recoverable with $\tilde{q} = O(q^2 \cdot \log(q/\varepsilon))$ and $\tilde{L} = O(q\log(1/\varepsilon))$. Moreover, the local list recovery algorithm can be implemented to run in* $\text{poly}(m, q, \log(1/\varepsilon))$ *time.*

For the proof of the above lemma we shall need the following two lemmas. The first lemma from [GS92] gives a local correction procedure for Reed-Muller codes (see. e.g., Proposition 2.6. in [Yek12]).

Lemma 4.6.1 (Local correction of Reed-Muller codes)**.** *There exists an absolute constant $r_0 > 0$ such that for any integers m, d, q which satisfy that $\frac{d}{q} \leq r_0$ the Reed-Muller code $RM(m,d,q)$ is $\left(q, \frac{1}{4}\right)$-locally correctable.*

In other words, there exists a randomized q-query algorithm Corr *such that given oracle access to a function $f : \mathbb{F}_q^m \to \mathbb{F}_q$ which agrees with a degree d polynomial $p : \mathbb{F}_q^m \to \mathbb{F}_q$ on at least $3/4$ fraction of inputs, and given $x \in \mathbb{F}_q^m$,*

$$\Pr[\text{Corr}^f(x) = p(x)] \geq \frac{2}{3},$$

where the probability is over the internal randomness of Corr*. Moreover* Corr *runs in* $\text{poly}(m, q)$ *time.*

The second lemma gives a tolerant local testing procedure for Reed-Muller codes. A tolerant local testing algorithm is a local testing algorithm that has the additional property of accepting all words which are sufficiently close to the code. Formally it is defined as follows.

Definition 4.6.2. *Let $0 < \alpha < \alpha' < 1$. We say that a code $C \subseteq \Sigma^n$ is (q, α, α')-tolerant locally testable if there exists a randomized algorithm A that satisfies the following requirements:*

- ***Input:*** *A gets oracle access to a string $w \in \Sigma^n$.*
- ***Query complexity:*** *A makes at most q queries to the oracle w.*
- ***Completeness:*** *If $\mathrm{dist}_H(w, C) \leq \alpha$, then A accepts with probability at least $\frac{2}{3}$.*
- ***Soundness:*** *If $\mathrm{dist}_H(w, C) \geq \alpha'$, then A rejects with probability at least $\frac{2}{3}$.*

Lemma 4.6.3 (Tolerant local testing of Reed-Muller codes)**.** *There exist absolute constants $r_0 > 0$ and $0 < \alpha_0 < 1/4$ such that for any integers m, d, q which satisfy that $\frac{d}{q} \leq r_0$ the Reed-Muller code $RM(m, d, q)$ is $(O(q), \alpha_0, 1/4)$-tolerant locally testable. Moreover the running time of the tester is $\mathrm{poly}(m, q)$.*

The proof of the above lemma is based on the robust local testing procedure for Reed-Muller codes from [FS95], and is deferred to Section 4.6.2. As an anonymous reviewer has pointed out, it is not necessary to use tolerant testing here. We can use a not necessarily tolerant tester to get local list recovery as in Lemma 4.5.3 but with a worse query complexity.

4.6.1 Proof of Lemma 4.5.3

The proof follows the lines of the algorithm for the local list decoding of Reed-Muller codes from [STV01], we need an additional local testing procedure that guarantees the soundness requirement in our definition of locally list recoverable codes (Definition 4.2.6). Let $S : \mathbb{F}_q^m \to \binom{\mathbb{F}_q}{\ell}$. We would like to construct a list of oracle algorithms which compute all codewords $p : \mathbb{F}_q^m \to \mathbb{F}_q$ such that $p(x) \in S(x)$ for at least $\beta := 1 - \alpha$ fraction of $x \in \mathbb{F}_q^m$. Moreover every oracle algorithm in the list should compute some codeword. Now we describe an algorithm for this task. We will begin by defining the following (deterministic) sub-algorithm which will be used in the main algorithm.

The algorithm receives as parameters $\beta \in [0, 1]$, $z \in \mathbb{F}_q^m$ and $a \in \mathbb{F}_q$.

Algorithm 1 $\mathcal{M}^S_{z,a,\beta}(x)$ given input $x \in \mathbb{F}_q^m$

```
1: Let $u_{z,x}(t) = (1 - t)z + tx$ denote the line through the points $z, x$. [13]
2: Find the list $h_1, \cdots, h_r$ that includes all univariate degree $d$ polynomials $p : \mathbb{F}_q \to \mathbb{F}_q$ such that $p(t) \in S(u_{z,x}(t))$ for at least $\beta/2$ fraction of $t \in \mathbb{F}_q$.
3: If there exists a unique $i$ such that $h_i(0) = a$, then output $h_i(1)$, else output 'FAIL'.
```

The parameters z, a in Algorithm 1 must be thought of as advice which tells us that the polynomial takes the value $a \in \mathbb{F}$ at the point $z \in \mathbb{F}_q^m$. The following claim makes this intuition precise.

Claim 4.6.4. *Let* $0 < \tau < 1$, $1 \geq \beta \geq \frac{16}{\tau}\sqrt{\ell d/q}$, *and let* $p : \mathbb{F}_q^m \to \mathbb{F}_q$ *be a degree* d *polynomial which agrees with* S *in at least* β *fraction of inputs, then the following are true.*

1. $\mathcal{M}^S_{z,a,\beta}$ *makes at most* q *queries to* S *and runs in* $\mathrm{poly}(m, q)$ *time.*
2. $\Pr_z\left[\Pr_x\left[\mathcal{M}^S_{z,p(z),\beta} \text{ computes } p \text{ at } x\right] \geq 1 - \tau\right] \geq \frac{1}{2}$.

Proof. The number of queries is q since the algorithm only queries points on a line. Also Step 2 of the algorithm, which is the most expensive step, can be implemented

[13] If $z = x$, choose a random line through z.

in $\mathrm{poly}(m, q)$ time by Theorem 4.2.2. Now we will prove (2). By Markov inequality,

$$\begin{aligned}&\Pr_z\left[\Pr_x\left[\mathcal{M}^S_{z,p(z),\beta} \text{ does not compute } p \text{ at } x\right] \geq \tau\right]\\ &\leq \frac{1}{\tau}\Pr_{z,x}\left[\mathcal{M}^S_{z,p(z),\beta} \text{ does not compute } p \text{ at } x\right].\end{aligned}$$

To bound the probability that $\mathcal{M}^S_{z,p(z),\beta}$ does not compute p at x, let us define the following two bad events and bound their probabilities.

Event A: $\nexists i \in [r]$ *s.t.* $h_i = p|_{u_{z,x}}$

This will happen only if p does not agree with S on at least $\beta/2$ fraction of points on the line $u_{z,x}$. But we know that p has agreement at least β with S on the entire space. Since z, x are uniformly random, the set of points on the line $u_{z,x}$ are pairwise independent. So we can use Chebychev's inequality to bound the probability of this event. Let $X_1, \ldots, X_q$ be indicator random variables where $X_i = 1$ iff p agrees with S on the i^{th} point of the line $u_{z,x}$. Let μ be the fraction of points where p agrees with S, we know that $\mathbb{E}[X_i] = \mu \geq \beta$ and $\mathrm{Var}(X_i) \leq \mathbb{E}[X_i]$.

$$\begin{aligned}\Pr[A] &= \Pr[\nexists i \in [r] \ s.t. \ h_i = p|_{u_{z,x}}]\\ &\leq \Pr\left[\frac{\sum_i X_i}{q} \leq \frac{\beta}{2}\right]\\ &\leq \Pr\left[\frac{\sum_i X_i}{q} \leq \frac{\mu}{2}\right]\\ &\leq \Pr\left[\left|\frac{\sum_i X_i}{q} - \mu\right| \leq \frac{\mu}{2}\right]\\ &\leq \frac{4\mathrm{Var}(X_1)}{q\mu^2} \leq \frac{4}{q\mu}\\ &\leq \frac{4}{q\beta} \leq \frac{\tau}{4\sqrt{\ell d q}} \leq \frac{\tau}{4}.\end{aligned}$$

Event B: $\exists i \in [r]$ *s.t.* $h_i \neq p|_{u_{z,x}}$ *and* $h_i(0) = p(z)$

Since the list of polynomials $h_1, \cdots, h_r$ depends only on the line through z, x, we can think of the random process as first picking a random line u and then picking two random points $t_1, t_2 \in \mathbb{F}_q$ and letting $z = u(t_1), x = u(t_2)$. If $h_i \neq p|_u$, then they agree on at most d points of u, so $\Pr_{t_1}[h_i(t_1) = p(u(t_1))] \leq \frac{d}{q}$. By union bound,

$$\begin{aligned}\Pr[B] &= \Pr\left[\exists i \in [r] \ s.t. \ h_i \neq p|_{u_{z,x}} \ and \ h_i(0) = p(z)\right] \\ &\leq \frac{rd}{q}.\end{aligned}$$

By applying the Johnson bound for list recovery (Lemma 4.5.2) to the Reed-Solomon code of degree d on the line u ,

$$r \leq \frac{\ell}{(\beta/2)^2 - \ell d/q} \leq \frac{q/d}{(8/\tau)^2 - 1}.$$

Combining the above bounds we get,

$$\Pr[B] \leq 1/((8/\tau)^2 - 1) \leq \tau/8.$$

Clearly, if events A, B do not happen, then $\mathcal{M}^S_{z,p(z),\beta}$ will compute p at x. Therefore

$$\begin{aligned}&\Pr_{z,x}\left[\mathcal{M}^S_{z,p(z),\beta} \text{ does not compute } p \text{ at } x\right] \\ &\leq \Pr[A] + \Pr[B] \leq \frac{\tau}{2}.\end{aligned}$$

Therefore,

$$\Pr_z\left[\Pr_x\left[\mathcal{M}^S_{z,p(z),\beta} \text{ does not compute } p \text{ at } x\right] \geq \tau\right] \leq \frac{1}{2}.$$

□

Algorithm 2 Local list recovery algorithm $\mathcal{R}(S, \beta)$

```
1: Sample $z_1, \cdots, z_t \in \mathbb{F}_q^m$ uniformly at random where $t = \log(2/\varepsilon)$.
2: Let $\mathcal{L}$ be the list of all oracle algorithms $\mathcal{M}^S_{z_i,a,\beta}$ for $i \in [t]$ and $a \in \mathbb{F}$.
3: Run the tolerant local tester $\mathcal{T}$ from Lemma 4.6.3 on each algorithm in $\mathcal{L}$ for
   $t' = 100 \log(2qt/\varepsilon)$ times and remove from $\mathcal{L}$ any algorithm which fails a majority
   of the tests.[14]
4: For every $\mathcal{M} \in \mathcal{L}$, include the oracle algorithm $\mathrm{Corr}^{\mathcal{M}}$ in the output list where
   Corr is the corrector from Lemma 4.6.1.
```

The following claim essentially proves Lemma 4.5.3.

Claim 4.6.5. *Let $1 \geq \beta > c'\sqrt{\ell d/q}$ where c' is some sufficiently large absolute constant. Let $\mathcal{L}_{out}$ be the list of oracle algorithms output by $\mathcal{R}(S, \beta)$. Then the following statements are true.*

1. *The size of the list $|\mathcal{L}_{out}| = O(q \log(1/\varepsilon))$.*

2. *The algorithm $\mathcal{R}(S, \beta)$ makes at most $O(q^2 \log(q/\varepsilon))$ queries to oracle S and runs in $\mathrm{poly}(m, q, \log(1/\varepsilon))$ time.*

3. *Each algorithm in $\mathcal{L}_{out}$ makes at most q^2 queries to S and runs in $\mathrm{poly}(m, q)$ time.*

4. *Let $p : \mathbb{F}_q^m \to \mathbb{F}_q$ be a degree d polynomial which agrees with S on at least β fraction of inputs, then with probability at least $1 - \varepsilon$, there exists $\mathcal{A} \in \mathcal{L}_{out}$ which computes p.*

5. *With probability at least $1 - \varepsilon$, every $\mathcal{A} \in \mathcal{L}_{out}$ computes some degree d polynomial.*

[14] As an anonymous reviewer has pointed out, it is not necessary to use tolerant testing here. For each $\mathcal{M} \in \mathcal{L}$ in Step 3, we could run a (not necessarily tolerant) tester on $\mathrm{Corr}^{\mathcal{M}}$ and if it accepts, then we add $\mathrm{Corr}^{\mathrm{Corr}^{\mathcal{M}}}$ in final output list. We use a tolerant tester because it is more natural here and gives better query complexity.

Proof. (1) is trivially true since the list only gets smaller after Step 2 and at the end of Step 2 we have at most $t \cdot q = O(q \log(1/\varepsilon))$ algorithms in the list. To prove (2), note that $\mathcal{R}$ makes queries to S only in Step 3. By Lemma 4.6.3, the tester $\mathcal{T}$ makes $O(q)$ queries to each algorithm $\mathcal{M}^S_{z_i,a,\beta} \in \mathcal{L}$, and each algorithm $\mathcal{M}^S_{z_i,a,\beta}$ makes at most q queries to S as in Algorithm 1. Since the test is repeated $t' = O(\log(q/\varepsilon))$ times, the total queries to S is $O(q^2 \log(q/\varepsilon))$. To analyze the running time, note that the tester $\mathcal{T}$ and the algorithms $\mathcal{M}^S_{z_i,a,\beta}$ run in $\text{poly}(m, q)$ time, so the total running time is $\text{poly}(m, q, \log(1/\varepsilon))$.

To prove (3), note that every algorithm in $\mathcal{L}_{out}$ looks like $\text{Corr}^{\mathcal{M}}$ for some $\mathcal{M}$ constructed in Step 2. By Lemma 4.6.1, Corr makes q queries to $\mathcal{M}$, and on each of them, each $\mathcal{M}$ makes q queries to S as in Algorithm 1. Thus the total queries $\text{Corr}^{\mathcal{M}}$ makes to S on any input is at most q^2. Also both Corr and $\mathcal{M}$ run in $\text{poly}(m, q)$ time, thus $\text{Corr}^{\mathcal{M}}$ also takes $\text{poly}(m, q)$ time.

To prove (4), observe that $\mathcal{M}^S_{z_1,p(z_1)}, \cdots, \mathcal{M}^S_{z_t,p(z_t)}$ are in the list $\mathcal{L}$. Let $0 < \alpha_0 < \frac{1}{4}$ be the constant that appears in Lemma 4.6.3 and let $c' > \frac{16}{\alpha_0}$. By Claim 4.6.4, with probability $\geq 1 - 1/2^t = 1 - \varepsilon/2$, at least one of these algorithms agree with p on $\geq 1 - \alpha_0$ fraction of inputs, call this algorithm $\mathcal{M}$. Therefore $\mathcal{M}$ will also pass the local testing in Step 3 with probability $1 - \varepsilon/2$ by Lemma 4.6.3 and Chernoff bound. Since $\alpha_0 < \frac{1}{4}$, by Lemma 4.6.1, $\text{Corr}^{\mathcal{M}}$ will compute p everywhere.

Finally to prove (5), by Lemma 4.6.3 and Chernoff bound, any $\mathcal{A} \in \mathcal{L}$ that is $1/4$ far from any degree d polynomial will remain in the list after Step 3 with probability at most $\frac{\varepsilon}{tq}$. By union bound over each algorithm in the list which is of size at most tq, with probability at least $1 - \varepsilon$, every algorithm $\mathcal{A}$ in $\mathcal{L}$ that remains after Step 3, will be $\frac{1}{4}$ close to some degree d polynomial p'. So by Lemma 4.6.1, $\text{Corr}^{\mathcal{A}}$ will compute p' everywhere. Therefore every algorithm in $\mathcal{L}_{out}$ computes some degree d polynomial.

□

4.6.2 Tolerant local testing of Reed-Muller codes - Proof of Lemma 4.6.3

For the proof of lemma 4.6.3 we shall use the following lemma from [FS95, Theorem 7] which gives a robust local testing procedure for Reed-Muller code. A robust local testing algorithm is a local testing algorithm such that its local view on words far from the code is far on average from an accepting view.

Lemma 4.6.6 (Robust local testing of Reed-Muller codes)**.** *There exists an absolute constant $r_0 > 0$ such that the following holds for any $\alpha > 0$ and integers m, d, q which satisfy that $\frac{d}{q} \leq r_0$. Suppose that $f : \mathbb{F}_q^m \to \mathbb{F}_q$ satisfies that $\text{dist}_H(f, RM(m,d,q)) \geq \alpha$. Then the expected relative distance of f from $RS_q(d,q)$ on a random line is at least $\frac{\alpha}{9}$.*

Proof of Lemma 4.6.3. Say we are given a function $f : \mathbb{F}_q^m \to \mathbb{F}_q$ and we need to test if it is close to a degree d polynomial. Let $0 < \tau < 1 - \sqrt{d/q}$ be some threshold parameter to be chosen later. The test is to choose a random line u in $\mathbb{F}_q^m$ and find if there is a univariate degree d polynomial which is τ-close to $f|_u$. If yes, then accept, else reject. Clearly this test makes only q queries. Also by Theorem 4.2.2, when $\tau < 1 - \sqrt{d/q}$, this can be implemented in poly(m,q) time. Now we will show that for an appropriate choice of τ, this is a $(O(q), \alpha_0, 1/4)$ tolerant test for some $\alpha_0 > 0$.

Let $f : \mathbb{F}_q^m \to \mathbb{F}_q$ be some function which is α_0-close to a degree d polynomial p. Since points on a random line are uniform over $\mathbb{F}_q^m$, by Markov inequality, the probability that $f|_u$ is τ-far from any univariate degree d polynomial is at most α_0/τ. So the probability that the test rejects f is at most $\beta_0 = \alpha_0/\tau$.

Let $g : \mathbb{F}_q^m \to \mathbb{F}_q$ be some function which is 1/4-far from any degree d polynomial. Then by Lemma 4.6.6, the expected distance of $g|_u$ to $RS_q(d,q)$ is at least 1/36. The probability that $g|_u$ is τ-far from $RS_q(d,q)$ is at least $\beta_1 = \frac{1/36-\tau}{1-\tau}$. When d/q is sufficiently small, we can choose α_0 and τ to be some absolute constants such that $0 < \tau < 1 - \sqrt{d/q}$ and $\beta_0 < \beta_1$.

Finally to get the acceptance and rejection probabilities to 2/3 as in the definition of tolerant locally testable codes, we repeat the above t times and accept a function if it is accepted in at least $\frac{\beta_0+\beta_1}{2}$ fraction of the tests. When t is large enough (but still some absolute constant), by Chernoff bound, the new test will have the required soundness and completeness.

□

4.7 Distance amplification for local list recovery

In this section we prove Lemma 4.5.4 which we recall here.

Lemma 4.5.4 (Distance amplification for local list recovery)**.** *Suppose the codes C_{out} and C_{in} exist with the following parameters:*

- *C_{out} is an $\mathbb{F}$-linear code of block length n_{out}, alphabet size σ_{out}, rate r_{out}, and relative distance δ_{out} that is $(q, \alpha_{out}, \varepsilon, \ell_{out}, L_{out})$-locally list recoverable.*

- *C_{in} is an $\mathbb{F}$-linear code of block length n_{in}, alphabet size σ_{in}, rate r_{in}, and relative distance δ_{in} that is $(\alpha_{in}, \ell_{in}, L_{in})$-(globally) list recoverable.*

There exists a $d = d(\delta_{out}, \alpha_{out}, \gamma) = (1/\delta_{out} + 1/\alpha_{out} + 1/\gamma)^{O(1)}$ such that if the parameters of C_{out} and C_{in} satisfy $n_{in} \geq d$, $\sigma_{out} = \sigma_{in}^{r_{in} \cdot n_{in}}$ and $L_{in} \leq \ell_{out}$, then there exists an $\mathbb{F}$-linear code C of block length n_{out}, alphabet size $\sigma_{in}^{n_{in}}$, rate $r_{in} \cdot r_{out}$ and relative distance at least $\delta_{in} - 2\gamma$ that is $(O(q \cdot n_{in}^2 \cdot \log(n_{in})), \alpha_{in} - \gamma, \varepsilon, \ell_{in}, L_{out})$-locally list recoverable.

Moreover,

- *If the running time of the local list recovery algorithm for C_{out} is T_{out} and the running time of the global list recovery algorithm for C_{in} is T_{in} then the running* *time of the local list recovery algorithm for C is*

$$O(T_{out}) + O(q \cdot T_{in}) + \mathrm{poly}(q, n_{in}, \ell_{in}).$$

- *If the encoding times of C_{out}, C_{in} are $\hat{T}_{out}, \hat{T}_{in}$, respectively, then the encoding time of C is*

$$O(\hat{T}_{out} + n_{out} \cdot \hat{T}_{in}) + n_{out} \cdot \mathrm{poly}(n_{in}, \log(n_{out})).$$

The construction of the code C and analysis closely follow that of the high rate locally correctable codes from [KMRS17].

One important ingredient in our construction will be a family of bipartite expanders which have the property of being good *samplers.* We define samplers below and state a lemma (very closely related to that from [KMRS17]) showing the existence of the kind of samplers we will need.

For a graph G, a vertex s and a set of vertices T, let $E(s,T)$ denote the set of edges that go from s into T. Roughly speaking, a sampler is a bipartite d-regular graph in which the density of any subset T of right vertices can be approximated by the value of $E(s,T)/d$ for a uniform random left vertex s.

Definition 4.7.1. *Let $G = (U \cup V, E)$ be a bipartite d-regular graph with $|U| = |V| = n$. We say that G is an (α, γ)-sampler if the following holds for every $T \subseteq V$: For at least $1 - \alpha$ fraction of the vertices $s \in U$ it holds that*

$$\frac{|E(s,T)|}{d} - \frac{|T|}{n} \leq \gamma.$$

Lemma 4.7.2. *For every $\alpha, \gamma > 0$, there exists $\hat{d} = \mathrm{poly}(\frac{1}{\alpha\gamma})$ such that for every sufficiently large n and for every $d > \hat{d}$ there exists a bipartite d-regular graph $G_{n,d,\alpha,\gamma} = (U \cup V, E)$ with $|U| = |V| = n$ such that $G_{n,d,\alpha,\gamma}$ is an (α, γ)-sampler. Fur-*

thermore, there exists an algorithm that takes as inputs n, d, α, γ and a vertex w of $G_{n,d,\alpha,\gamma}$, and computes the list of the neighbors of w in $G_{n,d,\alpha,\gamma}$ in time $\mathrm{poly}(\frac{\log n \cdot d}{\alpha \cdot \gamma})$.

The proof of the lemma above follows the outline presented in [KMRS15, Section 2.4] and we omit the details here. The only difference from [KMRS15] is that here we require the degree to be any $d > \hat{d}$, whereas in [KMRS15] it was constructed for specific $d = \mathrm{poly}\left(\frac{1}{\alpha \cdot \gamma}\right)$. However it is easy to see that the proof can be modified to make it work for larger degrees as well, by first constructing an $(\alpha, \gamma/2)$ sampler for a pretty large degree and then adding matchings to the graph to get the required degree and not hurting the sampling property too much.

With these samplers in hand, we are now ready to prove the lemma. We begin with a high level overview of the proof:

Proof overview Given a code C_{out} which is locally list-recoverable from a small $\alpha_{out} \ll 1$ fraction of errors, and a small code C_{in} which is (globally) list-recoverable from a large fraction of errors α_{in}, they can be combined using AEL transformation to get a new code C which is locally list-recoverable from almost α_{in} fraction of errors but doesn't use many more queries than C_{out}, and other code parameters are not significantly affected. Thus this procedure amplifies the distance from which we can list-recover without significantly worsening other parameters.

The AEL transformation works as follows: Given a codeword $c_{out} \in C_{out}$, we encode each symbol of c_{out} using an inner code C_{in} which is (globally) list recoverable from a large fraction α_{in} of errors. Now suppose we have errors in $\alpha_{in} - \gamma$ fraction of places which are randomly chosen, then by Chernoff bounds, we can say that almost all (except for at most α_{out} fraction) the inner encodings will have at most α_{in} fraction of errors. So we can list recover most of the inner encodings (except for at most α_{out} fraction). Finally, we can list recover C_{out} from α_{out} fraction of errors. Since we are interested in locally list recovering C_{out}, we only need to list recover those inner encodings which are queried by the the local list recovering algorithm for C_{out}. Thus we will not lose much in locality, as long as the length of the inner encodings are small.

But we need to deal with adversarial errors, not random. They can completely wipe α_{in} fraction of inner encodings. To overcome this problem, we use samplers. The final effect of this will be to reduce adversarial errors to random looking errors which we know how to deal with. We will choose the degree of the regular bipartite graph (sampler) to be equal the length of the inner code. We associate each inner encoding with a left vertex of the graph and distribute its symbols to each of the neighbors. The right vertices collect these symbols from their neighbors and repackage them as one symbol over a larger alphabet. This will be the final codeword which will have same length as C_{out} but over a larger alphabet. The property of the sampler will ensure that whenever $\alpha_{in} - \gamma$ fraction of symbols in the final codeword are corrupted, then after undoing the permutation of the sampler, almost all (except for at most α_{out} fraction) the inner encodings will have at most α_{in} fraction of symbols errors, and now we proceed as before.

Proof of Lemma 4.5.4. First, we describe the construction of the code C using the samplers above.

Construction of code C We construct C by giving a bijection from C_{out} to C. Let $\Sigma_{out}, \Sigma_{in}$ denote the alphabets of C_{out}, C_{in} respectively. Given a codeword $c_{out} \in C_{out}$, one obtains the corresponding codeword $c \in C$ as follows:

- View each codeword symbol in Σ_{out} as a vector of length $r_{in} \cdot n_{in}$ over Σ_{in} and encode it via the code C_{in}. Each codeword symbol gets mapped to a string in $\Sigma_{in}^{n_{in}}$. We denote the resulting string by $c' \in \Sigma_{in}^{n_{in} \cdot n_{out}}$ and the various resulting codewords of C_{in} by $B_1, B_2, \ldots B_{n_{out}} \in \Sigma_{in}^{n_{in}}$.

- Next, we apply a "pseudorandom" permutation to the coordinates of c' as follows: Let $G_{n_{out}}$ be a graph from the infinite family of n_{in}-regular $(\min\{\alpha_{out}, \delta_{out}/2\}, \gamma)$ samplers above and let $U = \{u_1, \ldots, u_{n_{out}}\}$ and $V = \{v_1, \ldots, v_{n_{out}}\}$ be the left and right vertices of $G_{n_{out}}$ respectively. For each $i \in [n_{out}]$ and $j \in [n_{in}]$, we write the j-th symbol of B_i on the j-th edge of u_i. Then, we construct new blocks $D_1, \ldots, D_{n_{out}} \in \Sigma_{in}^{n_{in}}$, by setting the j-th symbol of D_i to be the symbol written on the j-th edge of v_i. We reinterpret each of these blocks to be a symbol of the new alphabet $\Sigma \overset{\text{def}}{=} \Sigma_{in}^{n_{in}}$.

- Finally, we define the codeword c of $C \subseteq \Sigma^{n_{out}}$ as follows: the i-th coordinate c_i is the block D_i, reinterpreted as a symbol of the alphabet Σ. We choose c to be the codeword in C that corresponds to the codeword c_{out} in C_{out}.

This completes the definition of the bijection. It follows that C is an $\mathbb{F}$-linear code of blocklength n_{out} and alphabet size $\sigma_{in}^{n_{in}}$. The rate of C is

$$\begin{aligned}
\frac{\log|C|}{n_{out}\cdot\log|\Sigma|} &= \frac{\log|C_{out}|}{n_{out}\cdot n_{in}\cdot\log|\Sigma_{in}|}\\
&= \frac{r_{out}\cdot n_{out}\cdot\log|\Sigma_{out}|}{n_{out}\cdot n_{in}\cdot\log|\Sigma_{in}|}\\
&= \frac{r_{out}\cdot\log|\Sigma_{out}|}{n_{in}\cdot\log|\Sigma_{in}|}\\
&= \frac{r_{out}\cdot r_{in}\cdot n_{in}\cdot\log|\Sigma_{in}|}{n_{in}\cdot\log|\Sigma_{in}|}\\
&= r_{out}\cdot r_{in}.
\end{aligned}$$

It remains to show that the relative distance of C is at least $\delta_{in} - 2\gamma$ and that C is $(\tilde{q}, \alpha, \varepsilon, \ell, L)$-locally list recoverable for $\tilde{q} = O(q \cdot n_{in}^2 \cdot \log(n_{in}))$, $\alpha = \alpha_{in} - \gamma$, $\ell = \ell_{in}$, and $L = L_{out}$.

Once we prove the portion of the theorem that shows that C is $(O(q \cdot n_{in}^2 \cdot \log(n_{in})), \alpha_{in} - \gamma, \varepsilon, \ell_{in}, L_{out})$-locally list recoverable, it will follow almost in a black-box manner that the relative distance of C is at least $\delta_{in} - 2\gamma$ for the following reason.

Notice that it will suffice to show that C can be uniquely decoded from $\frac{\delta_{in}}{2} - \gamma$ fraction of errors. Since C_{in} has relative distance at least δ_{in}, C_{in} can be uniquely decoded from $\frac{\delta_{in}}{2}$ fraction of errors and in other words C_{in} is $(\delta_{in}/2, 1, 1)$-(globally) list recoverable. Also C_{out} can be uniquely decoded from $\frac{\delta_{out}}{2}$ fraction of errors and is hence trivially $(n_{out}, \delta_{out}/2, 0, 1, 1)$-locally list recoverable.

Thus by the same construction (i.e same choice of samplers), the code C is $(O(n_{out} \cdot n_{in}^2 \cdot \log(n_{in})), \delta_{in}/2 - \gamma, 0, 1, 1)$-locally list recoverable. In other words C is uniquely decodable from $\delta_{in}/2 - \gamma$ fraction of errors and hence has relative distance at least $\delta_{in} - 2\gamma$.

We now prove that C is $(O(q \cdot n_{in}^2 \cdot \log(n_{in})), \alpha_{in} - \gamma, \varepsilon, \ell_{in}, L_{out})$-locally list recoverable.

Local list recoverability We will now describe the $(O(q \cdot n_{in}^2 \cdot \log(n_{in})), \alpha_{in} - \gamma, \varepsilon, \ell_{in}, L_{out})$-local list recovery algorithm A for the code C. This is based on the following algorithm A which locally list recovers coordinates of C_{out} (instead of coordinates of C, as required of A).

Lemma 4.7.3. *There exists a randomized algorithm $\tilde{A}$ that on oracle access to an $S \in \binom{\Sigma}{\ell_{in}}^{n_{out}}$ makes at most $O(q \cdot n_{in} \cdot \log(n_{in}))$ queries to S and outputs a list of L_{out} randomized algorithms $\tilde{A}_1$, ..., $\tilde{A}_{L_{out}}$ which satisfy the following:*

- *Each $\tilde{A}_j$ takes as input coordinate $i \in [n_{out}]$ and also gets oracle access to the tuple S. $\tilde{A}_j$ makes at most $O(q \cdot n_{in} \cdot \log(n_{in}))$ queries to S and outputs a symbol $\tilde{A}_j^S(i) \in \Sigma_{out}$.*

- *(Completeness) For each $c_{out} \in C_{out}$ such that the corresponding codeword c of C (as given by the bijection above) satisfies $\mathrm{dist}_H(c, S) \leq \alpha_{in} - \gamma$, with probability at least $1 - \varepsilon$ over the randomness of $\tilde{A}$, there exists some $j \in [L_{out}]$ such that $\Pr_{\tilde{A}_j}\left[\tilde{A}_j^S(i) = c_{out_i}\right] \geq 1 - \frac{1}{3n_{in}}$ for all $i \in [n]$.*

- *(Soundness) With probability at least* $1-\varepsilon$ *over the randomness of* $\tilde{A}$*, for every* $j \in [L_{out}]$*, there exists some* $c_{out} \in C_{out}$ *such that* $\Pr_{\tilde{A}_j}\left[\tilde{A}_j^S(i) = c_{out_i}\right] \geq 1 - \frac{1}{3n_{in}}$ *for all* $i \in [n]$.

Given such an algorithm $\tilde{A}$ guaranteed by Lemma 4.7.3, we show how to construct the required algorithm A. The algorithm A is given oracle access to an $S \in \binom{\Sigma}{\ell_{in}}^{n_{out}}$, and needs to locally list recover all codewords $c \in C$ that "disagree" with S in at most $\alpha_{in} - \gamma$ fraction of coordinates. A outputs a list of L_{out} randomized algorithms A_1, ..., $A_{L_{out}}$ which work as follows.

Each A_j takes as input a coordinate $i \in [n_{out}]$ and also gets oracle access to the tuple S. Note that by the above lemma, with probability at least $1-\varepsilon$ over the randomness of $\tilde{A}$, for each $\tilde{A}_j$ there exists some $c_{out} \in C_{out}$ such that $\Pr_{\tilde{A}_j}\left[\tilde{A}_j^S(i) = c_{out_i}\right] \geq 1 - \frac{1}{3n_{in}}$ for all $i \in [n]$. Let the corresponding codeword in C be c. We will use $\tilde{A}_j$ to design A_j that will output the coordinates of c. Let $B_1, \ldots, B_{n_{out}}$ and $D_1, \ldots, D_{n_{out}}$ be the corresponding blocks that arise in the construction of c from c_{out}. In order for $A_j(i)$ to be able to decode the value of c_i, it should be able to correctly decode all the symbols in the block D_i. Let $u_{i_1}, \ldots, u_{i_{n_{in}}}$ be the neighbors of v_i in the graph $G_{n_{out}}$. Each symbol of D_i belongs to one of the blocks $B_{i_1}, \ldots, B_{i_{n_{in}}}$, and therefore it suffices to retrieve these blocks. Each of these blocks B_{i_j} is the encoding of $c_{out_{i_j}}$ (the i_jth symbol of c_{out}) via the code C_{in}. Thus to recover $B_{i_1}, \ldots, B_{i_{n_{in}}}$, it suffices to recover $c_{out_{i_1}}, \ldots, c_{out_{i_{n_{in}}}}$. The algorithm A_j invokes the algorithm $\tilde{A}_j$ to recover each of $c_{out_{i_1}}, \ldots, c_{out_{i_{n_{in}}}}$, and by the union bound, it recovers all of them correctly with probability at least $1 - n_{in} \cdot \frac{1}{3n_{in}} = 2/3$. Whenever this happens, the algorithm A_j correctly retrieves the blocks $B_{i_1}, \ldots, B_{i_{n_{in}}}$ and hence also D_i and hence c_i.

Clearly the query complexity of A_j is n_{in} times the query complexity of $\tilde{A}_j$, and is hence at most $O(q \cdot n_{in}^2 \cdot \log(n_{in}))$. The completeness and soundness of A follow from the completeness and soundness of $\tilde{A}$.

It can be verified that the local list recovery algorithms $\tilde{A}$ and A can be implemented efficiently as required by the "moreover" part of the lemma.

□

We now prove Lemma 4.7.3.

Proof of Lemma 4.7.3. Let $\bar{A}$ be the local list recovery algorithm for C_{out}. $\bar{A}$ is a randomized algorithm that on oracle access to a tuple $\overline{S} \in \binom{\Sigma_{out}}{\ell_{out}}^{n_{out}}$ outputs a list of L_{out} randomized algorithms $\bar{A}_1, \bar{A}_2, \ldots, \bar{A}_{L_{out}}$. By amplification we may assume that for each $i \in [n_{out}]$, $\bar{A}_j(i)$ errs with probability at most $\frac{1}{3 \cdot n_{in}}$, and this incurs a factor of at most $O(\log(n_{in}))$ to its query complexity. Thus the query complexity is at most $O(q \cdot \log(n_{in}))$.

We now describe $\tilde{A}$. Suppose the algorithm $\tilde{A}$ is invoked on a tuple $S = (S_1, \ldots, S_{n_{out}}) \in \binom{\Sigma}{\ell_{in}}^{n_{out}}$, the algorithm $\tilde{A}$ invokes the algorithm $\bar{A}$ and emulates $\bar{A}$ in the natural way. Recall that $\bar{A}$ expects to be given a tuple $\bar{S} \in \binom{\Sigma_{out}}{\ell_{out}}^{n_{out}}$. On input coordinate i, $\bar{A}$ makes queries to this sequence and outputs a value $\bar{A}(i)$. For any $k \in [n_{out}]$, whenever $\bar{A}$ queries the kth element of the sequence $\bar{S}_1, \ldots, \bar{S}_{n_{out}} \in \binom{\Sigma_{out}}{\ell_{out}}$, the algorithm $\tilde{A}$ performs the following steps.

1. In the first step, for each coordinate $r \in [n_{in}]$ of B_k, $\tilde{A}$ will find a list $S^{(k,r)} \in \binom{\Sigma_{in}}{\ell_{in}}$ and associate that list with the rth coordinate of B_k. The list $S^{(k,r)}$ is defined as follows: Suppose that v_{k_r} is the rth neighbor of the vertex u_k in $G_{n_{out}}$. Suppose that u_k is the $\hat{r}$th neighbor of the vertex v_{k_r}. Then in the construction of the codeword c from c_{out}, the value of the rth coordinate of B_k is stored in the $\hat{r}$th coordinate of D_{k_r}. Now $S_{k_r} \in \binom{\Sigma}{\ell_{in}} = \binom{\Sigma_{in}^{n_{in}}}{\ell_{in}}$ is the input list associated with the k_rth coordinate. Note that each element $s \in S_{k_r}$ can be viewed as an n_{in}-tuple of elements from Σ_{in}. Let the $\hat{r}$th element of this tuple be $s^{(\hat{r})}$. Then $S^{(k,r)}$ is defined to be the set in $\binom{\Sigma_{in}}{\ell_{in}}$ obtained by taking the $\hat{r}$th element of

each member of the set S_{k_r}. $\tilde{A}$ can find this set by making a single query to the k_rth element of S to obtain S_{k_r}, and from it find $S^{(k,r)}$.

2. $\tilde{A}$ then invokes the global-list recovery algorithm for C_{in} with the lists $S^{(k,r)}$ for each $r \in [n_{in}]$. The output of this algorithm is a list of size at most L_{in} with elements from $\Sigma_{in}^{n_{in}}$. We denote by $\bar{S}_k$ the set of messages in $\Sigma_{in}^{r_{in}n_{in}} = \Sigma_{out}$ corresponding to the codewords in this list. This is what $\tilde{A}$ feeds to $\bar{A}$.

It is not hard to see that the query complexity of $\tilde{A}$ is at most n_{in} times the query complexity of $\bar{A}$, and hence it is at most $O(q \cdot n_{in} \cdot \log(n_{in}))$. It remains to show that $\tilde{A}$ satisfies the completeness and soundness requirements. We first show the completeness.

Completeness: Let $c_{out} \in C_{out}$ be such that the corresponding codeword c of C (as given by the bijection above) satisfies $\text{dist}_H(c, S) \leq \alpha_{in} - \gamma$. From the following claim, completeness of $\tilde{A}$ will follows from the completeness of $\bar{A}$.

Claim 4.7.4. *The tuple $\bar{S} := (\bar{S}_1, \bar{S}_2, \ldots, \bar{S}_{n_{out}})$ as defined above satisfies $\text{dist}_H(c_{out}, \bar{S}) \leq \alpha_{out}$.*

Proof. Let $T = \{k \in [n_{out}] \mid c_k \notin S_k\}$. Then we know that $|T| \leq (\alpha_{in} - \gamma)n_{out}$. Let *Good* be the set of all $i \in [n_{out}]$ such that in the graph $G_{n_{out}}$, u_i has at most α_{in} fraction of its neighbors v_j with $j \in T$. By the sampling property of $G_{n_{out}}$, it holds that $|Good| \geq (1 - \alpha_{out}) \cdot n_{out}$.

We will now show that for all $k \in Good$, $c_{out_k} \in \bar{S}_k$. Since $|Good| \geq (1-\alpha_{out}) \cdot n_{out}$, this shows that $\text{dist}_H(c_{out}, \bar{S}) \leq \alpha_{out}$ and thus proves the claim.

Let $k \in Good$. For each $r \in [n_{in}]$, let $S^{(k,r)} \in \binom{\Sigma_{in}}{\ell_{in}}$ be the set assigned to the rth coordinate of B_k (as described above). To show that $c_{out_k} \in \bar{S}_k$, it suffices to show that the encoding of c_{out_k} via the code C_{in} (which we call B_k) agrees with various

$S^{(k,r)}$ for at least $1-\alpha_{in}$ fraction of coordinates $r \in [n_{in}]$, since then the global list recovery algorithm of C_{in} succeeds in outputting c_{out_k}.

Now let r be any coordinate such that the rth neighbor of u_k in $G_{n_{out}}$ is a vertex v_{k_r} where $k_r \notin T$. Thus $c_{k_r} \in S_{k_r}$. Hence, by the definition of $S^{(k,r)}$, it holds that the rth coordinate of B_k agrees with $S^{(k,r)}$. Since at most α_{in} fraction of the r's could have been such that $k_r \in T$, thus for at least $1-\alpha_{in}$ fraction of coordinates $r \in [n_{in}]$, B_k agrees with $S^{(k,r)}$, and hence $c_{out_k} \in \bar{S}_k$.

□

Soundness: The soundness of $\tilde{A}$ follows from the soundness of $\bar{A}$. This is because for any $S \in \binom{\Sigma}{\ell_{in}}^{n_{out}}$ and $i \in L_{out}$, algorithm $\tilde{A}_i^S$ behaves exactly like $\bar{A}_i^{\bar{S}}$ where $\bar{S}$ is as defined above, in particular they have the same output. But we know that, with probability $\geq 1-\varepsilon$ over the randomness of $\bar{A}$, each algorithm $\bar{A}_i^{\bar{S}}$ output by the list recovery algorithm $\bar{A}^{\bar{S}}$ computes some $c_{out} \in C_{out}$. Thus with probability $\geq 1-\varepsilon$, each $\tilde{A}_i^S$ also computes some codeword in C_{out}.

□

4.8 Johnson Bound for List Recovery

In this section we prove Lemma 4.5.2 restated below.

Lemma 4.5.2 (Johnson bound for list recovery)**.** *Let $C \subseteq \Sigma^n$ be a code of relative distance at least δ. Then C is (α, ℓ, L)-list recoverable for any $\alpha < 1 - \sqrt{\ell \cdot (1-\delta)}$ with $L = \frac{\delta\ell}{(1-\alpha)^2 - \ell(1-\delta)}$.*

Proof. The proof is a simple adaptation of the proof of the Johnson bound for list decoding from [Gur06, Theorem 3.3].

Let $|\Sigma| = q$, let $S \in \binom{\Sigma}{\ell}^n$ be a tuple, and let $\mathcal{N} := \{c \in C \mid \mathrm{dist}_H(c, S) \leq \alpha\}$. Our goal will be to show that $L =| \mathcal{N} | \leq \frac{\delta\ell}{(1-\alpha)^2 - \ell(1-\delta)}$. As the minimum relative distance of the code C is δ and each $c \in \mathcal{N}$ has relative distance at most α from the tuple S, we have

$$\delta \leq \mathop{\mathbb{E}}_{\{\mathbf{x},\mathbf{y}\}\sim\binom{\mathcal{N}}{2}}\left[\frac{\Delta(\mathbf{x},\mathbf{y})}{n}\right] \quad \text{and} \quad \alpha \geq \varepsilon := \mathop{\mathbb{E}}_{\substack{\mathbf{x}\sim\mathcal{N}\\ i\sim[n]}}\left[1_{x_i \notin S_i}\right]. \tag{4.6}$$

Let $\mathbf{x}, \mathbf{y}$ be two distinct words in $\mathcal{N}$, chosen uniformly at random. We will obtain a lower bound on the expected fraction of coordinates where $\mathbf{x}$ and $\mathbf{y}$ agree (in terms of L, α and ℓ). We know that this expectation is at most $1-\delta$. The theorem will follow by comparing these two quantities.

For $i \in [n]$, and $z \in \Sigma$, let

$$k_i(z) = |\{\mathbf{x} \in \mathcal{N} \mid x_i = z\}|.$$

Then we have that

$$\begin{aligned}\Pr_{\{\mathbf{x},\mathbf{y}\}\sim\binom{\mathcal{N}}{2}}[x_i = y_i] &= \binom{L}{2}^{-1} \cdot \sum_{z\in\Sigma}\binom{k_i(z)}{2} \\ &= \binom{L}{2}^{-1} \cdot \left[\sum_{z\in S_i}\binom{k_i(z)}{2} + \sum_{z\in\Sigma\setminus S_i}\binom{k_i(z)}{2}\right] \\ &\geq \binom{L}{2}^{-1} \cdot \left[\ell \cdot \binom{k_i}{2} + (q-\ell)\binom{\frac{L-\ell k_i}{q-\ell}}{2}\right]\end{aligned}$$

where $k_i = \dfrac{1}{\ell} \cdot \displaystyle\sum_{z\in S_i} k_i(z)$ and we used Jensen's inequality.

Hence, the expected fraction of coordinates where $\mathbf{x}$ and $\mathbf{y}$ agree is bounded by

$$\begin{aligned}&\frac{1}{n} \cdot \sum_{i=1}^{n} \Pr_{\{\mathbf{x},\mathbf{y}\}\sim\binom{\mathcal{N}}{2}}[x_i = y_i] \\ &\geq \frac{1}{n} \cdot \binom{L}{2}^{-1} \cdot \sum_{i=1}^{n}\left[\ell \cdot \binom{k_i}{2} + (q-\ell)\binom{\frac{L-\ell k_i}{q-\ell}}{2}\right] \\ &\geq \binom{L}{2}^{-1} \cdot \left[\ell \cdot \binom{t}{2} + (q-\ell)\binom{\frac{L-t\ell}{q-\ell}}{2}\right]\end{aligned}$$

where $t = \frac{1}{n} \cdot \sum_{i=1}^{n} k_i$ and we again used Jensen's inequality. Since the left hand side is bounded from above by $1 - \delta$, after some rearrangement, we have:

$$(1-\delta) \cdot \binom{L}{2} \geq \ell \cdot \binom{t}{2} + (q-\ell)\binom{\frac{L-t\ell}{q-\ell}}{2} \tag{4.7}$$

Since ε is the expected fraction of disagreement between the words in $\mathcal{N}$ and S, we have that $Ln\varepsilon$ is the total amount of disagreement between $\mathcal{N}$ and S. We can also count the amount of disagreement in the following way: $\ell k_i = \sum_{z \in S_i} k_i(z)$ is the amount of agreement between the words of $\mathcal{N}$ and the input list S_i. Hence, the total agreement between the words of $\mathcal{N}$ and S is $\sum_{i=1}^{n} \ell k_i = t\ell n$. This implies that the total disagreement is $Ln - t\ell n = Ln\varepsilon$. Thus, we obtain that $\frac{t\ell}{L} = 1 - \varepsilon$.

Substituting $\frac{t\ell}{L} = 1 - \varepsilon$ in equation (4.7), and rearranging terms, we have

$$\begin{aligned}(1-\delta) \cdot \frac{L(L-1)}{2} &\geq \ell \cdot \frac{t(t-1)}{2} + \frac{(L-t\ell)(L-t\ell-q+\ell)}{2(q-\ell)} \\ &= \frac{\ell t^2}{2} - \frac{\ell t}{2} + \frac{(L-t\ell)^2}{2(q-\ell)} - \frac{L-t\ell}{2} \\ &= \frac{(1-\varepsilon)^2 L^2}{2\ell} - \frac{(1-\varepsilon)L}{2} + \frac{(L\varepsilon)^2}{2(q-\ell)} - \frac{L\varepsilon}{2},\end{aligned}$$

which gives

$$\begin{aligned}(1-\delta) \cdot (L-1) &\geq \frac{(1-\varepsilon)^2}{\ell} L - (1-\varepsilon) + \varepsilon\left(\frac{\varepsilon L}{q-\ell} - 1\right) \\ &= \frac{(1-\varepsilon)^2}{\ell} L + \frac{\varepsilon^2 L}{q-\ell} - 1.\end{aligned}$$

By grouping the terms with L and rearranging the inequality above, we get that

$$\begin{aligned} L &\leq \frac{\delta}{\frac{(1-\varepsilon)^2}{\ell} + \frac{\varepsilon^2}{q-\ell} - (1-\delta)} \\ &= \frac{\delta \ell}{(1-\varepsilon)^2 + \frac{\ell\varepsilon^2}{q-\ell} - \ell(1-\delta)} \\ &\leq \frac{\delta\ell}{(1-\varepsilon)^2 - \ell(1-\delta)} \leq \frac{\delta\ell}{(1-\alpha)^2 - \ell(1-\delta)}, \end{aligned}$$

where the last inequality follows since $\varepsilon \geq \alpha$. □

Chapter 5

LDCs from Outlaw distributions

5.1 Introduction

Despite their many applications, our knowledge of LDCs is very limited; the best-known constructions are far from what is currently known about their limits. Although standard random (linear) ECCs do allow for some weak local-decodability, they are outperformed by even the earliest explicit constructions [KS07]. All the known constructions of LDCs were obtained by explicitly designing such codes using some algebraic objects like low-degree polynomials or matching vectors [Yek12].

In this paper, we give a characterization of LDCs in probabilistic and geometric terms, making them amenable to probabilistic constructions. On the flip side, these characterizations might also be easier to work with for the purpose of showing lower bounds. We will make this precise in the next section.

5.1.1 LDCs from distributions over smooth Boolean functions

Our main result shows that LDCs can be obtained from "outlaw" distributions over "smooth" functions. The term outlaw refers to the Law of Large Numbers, which says that the average of independent samples tends to the expectation of the distribution from which they are drawn. Roughly speaking, a probability distribution is an outlaw if many samples are needed for a good estimation of the expectation and a smooth function over the n-dimensional Boolean hypercube is one that has no influential variables. Paradoxically, while many instances of the probabilistic method use the fact that sample means of a small number of independent random variables tend to concentrate around the true mean, as captured for example by the Chernoff bound, our main result requires precisely the opposite. We show that if *at least k* samples from a distribution over smooth functions are needed to approximate the mean, then there exists an $O(1)$-query LDC sending $\{0,1\}^{\Omega(k)}$ to $\{0,1\}^n$, where the hidden constants depend only the smoothness and mean-estimation parameters.

To make this precise, we now formally define smooth functions and outlaw distributions. Given a function $f : \{-1,1\}^n \to \mathbb{R}$, its *spectral norm* (also known as the *algebra norm* or *Wiener norm*) is defined as

$$\|f\|_A = \sum_{S \subset [n]} |\widehat{f}(S)|,$$

where $\widehat{f}(S)$ are the Fourier coefficients of f (see Section 5.2.1 for basics on Fourier analysis). Note that $\|f\|_A = \|\hat{f}\|_{\ell_1}$. By Young's inequality for convolutions, it follows that for any f, g,

$$\|fg\|_A \leq \|f\|_A \, \|g\|_A \, ,$$

which justifies the term *Algebra norm.* We also consider the supremum norm, $\|f\|_{L_\infty} = \sup\{|f(x)| : x \in \{-1,1\}^n\}$. It follows from the Fourier inversion formula that $\|f\|_{L_\infty} \leq \|f\|_A$. The *ith discrete derivative* of f is the function $(D_i f)(x) = (f(x) - f(x^i))/2$, where x^i is the point that differs from x on the ith coordinate. The Fourier expansion of $D_i f$ is given by $D_i f(x) = \sum_{S \ni i} \widehat{f}(S)\chi_S(x)$.

Hence it follows that

$$\|D_i f\|_A = \sum_{S \ni i} \left| \widehat{f}(S) \right| .$$

Smooth functions are functions whose discrete derivatives have small spectral norms.

Definition 5.1.1 (σ-smooth functions)**.** *For $\sigma > 0$, a function $f : \{-1,1\}^n \to \mathbb{R}$ is σ-smooth, if for every $i \in [n]$, we have $\|D_i f\|_A \leq \sigma/n$.*

Intuition for the above definition may be gained from the fact that smooth functions have no influential variables. The influences, $(\mathbb{E}_{x \in \{-1,1\}^n}[(D_i f)(x)^2])^{1/2}$, measure the extent to which changing the ith coordinate of a randomly chosen point changes the value of f. Since $\|D_i f\|_{L_\infty} \leq \|D_i f\|_A$, the directional derivatives of σ-smooth functions are uniformly bounded by σ/n, which is a much stronger condition than saying that the derivatives are small on average. Outlaws are defined as follows.

Definition 5.1.2 (Outlaw)**.** *Let n be a positive integer and μ be a probability distribution over real-valued functions on $\{-1,1\}^n$. For a positive integer k and $\varepsilon > 0$, say that μ is a (k, ε)-*outlaw *if for independent random μ-distributed functions $f_1, \ldots, f_k$ and $\bar{f} = \mathbb{E}_\mu[f]$,*

$$\mathbb{E}\left[\left\| \frac{1}{k} \sum_{i=1}^{k} (f_i - \bar{f}) \right\|_{L_\infty} \right] \geq \varepsilon.$$

Denote by $\kappa_\mu(\varepsilon)$ the largest integer k such that μ is a (k, ε)-outlaw.

To approximate the true mean of an outlaw μ to within ε on average in the L_∞-distance, one thus needs $\kappa_\mu(\varepsilon) + 1$ samples. Note that if μ is a distribution over σ-smooth functions, then the distribution $\tilde{\mu}$ obtained by scaling functions in the support of μ by $1/\sigma$ is a distribution over 1-smooth functions and $\kappa_{\tilde{\mu}}(\varepsilon/\sigma) = \kappa_\mu(\varepsilon)$.

Our main result is then as follows.

Theorem 5.1.3 (Main theorem)**.** *Let n be a positive integer and $\varepsilon > 0$. Let μ be a probability distribution over 1-smooth functions on $\{-1,1\}^n$ and $k = \kappa_\mu(\varepsilon)$. Then, there exists a (q, δ, η)-LDC sending $\{0,1\}^l$ to $\{0,1\}^n$ where $l = \Omega(\varepsilon^2 k / \log(1/\varepsilon))$,*

$q = O(1/\varepsilon)$, $\delta = \Omega(\varepsilon)$ and $\eta = \Omega(\varepsilon)$. Additionally, if μ is supported on degree-d functions, then we can take $q = d$.

Note that the smoothness requirement is essential. For example the uniform distribution over the n dictator functions $f_i(x) = x_i$ for $i \in [n]$ is an $(n/2, 1)$-outlaw, but it cannot imply constant rate, constant query LDCs which we know do not exist. In fact we establish a converse to Theorem 5.1.3, showing that its hypothesis is essentially equivalent to the existence of LDCs in the small query complexity regime.

Theorem 5.1.4. *If $C : \{0,1\}^k \to \{0,1\}^n$ is a (q, δ, η)-LDC, then there exists a probability distribution μ over 1-smooth degree-q functions on $\{-1,1\}^n$ such that*

$$\kappa_\mu(\varepsilon) \geq (\eta/2)k$$

where $\varepsilon = \eta\delta/(2q2^{q/2})$.

Theorem 5.1.4 can in turn convert the problem of proving lower bounds on the length of LDCs to a problem in Banach space geometry. In particular, for a distribution μ over 1-smooth degree-q functions on $\{0,1\}^n$, one can upper bound $\kappa_\mu(\varepsilon)$ in terms of type constants of the space of q-linear forms on ℓ_q^{n+1} [Bri16].

Candidate outlaws One scenario in which outlaw distributions can be obtained is using incidence geometry in finite fields. In particular, the following result can be derived from our main theorem (stated a bit informally here, see Section 5.5.1 for the formal version).

Corollary 5.1.5. *Let $p > 2$ be a fixed prime. Suppose that for a random set of directions $D \subset \mathbb{F}_p^n$ of size $|D| \leq k$, with probability at least $1/2$, there exists a set $B \subset \mathbb{F}_p^n$ of size $|B| \geq \Omega(p^n)$ which does not contain any lines with direction in D. Then, there exists a p-query LDC sending $\{0,1\}^{\Omega(k)}$ to $\{0,1\}^{p^n}$.*

The assumption in Corollary 5.1.5 that D be random is essential for it to be potentially interesting for LDCs. If we instead ask that every set of directions D satisfies the condition—as we did in the conference version of this paper—then letting D be a subspace shows that k must be smaller than a constant depending only on p and ε by Szemerédi's Theorem (Theorem 5.5.2 below) [Fox17].

The analogue of Corollary 5.1.5 in $\mathbb{Z}/N\mathbb{Z}$ where lines correspond to arithmetic progressions and directions correspond to common differences can also be used to construct LDCs. This question was studied in [FLW16a], where they show that if D is a random subset of $\mathbb{Z}/N\mathbb{Z}$ of size $\omega(N^{1-1/p})$, then almost surely every dense subset of $\mathbb{Z}/N\mathbb{Z}$ contains a p-term arithmetic progression with common difference in D. Our main result, together with the best-known lower bounds on LDCs show that the bound of [FLW16a] can be improved to $\tilde{\omega}(N^{1-1/(\lceil p/2\rceil-1)})$.

Another setting in which our approach leads to interesting open problems is in relation to pseudorandom hypergraphs. Consider a partition of the complete bipartite graph $K_{n,n}$ into n perfect matchings. It is known that picking $k = O(\log n)$ of these matchings at random will give us a pseudorandom (expander) graph (of degree k). For some particular partitions (e.g., given by an Abelian group) this bound is tight. The questions arising from our approach can be briefly summarized as follows: Can one find an n-vertex hypergraph H (say three uniform to be precise) and a partition of H into matchings so that, to get a pseudorandom hypergraph (defined appropriately) one needs at least k random matchings. This would give a code sending $\Omega(k)$-bit messages with encoding length $O(n)$ and so, becomes interesting when k is super poly-logarithmic in n. We elaborate on this in Section 5.5.2

5.1.2 Techniques

Our proof of Theorem 5.1.3 proceeds in two steps. The first step consists of turning an outlaw over smooth functions into a seemingly crude type of LDC that is only required to work on average over a uniformly distributed message and a uniformly distributed message index. We call such codes average-case smooth codes (see Section 2.3.2). The second step consists of showing that such codes are in fact not much weaker than honest LDCs.

From outlaws to average-case smooth codes The key ingredient for the first step is *symmetrization*, a basic technique from high-dimensional probability. We briefly sketch how this is used (we refer to Section 5.3 for the full proof). Suppose that $f_1, \dots, f_k$ are independent smooth functions distributed according to a (k, ε)-outlaw with expectation $\bar{f}$. We introduce an independent copy[1] f_i' of f_i for each $i \in [k]$ and consider the symmetrically distributed random functions $f_i - f_i'$. Since $\bar{f} = \mathbb{E}[f_i']$ for each $i \in [k]$, Jensen's inequality and Definition 5.1.2 imply that

$$\mathbb{E}\Big[\|(f_1 - f_1') + \cdots + (f_k - f_k')\|_{L_\infty}\Big] \geq \mathbb{E}\Big[\|(f_1 - \mathbb{E}[f_1']) + \cdots + (f_k - \mathbb{E}[f_k'])\|_{L_\infty}\Big] \geq \varepsilon k.$$

Since the random functions $f_i - f_i'$ are independent and symmetric, we get that for independent uniformly random signs $x_1, \dots, x_k \in \{-1, 1\}$, the above left-hand side equals

$$\mathbb{E}\Big[\|x_1(f_1 - f_1') + \cdots + x_k(f_k - f_k')\|_{L_\infty}\Big].$$

The triangle inequality and the Averaging Principle then give that there exist *fixed* smooth functions $f_1^\star, \dots, f_k^\star$ such that on average over the random signs, we have

$$\mathbb{E}\Big[\|x_1 f_1^\star + \cdots + x_k f_k^\star\|_{L_\infty}\Big] \geq \varepsilon k/2. \tag{5.1}$$

To get an average-case smooth code out of this, we view each sequence $x = (x_1, \dots, x_k)$ as a k-bit message and choose an arbitrary n-bit string for which the L_∞-norm in (5.1) is achieved to be the its encoding, $C(x)$. This gives a map $C : \{-1, 1\}^k \to \{0, 1\}^n$

[1] in this context sometimes referred to as a "ghost copy" as it will later disappear again

satisfying

$$\mathbb{E}\Big[x_1 f_1^\star(C(x)) + \cdots x_k f_k^\star(C(x))\Big] \geq \varepsilon k/2.$$

Equivalently, for uniform x and i, we have $\Pr[f_i^\star(C(x)) = x_i] \geq \frac{1}{2} + \frac{\varepsilon}{4}$. Finally, we use the smoothness property to transform the $f_i^\star$ into decoders with the desired properties. This is done in Section 5.3. It is in the application of the Averaging Principle where the probabilistic method appears in our construction of LDCs.

Average-case smooth codes are LDCs Our second step in the proof of Theorem 5.1.3 is an average-case to worst-case reduction showing that smooth LDCs which only work on average i.e. for a random message and random decoding bit, can be converted into smooth LDCs that work for every message and every decoding bit. See Section 2.3.2 for the relevant definitions and Theorem 2.3.7 showing such a reduction. Combining it with Proposition 2.3.5 to convert the resulting smooth LDC into an LDC, we get the following lemma.

Lemma 5.1.6. *Let $C : \{0,1\}^k \to \{0,1\}^n$ be a (q, c, η)-average-case smooth code. Then, there exists an $(q, \Omega(\eta/cq), \Omega(\eta))$-LDC sending $\{0,1\}^l$ to $\{0,1\}^n$ where $l = \Omega(\eta^2 k/\log(1/\eta))$.*

5.1.3 Organization

In Section 5.3, we prove our main theorem (Theorem 5.1.3) by first showing that outlaw distributions over smooth functions imply existence of average-case smooth codes and using Lemma 5.1.6 to convert them to LDCs. In Section 5.4, we show the converse to our main theorem (Theorem 5.1.4) showing how to get outlaw distributions over smooth functions from LDCs. Finally in Section 5.5, we give some candidate constructions of outlaw distributions over smooth functions using incidence geometry and hypergraph pseudorandomness.

5.2 Preliminaries

5.2.1 Fourier analysis on the Boolean cube

We recall a few basic definitions and facts from Fourier analysis over the n-dimensional Boolean hypercube $\{-1,1\}^n$. Equipped with the coordinate-wise multiplication operation, the hypercube forms an Abelian group whose group of characters is formed by the functions $\chi_S(x) = \prod_{i\in S} x_i$ for all $S \subseteq [n]$. The characters form a complete orthonormal basis for the space of real-valued functions on $\{-1,1\}^n$ endowed with the inner product $\langle f, g\rangle = \mathbb{E}_{x\in\{-1,1\}^n}[f(x)g(x)]$, where we use the notation $\mathbb{E}_{a\in S}$ to denote the expectation with respect to a uniformly distributed element a over a set S. The *Fourier transform* of a function $f : \{-1,1\}^n \to \mathbb{R}$ is the function $\widehat{f} : 2^{[n]} \to \mathbb{R}$ defined by $\widehat{f}(S) = \langle f, \chi_S\rangle$. The Fourier inversion formula (which follows from orthonormality of the character functions) asserts that

$$f = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S.$$

Parseval's Identity relates the L_2-norms of f and its Fourier transform by

$$\Big(\mathbb{E}_{x\in\{-1,1\}^n}[f(x)^2]\Big)^{1/2} = \Big(\sum_{S\subseteq[n]} |\widehat{f}(S)|^2\Big)^{1/2}.$$

A function f has *degree q* if $\widehat{f}(S) = 0$ when $|S| > q$ and the *degree-q truncation* of f, denoted $f^{\le q}$, is the degree-q function defined by

$$f^{\le q} = \sum_{|S|\le q} \widehat{f}(S)\chi_S.$$

A function f is a *q-junta* if it depends only on a subset of q of its variables, or equivalently, if there exists a subset $T \subseteq [n]$ of size $|T| \le q$ such that $\widehat{f}(S) = 0$ for every $S \not\subseteq T$. The *ith discrete derivative* $D_i f$ is the function

$$(D_i f)(x) = \frac{f(x) - f(x^{(i)})}{2},$$

where $x^{(i)}$ is the point that differs from x on the ith coordinate. It is easy to show that the ith discrete derivative in of a function f is given by

$$D_i f = \sum_{S \ni i} \widehat{f}(S)\chi_S.$$

5.3 From outlaws to average-case smooth codes

In this section we prove Theorem 5.1.3. For convenience, in the remainder of this paper, we switch the message and codeword alphabets of all codes from $\{0,1\}^n$ to $\{-1,1\}^n$. We begin by showing that outlaw distributions over degree-q functions give q-query average-case smooth codes. Combined with Lemma 5.1.6, this implies the second part of Theorem 5.1.3.

Theorem 5.3.1. *Let μ be a probability distribution on 1-smooth degree-q functions on $\{-1,1\}^n$, let $\varepsilon \in (0,1]$ and let $k = \kappa_\mu(\varepsilon)$. Then, there exists a $(q, 1/q, \varepsilon/2)$-average-case smooth code sending $\{-1,1\}^k$ to $\{-1,1\}^n$.*

Proof. The proof uses a symmetrization argument. Let $\mathcal{F} = (f_1, \dots, f_k)$ and $\mathcal{F}' = (f_1', \dots, f_k')$ be two k-tuples of independent μ-distributed random variables and let $\bar{f} = \mathbb{E}_\mu[f]$. Then, by definition of $\kappa_\mu(\varepsilon)$ and Jensen's inequality,

$$\begin{aligned}
\varepsilon &\le \mathbb{E}_{\mathcal{F}}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_i - \bar{f})\Big\|_{L_\infty}\Big] \\
&= \mathbb{E}_{\mathcal{F}}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}\Big(f_i - \mathbb{E}_{\mathcal{F}'}[f_i']\Big)\Big\|_{L_\infty}\Big] \\
&\le \mathbb{E}_{\mathcal{F},\mathcal{F}'}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_i - f_i')\Big\|_{L_\infty}\Big].
\end{aligned}$$

The random variables $f_i - f_i'$ are symmetrically distributed, which is to say that they have the same distribution as their negations $f_i' - f_i$. Since they are independent, it follows that for every $x \in \{-1,1\}^k$, the random variable $x_1(f_1 - f_1') + \cdots + x_k(f_k - f_k')$ has the same distribution as $(f_1 - f_1') + \cdots + (f_k - f_k')$. Therefore,

$$\begin{aligned}\mathbb{E}_{\mathcal{F},\mathcal{F}'}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_i - f_i')\Big\|_{L_\infty}\Big] &= \mathbb{E}_{x\in\{-1,1\}^k}\Big[\mathbb{E}_{\mathcal{F},\mathcal{F}'}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}x_i(f_i - f_i')\Big\|_{L_\infty}\Big]\Big] \\ &\leq 2\mathbb{E}_{\mathcal{F}}\Big[\mathbb{E}_{x\in\{-1,1\}^k}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}x_i f_i\Big\|_{L_\infty}\Big]\Big].\end{aligned}$$

Applying the Averaging Principle to the outer expectation, we find that there exist 1-smooth degree-q functions $f_1^\star, \ldots, f_k^\star : \{-1,1\}^n \to \mathbb{R}$ such that

$$\mathbb{E}_{x\in\{-1,1\}^k}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}x_i f_i^\star\Big\|_{L_\infty}\Big] \geq \frac{\varepsilon}{2}. \tag{5.2}$$

Define the code $C : \{-1,1\}^k \to \{-1,1\}^n$ such that for each $x \in \{-1,1\}^k$, we have

$$\frac{1}{k}\sum_{i=1}^{k}x_i f_i^\star(C(x)) = \Big\|\frac{1}{k}\sum_{i=1}^{k}x_i f_i^\star\Big\|_{L_\infty}. \tag{5.3}$$

For each $i \in [k]$, define the decoder $\mathcal{A}_i$ as follows. Let $\nu_i : 2^{[n]} \to [0,1]$ be the probability distribution defined by $\nu_i(S) = |\widehat{f_i^\star}(S)| / \|f_i^\star\|_A$. Given a string $z \in \{-1,1\}^n$, with probability $1 - \|f_i^\star\|_A$, the decoder $\mathcal{A}_i$ returns a uniformly random sign, and with probability $\|f_i^\star\|_A$, it samples a set $S \subseteq [n]$ according to ν_i and returns $\chi_S(z)$. This is a valid probability distribution since for any 1-smooth function f, we have

$$\|f\|_A = \sum_{S\subset[n]}|\widehat{f}(S)| \leq \sum_{S\subset[n]}|S||\widehat{f}(S)| = \sum_{i=1}^{n}\sum_{S\ni i}|\widehat{f}(S)| \leq n\cdot\frac{1}{n} = 1.$$

Then, $\mathcal{A}_i$ queries at most q coordinates of z and since $f_i^\star$ is 1-smooth, the probability that it queries any coordinate $j \in [n]$ is at most $\|D_j f_i^\star\|_A \leq 1/n$. Since the queries can be presented in a random order, the probability that t^{th} query is j is $\leq 1/qn$. We also have $\mathbb{E}[\mathcal{A}_i(z)] = f_i^\star(z)$. Therefore, by (5.2) and (5.3), we have

$$\begin{aligned}
\mathbb{E}_{x\in\{-1,1\}^k, i\in[k]}\left[\Pr[x_i = \mathcal{A}_i(C(x))]\right] &= \frac{1}{2} + \frac{1}{2}\mathbb{E}_{x\in\{-1,1\}^k, i\in[k]}\left[x_i \mathbb{E}[\mathcal{A}_i(C(x))]\right] \\
&= \frac{1}{2} + \frac{1}{2}\mathbb{E}_{x\in\{-1,1\}^k, i\in[k]}\left[x_i f_i^\star(C(x))\right] \\
&= \frac{1}{2} + \frac{1}{2}\mathbb{E}_{x\in\{-1,1\}^k}\left[\left\|\frac{1}{k}\sum_{i=1}^k x_i f_i^\star\right\|_{L_\infty}\right] \\
&\geq \frac{1}{2} + \frac{\varepsilon}{4}.
\end{aligned}$$

Hence, C is a $(q, 1/q, \varepsilon/2)$-average-case smooth code. □

The final step before the proof of Theorem 5.1.3 is to show that for any distribution μ over smooth functions, there exists a distribution $\tilde{\mu}$ over smooth functions of bounded degree that is not much more concentrated than μ.

Lemma 5.3.2. *Let μ be a probability distribution over 1-smooth functions and let $\varepsilon > 0$. Then, there exists a probability distribution $\tilde{\mu}$ over 1-smooth functions of degree $q = 4/\varepsilon$ such that $\kappa_{\tilde{\mu}}(\varepsilon/2) \geq \kappa_\mu(\varepsilon)$.*

Proof. We first establish that smooth functions have low-degree approximations in the supremum norm. If $f : \{-1,1\}^n \to \mathbb{R}$ is 1-smooth, then

$$q \sum_{|S|>q} |\widehat{f}(S)| \leq \sum_{S\subset[n]} |S||\widehat{f}(S)| = \sum_{i=1}^n \sum_{S\ni i} |\widehat{f}(S)| = \sum_{i=1}^n \|D_i f\|_A \leq 1.$$

It follows that the degree-q truncation $f^{\leq q}$ satisfies

$$\left\|f - f^{\leq q}\right\|_{L_\infty} \leq \sum_{|S|>q} |\widehat{f}(S)| \leq \frac{1}{q} = \frac{\varepsilon}{4}. \tag{5.4}$$

Define $\tilde{\mu}$ as follows: sample f according to μ and output $f^{\leq q}$. Clearly, $\tilde{\mu}$ is also a distribution over 1-smooth functions. For $k = \kappa_\mu(\varepsilon)$, we have

$$\mathbb{E}_{f_1,\dots,f_k \sim \mu}\left[\left\|\frac{1}{k}\sum_{i=1}^{k}\Big(f_i - \mathbb{E}[f_i]\Big)\right\|_{L_\infty}\right] \geq \varepsilon.$$

Hence, by the triangle inequality and (5.4), we have

$$\mathbb{E}_{f_1,\dots,f_k \sim \tilde{\mu}}\left[\left\|\frac{1}{k}\sum_{i=1}^{k}\Big(f_i - \mathbb{E}[f_i]\Big)\right\|_{L_\infty}\right] \geq \frac{\varepsilon}{2},$$

giving the claim. □

Proof of Theorem 5.1.3. By applying Lemma 5.3.2 to μ, we get a distribution $\tilde{\mu}$ over 1-smooth degree $q = O(1/\varepsilon)$ functions with $k' = \kappa_{\tilde{\mu}}(\varepsilon/2) \geq \kappa_\mu(\varepsilon) = k$. By Theorem 5.3.1, we get a $(q, 1/q, \Omega(\varepsilon))$-average-case smooth code $C' : \{-1,1\}^{k'} \to \{-1,1\}^n$. Finally we use Lemma 5.1.6 to convert C' to a $(q, \Omega(\varepsilon), \Omega(\varepsilon))$-LDC $C : \{-1,1\}^\ell \to \{-1,1\}^n$ where $\ell = \Omega(\varepsilon^2 k'/\log(1/\varepsilon))$. For the last part of the theorem we can simply apply Theorem 5.3.1 directly. □

5.4 From LDCs to outlaws

In this section we prove Theorem 5.1.4, the converse of our main result.

Proof of Theorem 5.1.4. By Proposition 2.3.5, the map $C : \{-1,1\}^k \to \{-1,1\}^n$ is also a $(q, 1/\delta, \eta)$-smooth code. For each $i \in [k]$, let $\mathcal{B}_i$ be its decoder for the ith index. Let $\nu_i : 2^{[n]} \to [0,1]$ be the probability distribution used by $\mathcal{B}_i$ to sample a set $S \subseteq [n]$ of at most q coordinates and let $f_{i,S} : \{-1,1\}^n \to [-1,1]$ be function whose value at $y \in \{-1,1\}^n$ is the expectation of the random sign returned by $\mathcal{B}_i(y)$ conditioned on the event that it samples S. Since this value depends only on the coordinates in S, the function $f_{i,S}$ is a q-junta.

Fix an $i \in [k]$ and let $f_i : \{-1,1\}^n \to [-1,1]$ be the function given by $f_i = \mathbb{E}_{S\sim\nu_i}[f_{i,S}]$. Then, since a q-junta has degree at most q, so does f_i. We claim that f_i is $\delta/(q2^{q/2})$-smooth. Since the functions $f_{i,S} : \{-1,1\}^n \to \{-1,1\}$ are q-juntas, it follows from Parseval's identity that they have spectral norm at most $2^{q/2}$. Moreover, for each $j \in [n]$, we have $\Pr_{S\sim\nu_i}[j \in S] \le q/(\delta n)$. Hence, since $f_{i,S}$ depends only on the coordinates in S, we have

$$\|D_j f_i\|_A \le \sum_{S\ni j} \nu_i(S)\, \|f_{i,S}\|_A \le \frac{q2^{q/2}}{\delta n},$$

which gives the claim. By (2.3), it holds for every $x \in \{-1,1\}^k$ and every $i \in [k]$ that

$$x_i f_i\Big(C'(x)\Big) \ge \eta. \tag{5.5}$$

Define the distribution μ to correspond to the process of sampling $i \in [k]$ uniformly at random and returning f_i. Let $\bar{g} = (f_1 + \cdots + f_k)/k$ be the mean of μ. We show that $\kappa_\mu(\eta/2) \ge \eta k/2$. To this end, let $l = \eta k/2$, let $\sigma : [l] \to [k]$ be an arbitrary map and define the functions $g_1, \dots, g_l$ by $g_i = f_{\sigma(i)}$. Let $x \in \{-1,1\}^k$ be such that for each $i \in [l]$, we have $x_{\sigma(i)} = 1$ and $x_j = -1$ elsewhere. It follows from (5.5) that $f_{\sigma(i)}\Big(C(x)\Big) \in [\eta, 1]$ for every $i \in [l]$ and that $f_i\Big(C(x)\Big) \le 0$ for every other $i \in [k]$. Hence,

$$\begin{aligned}
\Big\|\frac{1}{l}\sum_{i=1}^{l}(g_i - \bar{g})\Big\|_{L_\infty} &\ge \Big(\frac{1}{l}\sum_{i=1}^{l}(g_i - \bar{g})\Big)\Big(C(x)\Big) \\
&= \frac{1}{l}\sum_{i=1}^{l} f_{\sigma(i)}\Big(C(x)\Big) - \frac{1}{k}\sum_{i=1}^{k} f_i\Big(C(x)\Big) \\
&\ge \eta - \frac{l}{k} = \eta/2.
\end{aligned}$$

If σ maps each element in $[l]$ to a uniformly random element in $[k]$, then $g_1, \ldots, g_l$ are independent, μ-distributed and satisfy

$$\mathbb{E}\left[\left\|\frac{1}{l}\sum_{i=1}^{l}(g_i - \bar{g})\right\|_{L_\infty}\right] \geq \eta/2,$$

which shows that $\kappa_\mu(\eta/2) \geq l$. Finally we can scale all the functions in μ to make them 1-smooth, and get a distribution $\tilde{\mu}$ over 1-smooth functions with $\kappa_{\tilde{\mu}}(\eta\delta/(2q2^{q/2})) \geq \eta k/2$. □

5.5 Candidate outlaws

In this section we elaborate on the candidate outlaws mentioned in the introduction.

5.5.1 Incidence geometry

We begin by describing a variant of Corollary 5.1.5 based on a slightly different assumption and show conditions under which this assumption holds. Let p be an odd prime, let $\mathbb{F}_p$ be a finite field with p elements and let n be a positive integer. For $x, y \in \mathbb{F}_p^n$, the *line* with origin x in direction y, denoted $\ell_{x,y}$, is the sequence $(x + \lambda y)_{\lambda \in \mathbb{F}_p}$. A line is nontrivial if $y \neq 0$.

Corollary 5.5.1. *For every odd prime p and $\varepsilon \in (0,1]$, there exist a positive integer $n_1(p,\varepsilon)$ and a $c = c(p,\varepsilon) \in (0,1/2]$ such that the following holds. Let $n \geq n_1(p,\varepsilon)$ and k be positive integers. Assume that for independent uniformly distributed elements $z_1, \ldots, z_k \in \mathbb{F}_p^n$, with probability at least $1/2$, there exists a set $B \subseteq \mathbb{F}_p^n$ of size εp^n such that every nontrivial line through the set $\{z_1, \ldots, z_k\}$ contains at most $p-2$ points of B. Then, there exists a $(p-1, c, c)$-LDC sending $\{0,1\}^l$ to $\{0,1\}^{p^n}$, where $l = \Omega(c^2 k/\log(1/c))$.*

The proof uses the following version of Szemerédi's Theorem [Tao12, Theorem 1.5.4] and its standard "Varnavides-type" corollary (see for example [TV06, Exercise 10.1.9]).

Theorem 5.5.2 (Szemerédi's theorem)**.** *For every odd prime p and any $\varepsilon \in (0,1]$, there exists a positive integer $n_0(p,\varepsilon)$ such that the following holds. Let $n \geq n_0(p,\varepsilon)$ and let $S \subseteq \mathbb{F}_p^n$ be a set of size $|S| \geq \varepsilon p^n$. Then, S contains a nontrivial line.*

Corollary 5.5.3. *For every odd prime p and any $\varepsilon \in (0,1]$, there exists a positive integer $n_1(p,\varepsilon)$ and a $c(p,\varepsilon) \in (0,1]$ such that the following holds. Let $n \geq n_1(p,\varepsilon)$ and let $S \subseteq \mathbb{F}_p^n$ be a set of size $|S| \geq \varepsilon p^n$. Then, S contains at least $c(p,\varepsilon)p^{2n}$ nontrivial lines, that is,*

$$\Pr_{x\in\mathbb{F}_p^n, y\in\mathbb{F}_p^n\setminus\{0\}}\Big[\{(x+\lambda y)_{\lambda=0}^{p-1}\} \subset S\Big] \geq c(p,\varepsilon).$$

Proof of Corollary 5.5.1. Abusing notation, we identify functions $f : \mathbb{F}_p^n \to \{-1,1\}$ with vectors in $\{-1,1\}^{\mathbb{F}_p^n}$. Let $\phi : \{-1,1\} \to \{0,1\}$ be the map $\phi(\alpha) = (\alpha+1)/2$. For a function $f : \mathbb{F}_p^n \to \{-1,1\}$, let $\phi(f) : \mathbb{F}_p^n \to \{0,1\}$ be the function $\phi(f)(x) = \phi(f(x))$ and for $f : \mathbb{F}_p^n \to \{0,1\}$, define $\phi^{-1}(f) : \mathbb{F}_p^n \to \{-1,1\}$ analogously.

For every $x \in \mathbb{F}_p^n$, let $F_x : \{-1,1\}^{\mathbb{F}_p^n} \to \mathbb{R}$ be the degree-$(p-1)$ function

$$F_x(f) = \mathbb{E}_{y\in\mathbb{F}_p^n\setminus\{0\}}\left[\prod_{\lambda\in\mathbb{F}_p^*} \phi(f)(x+\lambda y)\right]. \tag{5.6}$$

Then, for a set $B \subseteq \mathbb{F}_p^n$, the value $F_x(\phi^{-1}(1_B))$ equals the fraction of all nontrivial lines $\ell_{x,y}$ through x of which B contains the $p-1$ points $\{x+\lambda y : \lambda \in \mathbb{F}_p^*\}$. If B has size at least εp^n, it follows from Corollary 5.5.3 that $\mathbb{E}_{x\in\mathbb{F}_p^n}[F_x(\phi^{-1}(1_B))] \geq c(p,\varepsilon)$. Moreover, since the monomials in the expectation of (5.6) involve disjoint sets of variables and can be expanded as

$$\prod_{\lambda\in\mathbb{F}_p^*}\phi(f)(x+\lambda y)=\frac{1}{2^q}\sum_{S\subseteq\mathbb{F}_p^*}\prod_{\lambda\in S}f(x+\lambda y),$$

it follows that each F_x is $2(1-p^{-n})$-smooth.

Let μ be the uniform distribution over F_x. We claim that $\kappa_\mu(c(p,\varepsilon))\geq k$, which implies the result by Theorem 5.1.3 since μ is supported by degree $(p-1)$-functions. For every set $A\subseteq\mathbb{F}_p^n$, let $B_A\subseteq\mathbb{F}_p^n$ be a maximal set such that every nontrivial line through A contains at most $p-2$ points of B_A, and let $f_A=\phi^{-1}(1_{B_A})$. Let z be a uniformly distributed random variable over $\mathbb{F}_p^n$, let $z_1,\dots,z_k$ be independent copies of z and let $A=\{z_1,\dots,z_k\}$. Then, $F_{z_1},\dots,F_{z_k}$ are independent μ-distributed random functions. Moreover, in the event that both $|B_A|\geq\varepsilon p^n$ and every nontrivial line through A meets B_A in at most $p-2$ points, we have

$$|(F_{z_i}-\mathbb{E}[F_z])(f_A)|=\mathbb{E}\Big[F_z(\phi^{-1}(1_{B_A}))\Big]-F_{z_i}(\phi^{-1}(1_{B_A}))\geq c(p,\varepsilon)$$

for every $i\in[k]$. Since this event happens with probability at least $1/2$, we have

$$\mathbb{E}\Bigg[\Big\|\frac{1}{k}\sum_{i=1}^k\Big(F_{z_i}-\mathbb{E}[F_z]\Big)\Big\|_{L_\infty}\Bigg]\geq\mathbb{E}\Bigg[\Big|\frac{1}{k}\Big(\sum_{i=1}^k\Big(F_{z_i}-\mathbb{E}[F_z]\Big)\Big)(f_A)\Big|\Bigg]\geq\frac{c(p,\varepsilon)}{2},$$

which gives the claim. □

The proof of the formal version of Corollary 5.1.5 (given below) is similar to that of Corollary 5.5.1, so we omit it. In the following, $\mathbb{PF}_p^{n-1}$ is the projective space of dimension $n-1$, which is the space of directions in $\mathbb{F}_p^n$. The formal version of Corollary 5.1.5 is then as follows.

Corollary 5.5.4. *For every odd prime p and $\varepsilon\in(0,1]$, there exist a positive integer $n_1(p,\varepsilon)$ and a $c=c(p,\varepsilon)\in(0,1/2]$ such that the following holds. Let $n\geq n_1(p,\varepsilon)$ and k be positive integers. Suppose that for independent uniformly distributed elements*

$z_1, \ldots, z_k \in \mathbb{PF}_p^{n-1}$, with probability at least $1/2$, there exists a set $B \subset \mathbb{F}_p^n$ of size $|B| \geq \varepsilon p^n$ which does not contain any lines with direction in $\{z_1, \ldots, z_k\}$. Then, there exists a (p, c, c)-LDC sending $\{0,1\}^l$ to $\{0,1\}^{p^n}$, where $l = \Omega(c^2 k/\log(1/c))$.

Feasible parameters for Corollary 5.5.1 Proving lower bounds on k for which the assumption of Corollary 5.5.1 holds true thus allows one to infer the existence of $(p-1)$-query LDCs with rate $\Omega(k/N)$ for $N = p^n$, provided p and ε are constant with respect to n. We establish the following bounds, which imply the (well-known) existence of $(p-1)$-query LDCs with message length $k = \Omega((\log N)^{p-2})$.

Theorem 5.5.5. *For every odd prime p there exists an $\varepsilon(p) \in (0, 1]$ such that the following holds. For every set $A \subseteq \mathbb{F}_p^n$ of size $|A| \leq \binom{n+p-3}{p-2} - 1$, there exists a set $B \subseteq \mathbb{F}_p^n$ of size $\varepsilon(p)p^n$ such that every line through A contains at most $p-2$ points of B.*

The proof uses some basic properties of polynomials over finite fields. For an n-variate polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ denote $Z(f) = \{x \in \mathbb{F}_p^n : f(x) = 0\}$. The starting point of the proof is the following standard result (see for example [Tao14]), showing that small sets can be 'captured' by zero-sets of nonzero, homogeneous polynomials of low degree.

Lemma 5.5.6 (Homogeneous Interpolation)**.** *For every $A \subseteq \mathbb{F}_p^n$ of size $|A| \leq \binom{n+d-1}{d} - 1$, there exists a nonzero homogeneous polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ of degree d such that $A \subseteq Z(f)$.*

The next two lemmas show that if f is nonzero, homogeneous and degree d, and if $a \in \mathbb{F}_p^*$ is such that $f^{-1}(a)$ is nonempty, then lines through $Z(f)$ meet $f^{-1}(a)$ in at most d points.

Lemma 5.5.7. *Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero homogeneous polynomial of degree d. Let $a \in \mathbb{F}_p^*$ be such that the set $f^{-1}(a)$ is nonempty. Then, every line that meets $f^{-1}(a)$ in $d+1$ points must have direction in $Z(f)$.*

Proof. The univariate polynomial $g(\lambda) = f(x + \lambda y)$ formed by the restriction of f to a line $\ell_{x,y}$ has degree at most d. By the Factor Theorem, such a polynomial must be the constant polynomial $g(\lambda) = a$ to assume the value a for $d + 1$ values of λ. Since f is homogeneous, the coefficient of λ^d, which must be zero, equals $f(y)$, giving the result. □

The following lemma is essentially contained in [BR16].

Lemma 5.5.8 (Briët–Rao)**.** *Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero homogeneous polynomial of degree d. Let $a \in \mathbb{F}_p^*$ be such that $f^{-1}(a)$ is nonempty. Then, there exists no line that intersects $Z(f)$, meets $f^{-1}(a)$ in at least d points and has direction in $Z(f)$.*

Proof. For a contradiction, suppose there exists a line $\ell_{x,y}$ through $Z(f)$ that meets $f^{-1}(a)$ in d points and has direction $y \in Z(f)$. Observe that for every $\lambda \in \mathbb{F}_p$, the shifted line $\ell_{x+\lambda y,y}$ also meets $f^{-1}(a)$ in d points. Hence, without loss of generality we may assume that the line starts in $Z(f)$, that is $x \in Z(f)$. Let $g(\lambda) = a_0 + a_1\lambda + \cdots + a_d\lambda^d = f(x + \lambda y) \in \mathbb{F}_p[\lambda]$ be the restriction of f to $\ell_{x,y}$. It follows that $a_0 = g(0) = f(x) = 0$ and, since f is homogeneous, that $a_d = f(y) = 0$. Moreover, there exist distinct elements $\lambda_1, \ldots, \lambda_d \in \mathbb{F}_p^*$ such that $g(\lambda_i) = f(x + \lambda_i y) = a$ for every $i \in [d]$. Then $g(\lambda) - a$ is a degree $d - 1$ polynomial with d distinct roots. But it cannot be the zero polynomial since it takes value $-a$ when $\lambda = 0$. □

The final ingredient for the proof of Theorem 5.5.5 is the DeMillo–Lipton–Schwartz–Zippel Lemma, as it appears in [CT14].

Lemma 5.5.9 (DeMillo–Lipton–Schwartz–Zippel)**.** *Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero polynomial of degree d and denote $r = |\mathbb{F}_p|$. Then,*

$$|Z(f)| \leq \Big(1 - \frac{1}{r^{d/(r-1)}}\Big) r^n.$$

Proof of Theorem 5.5.5. Let $A \subseteq \mathbb{F}_p^n$ be a set of size $|A| \leq \binom{n+p-3}{p-2} - 1$. Let $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ be a nonzero degree-$(p-2)$ homogeneous polynomial such that $A \subseteq Z(f)$, as promised to exist by Lemma 5.5.6. By Lemma 5.5.9, there exists an $a \in \mathbb{F}_p^*$ such that the set $B = f^{-1}(a)$ has size at least $|B| \geq p^n / p^{(2p-3)/(p-1)}$. By Lemma 5.5.7, every line that meets B in $p-1$ points must have direction in $Z(f)$, but by Lemma 5.5.8 no such line can pass through $Z(f)$. Hence, every line through A meets B in at most $p-2$ points. □

5.5.2 Hypergraph pseudorandomness

A second candidate for constructing outlaws comes from special types of hypergraphs. A hypergraph $H = (V, E)$ is a pair consisting of a finite vertex set V and an edge set E of subsets of V that allows for parallel (repeated) edges. A hypergraph is t-uniform if all its edges have size t. For subsets $W_1, \ldots, W_t \subseteq V$, define the induced edge count by

$$e_H(W_1, \ldots, W_t) = \sum_{v_1 \in W_1} \cdots \sum_{v_t \in W_t} 1_E(\{v_1, \ldots, v_t\}).$$

A perfect matching in a t-uniform hypergraph is a family of vertex-disjoint edges that intersects every vertex. We shall use the following notion of pseudorandomness.

Definition 5.5.10 (Relative pseudorandomness)**.** *Let $H = (V, E)$, $J = (V, E')$ be t-uniform hypergraphs with identical vertex sets. Then J is ε-pseudorandom relative to H if for all $W_1, \ldots, W_t \subseteq V$, we have*

$$\left| \frac{e_J(W_1, \ldots, W_t)}{|E'|} - \frac{e_H(W_1, \ldots, W_t)}{|E|} \right| < \varepsilon. \tag{5.7}$$

The left-hand side of (5.7) compares the fraction of edges that the sets $W_1, \ldots, W_t$ induce in J with the fraction of edges they induce in H. Standard concentration arguments show that if $|E| \geq |V|$, then a random hypergraph J whose edge set E' is formed by independently putting each edge of E in E' with probability $p = p(\varepsilon, t)$, is ε-pseudorandom relative to H with high probability. A deterministic hypergraph J is thus pseudorandom relative to H if it mimics this property of truly random sub-hypergraphs. For graphs, relative ε-pseudorandomness turns into a common notion sometimes referred to as ε-uniformity when H is the complete graph with all loops, in which case (5.7) says that the number of edges induced by a pair of vertex-subsets W_1, W_2 is roughly equal to the product of their densities $(|W_1|/|V|)(|W_2|/|V|)$. Uniformity in graphs is closely connected to the perhaps better-known notion of spectral expansion [HLW06]. These two notions were recently shown to be equivalent (up-to universal constants) for all vertex-transitive graphs [CZ17].

We shall be interested in hypergraphs whose edge set can be partitioned into a family of "blocks", such that randomly removing relatively few of the blocks likely leaves a hypergraph that is *not* pseudorandom relative to the original. (Think of a Jenga tower[2] that's already in a delicate balance, so that there are only few ways, or perhaps even no way, to remove many blocks without having it collapse.) Our blocks will be formed by perfect matchings. For technical reasons, the formal definition takes the view of building a new hypergraph out of randomly selected matchings, as opposed to obtaining one by randomly removing matchings.

Definition 5.5.11 (Jenga hypergraph)**.** *A t-uniform hypergraph H is (k, ε)-jenga if its edge set can be partitioned into a family $\mathcal{M}$ of perfect matchings such that, with probability at least $1/2$, the disjoint union of k independent uniformly distributed matchings from $\mathcal{M}$ forms a hypergraph which is* not *ε-pseudorandom relative to H.*

We have the following simple corollary to Theorem 5.1.3.

[2] *Jenga®* is a game of dexterity in which players begin with a tower of wooden blocks and take turns trying to remove a block without making the tower collapse.

Corollary 5.5.12. *Let n, k, t be positive integers and $\varepsilon \in (0,1]$. Assume that there exists a t-uniform n-vertex hypergraph that is (k,ε)-jenga. Then, there exists a (t,η,η)-LDC sending $\{0,1\}^l$ to $\{0,1\}^{tn}$, where*

$$\eta = \Omega(\varepsilon/t^2) \text{ and } l = \Omega(\varepsilon^2 k/t^4 \log(t^2/\varepsilon)).$$

Proof. Let $H = (V,E)$ be a hypergraph as assumed in the corollary. Let $\mathcal{M}$ be a partition of E into perfect matchings such that if $M_1, \ldots, M_k$ are independent and uniformly distributed over $\mathcal{M}$, then with probability at least $1/2$, the hypergraph $J = (V, M_1 \uplus \cdots \uplus M_k)$ is not ε-pseudorandom relative to H.

Let $V_1, \ldots, V_t$ be copies of V. For each $M \in \mathcal{M}$, define $f_M : \mathbb{R}^{V_1 \cup \cdots \cup V_t} \to \mathbb{R}$ by

$$f_M(x[1], \ldots, x[t]) = \frac{1}{|M|} \sum_{v_1 \in V_1} \cdots \sum_{v_t \in V_t} 1_M(\{v_1, \ldots, v_t\})\, x[1]_{v_1} \cdots x[t]_{v_t}, \quad x[i] \in \mathbb{R}^{V_i}.$$

The function f_M is a degree-t polynomial. Since every one of the tn variables appears in exactly one monomial and $|M| = n/t$, the restriction of f_M to $\{-1,1\}^{V_1 \cup \cdots \cup V_t}$ is t^2-smooth. Moreover, for $J = (V, M)$ and $W_1, \ldots, W_t \subseteq V$, we have

$$f_M(1_{W_1}, \ldots, 1_{W_t}) = \frac{e_J(W_1, \ldots, W_t)}{|M|}.$$

Let $M_1, \ldots, M_k$ be independent uniformly distributed matchings from $\mathcal{M}$ and consider the random hypergraph $J = (V, M_1 \uplus \cdots \uplus M_k)$. Let $\bar{f} = \mathbb{E}[f_{M_1}]$ be the expectation of the random function f_{M_1} and note that $\mathbb{E}[f_{M_i}] = \bar{f}$ for each $i \in [k]$.

Then, since the functions $f_{M_i} - \bar{f}$ are multilinear,

$$\begin{aligned}
\mathbb{E}\Big[\Big\|\frac{1}{k}\sum_{i=1}^{k}(f_{M_i} - \bar{f})\Big\|_{L_\infty}\Big] &\geq \mathbb{E}\Big[\max_{W_1,\dots,W_t \subseteq V}\Big|\frac{1}{k}\sum_{i=1}^{k}(f_{M_i} - \bar{f})(1_{W_1},\dots,1_{W_t})\Big|\Big] \\
&= \mathbb{E}\Big[\max_{W_1,\dots,W_t \subseteq V}\Big|\frac{e_J(W_1,\dots,W_t)}{k|M|} - \frac{e_H(W_1,\dots,W_t)}{|E|}\Big|\Big] \\
&\geq \frac{\varepsilon}{2}.
\end{aligned}$$

The result now follows from Theorem 5.1.3. □

In the context of outlaws and LDCs, the relevant question concerning Jenga hypergraphs is the following. Let $\kappa^J(n,t,\varepsilon)$ denote the maximum integer k such that there exists an n-vertex t-uniform hypergraph that is (k,ε)-jenga.

Question 5.5.13. *For integer $t \geq 2$ and parameter $\varepsilon \in (0,1]$, what is the growth rate of $\kappa^J(n,t,\varepsilon)$ as a function of $n \in \mathbb{N}$?*

For $t = 2$ (graphs), the answer to Question 5.5.13 follows from famous work of Alon and Roichman [AR94] on expansion of random Cayley graphs, which implies that for constant $\varepsilon \in (0,1]$, we have $\kappa(n,2,\varepsilon) = \Theta(\log n)$. The lower bound follows for instance by partitioning the edge set of the complete graph with vertex set $V = \mathbb{F}_2^m$ into the collection of matchings of the form $M_y = \big\{\{x, x+y\} : x \in \mathbb{F}_2^m\big\}$ for each $y \in \mathbb{F}_2^m \setminus \{0\}$. Any $m-1$ of such matchings give a graph with two disconnected components of equal size, making it $(m-1,\frac{1}{4})$-jenga. Via Corollary 5.5.12, this arguably gives the most round-about way to prove the existence of 2-query LDCs matching the paramaters of the Hadamard code! Generalizing the above example, [BR16] considered the p-uniform hypergraph on $\mathbb{F}_p^m$ whose edges are the (unordered) nontrivial lines. It was shown that this hypergraph is (m^{p-1},ε)-jenga for some $\varepsilon = \varepsilon(p)$ depending on p only, by partitioning the edge set according to the directions of the lines, that is, partitioning it with the matchings $M_y = \big\{\{x+\lambda y : \lambda \in \mathbb{F}_p\} : x \in \mathbb{F}_p^m\big\}$,

$y \in \mathbb{F}_p^m \setminus \{0\}$. To the best of our knowledge, the best upper bounds on $\kappa^J(n, t, \varepsilon)$ for constant $t \geq 3$ and $\varepsilon \in (0, 1]$ follow from upper bounds on LDCs, via Corollary 5.5.12.

We end with the following natural question concerning Jenga hypergraphs.

Question 5.5.14. *Is $\kappa^J(n, t, \varepsilon)$ largest for the complete hypergraph?*

Chapter 6

Lower bounds for affine invariant local codes

6.1 Introduction

We restrict ourselves throughout this chapter to the setting where the query complexity is a constant (independent of the length of the code) and consider the tradeoff between query complexity and code length. The current best constant-query LCCs have exponential length, while the current best constant-query LTCs have near-linear length but they are quite complicated [BS08, Din07, Mei09, Vid15]. Getting subexponential length LCCs or linear length LTCs with constant query complexity are major open problems in the area.

Intuitively, for LCCs and LTCs with constant query complexity, there must be a lot of redundancy in the code, since every symbol of the codeword must satisfy local constraints with most other symbols in the codeword. A systematic way to generate redundancy is to make sure that the code has a large group of *invariances*[1].

[1] A quite different way to generate redundancy is through *tensoring*; see [BSS06]. Invariances and tensoring are essentially the only two "generic" reasons known to cause local correctability/testability.

Formally, given a code $\mathcal{C} \subset \Sigma^N$ of length N over alphabet Σ, a codeword $c \in \mathcal{C}$ can be naturally viewed as a function $c : [N] \to \Sigma$. Then, we say that $\mathcal{C}$ is invariant under a set[2] $G \subset \{[N] \to [N]\}$ if for every $\pi \in G$ and codeword $c \in \mathcal{C}$, $c \circ \pi$ also describes a codeword $c' \in \mathcal{C}$. Now, the key observation is that if for every codeword $c \in \mathcal{C}$, if there is a constraint among $c(i_1), \ldots, c(i_k)$ for some $i_1, \ldots, i_k \in [N]$, then for every $c \in \mathcal{C}$, there must also be a constraint among $c(\pi(i_1)), \ldots, c(\pi(i_k))$ for any π in the invariance set G, since $c \circ \pi$ is itself another codeword. Hence if G is large, the presence of one local constraint immediately implies presence of many and suggests the possibility of local algorithms for the code. This connection between invariance and correctability/testability was first explicitly examined by Kaufman and Sudan [KS08]. One is then motivated to understand more clearly the possibilities and limitations of local correctors/testers for codes possessing natural symmetries.

We focus on affine-invariant codes, for which the domain $[N]$ is an n-dimensional vector space $\mathbb{K}^n$ over a finite field $\mathbb{K}$ and the code $\mathcal{C} \subset \{\mathbb{K}^n \to \Sigma\}$ is invariant under affine transformations $A : \mathbb{K}^n \to \mathbb{K}^n$. Affine invariance is a very natural symmetry for "algebraic codes" and has long been studied in coding theory [KLP67]. The study of affine-invariant LCCs and LTCs was initiated in [KS08] and has been investigated in several follow-up works [BSS11, Guo13, BSRZS12, GSVW14]. The hope is that because affine-invariant codes have a large group of invariance and, at the same time, are conducive to non-trivial algebraic constructions, they may contain a code that improves current constructions of LCCs or LTCs.

The current best parameters for constant-query affine-invariant LCCs and LTCs are achieved by the lifted codes of Guo, Kopparty and Sudan [GKS13]. They construct an affine-invariant code $\mathcal{F} \subset \{\mathbb{F}_{2^\ell}^n \to \mathbb{F}_2\}$ with $\exp(\Theta(n^{r-2}))$ codewords that is an $(r-1)$-query LCC and an r-query LTC, where $r = 2^\ell$. The $\Theta(\cdot)$ notation hides factors that depend on r but not n. For LCCs, the same asymptotic tradeoff between

[2] $\{A \to B\}$ and B^A denote the set of all functions from A to B.

query complexity and code length is achieved by the Reed-Muller code. For every $r \geq 2$, the Reed-Muller code of order $r-1$ (i.e., polynomials over $\mathbb{F}_q$ on n variables of total degree $\leq r-1$ with $q > r$) is an affine-invariant r-query LCC with $\exp(\Theta(n^{r-1}))$ codewords. In fact, even if we drop the affine-invariance requirement, Reed-Muller codes and the construction of [GKS13] achieve the best known codeword length for constant query LCCs[3].

In this work, we show that the parameters for the lifted codes of [GKS13] are, in fact, tight for affine-invariant LCCs/LTCs in $\{\mathbb{K}^n \to \Sigma\}$ for any fixed finite field $\mathbb{K}$ and any fixed finite alphabet Σ.

Theorem 6.1.1 (Main Result, informal)**.**

(i) Let $\mathcal{C} \subset \{\mathbb{K}^n \to \Sigma\}$ be an r-query affine-invariant LCC. Then $|\mathcal{C}| \leq \exp\left(O_{\mathbb{K},r,|\Sigma|}(n^{r-1})\right)$.

(ii) Let $\mathcal{C} \subset \{\mathbb{K}^n \to \Sigma\}$ be an r-query affine-invariant LTC. Then $|\mathcal{C}| \leq \exp\left(O_{\mathbb{K},r,|\Sigma|}(n^{r-2})\right)$.

Note that a local constraint among t coordinates can be used to correct one of the coordinate using $t-1$ queries by a local corrector whereas a local tester needs to make t queries to check the constraint. This explains the difference in the dependence of r in the bounds for LCCs and LTCs.

6.1.1 Related Work

Ben-Sasson and Sudan in [BSS11] obtained a similar result as Theorem 6.1.1, when the code is assumed to be linear, i.e., when the codewords form a vector space. They showed that if $\mathcal{C} \subset \{\mathbb{K}^n \to \mathbb{F}\}$ is an $(r-1)$-query locally correctable or r-query locally

[3] In contrast, there exist non-affine-invariant LTCs of constant query complexity and inverse polylogarithmic rate. This corresponds to a constant query LTC $\mathcal{C} \subset \{\{0,1\}^n \to \{0,1\}\}$ with $\exp(2^n/\mathrm{poly}(n))$ codewords, while the affine-invariant LTC of [GKS13] and Reed-Muller codes have $\exp(\mathrm{poly}(n))$ codewords for constant query complexity.

testable *linear*, affine-invariant code, where $\mathbb{K}$ and $\mathbb{F}$ are finite fields of characteristic $p > 0$ with $\mathbb{K}$ an extension of $\mathbb{F}$, then the dimension of $\mathcal{C}$ as a vector space over $\mathbb{F}$ is at most $(n \log_p |\mathbb{K}|)^{r-2}$. When $\mathbb{K}$ is fixed (as in [GKS13]'s construction of constant query LCCs/LTCs), the result of [BSS11] is a very special case of our Theorem 6.1.1. On the other hand, [BSS11]'s result also applies when the size of $\mathbb{K}$ is growing (as long as $\mathbb{K}$ extends $\mathbb{F}$), whereas ours does not.

Since LCCs are stronger than LDCs, lower bounds for LDCs also apply to LCCs. Unfortunately, stronger lower bounds are not known. For general (non-affine-invariant) LCCs, tight lower bounds are known only for 2-query LCCs. In [KW04, WdW05a], it was shows that if $\mathcal{C} \subset \{\{0,1\}^n \to \Sigma\}$ is a 2-query LCC[4], then $|\mathcal{C}| \leq \exp(O(n|\Sigma|^2))$. This is tight for constant Σ and achieved by the Hadamard code. For r-query LCCs where $r > 2$, the lower bounds known are much weaker. The best known bounds, due to [KW04, Woo07], show that if $\mathcal{C} \subset \{\{0,1\}^n \to \{0,1\}\}$ is an r-query LCC, then

$$|\mathcal{C}| \leq \exp\left(2^{n/(1+1/(\lceil r/2 \rceil+1))+o(n)}\right).$$

See Section 2.3.4, Section 2.4.3 and Section 2.5.1 for more information about lower bounds.

Higher-order Fourier analysis was applied to other problems in coding theory in [BL18, TW14].

6.1.2 Proof Overview

Our arguments are based on standard techniques from higher-order Fourier analysis [Tao12], but they are new in this context. We show that if an affine-invariant code is an r-query LCC, then its codewords are far from each other in the U^r-norm, the

[4] The lower bound also holds for the weaker notion of locally decodable code (LDC)

Gowers norm of order r. Similarly, we show that the codewords of an affine-invariant r-query LTC are far from each other in the U^{r-1}-norm. Therefore, we can upper bound the number of LCC/LTC codewords in terms of the size of a net that is fine enough with respect to the Gowers norm of an appropriate order. We bound the size of such a net by explicitly constructing one using a standard decomposition theorem (analogous to Szemerédi's regularity lemma): any bounded function $f : \mathbb{K}^n \to \mathbb{C}$ can be approximated, up to a small error in the Gowers norm, by a composition of a bounded number of low-degree non-classical polynomials [TZ12].

The way we argue that two codewords f and g of an r-query LCC are far in the Gowers norm is that if $\|f - g\|_{U^r} < \varepsilon$, then for small enough ε (with respect to r, $|\Sigma|$ and correctness probability), the local corrector when applied to f can act as if it is applied to g. The argument is, briefly, as follows. On the one hand, the codewords f and g must be far in Hamming distance, because the definition of LCC implies that there is a unique codeword close to any string. So, with constant probability over choice of $y \in \mathbb{K}^n$, the local corrector's guess for $f(y)$ must differ from $g(y)$. On the other hand, we can lower bound by a constant the probability of the event that the corrector outputs $g(y)$ when it queries coordinates of f, because f and g are close in the $\|\cdot\|_{U^r}$ norm. This last calculation uses the affine invariance of the code and the *generalized von Neumann inequality*, which bounds by $\|f_0\|_{U^k}$ the expectation over $z_1, \ldots, z_m \in \mathbb{K}^n$ of the product $\prod_{i=0}^{k} f_i(\mathcal{L}_i(z_1, \ldots, z_m))$, where the $\mathcal{L}_i$'s are arbitrary linear forms so that no two are linearly dependent and $f_i : \mathbb{K}^n \to \mathbb{C}$ are arbitrary functions with $|f_i| \leq 1$.

The argument for r-query LTCs is similar. Suppose f and g are close in the $\|\cdot\|_{U^{r-1}}$ norm. Consider the random function H such that for every x independently, $H(x)$ equals $f(x)$ with probability 1/2 and $g(x)$ with probability 1/2. H itself is far from a codeword with high probability. But we show that since the local tester accepts f, it will also accept $H \circ \ell$ for a random invertible affine map $\ell : \mathbb{K}^n \to \mathbb{K}^n$ with good probability. This implies that with good probability, $H \circ \ell$ is close to a codeword and by affine-invariance, H itself is close to a codeword which gives a contradiction. To draw this conclusion, we again use the generalized von Neumann inequality as well as a hybrid argument.

Organization Section 6.2 contains preliminaries that lay the foundations of our analysis. Section 6.3 proves the first part of our main result about LCCs, while Section 6.4 proves the second part about LTCs.

6.2 Preliminaries

6.2.1 Error-correcting codes and affine invariance

Here we recall a few definitions about error correcting codes from Section 2.2 and setup some notation. Let $\mathcal{X}$ be a finite set called the set of coordinates and Σ be an other finite set called the alphabet. Let $\Sigma^{\mathcal{X}}$ denote the set of all functions from $\mathcal{X} \to \Sigma$. A subset $\mathcal{C} \subset \Sigma^{\mathcal{X}}$ is called a code and its elements are called codewords. Given $f, g \in \Sigma^{\mathcal{X}}$, the (normalized) Hamming distance between f and g is

$$\mathrm{dist}_H(f, g) := \Pr_{x \in \mathcal{X}}[f(x) \neq g(x)]$$

where x is uniformly chosen from $\mathcal{X}$. For a code $\mathcal{C} \subset \Sigma^{\mathcal{X}}$, the minimum distance of $\mathcal{C}$ as $\min_{f,g \in \mathcal{C}, f \neq g} \mathrm{dist}_H(f, g)$.

Let $\blacktriangle_\Sigma = \{q : \Sigma \to \mathbb{R}_{\geq 0} : \sum_{i \in \Sigma} q(i) = 1\}$ denote the probability simplex on Σ. We embed Σ into $\blacktriangle_\Sigma$ by sending $i \in \Sigma$ to e_i which is the i^{th} coordinate vector in $\mathbb{R}^\Sigma$. This also lets us extend functions $f : \mathcal{X} \to \Sigma$ to $\hat{f} : \mathcal{X} \to \blacktriangle_\Sigma$ using the embedding. We call $\hat{f}$ the simplex extension of f. Now given $f, g \in \Sigma^{\mathcal{X}}$, we can write the Hamming distance between them as

$$\mathrm{dist}_H(f,g) = 1 - \Pr_{x\in\mathcal{X}}[f(x) = g(x)] = 1 - \mathbb{E}_{x\in\mathcal{X}}\langle \hat{f}, \hat{g}\rangle$$

where $\langle\cdot,\cdot\rangle$ is the standard inner product in $\mathbb{R}^{\Sigma}$.

Definition 6.2.1 (Affine invariance)**.** *Let $\mathcal{X}$ be a finite dimensional vector space over some finite field $\mathbb{K}$, then $\mathcal{C} \subset \Sigma^{\mathcal{X}}$ is called affine invariant if for every $f \in \mathcal{C}$ and every invertible affine map $\ell : \mathcal{X} \to \mathcal{X}$, $f \circ \ell \in \mathcal{C}$.*

Locally correctable and testable codes are defined formally in Sections 6.3 and 6.4 respectively.

6.2.2 Higher order Fourier analysis

Fix a finite field $\mathbb{F}_p$ of prime order p, and let $\mathbb{K} = \mathbb{F}_q$ where $q = p^t$ for a positive integer t. $\mathbb{K}$ is then a vector space of dimension t over $\mathbb{F}_p$. We denote by $\mathrm{Tr} : \mathbb{K} \to \mathbb{F}_p$ the *trace function*:

$$\mathrm{Tr}(x) = x + x^p + x^{p^2} + \cdots + x^{p^{t-1}}.$$

Also, we use $|\cdot|$ to denote the obvious map from $\mathbb{F}_p$ to $\{0, 1, \ldots, p-1\}$.

Given functions $f, g : \mathbb{K}^n \to \mathbb{C}$, we define their inner product as $\langle f, g\rangle = \mathbb{E}_x[\overline{f(x)}g(x)]$ where x is chosen uniformly from $\mathbb{K}^n$. We define $\|\cdot\|_p$-norm on such functions as $\|f\|_p = \mathbb{E}_x[|f(x)|^p]^{1/p}$. We say a function $f : \mathbb{K}^n \to \mathbb{C}$ is *bounded* if $|f| \le 1$. Let $\mathbb{T}$ denote the circle group $\mathbb{R}/\mathbb{Z}$ and $e : \mathbb{T} \to \mathbb{C}$ be the map given by $e(x) = \exp(2\pi i x)$.

Definition 6.2.2 (Non-classical Polynomials)**.** *A* non-classical polynomial of degree *$< d$ is a function $f : \mathbb{K}^n \to \mathbb{T}$ if*

$$\forall h_1, h_2 \cdots, h_d \in \mathbb{K}^n \quad D_{h_1}D_{h_2}\cdots D_{h_d}f = 0$$

where D_h is the difference operator defined as $D_hf(x) = f(x+h) - f(x)$. For such an f, the function $e(f)$ is called a non-classical phase polynomial of degree $< d$.

Note that the derivative operator is linear, and so, the multiplicative structure of the field $\mathbb{K}$ is ignored here. Because as an additive group, $\mathbb{K}^n$ is isomorphic to $\mathbb{F}^{tn}$, a non-classical polynomial $P : \mathbb{K}^n \to \mathbb{T}$ over $\mathbb{K}$ can also be identified as a non-classical polynomial $P : \mathbb{F}^{tn} \to \mathbb{T}$ over $\mathbb{F}$.

Let $\alpha_1, \cdots, \alpha_t \in \mathbb{K}$ be a basis for $\mathbb{K}$ when viewed as a vector space over $\mathbb{F}_p$. It is known [TZ12, BB15] that non-classical polynomials of degree $\leq d$ are exactly those functions $P : \mathbb{K}^n \to \mathbb{T}$ which have the following form:

$$\begin{aligned} &P(x_1, \ldots, x_n) \\ &= \theta + \sum_{k\geq 0} \sum_{\substack{0\leq d_{i,j}<p \ \forall i\in[n],j\in[t]; \\ 0<\sum_{i=1}^{n}\sum_{j=1}^{t} d_{i,j}\leq d-k(p-1)}} \frac{c_{d_{1,1},\ldots,d_{n,t},k} \prod_{i=1}^{n}\prod_{j=1}^{t} |\mathrm{Tr}(\alpha_j x_i)|^{d_{i,j}}}{p^{k+1}} \pmod 1 \end{aligned} \tag{6.1}$$

for some $c_{d_{1,1},\ldots,d_{n,t},k} \in \{0, 1, \cdots, p-1\}$ and $\theta \in \mathbb{T}$. Next, we define the Gowers norm for arbitrary functions $f : \mathbb{K}^n \to \mathbb{C}$.

Definition 6.2.3 (Gowers uniformity norm [Gow01])**.** *For a function $f : \mathbb{K}^n \to \mathbb{C}$, the* Gowers norm of order r*, denoted by $\| \cdot \|_{U^r}$, is defined as*

$$\|f\|_{U^r} = \left(\mathbb{E}_{x,h_1,\cdots,h_r\in\mathbb{K}^n}[\Delta_{h_1}\Delta_{h_2}\cdots\Delta_{h_r}f(x)]\right)^{1/2^r}$$

where Δ_h is the multiplicative difference operator defined as $\Delta_h f(x) = f(x+h)\overline{f(x)}$.

The Gowers norm is an actual norm when $r \geq 2$. It also satisfies a useful monotonicity property: for any function $f : \mathbb{K}^n \to \mathbb{C}$,

$$|\mathbb{E}[f(x)]| = \|f\|_{U^1} \leq \|f\|_{U^2} \leq \cdots \leq \|f\|_{U^r} \leq \cdots \leq \|f\|_{\infty}.$$

See [Tao12] for more on Gowers norm. Observe that if $f : \mathbb{K}^n \to \mathbb{C}$ is a non-classical phase polynomial of degree $< r$ then $\|f\|_{U^r} = 1$. The inverse Gowers theorem is a partial converse to this. It shows that the Gowers norm of order r of a function is in direct correspondence with its correlation with non-classical phase polynomials of degree $< r$. In particular:

Lemma 6.2.4 (Inverse Gowers theorem [TZ12]). *For any bounded*[5] $f : \mathbb{K}^n \to \mathbb{C}$*, if* $\|f\|_{U^r} > \delta$ *then there exists a non-classical polynomial* P *of degree* $< r$ *such that*

$$| \langle f, e(P) \rangle | \geq c(\delta, \mathbb{K}, r)$$

where $c(\delta, \mathbb{K}, r)$ *is a constant depending only on* $\delta, \mathbb{K}, r$*.*

A linear form on m variables is a vector $\mathcal{L} = (w_1, \cdots, w_m) \in \mathbb{K}^m$ that is interpreted as a function $\mathcal{L} : (\mathbb{K}^n)^m \to \mathbb{K}^n$ via the map $(x_1, \cdots, x_m) \mapsto \sum_{i=1}^m w_i x_i$. A key reason that the Gowers norm is useful in applications is that if a function has small Gowers norm of the appropriate order, then it behaves pseudorandomly in a certain way with respect to linear forms.

Lemma 6.2.5 (Generalized von Neumann inequality (Exercise 1.3.23 in [Tao12])). *Let* $f_0, f_1, f_2, \cdots, f_k : \mathbb{K}^n \to \mathbb{C}$ *be bounded functions and let* $\mathcal{L} = \{\mathcal{L}_0, \mathcal{L}_1, \cdots, \mathcal{L}_k\}$ *be a system of* $k + 1$ *linear forms in* m *variables such that no form is a multiple of another. Then*

$$|\mathbb{E}_{z_1, \cdots, z_m \in \mathbb{K}^n} [\prod_{i=0}^{k} f_i(\mathcal{L}_i(z_1, \cdots, z_m))]| \leq \min_{0 \leq i \leq k} \|f_i\|_{U^k}.$$

See Appendix 6.5 for proof.

[5] Note that bounded means $|f| \leq 1$.

6.2.3 A net for Gowers norm

The goal of this section is to establish the following claim.

Theorem 6.2.6 (ε-net for U^r norm)**.** *The metric induced by the $\|\cdot\|_{U^r}$ norm on the space of all bounded functions $\{f : \mathbb{K}^n \to \mathbb{C}\}$ has an ε-net of size* $\exp(O_{\varepsilon,\mathbb{K},r}(n^{r-1}))$.

For the proof, we need the following definitions.

Definition 6.2.7 (Polynomial factors)**.** *A polynomial factor $\mathcal{B}$ is a sequence of non-classical polynomials $P_1, ..., P_k : \mathbb{K}^n \to \mathbb{T}$. We also identify it with the function $\mathcal{B} : \mathbb{K}^n \to \mathbb{T}^k$ mapping $x \mapsto (P_1(x), ..., P_k(x))$. The partition induced by $\mathcal{B}$ is the partition of $\mathbb{K}^n$ given by $\{\mathcal{B}^{-1}(y) : y \in \mathbb{T}^k\}$. The complexity of $\mathcal{B}$ is the number of defining polynomials, $|\mathcal{B}| = k$. The degree of $\mathcal{B}$ is the maximum degree among its defining polynomials $P_1, \cdots, P_k$. A function $f : \mathbb{K}^n \to \mathbb{C}$ is called $\mathcal{B}$-measurable if it is constant in each cell of the partition induced by $\mathcal{B}$ or equivalently f can be written as a $\tau(P_1, \cdots, P_k)$ for some function $\tau : \mathbb{T}^k \to \mathbb{C}$.*

Definition 6.2.8 (Conditional expectations)**.** *Given a polynomial factor $\mathcal{B}$, the conditional expectation of $f : \mathbb{K}^n \to \mathbb{C}$ over $\mathcal{B}$, denoted by $\mathbb{E}[f|\mathcal{B}]$, is the $\mathcal{B}$-measurable function defined by*

$$\mathbb{E}[f|\mathcal{B}](x) = \mathbb{E}_{y \in \mathcal{B}^{-1}(\mathcal{B}(x))}[f(y)].$$

Definition 6.2.9 (Factor refinement)**.** *Given two polynomial factors $\mathcal{B}, \mathcal{B}'$, we say $\mathcal{B}'$ is a refinement of $\mathcal{B}$, denoted by $\mathcal{B}' \preceq \mathcal{B}$, if every cell in the partition induced by $\mathcal{B}'$ is contained in some cell in the partition induced by $\mathcal{B}$.*

The definition of refinement immediately implies:

Lemma 6.2.10 (Pythagoras theorem)**.** *Let $\mathcal{B}, \mathcal{B}'$ be polynomial factors such that $\mathcal{B}' \preceq \mathcal{B}$, then for any function $f : \mathbb{K}^n \to \mathbb{C}$,*

$$\|\mathbb{E}[f|\mathcal{B}']\|_2^2 = \|\mathbb{E}[f|\mathcal{B}]\|_2^2 + \|\mathbb{E}[f|\mathcal{B}'] - \mathbb{E}[f|\mathcal{B}]\|_2^2 .$$

The next claim shows that any bounded function is "close" to being measurable by a polynomial factor of bounded complexity. Precisely:

Lemma 6.2.11 (Decomposition Theorem)**.** *Any bounded $f : \mathbb{K}^n \to \mathbb{C}$ can be approximated in $\| \cdot \|_{U^r}$ by a function of a small number of degree $< r$ non-classical polynomials i.e. for any $\varepsilon > 0$, there exists non-classical polynomials $P_1, P_2, \cdots, P_k$ of degree $< r$ with $P_i(\bar{0}) = 0$ $\forall i$ and a bounded function $\tau : \mathbb{T}^k \to \mathbb{C}$ such that*

$$\|f - \tau(P_1, P_2, \cdots, P_k)\|_{U^r} \leq \varepsilon$$

where $k = k(\varepsilon, \mathbb{K}, r)$ is a constant depending only on $\varepsilon, \mathbb{K}, r$.

Proof. The proof is similar to the proof of the Quadratic Koopman-von Neumann decompostion which is Prop 3.7 in [Gre06] but using the full Inverse Gowers Theorem (Lemma 6.2.4) and similar claims are implicit elsewhere, but for completeness, we give the proof.

The main idea is to approximate the function f using its conditional expectation over a suitable polynomial factor $\mathcal{B}$ of degree $< r$. We will start with the trivial factor $\mathcal{B}_0 = (1)$ and iteratively construct more refined partitions $\mathcal{B}_i \preceq \mathcal{B}_{i-1}$ until we find a factor $\mathcal{B}_k$ which satisfies $\|f - \mathbb{E}[f|\mathcal{B}_k]\|_{U^r} \leq \varepsilon$. To bound the number of iterations needed to achieve this, we will show that the energy $\|\mathbb{E}[f|\mathcal{B}_i]\|_2^2$ which is bounded above by 1, increases by a fixed constant in every step.

Suppose that after step $i - 1$, we still have $\|f - \mathbb{E}[f|\mathcal{B}_{i-1}]\|_{U^r} > \varepsilon$. Let $g = f - \mathbb{E}[f|\mathcal{B}_{i-1}]$, then by the inverse Gowers theorem (Lemma 6.2.4), we have some non-classical polynomial P_i of degree $< r$ such that $|\langle g, e(P_i)\rangle| \geq \kappa = c(\varepsilon, p, r)$. We can assume that $P_i(\bar{0}) = 0$. Refine the factor $\mathcal{B}_{i-1}$ by adding the polynomial P_i to obtain $\mathcal{B}_i \preceq \mathcal{B}_{i-1}$. Now consider the energy increment,

$$\|\mathbb{E}[f|\mathcal{B}_i]\|_2^2 - \|\mathbb{E}[f|\mathcal{B}_{i-1}]\|_2^2 = \|\mathbb{E}[f|\mathcal{B}_i] - \mathbb{E}[f|\mathcal{B}_{i-1}]\|_2^2 = \|\mathbb{E}[g|\mathcal{B}_i]\|_2^2$$

where we used the Pythagoras theorem(Lemma 6.2.10) and the fact that

$$\mathbb{E}\Big[\mathbb{E}[f|\mathcal{B}_{i-1}]\Big|\mathcal{B}_i\Big] = \mathbb{E}[f|\mathcal{B}_{i-1}]$$

since $\mathcal{B}_i \preceq \mathcal{B}_{i-1}$. So

$$\begin{aligned} \kappa^2 \leq |\mathbb{E}[g \cdot e(P_i)]|^2 &= \Big|\mathbb{E}\Big[\mathbb{E}[g \cdot e(P_i)|\mathcal{B}_i]\Big]\Big|^2 = \Big|\mathbb{E}\Big[e(P_i)\mathbb{E}[g|\mathcal{B}_i]\Big]\Big|^2 \\ &\leq \|\mathbb{E}[g|\mathcal{B}_i]\|_1^2 \leq \|\mathbb{E}[g|\mathcal{B}_i]\|_2^2 = \|\mathbb{E}[f|\mathcal{B}_i]\|_2^2 - \|\mathbb{E}[f|\mathcal{B}_{i-1}]\|_2^2 . \end{aligned}$$

Thus the energy increases by κ^2 every step. But since the energy is bounded above by 1, the process should end in a finite number of steps $k \leq \frac{1}{\kappa^2}$. So $\|f - \mathbb{E}[f|\mathcal{B}_k]\|_{U^r} \leq \varepsilon$, but since $\mathbb{E}[f|\mathcal{B}_k]$ is $\mathcal{B}_k$-measurable, we can write $\mathbb{E}[f|\mathcal{B}_k] = \tau(P_1, \cdots, P_k)$ for some function τ with $|\tau| = |\mathbb{E}[f|\mathcal{B}_k]| \leq |f| \leq 1$. □

We are now ready to prove Theorem 6.2.6.

Proof of Theorem 6.2.6. Recall that $\mathbb{K}$ is an extension field of dimension t over a prime field $\mathbb{F}_p$. The ε-net will be the set $\mathcal{N}$ of all functions of the form $\tau(P_1, \cdots, P_k)$ where $P_1, \cdots, P_k$ are degree $< r$ non-classical polynomials with zero constant terms, $\tau : \mathbb{T}^k \to \mathbb{C}$ is a bounded function and $k = k(\varepsilon, p, r)$ is the constant given by Lemma 6.2.11. But we will not include all possible bounded $\tau : \mathbb{T}^k \to \mathbb{C}$. Firstly by Equation 6.1, $P_1, \cdots, P_k$ take values only in $\frac{1}{p^r}\mathbb{Z}/\mathbb{Z}$. Next we will discretize the set $\{z \in \mathbb{C} : |z| \leq 1\}$ into the ε-lattice i.e. we will only consider maps $\tau : (\frac{1}{p^r}\mathbb{Z}/\mathbb{Z})^k \to \{z \in \mathbb{C} : |z| \leq 1\} \cap \varepsilon(\mathbb{Z} + i\mathbb{Z})$. The number of such maps is bounded by $(4/\varepsilon^2)^{p^{rk}}$.

By Equation 6.1, a non-classical polynomial of degree $< r$ in n variables with zero constant term can be represented by $\leq \binom{nt+r-1}{r-1} r$ coefficients in $\{0, 1, \cdots, p-1\}$. So the number of such non-classical polynomials is bounded by $\exp\left(O_{r,\mathbb{K}}(n^{r-1})\right)$.

Combining both the bounds,

$$|\mathcal{N}| \leq \exp\left(O_{r,\mathbb{K}}(n^{r-1})\right)^k \cdot (4/\varepsilon^2)^{p^{rk}} = \exp\left(O_{\varepsilon,\mathbb{K},r}(n^{r-1})\right).$$

We will now prove that $\mathcal{N}$ is a 3ε-net. Given any $f : \mathbb{K}^n \to [-1,1]$, using Lemma 6.2.11, there is a function $\tau(P_1, \cdots, P_k)$ such that

$$\|f - \tau(P_1, P_2, \cdots, P_k)\|_{U^r} \leq \varepsilon.$$

If we consider the $\tilde{\tau} \in \mathcal{N}$ by rounding values real and imaginary parts of τ to the nearest multiple of ε, we get

$$\begin{aligned}
&\|f - \tilde{\tau}(P_1, P_2, \cdots, P_k)\|_{U^r} \\
&\leq \|f - \tau(P_1, P_2, \cdots, P_k)\|_{U^r} + \|\tau(P_1, P_2, \cdots, P_k) - \tilde{\tau}(P_1, P_2, \cdots, P_k)\|_{U^r} \\
&\leq \varepsilon + \|\tau(P_1, P_2, \cdots, P_k) - \tilde{\tau}(P_1, P_2, \cdots, P_k)\|_{\infty} \leq 3\varepsilon.
\end{aligned}$$

□

6.3 Locally Correctable Codes

We will recall here the definition of a locally correctable code from Section 2.4. An (r, δ, η)-LCC is a code $\mathcal{C} \subset \Sigma^{\mathcal{X}}$ with the following property:

For each $x \in \mathcal{X}$ there is a distribution $\mathcal{M}_x$ over r-tuples of distinct[6] coordinates such

[6] Without loss of generality, we can assume the tuples have distinct coordinates by adding dummy coordinates and modifying the decoding functions $\mathcal{D}_{x,y_1,\cdots,y_r}$

that whenever $\tilde{f} \in \Sigma^{\mathcal{X}}$ is δ-close to some codeword $f \in \mathcal{C}$ in Hamming distance,

$$\Pr_{(y_1,\cdots,y_r)\sim\mathcal{M}_x}[\mathcal{D}_{x,y_1,\cdots,y_r}(\tilde{f}(y_1), \tilde{f}(y_2), \cdots, \tilde{f}(y_r)) = f(x)]$$
$$\geq \Pr_{(y_1,\cdots,y_r)\sim\mathcal{M}_x}[\mathcal{D}_{x,y_1,\cdots,y_r}(\tilde{f}(y_1), \tilde{f}(y_2), \cdots, \tilde{f}(y_r)) = \sigma] + \eta$$

for every $\sigma \in \Sigma$ such that $\sigma \neq f(x)$ where $\mathcal{D}_{x,y_1,\cdots,y_r} : \Sigma^r \to \Sigma$, called the decoding operator, depends only on $x, y_1, \cdots, y_r$.

If furthermore $\mathcal{X}$ is a vector space and $\mathcal{C}$ is affine invariant then we call it an affine invariant LCC.

Remark 6.3.1. *Let $|\Sigma| = m$, Without loss of generality, we can assume that $\Sigma = \{1, 2, \cdots, m\}$. Then we can extend functions $f : \mathcal{X} \to \Sigma$ to $\hat{f} : \mathcal{X} \to \blacktriangle_m$. The decoding operators $\mathcal{D} : \Sigma^r \to \Sigma$ can also be extended to $\widehat{\mathcal{D}} : \blacktriangle_m^r \to \blacktriangle_m$ as follows: For $z_1, \cdots, z_r \in \blacktriangle_m$ define*

$$\widehat{\mathcal{D}}(z_1, \cdots, z_r) = \sum_{1 \leq i_1, \cdots, i_r \leq m} e_{\mathcal{D}(i_1,\cdots,i_r)}(z_1)_{i_1} \cdots (z_r)_{i_r}$$

where e_j stands for the j^{th} coordinate vector in $\mathbb{R}^m$ and $(z_j)_i$ is the i^{th} coordinate of the vector z_j. Now we the decoding condition implies that:

$$\mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}_x}\left[\left\langle \hat{f}(x), \widehat{\mathcal{D}}_{x,y_1,\cdots,y_r}(\hat{f}(y_1), \hat{f}(y_2), \cdots, \hat{f}(y_r))\right\rangle\right] \geq \eta.$$

Now, we are ready to prove our main result of this section.

Theorem 6.3.2 (Lower bound for LCCs)**.** *Let $\mathcal{C} \subset \Sigma^{\mathbb{K}^n}$ be an (r, δ, η) affine-invariant LCC where $\eta > 1 - \frac{2\delta}{3}$. Then $|\mathcal{C}| \leq \exp\left(O_{\delta,\mathbb{K},r,|\Sigma|}(n^{r-1})\right)$.*

Proof. Let $|\Sigma| = m$. Let $\mathcal{N}$ be an $\varepsilon/2$-net for the space of all bounded functions $\{h : \mathbb{K}^n \to \mathbb{C}\}$ with the metric induced by $\|\cdot\|_{U^r}$-norm where $\varepsilon = \frac{2\delta}{3m^r}$. Given a bounded $h : \mathbb{K}^n \to \mathbb{C}$, define

$$\phi(h) := \mathrm{argmin}_{h' \in \mathcal{N}} \|h - h'\|_{U^r}$$

(break ties arbitrarily). Since $\mathcal{N}$ is an $\varepsilon/2$ net, we have $\|h - \phi(h)\|_{U^r} \leq \varepsilon/2$. Define $\Psi : \mathcal{C} \to \mathcal{N}^m$ as

$$\Psi(f) := (\phi(\hat{f}_1), \cdots, \phi(\hat{f}_m))$$

where $\hat{f}_i : \mathbb{K}^n \to \mathbb{R}_{\geq 0}$ is the i^{th} coordinate function of the simplex extension $\hat{f} : \mathbb{K}^n \to \blacktriangle_m$ of $f : \mathbb{K}^n \to \Sigma$. We claim that Ψ is an injection which implies that $|\mathcal{C}| \leq |\mathcal{N}|^m$. Now using Theorem 6.2.6, the required bound follows. Suppose that Ψ is not an injection. Let $f, g \in \mathcal{C}$ be two distinct codewords such that $\Psi(f) = \Psi(g)$. This implies that

$$\forall\ i \in [m]\ \|\hat{f}_i - \hat{g}_i\|_{U^r} \leq \|\hat{f}_i - \phi(\hat{f}_i)\|_{U^r} + \|\hat{g}_i - \phi(\hat{g}_i)\|_{U^r} \leq \varepsilon.$$

By affine invariance of $\mathcal{C}$, $f \circ \ell \in \mathcal{C}$ for all invertible affine maps $\ell : \mathbb{K}^n \to \mathbb{K}^n$. So by the local correction property,

$$\Pr_{\ell, y_0, (y_1, \cdots, y_r) \sim \mathcal{M}_{y_0}} [f \circ \ell(y_0) = \mathcal{D}_{y_0, y_1, \cdots, y_r}(f \circ \ell(y_1), \cdots, f \circ \ell(y_r))] \geq \eta$$

where ℓ ranges uniformly over all invertible affine maps from $\mathbb{K}^n \to \mathbb{K}^n$ and y_0 ranges uniformly over $\mathbb{K}^n$. Now consider the following difference:

$$\begin{aligned}
&\Pr_{\ell, y_0, (y_1, \cdots, y_r) \sim \mathcal{M}_{y_0}} [f \circ \ell(y_0) = \mathcal{D}_{y_0, y_1, \cdots, y_r}(f \circ \ell(y_1), \cdots, f \circ \ell(y_r))] \\
&\qquad - \Pr_{\ell, y_0, (y_1, \cdots, y_r) \sim \mathcal{M}_{y_0}} [g \circ \ell(y_0) = \mathcal{D}_{y_1, \cdots, y_r}(f \circ \ell(y_1), \cdots, f \circ \ell(y_r))] \\
&= \mathbb{E}_\ell \mathbb{E}_{y_0} \mathbb{E}_{(y_1, \cdots, y_r) \sim \mathcal{M}_{y_0}} \Big[\Big\langle \hat{f} \circ \ell(y_0), \widehat{\mathcal{D}}_{y_0, y_1, \cdots, y_r}(\hat{f} \circ \ell(y_1), \cdots, \hat{f} \circ \ell(y_r)) \Big\rangle \\
&\qquad - \Big\langle \hat{g} \circ \ell(y_0), \widehat{\mathcal{D}}_{y_1, \cdots, y_r}(\hat{f} \circ \ell(y_1), \cdots, \hat{f} \circ \ell(y_r)) \Big\rangle \Big] \\
&= \mathbb{E}_{y_0} \mathbb{E}_{(y_1, \cdots, y_r) \sim \mathcal{M}_{y_0}} \Big[\mathbb{E}_\ell \Big[\Big\langle \hat{f} \circ \ell(y_0) - \hat{g} \circ \ell(y_0), \widehat{\mathcal{D}}_{y_0, y_1, \cdots, y_r}(\hat{f} \circ \ell(y_1), \cdots, \hat{f} \circ \ell(y_r)) \Big\rangle \Big] \Big]
\end{aligned}$$

Now we fix $y_0, y_1, \cdots, y_r$ and show that inner expectation is small for each tuple $(y_0, y_1, \cdots, y_r)$. Let us denote $\mathcal{D} = \mathcal{D}_{y_0, y_1, \cdots, y_r}$ for brevity. Let $t = \text{rank}(y_0, y_1, \cdots, y_r)$[7], thus there exist independent vectors $v_1, \cdots, v_t \in \mathbb{K}^n$ such that for every $0 \le i \le r$, $y_i = \sum_{j=1}^t \lambda_{ij} v_j$ for some fixed $\lambda_{ij} \in \mathbb{K}$. The action of a random invertible affine map ℓ can be approximated by sampling $z_0, z_1, \cdots, z_t \in \mathbb{K}^n$ uniformly and mapping $y_i \mapsto z_0 + \sum_{j=1}^t \lambda_{ij} z_j$ since with probability $1 - o_n(1)$,

[7] $\text{rank}(y_0, y_1, \cdots, y_r)$ is the dimension of the subspace spanned by the vectors $y_0, y_1, \cdots, y_r$.

$z_1, \cdots, z_t$ will be independent. Therefore,

$$\begin{aligned}
&\mathbb{E}_{\ell}\left[\left\langle \hat{f}\circ\ell(y_0) - \hat{g}\circ\ell(y_0), \widehat{\mathcal{D}}_{y_0,y_1,\cdots,y_r}(\hat{f}\circ\ell(y_1),\cdots,\hat{f}\circ\ell(y_r))\right\rangle\right] \\
&= o_n(1) + \mathbb{E}_{z_0,z_1,\cdots,z_t\in\mathbb{K}^n}\Bigg[\Bigg\langle (\hat{f}-\hat{g})(z_0+\sum_{j=1}^{t}\lambda_{0j}z_j), \\
&\qquad\qquad \widehat{\mathcal{D}}\left(\hat{f}(z_0+\sum_{j=1}^{t}\lambda_{1j}z_j),\cdots,\hat{f}(z_0+\sum_{j=1}^{t}\lambda_{rj}z_j)\right)\Bigg\rangle\Bigg] \\
&\qquad\qquad\qquad\qquad \text{(we can ignore the } o_n(1) \text{ term)} \\
&= \mathbb{E}_{z_0,z_1,\cdots,z_t\in\mathbb{K}^n}\Bigg[\Bigg\langle (\hat{f}-\hat{g})(z_0+\sum_{j=1}^{t}\lambda_{0j}z_j), \\
&\qquad\qquad \left(\sum_{1\le i_1,\cdots,i_r\le m} e_{\mathcal{D}(i_1,\cdots,i_r)}\prod_{k=1}^{r}\hat{f}_{i_k}(z_0+\sum_{j=1}^{t}\lambda_{kj}z_j)\right)\Bigg\rangle\Bigg] \\
&= \mathbb{E}_{z_0,z_1,\cdots,z_t\in\mathbb{K}^n}\Bigg[\Bigg(\sum_{1\le i_1,\cdots,i_r\le m} \\
&\qquad\qquad (\hat{f}-\hat{g})_{\mathcal{D}(i_1,\cdots,i_r)}(z_0+\sum_{j=1}^{t}\lambda_{0j}z_j)\cdot\prod_{k=1}^{r}\hat{f}_{i_k}(z_0+\sum_{j=1}^{t}\lambda_{kj}z_j)\Bigg)\Bigg] \\
&\le \left(\sum_{0\le i_1,\cdots,i_r\le m-1}\|(\hat{f}-\hat{g})_{\mathcal{D}(i_1,\cdots,i_r)}\|_{U^r}\right) \le m^r\varepsilon
\end{aligned}$$

where the first inequality is obtained by applying generalized von Neumann inequality (Lemma 6.2.5) to each term. Therefore

$$\begin{aligned}
&\Pr_{\ell,y_0,(y_1,\cdots,y_r)\sim\mathcal{M}_{y_0}}[g\circ\ell(y_0) = \mathcal{D}_{y_1,\cdots,y_r}(f\circ\ell(y_1),\cdots,f\circ\ell(y_r))] \\
&\ge \Pr_{\ell,y_0,(y_1,\cdots,y_r)\sim\mathcal{M}_{y_0}}[f\circ\ell(y_0) = \mathcal{D}_{y_1,\cdots,y_r}(f\circ\ell(y_1),\cdots,f\circ\ell(y_r))] - m^r\varepsilon \\
&\ge \eta - 2\delta/3.
\end{aligned}$$

On the other hand,

$$\begin{aligned}
&\Pr_{\ell,y_0,(y_1,\cdots,y_r)\sim\mathcal{M}_{y_0}}[g\circ\ell(y_0)=\mathcal{D}_{y_1,\cdots,y_r}(f\circ\ell(y_1),\cdots,f\circ\ell(y_r))]\\
&\leq \Pr_{\ell,y_0,(y_1,\cdots,y_r)\sim\mathcal{M}_{y_0}}[g\circ\ell(y_0)=f\circ\ell(y_0)]\\
&\qquad\qquad+\Pr_{\ell,y_0,(y_1,\cdots,y_r)\sim\mathcal{M}_{y_0}}[f\circ\ell(y_0)\neq\mathcal{D}_{y_1,\cdots,y_r}(f\circ\ell(y_1),\cdots,f\circ\ell(y_r))]\\
&\leq \Pr_x[f(x)=g(x)]+1-\eta\\
&\leq 1-2\delta+1-\eta && \text{(By Lemma 2.4.2)}
\end{aligned}$$

This is a contradiction when $\eta > 1-\frac{2\delta}{3}$.

□

6.4 Locally Testable Codes

We start by defining a weaker definition of locally testable codes than the one given in Section 2.5.

Definition 6.4.1 ((weak) Locally Testable Code (LTC)). *An (r,δ,τ)-weak LTC is a code $\mathcal{C}\subset\Sigma^{\mathcal{X}}$ with minimum distance at least δ and the following property:*
There is a distribution $\mathcal{M}$ over r-tuples of distinct[8] *coordinates such that for each codeword $f\in C$,*

$$\Pr_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(f(y_1),f(y_2),\cdots,f(y_r))=1]\geq 3/4$$

and for every $g\in\Sigma^{\mathcal{X}}$ which is τ-far away from every codeword,

$$\Pr_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(g(y_1),g(y_2),\cdots,g(y_r))=1]\leq 1/4$$

[8] Again, without loss of generality, we can assume the tuples have distinct coordinates by adding dummy coordinates and modifying the decoding functions $\mathcal{D}_{y_1,\cdots,y_r}$

where $\mathcal{D}_{y_1,\cdots,y_r} : \Sigma^r \to \{0,1\}$, called the testing operator, depends only on $y_1, \cdots, y_r$. If furthermore $\mathcal{X}$ is a vector space and $\mathcal{C}$ is affine-invariant then we call it an affine invariant weak LTC.

Note that a (r, δ, ρ) (strong) LTC as defined in Section 2.5 is also a $(r, \delta, \frac{3}{4\rho})$-weak LTC. Therefore our lower bounds also apply to (strong) LTCs with appropriate parameters.

Remark 6.4.2. *Let $|\Sigma| = m$, Without loss of generality, we can assume that $\Sigma = \{1, 2, \cdots, m\}$. We can extend $f : \mathcal{X} \to \Sigma$ to $\hat{f} : \mathcal{X} \to \blacktriangle_m$. The testing operator $\mathcal{D} : \Sigma^r \to \{0,1\}$ can also be extended to $\widehat{\mathcal{D}} : \blacktriangle_m^r \to [0,1]$ as follows: For $z_1, \cdots, z_r \in \blacktriangle_m$ define*

$$\widehat{\mathcal{D}}(z_1, \cdots, z_r) = \sum_{1 \le i_1, \cdots, i_r \le m} \mathcal{D}(i_1, \cdots, i_r)(z_1)_{i_1} \cdots (z_r)_{i_r}. \tag{6.2}$$

Now we can rewrite the probability in terms of expectation as:

$$\begin{aligned} &\Pr_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(f(y_1), \cdots, f(y_r)) = 1] \\ &= \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{f} \circ \ell(y_1), \cdots, \hat{f} \circ \ell(y_r))]. \end{aligned}$$

We are now ready to prove the main result of this section.

Theorem 6.4.3 (Lower bound for LTC's)**.** *Let $\mathcal{C} \subset \Sigma^{\mathbb{K}^n}$ be an $(r, \delta, \delta/3)$ affine invariant weak LTC, then $|\mathcal{C}| \le \exp\left(O_{\delta,\mathbb{K},r,|\Sigma|}(n^{r-2})\right)$.*

Proof. Let $|\Sigma| = m$. The proof is very similar to that of Theorem 6.3.2. Let $\mathcal{N}$ be an $\varepsilon/2$-net for the space of all bounded functions $\{f : \mathbb{K}^n \to \mathbb{C}\}$ with the metric induced by $\|\cdot\|_{U^{r-1}}$-norm where $\varepsilon = 1/2rm^r$. Define $\Psi : \mathcal{C} \to \mathcal{N}^m$ as in the proof of Theorem 6.3.2, it is enough to show that Ψ is an injection. Suppose that Ψ is not an injection, then there exists $f, g \in \mathcal{C}$ which are distinct such that $\Psi(f) = \Psi(g)$. This implies that

$$\forall\ i \in [m]\ \|\hat{f}_i - \hat{g}_i\|_{U^{r-1}} \le \varepsilon.$$

By affine invariance of $\mathcal{C}$, $f \circ \ell \in \mathcal{C}$ for all invertible affine maps $\ell : \mathbb{K}^n \to \mathbb{K}^n$. So

$$\mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(f\circ\ell(y_1), f\circ\ell(y_2),\cdots, f\circ\ell(y_r))] \geq 3/4$$

where ℓ ranges over all invertible affine maps from $\mathbb{K}^n \to \mathbb{K}^n$. Let $H \in \Sigma^{\mathcal{X}}$ be a random word where for each coordinate $x \in \mathcal{X}$ independently,

$$H(x) = \begin{cases} f(x) & \text{with probability } 1/2 \\ g(x) & \text{with probability } 1/2 \end{cases}.$$

Define $\hat{h} : \mathcal{X} \to \blacktriangle_m$ as $\hat{h}(x) = \mathbb{E}_H[\widehat{H}(x)] = \frac{\hat{f}(x)+\hat{g}(x)}{2}$ where $\hat{f}, \hat{g}$ are the simplex extensions of the original f, g. So $\forall\ i \in [m]$ $\|\hat{f}_i - \hat{h}_i\|_{U^{r-1}} = \|\hat{f}_i - \hat{g}_i\|_{U^{r-1}}/2 \leq \varepsilon/2$. We will now show that the test accepts $H \circ \ell$ with good probability when ℓ is a random invertible affine map from $\mathbb{K}^n \to \mathbb{K}^n$.

$$\begin{aligned}
&\mathbb{E}_H \mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(f\circ\ell(y_1),\cdots,f\circ\ell(y_r)) \\
&\qquad\qquad\qquad\qquad - \mathcal{D}_{y_1,\cdots,y_r}(H\circ\ell(y_1),\cdots,H\circ\ell(y_r))] \\
&= \mathbb{E}_H \mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{f}\circ\ell(y_1),\cdots,\hat{f}\circ\ell(y_r)) \\
&\qquad\qquad\qquad\qquad - \widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\widehat{H}\circ\ell(y_1),\cdots,\widehat{H}\circ\ell(y_r))] \\
&= \mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{f}\circ\ell(y_1),\cdots,\hat{f}\circ\ell(y_r)) \\
&\qquad\qquad\qquad\qquad - \widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{h}\circ\ell(y_1),\cdots,\hat{h}\circ\ell(y_r))]
\end{aligned}$$

(using multilinear expansion of $\widehat{\mathcal{D}}_{y_1,\cdots,y_r}$(Equation 6.2) and taking expectation over H)

$$\begin{aligned}
&= \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}\Big[\mathbb{E}_\ell\Big[\widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{f}\circ\ell(y_1),\cdots,\hat{f}\circ\ell(y_r)) \\
&\qquad\qquad\qquad\qquad -\widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{h}\circ\ell(y_1),\cdots,\hat{h}\circ\ell(y_r))\Big]\Big]
\end{aligned}$$

Now we fix $y_1,\cdots,y_r$ and show that inner expectation is small for each tuple $(y_1,\cdots,y_r)$. Let us denote $\mathcal{D} = \mathcal{D}_{y_1,\cdots,y_r}$ for brevity. Let $t = \text{rank}(y_1,\cdots,y_r)$, thus there exist independent vectors $v_1, \cdots, v_t \in \mathbb{K}^n$ such that for every $1 \leq i \leq r$, $y_i = \sum_{j=1}^{t} \lambda_{ij} v_j$ for some fixed $\lambda_{ij} \in \mathbb{K}$. The action of a random invertible affine map ℓ can be approximated by sampling $z_0, z_1, \cdots, z_t \in \mathbb{K}^n$ uniformly and mapping $y_i \mapsto z_0 + \sum_{j=1}^{t} \lambda_{ij} z_j$ since with probability $1 - o_n(1)$, $z_1, \cdots, z_t$ will be independent. Therefore,

$$
\begin{aligned}
&\mathbb{E}_\ell \left[\widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{f} \circ \ell(y_1), \cdots, \hat{f} \circ \ell(y_r)) - \widehat{\mathcal{D}}_{y_1,\cdots,y_r}(\hat{h} \circ \ell(y_1), \cdots, \hat{h} \circ \ell(y_r)) \right] \\
&= o_n(1) + \mathbb{E}_{z_0,\cdots,z_t \in \mathbb{K}^n} \Bigg[\widehat{\mathcal{D}}(\hat{f}(z_0 + \sum_{j=1}^{t} \lambda_{1j} z_j), \cdots, \hat{f}(z_0 + \sum_{j=1}^{t} \lambda_{rj} z_j)) \\
&\qquad\qquad - \mathcal{D}(\hat{h}(z_0 + \sum_{j=1}^{t} \lambda_{1j} z_j), \cdots, \hat{h}(z_0 + \sum_{j=1}^{t} \lambda_{rj} z_j)) \Bigg] \\
&= \mathbb{E}_{z_0,z_1,\cdots,z_t \in \mathbb{K}^n} \Bigg[\sum_{1 \leq i_1,\cdots,i_r \leq m} \\
&\qquad \mathcal{D}(i_1, \cdots, i_r) \left(\prod_{k=1}^{r} \hat{f}_{i_k}(z_0 + \sum_{j=1}^{t} \lambda_{kj} z_j) - \prod_{k=1}^{r} \hat{h}_{i_k}(z_0 + \sum_{j=1}^{t} \lambda_{kj} z_j) \right) \Bigg] \\
&\leq r \cdot m^r \cdot \frac{\varepsilon}{2} = \frac{1}{4}
\end{aligned}
$$

where the last line is obtained by forming hybrids i.e. writing

$$
\begin{aligned}
&\hat{f}_{i_1} \cdot \hat{f}_{i_2} \cdots \hat{f}_{i_r} - \hat{h}_{i_1} \cdot \hat{h}_{i_2} \cdots \hat{h}_{i_r} \\
&= (\hat{f}_{i_1} - \hat{h}_{i_1}) \cdot \hat{f}_{i_2} \cdots \hat{f}_{i_r} + \hat{h}_{i_1} \cdot (\hat{f}_{i_2} - \hat{h}_{i_2}) \cdots \hat{f}_{i_r} + \cdots + \hat{h}_{i_1} \cdot \hat{h}_{i_2} \cdots (\hat{f}_{i_r} - \hat{h}_{i_r})
\end{aligned}
$$

and using Lemma 6.2.5 for each term. Therefore

$$
\begin{aligned}
&\mathbb{E}_H \mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}} [\mathcal{D}_{y_1,\cdots,y_r}(H \circ \ell(y_1), \cdots, H \circ \ell(y_r))] \\
&\geq \mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}} [\mathcal{D}_{y_1,\cdots,y_r}(f \circ \ell(y_1), \cdots, f \circ \ell(y_r))] - \frac{1}{4} \geq \frac{3}{4} - \frac{1}{4} = \frac{1}{2}.
\end{aligned}
$$

By Markov inequality,

$$\begin{aligned}\frac{1}{4} &\leq \Pr_H\left[\mathbb{E}_\ell \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(H\circ\ell(y_1),\cdots,H\circ\ell(y_r))] \geq \frac{1}{3}\right] \\ &\leq \Pr_H\left[\exists\ell\ \ \mathbb{E}_{(y_1,\cdots,y_r)\sim\mathcal{M}}[\mathcal{D}_{y_1,\cdots,y_r}(H\circ\ell(y_1),\cdots,H\circ\ell(y_r))] \geq \frac{1}{3}\right] \\ &\leq \Pr_H\left[\exists\ell\ \mathrm{dist}_H(H\circ\ell,\mathcal{C})] \leq \frac{\delta}{3}\right] \qquad\qquad \text{(by the soundness of the tester)} \\ &= \Pr_H\left[\mathrm{dist}_H(H,\mathcal{C})] \leq \frac{\delta}{3}\right] \qquad \text{(since ℓ is invertible and $\mathcal{C}$ is affine invariant)}\end{aligned}$$

Let $\mathcal{H} = \mathrm{supp}(H)$ be the set of words between f and g i.e. the set of all words $e \in \Sigma^{\mathbb{K}^n}$ such that $e(x) = f(x)$ or $e(x) = g(x)$ for all $x \in \mathbb{K}^n$. Let $\Delta = \mathrm{dist}_H(f,g)$. We have $|\mathcal{H}| = 2^{\Delta n}$. Since the distribution of H is uniform in $\mathcal{H}$, we proved that at least $\frac{1}{4}$ fraction of words in $\mathcal{H}$ contain a codeword in their $\delta/3$ neighborhood, let $\mathcal{H}' \subset \mathcal{H}$ denote this subset. Therefore the $\delta/6$ neighborhoods around the points in $\mathcal{H}'$ must be disjoint or else two distinct codewords will be $< \delta$ close to each other. The number of words in $\mathcal{H}$ which lie in a Hamming ball of radius $\delta/6$ around a point of $\mathcal{H}'$ is

$$\sum_{i=0}^{\delta n/6}\binom{\Delta n}{i} \geq 2^{H(\delta/6\Delta)\Delta n - o(n)} \geq 2^{H(\delta/6)\Delta n - o(n)}$$

where $H(\cdot)$ is the binary entropy function. By a packing argument, we can upper bound the size of $\mathcal{H}'$ as

$$|\mathcal{H}'| \leq \frac{2^{\Delta n}}{2^{H(\delta/6)\Delta n - o(n)}} = o(|\mathcal{H}|).$$

This contradicts the fact that $|\mathcal{H}'| \geq |\mathcal{H}|/4$.

□

6.5 Proof of generalized von Neumann inequality (Lemma 6.2.5)

Since the lemma is not stated in the form we want in [Tao12], we will include a proof here for completeness. To prove Lemma 6.2.5, we need the following lemma first.

Lemma 6.5.1 (Exercise 1.3.22 in [Tao12])**.** *Let $f : \mathbb{K}^n \to \mathbb{C}$ be a function, and for each $1 \leq i \leq k$, let $g_i : (\mathbb{K}^n)^k \to \mathbb{C}$ be a bounded function which is independent of the i^{th} coordinate of $(\mathbb{K}^n)^k$. Then,*

$$|\mathbb{E}_{x_1,\cdots,x_k\in\mathbb{K}^n}[f(x_1+x_2+\cdots+x_k)\prod_{i=1}^{k} g_i(x_1,\cdots,x_k)]| \leq \|f\|_{U^k}$$

Proof. The proof is by induction on k and using Cauchy-Schwarz inequality repeatedly. The case $k = 1$ is true by definition of $\|\cdot\|_{U^1}$.

$$\begin{aligned}
&\left|\mathbb{E}_{x_1,\cdots,x_k\in\mathbb{K}^n}\left[f(x_1+x_2+\cdots+x_k)\prod_{i=1}^{k} g_i(x_1,\cdots,x_k)\right]\right| \\
&= \left|\mathbb{E}_{x_2,\cdots,x_k}\left[g_1(x_1,\cdots,x_k)\mathbb{E}_{x_1}\left[f(x_1+x_2+\cdots+x_k)\prod_{i=2}^{k} g_i(x_1,\cdots,x_k)\right]\right]\right| \\
&\hfill \text{(since } g_1 \text{ doesn't depend on } x_1\text{)} \\
&\le \left|\mathbb{E}_{x_2,\cdots,x_k}\left[\mathbb{E}_{x_1'}\left[f(x_1'+x_2+\cdots+x_k)\prod_{i=2}^{k} g_i(x_1',x_2,\cdots,x_k)\right]\right.\right. \\
&\qquad\qquad \left.\left.\cdot\mathbb{E}_{x_1}\left[\bar{f}(x_1+x_2+\cdots+x_k)\prod_{i=2}^{k} \bar{g}_i(x_1,x_2,\cdots,x_k)\right]\right]\right|^{1/2} \\
&\hfill \text{(By Cauchy-Schwarz inequality and the fact that } |g_1|\le 1\text{)} \\
&= |\mathbb{E}_{x_1,h_1}\left[\mathbb{E}_{x_2,\cdots,x_k}\left[\Delta_{h_1} f(x_1+x_2+\cdots+x_k)\right.\right. \\
&\qquad\qquad \left.\left.\left.\cdot\prod_{i=2}^{k} g_i(x_1+h_1,x_2,\cdots,x_k)\bar{g}_i(x_1,x_2,\cdots,x_k)\right]\right]\right|^{1/2} \\
&\hfill \text{(By substituting } x_1' = x_1+h_1\text{)} \\
&\le \left|\mathbb{E}_{x_1,h_1}\left[\mathbb{E}_{h_2,\cdots,h_k,z}\left[\Delta_{h_k}\cdots\Delta_{h_1} f(x_1+z)\right]^{1/2^{k-1}}\right]\right|^{1/2} \\
&\hfill \text{(By induction hypothesis and the definition of Gowers norm)} \\
&\le |\mathbb{E}_{x_1,h_1,h_2,\cdots,h_k,z}\left[\Delta_{h_k}\cdots\Delta_{h_1} f(x_1+z)\right]|^{1/2^k} \qquad \text{(By Jensen's inequality)} \\
&= |\mathbb{E}_{h_1,h_2,\cdots,h_k,z}\left[\Delta_{h_k}\cdots\Delta_{h_1} f(z)\right]|^{1/2^k} = \|f\|_{U^k}
\end{aligned}$$

□

Proof of Lemma 6.2.5. By symmetry, it is enough to show that

$$|\mathbb{E}_{z_1,\cdots,z_m\in\mathbb{K}^n}[f_0(\mathcal{L}_0(z_1,\cdots,z_m))\prod_{i=1}^{k} f_i(\mathcal{L}_i(z_1,\cdots,z_m))]| \le \|f_0\|_{U^k}.$$

We will make a linear change of variables so that we can use Lemma 6.5.1 to get the required bound. For each $1 \le i \le k$, since $\mathcal{L}_0$ is not a multiple of $\mathcal{L}_i$, there exists a vector $v_i \in \mathbb{K}^m$ such that $\mathcal{L}_0(v_i) = 1$ and $\mathcal{L}_i(v_i) = 0$. Now we make the following change of variables: $(z_1, \cdots, z_m) \to (x_1, \cdots, x_m) + \sum_{i=1}^{k} y_i v_i^T$ where $x_1, \cdots, x_m$ and $y_1, \cdots, y_k$ are the new variables which range over $\mathbb{K}^n$.

$$
\begin{aligned}
&|\mathbb{E}_{z_1,\cdots,z_m \in \mathbb{K}^n}[f_0(\mathcal{L}_0(z_1, \cdots, z_m)) \prod_{i=1}^{k} f_i(\mathcal{L}_i(z_1, \cdots, z_m))]| \\
&= \left| \mathbb{E}_{x_1,\cdots,x_m,y_1,\cdots,y_k \in \mathbb{K}^n} \left[f_0 \left(\mathcal{L}_0(x_1, \cdots, x_m) + \sum_{j \in [k]} y_j \right) \right.\right. \\
&\qquad\qquad \left.\left. \prod_{i \in [k]} f_i \left(\mathcal{L}_i(x_1, \cdots, x_m) + \sum_{j \in [k] \setminus \{i\}} y_j \mathcal{L}_i(v_j) \right) \right] \right| \\
&\qquad\qquad \text{(By change of variables and linearity of } \mathcal{L}_i\text{)} \\
&\le \mathbb{E}_{x_1,\cdots,x_m \in \mathbb{K}^n} \left[\left| \mathbb{E}_{y_1,\cdots,y_k \in \mathbb{K}^n} \left[f_0 \left(\mathcal{L}_0(x_1, \cdots, x_m) + \sum_{j \in [k]} y_j \right) \right.\right.\right. \\
&\qquad\qquad \left.\left.\left. \prod_{i \in [k]} f_i \left(\mathcal{L}_i(x_1, \cdots, x_m) + \sum_{j \in [k] \setminus \{i\}} y_j \mathcal{L}_i(v_j) \right) \right] \right| \right] \\
&\le \|f_0\|_{U^k} \qquad\qquad \text{(By Lemma 6.5.1)}
\end{aligned}
$$

□

6.6 Conclusions

In this work, we proved tight lower bounds for constant query affine-invariant LCCs and LTCs when the number of queries r, underlying field $\mathbb{K}$ and the alphabet Σ are constant. However the constants in the bounds we obtain are of Ackermann-type in $r, |\mathbb{K}|, |\Sigma|$ because of the use of higher-order Fourier analysis. Improving the dependence on these parameters is an open problem which might require new ideas. In a recent work, Bhowmick and Lovett [BL15] obtain a "bias implies low rank" theorem for polynomials over growing fields. This might be a first step towards proving a variant of the inverse Gowers theorem (Lemma 6.2.4) for growing field size, which could then be used to make our lower bounds extend to the case of growing field size.

We also remark that our lower bounds work for any LCC or LTC where the queries are obtained as fixed linear combinations of uniformly chosen points from $\mathbb{K}^n$. Affine-invariant codes are a natural class of local codes where this is true. Relaxing these conditions to get lower bounds for a more general class of LCCs or LTCs is an open problem.

Chapter 7

Lower bounds for 2-query LCCs

7.1 Introduction

One particularly important feature of LDCs is their tight connection to *information-theoretic private information retrieval (PIR)* schemes as discussed in Section 3.2. 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. Conversely, a 2-query LDC $C : \{0,1\}^k \to \Sigma^n$ implies a 2-server PIR with communication cost $O(\log n + \log |\Sigma|)$. Since a 2-query LCC can be converted into a 2-query LDC with similar parameters as shown in Section 2.4.1, one can obtain 2-server PIR schemes from good LCCs as well.

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, $n \geq \Omega(\delta(k/\log|\Sigma|)^2)$. (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 $n \geq \exp(\delta k/|\Sigma|)$ for linear

2-query LDCs, while Kerenedis and de Wolf [KW04] (with further improvements in [WDW05b]) showed using quantum techniques that $n \geq \exp(\delta k/|\Sigma|^2)$ 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 as defined in Section 2.4 for which the corrector 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 will define them formally before stating our result.

Definition 7.1.1. *Let Σ be some finite alphabet. For positive integer n and parameters $\eta, \delta > 0$, a subset $C \subset \Sigma^n$ is a $(2, \delta, \eta)$-zero-error LCC if, for every $i \in [n]$, there exists a randomized* corrector *(a probabilistic algorithm) $\mathcal{A}_i$ such that:*

1. *For every codeword $c \in C$ and $z \in \Sigma^n$ such that $\mathrm{dist}_H(c, z) \leq \delta$,*

$$\Pr[\mathcal{A}_i(z) = c_i] \geq \Pr[\mathcal{A}_i(z) = \sigma] + \eta, \tag{7.1}$$

   for any $\sigma \in \Sigma$ such that $\sigma \neq x_i$.

2. *The decoder $\mathcal{A}_i(z)$ queries non-adaptively at most 2 coordinates of z.*

3. *If $c \in \mathcal{C}$, then for every $i \in [n]$, $\Pr[\mathcal{A}_i(c) = c_i] = 1$ 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 from Section 2.4 only in part (3) above. Whenever η is not mentioned, we assume that it is some fixed absolute constant. We now state our main lower bound for zero-error 2-query LCCs.

Theorem 7.1.2. *Let $\mathcal{C} \subset \Sigma^n$ be a $(2, \delta, \eta)$-LCC which is zero-error, then*

$$|\mathcal{C}| \leq \exp\left(O\left(\left(\tfrac{1}{\delta^4} + \tfrac{\log(1/\eta)}{\eta^2\delta^2}\right) \cdot \log n \cdot \log|\Sigma|\right)\right).$$

7.1.1 Discussion of Main Result

The lower bound in Theorem 7.1.2 is tight in its dependence on k and Σ. Specifically, Yekhanin in the appendix of [BDSS16] 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 $\delta \leq 1/6, \Sigma$ 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 7.1.2 shows $n \geq \exp(\delta^4 k/\log|\Sigma|)$ which is therefore tight upto poly(δ) 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, [BDSS16] showed that $n \geq \exp(\delta k/r) = \exp(\delta k/\log_p|\Sigma|)$ 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 [BDSS16] from linear codes to all zero-error LCCs. Moreover, this work is much more elementary and simple than [BDSS16] which uses non-trivial results from additive combinatorics.

It is important to note that Theorem 7.1.2 cannot be true for 2-query LDCs. Such a result would contradict the construction in Theorem 3.2.1 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 over large alphabet. 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 $n \geq \Omega(\delta k^2/\log^2|\Sigma|)$. And the quantum arguments of [KW04, WDW05b] give the lower bound $n \geq \exp(\delta k/|\Sigma|^2)$ which becomes trivial when $|\Sigma| = \Omega(n)$.

7.1.2 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,

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

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 $(x, y) \in M_u$ 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 $O_\delta(\log n)$[1] 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. [COOW12] 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.

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

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 $R(S)$ 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 $\mathrm{poly}(\delta) \cdot n$ 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.

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[2].

7.2 Matching lemma for zero-error LCCs

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, Yek12] 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 7.1.2 assumed that the set T_1 below is empty.

[2] 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.

Lemma 7.2.1. *Let $\mathcal{C} \subset \Sigma^n$ be a $(2, \delta, \eta)$-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 $\mathcal{R}_j^i$ for every $j \in [n] \cup \{\phi\}$ such that for every codeword $c \in \mathcal{C}$,*

$$\Pr_{j \sim \mathcal{D}_i}\left[\mathcal{R}_j^i(c_j) = c_i\right] \geq \Pr_{j \sim \mathcal{D}_i}\left[\mathcal{R}_j^i(c_j) = \sigma\right] + \eta$$

for any $\sigma \in \Sigma$ such that $\sigma \neq c_i$[3]. Moreover the distribution $\mathcal{D}_i$ is smooth over $[n]$ i.e. for every $j \in [n]$, $\Pr_{\mathcal{D}_i}[j] \leq \frac{4}{\delta n}$.

2. *For every $i \in T_2$, there exists a matching $\mathcal{M}_i$ of edges in $[n] \setminus \{i\}$ of size $|\mathcal{M}_i| \geq \frac{\delta}{4} n$ 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 $\mathcal{R}_{j,k}^i$ for every edge $(j,k) \in \mathcal{M}_i$ such that for every $c \in \mathcal{C}$,*

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

Proof. Fix $\varepsilon = \delta/4$. Let $\mathcal{A}_i$ be the local corrector algorithm of $\mathcal{C}$ for $i \in [n]$ 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.[4] 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 $\mathcal{R}_{j,k}^i(z_j, z_k)$ be the output[5] of $\mathcal{A}_i(z)$ when it samples (j,k) from the distribution $\mathcal{Q}_i$. So we have for every $\sigma \in \Sigma$,

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

[3] Here c_ϕ is an empty input defined for ease of notation.

[4] Wlog, we can assume $\mathcal{A}_i$ always queries two coordinates.

[5] Note that $\mathcal{R}_{j,k}^i$ might use additional randomness.

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

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

In this case we include $i \in T_1$. Since $\text{supp}(\mathcal{Q}_i)$ 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 $\mathcal{A}_i(z)$ i.e.

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

Clearly $|B_i| \leq 2\varepsilon n$ because $\mathcal{A}_i(z)$ makes at most two queries.

We now define a new one-query corrector for i, $\tilde{\mathcal{A}}_i(z)$ as follows: simulate $\mathcal{A}_i(z)$, but whenever $\mathcal{A}_i(z)$ queries z at a coordinate in $V_i \cup B_i$, $\tilde{\mathcal{A}}_i(z)$ doesn't query that coordinate and assumes that the queried coordinate is 0 (or some fixed symbol in Σ). Note that $\tilde{\mathcal{A}}_i(z)$ makes at most one query to z since V_i is a vertex cover for the support of $\mathcal{Q}_i$. Also $\tilde{\mathcal{A}}_i(c)$ behaves exactly like $\mathcal{A}_i(c')$ where c' is the word formed by zeroing out the $V_i \cup B_i$ coordinates of c. Since $|V_i \cup B_i| \leq 4\varepsilon n \leq \delta n$, we have

$$\Pr[\tilde{\mathcal{A}}_i(c) = c_i] = \Pr[\mathcal{A}^i(c') = c_i] \geq \Pr[\mathcal{A}^i(c') = \sigma] + \eta = \Pr[\tilde{\mathcal{A}}_i(c) = \sigma] + \eta$$

for any $\sigma \in \Sigma$ such that $\sigma \neq c_i$. Now define the distribution $\mathcal{D}_i$ over $[n] \cup \{\phi\}$ as:

$$\Pr_{\mathcal{D}_i}[j] = \Pr[\tilde{\mathcal{A}}_i(z) \text{ queries } j]$$

for $j \in [n]$ and

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

Since we never query elements of B_i, we have the required smoothness i.e. $\Pr_{\mathcal{D}_i}[j] \leq 1/(\varepsilon n)$ for all $j \in [n]$. Also define $\mathcal{R}^i_j(z_j)$ to be the output (can be randomized) of $\tilde{\mathcal{A}}_i(z)$ when it queries $j \in [n]$ and $\mathcal{R}^i_\phi(c_\phi)$ to be the output (can be randomized) of

$\tilde{\mathcal{A}}_i(z)$ when it doesn't make any query where c_ϕ is an empty input defined for ease of notation. By definition, we have

$$\Pr_{j\sim\mathcal{D}_i}[\mathcal{R}^i_j(c_j) = \sigma] = \Pr[\tilde{\mathcal{A}}_i(c) = \sigma]$$

for any $\sigma \in \Sigma$. This proves part (1). □

7.3 Proof of lower bound

7.3.1 An information theoretic lemma

The proof of Theorem 7.1.2 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 [CT12] 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:

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

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 $\Pr[\hat{X}(Y) \neq X] \leq p_e$ where p_e is small, then $H(X|Y)$ should be small as well (see [CT12] for a proof). More precisely,

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

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

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 $\geq 1/2$ given $X_{S(R)}, R$ where $X_{S(R)}$ is the restriction of X to $S(R)$. By Fano's inequality,

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

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

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

But we also have

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

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

□

7.3.2 Proof of Theorem 7.1.2

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 $O(\frac{1}{\delta^4} \cdot \log n)$ queries to c and outputs c with probability $\geq 1 - 1/n$. By Lemma 7.3.1, we get the required bound.

Let $[n] = T_1 \cup T_2$ be partition of coordinates given by Lemma 7.2.1.

Claim 7.3.2. *Algorithm $\mathcal{P}$ can learn $c|_{T_1}$ with probability $\geq 1 - 1/n$ by querying a uniformly random (sampled with repetitions) subset S of size $r = O(\frac{1}{\delta^2\eta^2} \cdot \log(n/\eta))$.*

Proof. Let $S = \{Z_1, \cdots, Z_r\}$ where each Z_i is a uniformly random element of $[n]$. By Lemma 7.2.1, for every $u \in T_1$, we have a smooth distribution $\mathcal{D}_u$ over $[n]$ and algorithms $\mathcal{R}_v^u$ for every $v \in [n]$. Let's fix $u \in T_1$ and let $p_v = \Pr_{\mathcal{D}_u}[v]$. By smoothness, $p_v \leq \frac{4}{\delta n}$ for every $v \in [n]$. The algorithm $\mathcal{P}$ estimates c_u as follows: Define the weight of σ to be

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

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

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

For $\sigma \in \Sigma$ and $v \in [n] \cup \{\phi\}$, let $f_v^\sigma = \Pr[\mathcal{R}_v^u(c_v) = \sigma]$. 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

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

Therefore W_σ is an unbiased estimator for $\Pr_{v \sim \mathcal{D}_u}[\mathcal{R}_v^u(c_v) = \sigma]$. By Lemma 7.2.1,

$$\mathbb{E}[W_{c_u}] \geq \mathbb{E}[W_\sigma] + \eta$$

for any $\sigma \in \Sigma$ such that $\sigma \neq c_u$. Now we will show that no other symbol can have higher weight than W_{c_u} except with probability $\frac{1}{n^2}$. Fix some $S \subset \Sigma$. Let us consider the random variable $W(S) = \sum_{\sigma \in S} W_\sigma$.

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

Note that this implies that $\mathbb{E}[W(\Sigma)] = p_\phi + \mathbb{E}[np_{Z_1}] = 1$. By smoothness, $np_{Z_i} \leq \frac{4}{\delta}$, so by applying Hoeffding's inequality,

$$\Pr\left[|W(S) - \mathbb{E}[W(S)]| \geq \frac{\eta}{100}\right] \leq \exp\left(-\Omega(r\delta^2\eta^2)\right). \tag{7.2}$$

Denote the expected weight of a symbol by $w_\sigma = \mathbb{E}[W_\sigma]$ and for $S \subset \Sigma$, denote the expected weight of symbols in S by $w(S) = \sum_{\sigma\in S} \mathbb{E}[W_\sigma]$. We know that $w(\Sigma) = 1$ and $w_{c_u} \geq w_\sigma + \eta$ for every $\sigma \neq c_u$.

Let $S_1 \cup S_2 \cup \cdots \cup S_t$ be a partition of $\Sigma \setminus \{c_u\}$ with smallest t such that for each S_i,

$$w(S_i) \leq w_{c_u} - \frac{\eta}{2}.$$

By minimality of t, all but one parts should have $w(S_i) \geq \eta/4$. Because if there are two parts with weight $< \eta/4$, we can merge them and the weight of the union is $< \eta/2 \leq w_u - \eta/2$, this contradicts the minimality of t. Since the total weight is at most 1, $t = O(1/\eta)$.

To show that $W_{c_u} \geq W_\sigma$ for every $\sigma \neq c_u$ w.p $\geq 1 - 1/n^2$, it is enough to show that for every part S_i, $W_{c_u} \geq W(S_i)$ w.p. $1 - 1/n^2$. Therefore by applying Equation 7.2 for each part S_i and then applying a union bound over the $t = O(1/\eta)$ parts,

$$\Pr\left[W_{c_u} \leq \max_{\sigma\neq c_u} W_\sigma\right] \leq t\exp(-\Omega(r\delta^2\eta^2)) \leq \frac{1}{n^2}$$

if $r \gg \frac{1}{\eta^2\delta^2}\log(n/\eta)$. Therefore with probability $\geq 1 - \frac{1}{n^2}$, c_u will be the symbol with maximum weight and the algorithm $\mathcal{P}$ will guess c_u correctly with probability $\geq 1 - \frac{1}{n^2}$. By union bound, we get that $\mathcal{P}$ can guess c_u correctly for all $u \in T_1$ with probability $\geq 1 - \frac{1}{n}$. □

We will now show that after learning $c|_{T_1}$, $\mathcal{P}$ can now learn $c|_{T_2}$ by querying a further $O_\delta(\log n)$ 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 7.3.2, we have shown that if S is a randomly chosen subset of size $O_{\delta,\eta}(\log n)$, then $T_1 \subseteq R(S)$ with probability $\geq 1 - \frac{1}{n}$. 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 $\geq 1 - \frac{1}{n}$. So we can assume that $T_2 \nsubseteq R(S)$. Our goal is to show that we can add a further $O(\text{poly}(1/\delta) \cdot \log n)$ 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 7.3.3. *There exists a set S of size $O((1/\delta)^4 \cdot \log n)$ such that $R(S \cup T_1) = V$.*

7.3.3 Proof of Claim 7.3.3

Claim 7.3.3 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 $(\delta/4)n$ given by Lemma 7.2.1. Let $\delta := \delta/4$. 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 7.2.1 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 7.3.4. *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 7.3.3 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 7.3.5 (Claim 7.3.3, 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 7.3.5. 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 7.3.6 (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 $|V'| \geq \delta|V|$ 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 $|E^*(V^*)| \leq |V^*| \cdot |V^*|/2$ 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 $0.9\delta|V| \cdot (|V^*| - 0.1\delta|V|) \geq 0.9^2\delta^2 \cdot |V|^2$. 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., $|E^*(V^*)| \geq 0.9^2\delta^2|V|^2 - \delta'|V|^2$. Combining both upper and lower bounds on $|E^*(V^*)|$ gives

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

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

Lemma 7.3.7 (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 $|R(S)| \geq d/2$.*

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 $|R(S)| \geq d/2$, 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

$$\left|E(A,B) \cap \bigcup_{i \in B} \mathcal{M}_i\right| = \sum_{a \in A} \left|E(a,B) \cap \bigcup_{i \in B} \mathcal{M}_i\right| = \sum_{a \in A} \left|E(a, V \setminus \{a\}) \cap \bigcup_{i \in B} \mathcal{M}_i\right| \quad (7.3)$$

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 $|E(a, V \setminus \{a\}) \cap \bigcup_{i \in B} \mathcal{M}_i| \geq d - |A| \geq d/2$. Plugging this estimate to Eq. (7.3) gives

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

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 $|A| = |R(S)| \leq d/2$ 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 $|R(S)| \geq (1 + \frac{d}{2|V|})^t$ or $|R(S)| \geq d/2$. Taking $t = O(\frac{|V|}{d} \cdot \log|V|)$ gives that after at most t iterations $|R(S)| \geq d/2$. □

Lemma 7.3.8 (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 7.3.9 (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 $|R(S)| \leq (1 - \delta)n$. Then, either*

1. *There exists an $i \in V \setminus R(S)$ such that $|R(S \cup \{i\})| \geq |R(S)| + 0.01 \cdot \delta^2 \cdot n$.*
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 $|R(S \cup \{i\})| \geq |R(S)| + 0.01\delta^2 \cdot n$.) 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 7.3.5.

Proof of Claim 7.3.5. 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 $|R(S \cup \{i\})| \geq |R(S)| + 0.01\delta^2 n$. 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 7.3.8, $|V'| \geq \delta n$. By Lemma 7.3.9, 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 $\delta' = 0.9\delta n/|V'| \geq \delta$. We apply Lemma 7.3.6 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 $d = \Omega((\delta')^2 \cdot |V'|) \geq \Omega(\delta^2 n)$. We apply Lemma 7.3.7 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 $|R_{G''}(S'')| \geq \Omega(d) \geq \Omega(\delta^2 n)$. 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 $O((1/\delta')^2 \cdot \log n) \leq O((1/\delta^2) \cdot \log n)$ elements are added to S. Overall, the size of S at the end of the process will be

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

Chapter 8

Applications to additive combinatorics

8.1 Introduction

In this chapter we will show a few applications of the theory of locally decodable codes to additive combinatorics. Specifically, we will show that techniques used to prove LDC lower bounds can be used to improve bounds on well-studied problems in additive combinatorics. Obtaining better LDC lower bounds, will very likely improve the bounds for these problems and conversely improvements to these bounds could lead to ideas that might be useful for improving LDC lower bounds. The structure underlying these connections is a way to bound Gaussian width of special point sets that arise as images of some special low-degree maps.

The *Gaussian width* of a point set $T \subseteq \mathbb{R}^k$ measures the expected maximum correlation between T and a standard Gaussian vector $g = N(0, I_k)$, and is given by

$$w(T) = \mathbb{E}\Big[\sup_{x\in T}\langle x, g\rangle\Big].$$

The terminology reflects the fact that the Gaussian width of a set is proportional to $\sqrt{k}$ times its average width in a random direction. While this quantity plays a central role in high-dimensional probability, it is notoriously hard to estimate in general; see for instance [Tal14b] for an extensive discussion of this problem.

Our main result gives upper bounds on the Gaussian width of sets that appear naturally in the context of probabilistic combinatorics. The relevant sets are given by the image of the n-dimensional Boolean hypercube under a certain polynomial mapping $\psi : \mathbb{R}^n \to \mathbb{R}^k$. In particular, we focus on the case where each coordinate $\psi_i : \mathbb{R}^n \to \mathbb{R}$ is a multilinear polynomial with 0-1 coefficients. Say that a polynomial has *multiplicity* t if each of its variables has a nonzero exponent in at most t monomials in its support.

Theorem 8.1.1. *Let $\psi : \mathbb{R}^n \to \mathbb{R}^k$ be a polynomial mapping such that each coordinate is multilinear, has 0-1 coefficients, and has degree at most d and multiplicity t. Then,*

$$w\Big(\psi(\{0,1\}^n)\Big) \lesssim_d nt\sqrt{kn^{1-\frac{1}{\lceil d/2\rceil}}\log n}.$$

The factor nt can be seen as a natural scaling due to the fact that each coordinate ψ_i maps the Boolean hypercube into $[0, nt]$ (which follows from a handshaking lemma). In the special case where ψ is linear, $\psi(x) = (\langle c_1, x\rangle, \ldots, \langle c_k, x\rangle)$, for some $c_1, \ldots, c_k \in \{0,1\}^N$, the set $\psi(\{0,1\}^n)$ is easily seen to be contained in the set $T = \{(\langle c_i, y\rangle)_{i=1}^k : \|y\|_{\ell_\infty} \leq 1\}$. The Gaussian width of the former set is thus at most that of the latter, which in turn is at most

$$\mathbb{E}\Big[\Big\|\sum_{i=1}^k g_i c_i\Big\|_{\ell_1}\Big] \lesssim n\sqrt{k},$$

as the sum is an n-dimensional Gaussian vector whose coordinates have variance at most k. Perhaps surprisingly, Theorem 8.1.1 shows that if ψ is quadratic and has constant multiplicity, then the Gaussian width is at most a factor $\sqrt{\log n}$ larger than the above upper bound. This turns out to be an easy consequence of a 1974 random matrix inequality due to Tomczak–Jeagermann [TJ74], which also forms the basis for our proof of the higher-degree cases. The proof of Theorem 8.1.1 (given in Section 8.2) proceeds in two steps: first we reduce to the case of homogeneous mappings of even degree, and then we reduce to the quadratic case. The first step is the reason for the ceiling in $\lceil d/2 \rceil$ appearing in the exponent and it would be interesting to know if one can remove this ceiling (i.e., does the result hold with the exponent $1 - 2/d$?). Finally, a close inspection of the proof of Theorem 8.1.1 shows that it also holds for polynomials with non-negative integer coefficients, for a suitable change of the definition of multiplicity. In the following four subsections we discuss two applications of this result and links with error correcting codes and the Banach space notion of type.

8.1.1 Random differences in Szemerédi's Theorem

In 1975 Szemerédi [Sze75] proved that any subset of the integers of positive upper density contains arbitrarily long arithmetic progressions, answering a famous open question of Erdős and Turán. It is well known that this is equivalent to the assertion that for every positive integer k and any $\alpha \in (0, 1)$, there exists an $N_0(k, \alpha) \in \mathbb{N}$ such that if $N \geq N_0(k, \alpha)$ and $A \subseteq \mathbb{Z}/N\mathbb{Z}$ is a set of size $|A| \geq \alpha N$, then A must contain a proper k-term arithmetic progression. Certain refinements of Szemerédi's theorem concern sets $D \subseteq \mathbb{N}$ for which the theorem still holds true when the arithmetic progressions are required to have common difference from D. Such sets are usually referred to as intersective sets in number theory, or recurrent sets in ergodic theory. More precisely, a set $D \subseteq \mathbb{N}$ is ℓ-intersective (or ℓ-recurrent) if any set $A \subseteq \mathbb{N}$ of positive upper density has an $(\ell + 1)$-term arithmetic progression with common difference in D. Szemerédi's theorem then states that $\mathbb{N}$ is ℓ-intersective for every $\ell \in \mathbb{N}$, but much smaller intersective sets exist. For example, for any $t \in \mathbb{N}$, the set $\{1^t, 2^t, 3^t, \dots\}$ is ℓ-intersective for every ℓ, which is a special case of more general results of Sárközy [Sár78a] when $\ell = 1$ and of Bergelson and Leibman [BL96] for all $\ell \geq 1$. The shifted primes $\{p - 1 : p \text{ is prime}\}$ and $\{p + 1 : p \text{ is prime}\}$ are also ℓ-intersective for every $\ell \in \mathbb{N}$, shown by Sárközy [Sár78b] when $\ell = 1$ and in a more general setting by Wooley and Ziegler [WZ12] for all $\ell \geq 1$.

It is natural to ask at what density, random sets become ℓ-intersective. To simplify the discussion, we will look at the analogous question in $\mathbb{Z}/N\mathbb{Z}$.

Definition 8.1.2. *Let ℓ be a positive integer and $\alpha \in (0, 1]$. A subset $D \subseteq \mathbb{Z}/N\mathbb{Z}$ is (ℓ, α)-intersective if any subset $A \subseteq \mathbb{Z}/N\mathbb{Z}$ of size $|A| \geq \alpha N$ contains a proper $(\ell + 1)$-term arithmetic progression with common difference in D.*

It was proved independently by Frantzikinakis et al. [FLW12] and Christ [Chr11] that for $\beta_\ell = \frac{1}{2^{\ell-1}}$ and $p \geq \omega(N^{-\beta_\ell} \log N)$, the random set $[\mathbb{Z}/N\mathbb{Z}]_p$ is (ℓ, α)-intersective with probability $1 - o(1)$, provided $N \geq N_1(\ell, \alpha)$. This was improved for all $\ell \geq 2$ in [FLW16b], where it was shown that the same result holds with $\beta_\ell = \frac{1}{\ell+1}$, though it was conjectured there that $\beta_\ell = 1$ suffices for all $\ell \geq 1$. Based on Theorem 8.1.1 we obtain the following result, which improves on the latter bounds.

Theorem 8.1.3. *For every $\ell \in \mathbb{N}$ and $\alpha \in (0, 1)$, there exists an $N_1(\ell, \alpha) \in \mathbb{N}$ such that the following holds. Let $N \geq N_1(\ell, \alpha)$ be an integer and let*

$$\beta_\ell = \frac{1}{\lceil \frac{\ell+1}{2} \rceil} \quad \textit{and} \quad p \geq \omega(N^{-\beta_\ell} \log N).$$

Then, with probability $1 - o(1)$, the set $[\mathbb{Z}/N\mathbb{Z}]_p$ is (ℓ, α)-intersective.

8.1.2 Large deviations for arithmetic progressions

Let $H = (V, E)$ be a hypergraph over a finite vertex set V of cardinality N and for $p \in (0, 1)$ denote by V_p the random binomial subset where each element of V

appears independently of all others with probability p. Let X be the number of edges in H that are induced by V_p. Important instances of the random variable X include the count of triangles in an Erdős–Rényi random graph and the count of arithmetic progressions of a given length in the random set $[\mathbb{Z}/N\mathbb{Z}]_p$.

The study of the asymptotic behavior of X when $p = p(N)$ is allowed to depend on N and N grows to infinity motivates a large body of research in probabilistic combinatorics. Of particular interest is the problem of determining the probability that X significantly exceeds its expectation $\Pr[X \geq (1+\delta)\mathbb{E}X]$ for $\delta > 0$, referred to as the *upper tail*. Despite the fact that standard probabilistic methods fail to give satisfactory bounds on the upper tail in general, advances were made recently for special instances, in particular for triangle counts [LZ17] and general subgraph counts [BGLZ17]. For more general hypergraphs, progress was made by Chatterjee and Dembo [CD16] using a novel nonlinear large deviation principle (LDP), which was improved by Eldan [Eld16] shortly after. The LDPs give precise estimates on the upper tail that are given in terms of a parameter ϕ_p whose value is determined by the solution to a certain variational problem. The range of values of p for which these estimates are actually valid depends on the underlying hypergraph H. This splits the problem of estimating the upper tail into two sub-problems: (1) determining for what range of p the estimate in terms of ϕ_p holds true and (2) solving the variational problem to determine the value of ϕ_p. The answer to problem (1) turns out to depend on the Gaussian width of a point set related to H.

This approach was pursued in [CD16] to estimate the upper tail of the number of 3-term arithmetic progressions in $[\mathbb{Z}/N\mathbb{Z}]_p$, for which the authors solved problem (1). The case of longer APs, asking for the upper tail probability of the count X_k of k-term arithmetic progressions in $[\mathbb{Z}/N\mathbb{Z}]_p$, was recently treated by Bhattacharya et al. [BGSZ18]. They solved the variational problem (2) for N prime and gave bounds for the relevant Gaussian width towards solving problem (1). Based on this, they showed that if $k \geq 3$ and $\delta > 0$ are fixed and p tends to zero sufficiently slowly as $N \to \infty$ along the primes, then

$$\Pr[X_k \geq (1+\delta)\mathbb{E}X_k] = p^{(1+o(1))\sqrt{\delta}p^{k/2}N}. \tag{8.1}$$

Similar results were shown for the analogous problem over $\{1,\dots,N\}$ (in which case N no longer needs to be prime), but we shall focus on the problem in $\mathbb{Z}/N\mathbb{Z}$ for ease of exposition. The rate at which p is allowed to decay for (8.1) to hold turns out to depend on Gaussian widths of the form featuring in Theorem 8.1.1. The bounds proved in [BGSZ18] imply that (8.1) holds provided $p \geq N^{-c_k}(\log N)^{\varepsilon_k}$ for

$$c_3 = \frac{1}{18}, \quad c_4 = \frac{1}{48} \quad \text{and} \quad c_k = \frac{1}{6k(k-1)} \quad \text{for } k \geq 5,$$

and absolute constants $\varepsilon_k \in (0,\infty)$ depending only on k. However, the authors conjecture that a probability p slightly larger than $N^{-1/(k-1)}$ suffices for all k. Some support for this conjecture is given by a result of Warnke [War16] showing that for all $p \geq (\log N/N)^{1/(k-1)}$, the logarithm of the upper tail (also referred to as the large deviation rate) of the k-AP count in $\{1,\dots,N\}_p$ is given by $\Theta_k(\sqrt{\delta}p^{k/2}N\log p)$, where the asymptotic notation hides constants depending only on k. Notice that (8.1) is more accurate than this result in that it (almost) determines those constants, though currently for a more narrow range of p.[1] Using Theorem 8.1.1, we widen the range of p for which (8.1) can be shown to hold for all $k \geq 5$.

Theorem 8.1.4. *For every integer $k \geq 3$ and*

$$c_k = \frac{1}{6k\left\lceil\frac{k-1}{2}\right\rceil},$$

[1] The main motivation for finding such precise estimates of the upper tail probability is not so much the problem itself as it is to understand structure of the set $[\mathbb{Z}/N\mathbb{Z}]_p$ conditioned on X_k being much larger than its expectation (see [BGSZ18]).

the estimate (8.1) *holds true, provided* $p \geq N^{-c_k}(\log N)$ *and* N *is prime.*

8.1.3 Relation to LDCs

There is a close connection between the Gaussian widths considered in Theorem 8.1.1 and LDCs. In Chapter 5, we showed that q-query LDCs from $\{0,1\}^{\Omega(k)}$ to $\{0,1\}^{O(n)}$ are equivalent to mappings $\psi : \mathbb{R}^n \to \mathbb{R}^k$ whose coordinates are degree-q, multiplicity-1 polynomials with 0-1 coefficients that are supported by $\Omega(n)$ monomials, and such that the set $\psi(\{0,1\}^n)$ has Gaussian width $\Omega(k)$. Because of this connection, the best-known lower bounds on the length $n = n(k)$ of q-query LDCs—proved using techniques from quantum information theory [KW04]—imply a slightly different but equivalent version of Theorem 8.1.3 (see Section 8.5). The proof of Theorem 8.1.1 is based on ideas from [KW04], but does not use quantum information theory. Not surprisingly, the LDC lower bounds of [KW04] are also implied by Theorem 8.1.1.

8.1.4 Gaussian width bounds from type constants

We observe that the Gaussian width in Theorem 8.1.1 can be bounded in terms of type constants of certain Banach spaces. Unfortunately, we do not have good enough bounds on the type constants of the required spaces to improve Theorem 8.1.1. But we hope that this connection will motivate progress on understanding these spaces.

A Banach space X is said to have (Rademacher) type $p > 0$ if there exists a constant $T < \infty$ such that for every k and $x_1, \ldots, x_k \in X$,

$$\mathbb{E}_\varepsilon \left\| \sum_{i=1}^k \varepsilon_i x_i \right\|_X^p \leq T^p \sum_{i=1}^k \|x_i\|_X^p , \tag{8.2}$$

where the expectation is over a uniformly random $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k) \in \{-1,1\}^k$. The smallest T for which (8.2) holds is referred to as the type-p constant of X, denoted $T_p(X)$. Type, and its dual notion cotype, play an important role in Banach space theory as they are tightly linked to local geometric properties (we refer to [LT79] and [Mau03] for extensive surveys). Some fundamental facts are as follows. It follows from the triangle inequality that every Banach space has type 1 and from the Khintchine inequality that no Banach space has type $p > 2$. The parallelogram law implies that Hilbert spaces have type 2. An easy but important fact is that ℓ_1 fails to have type $p > 1$. Indeed, a famous result of Maurey and Pisier [MP73] asserts that a Banach space fails to have type $p > 1$ if and only if it contains ℓ_1 uniformly. Finite-dimensional Banach spaces have type-p for all $p \in [1, 2]$.

Of importance to Theorem 8.1.1 are the actual type constants $T_p(X)$ of a certain family of finite-dimensional Banach spaces. Let $r_1, \dots, r_d \geq 1$ be such that $\sum_{i=1}^d \frac{1}{r_i} = 1$ and let $\mathcal{L}^n_{r_1,\dots,r_d}$ be the space of d-linear forms on $\mathbb{R}^n \times \cdots \times \mathbb{R}^n$ (d times) endowed with the norm

$$\|\Lambda\| = \sup\Big\{ \frac{|\Lambda(x_1, \dots, x_d)|}{\|x_1\|_{\ell_{r_1}} \cdots \|x_d\|_{\ell_{r_d}}} : x_1, \dots, x_d \in \mathbb{R}^n \setminus \{0\} \Big\}.$$

This space is also known as the injective tensor product of $\ell^n_{s_1}, \dots, \ell^n_{s_d}$ for $r_i^{-1} + s_i^{-1} = 1$ and as such plays an important role in the theory of tensor products of Banach spaces [Rya02]. The relevance of the type constants of this space to Theorem 8.1.1 is captured by the following lemma, proved in Section 8.7.

Lemma 8.1.5. *Let $\psi : \mathbb{R}^n \to \mathbb{R}^k$ be a polynomial mapping such that each coordinate is multilinear and has 0-1 coefficients, degree at most d and multiplicity t. Then for any $r_1, \dots, r_d \geq 1$ such that $\sum_{i=1}^d \frac{1}{r_i} = 1$ and any $p \in [1, 2]$,*

$$w\Big(\psi(\{0,1\}^n)\Big) \lesssim_d nt\, T_p(\mathcal{L}^n_{r_1,\dots,r_d})\, k^{1/p}.$$

Observe that the space $\mathcal{L}^n_{2,2}$ may be identified with the space of $n \times n$ matrices endowed with the spectral norm (or operator norm). A key ingredient in the proof of Theorem 8.1.1, Theorem 8.2.1 below, easily implies that the type-2 constant of this space is of order $O(\sqrt{\log n})$. A well-known lower bound of the same order follows for instance from the connection between Gaussian width and LDCs and a basic construction of a 2-query LDC known as the Hadamard code. More generally, lower bounds on the type constants of $\mathcal{L}^n_{r_1,\dots,r_d}$ are implied by d-query LDCs [BNR12, Bri16].

8.2 Proof of Theorem 8.1.1

In this section we prove Theorem 8.1.1. We begin by giving a high-level overview of the ideas. The main tool we use is the following random matrix inequality, which is a special case of a non-commutative version of the Khintchine inequality due to Tomczak-Jaegermann [TJ74, Theorem 3.1]. Let $\langle\cdot,\cdot\rangle$ be the standard inner product on $\mathbb{R}^N$ and denote by B_2^N the Euclidean unit ball in $\mathbb{R}^N$. Given a matrix $A \in \mathbb{R}^{N\times N}$, its operator norm (or spectral norm) is given by $\|A\| = \sup\{|\langle Ax, y\rangle| : x, y \in B_2^N\}$.

Theorem 8.2.1 (Tomczak-Jaegermann)**.** *There exists an absolute constant $C \in (0,\infty)$ such that the following holds. Let $A_1,\dots,A_k \in \mathbb{R}^{N\times N}$ be a collection of matrices and let $g_1,\dots,g_k$ be independent Gaussian random variables with mean zero and variance 1. Then,*

$$\mathbb{E}\Big[\Big\|\sum_{i=1}^k g_i A_i\Big\|\Big] \le C\sqrt{\log N}\Big(\sum_{i=1}^k \|A_i\|^2\Big)^{1/2}.$$

This result already suffices to prove Theorem 8.1.1 when the coordinate mappings ψ_i are quadratic forms, in which case there exist matrices $A_i \in \{0,1\}^{n\times n}$ such that $\psi_i(x) = \langle A_i x, x\rangle$. The assumption that each ψ_i has multiplicity t implies that each row and column of A_i has at most t ones. This in turn implies that $\|A_i\| \le t$ by a BirkhoffâĂŞ-von Neumann-type theorem. Since each $x \in \{0,1\}^n$ has Euclidean norm at most $\sqrt{n}$, we get

$$\begin{aligned} w\Big(\psi(\{0,1\}^n)\Big) &= \mathbb{E}\Big[\max_{x\in\{0,1\}^n}\sum_{i=1}^k g_i\langle A_i x, x\rangle\Big] \\ &= \mathbb{E}\Big[\max_{x\in\{0,1\}^n}\Big\langle\Big(\sum_{i=1}^k g_i A_i\Big)x, x\Big\rangle\Big] \le n\mathbb{E}\Big[\Big\|\sum_{i=1}^k g_i A_i\Big\|\Big]. \end{aligned}$$

By Theorem 8.2.1, the above is at most $Ctn\sqrt{k\log n}$.

The general case is proved via a reduction to the above quadratic case and consists of two steps. In the first step, we reduce to the case where each coordinate ψ_i is a homogeneous polynomial of degree $2\lceil d/2\rceil$. This is done in a straightforward way by adding at most dn variables in such a way so as to preserve the multiplicity. The second step consists of a reduction to the quadratic case. For this, it will be convenient to consider the hypergraphs associated with the monomial support of the coordinate mappings ψ_i.

Recall that an d-hypergraph $H = (V, E)$ consists of a vertex set V and a multiset E, also denoted $E(H)$, of subsets of V of size at most d, called the edges. A hypergraph is d-uniform if each edge has size exactly d. The degree of a vertex is the number of edges containing it and the degree of H, denoted $\Delta(H)$, is the maximum degree among its vertices. A *matching* is a hypergraph where no two edges intersect. Associate with a hypergraph $H = ([n], E)$, the multilinear polynomial $p_H \in \mathbb{R}[x_1, \dots, x_n]$ given by

$$p_H(x_1,\dots,x_n) = \sum_{e\in E}\prod_{i\in e} x_i. \tag{8.3}$$

The multiplicity of p_H is then exactly the degree $\Delta(H)$. Clearly the coordinate mappings ψ_i of the form featuring in Theorem 8.1.1 can be written as p_H for some d-hypergraph H of degree at most t. The reduction to the quadratic case is based on the following key lemma, in which for $x \in \mathbb{R}^n$ and $m \in \mathbb{N}$, the the mth tensor power x is defined as $x^{\otimes m} = (\prod_{i=1}^m x_{u_i})_{u \in [n]^m}$.

Lemma 8.2.2 (Matrix lemma)**.** *For every $r \in \mathbb{N}$ there exist a $C_r, c_r \in (0,\infty)$ and $n_0(r) \in \mathbb{N}$ such that the following holds. Let $n \geq n_0(r)$, $m = C_r n^{1-1/r}$ and $N = n^m$. Let $H = ([n], E)$ be a $2r$-uniform hypergraph and let p_H be the polynomial as in* (8.3)*. Then, there exists a matrix $A \in \mathbb{R}^{N\times N}$ such that $\|A\| \lesssim_r \Delta(H)$ and for every $x \in \{-1,1\}^n$,*

$$p_H(x) = \frac{n}{c_r N}\langle Ax^{\otimes m}, x^{\otimes m}\rangle.$$

Moreover, A is the adjacency matrix of a graph (with possible parallel edges).

With this lemma in hand, the proof of Theorem 8.1.1 is straightforward (see below). The idea behind Lemma 8.2.2 is to use decompositions into matchings and a generalization of the Birthday Paradox that says that for any n-vertex $2r$-matching, a random subset of $C_r n^{1-1/r}$ vertices contains r vertices of any fixed edge with probability c_r/n. To illustrate how this is used in the $r = 2$ case, let H be a 4-matching, let $m = C_2\sqrt{n}$ and $N = n^m$. It follows from the generalized Birthday Paradox that there are $c_2 N/n$ strings in $[n]^m$ containing at least two elements of a given edge. Now let G be the graph with vertex set $[n]^m$ whose edges are the pairs $\{u, v\}$ that *cover* some edge in H and *complement* each other, meaning: there are indices $i, j \in [m]$ such that $\{u_i, u_j, v_i, v_j\} \in E(H)$ and $u_\ell = v_\ell$ for all $\ell \notin \{i, j\}$. The main observation is that for every edge $\{u, v\} \in E(G)$ that covers an edge $e \in E(H)$ and every $x \in \{-1,1\}^n$, we have

$$(x^{\otimes m})_u (x^{\otimes m})_v = \prod_{\ell=1}^m x_{u_\ell} x_{v_\ell} = x_{u_i} x_{u_j} x_{v_i} x_{v_j} = \prod_{w \in e} x_w.$$

It follows that, modulo the relations $x_1^2 = 1, \ldots, x_n^2 = 1$, we have $p_G(x^{\otimes m}) = (c_2 N/n) p_H(x)$. The lemma would now follow by letting A be the appropriately scaled adjacency matrix of G, were it not for the issue that G could have very high degree, which would result in A having a large operator norm. To deal with this, we instead consider a pruned version of G in which we keep only edges that do not cover too many edges of H.

We now give the formal proof of Theorem 8.1.1. The following simple proposition is used for the first step, in which we homogenize the polynomials. Given two hypergraphs H, H', say that H' *majorizes* H if $V(H) \subseteq V(H')$ and if for each edge $e \in E(H)$, there is a unique edge $e' \in E(H')$ such that $e \subseteq e'$.

Proposition 8.2.3. *For any n-vertex d-hypergraph H, there is a d-uniform hypergraph H' on dn vertices that majorizes H and satisfies $\Delta(H') = \Delta(H)$.*

Proof. Let $t = \Delta(H)$. It follows from the handshaking lemma that $|E(H)| \leq tn$. Partition $E(H) = \{E_1, \dots, E_n\}$ into n pairwise disjoint sets of size at most t each. Add to $V(H)$ pairwise disjoint sets $W_1, \dots, W_n$ of $d-1$ new vertices each. For each $i \in [n]$, complete each edge $e \in E_i$ to a set of size d by adding vertices from W_i and let H' be the hypergraph thus obtained. Observe that we have not increased the degree of the vertices in $V(H)$. Since each E_i has size at most t, the new vertices in W_i also have degree at most t and therefore, $\Delta(H') = t$. It is trivial to verify that H' satisfies the other desired properties. □

of Theorem 8.1.1. Let $r = \lceil d/2 \rceil$ and for each $i \in [k]$, let H_i be the d-hypergraph of degree t such that $\psi_i = p_{H_i}$, with p_{H_i} as in (8.3). Assume that $n \geq n_0(r)$ for $n_0(r)$ as in Lemma 8.2.2. We start by reducing to the setting where each H_i is $2r$-uniform and of degree at most t. To this end, let $H_i' = ([n] \cup [(2r-1)n], E_i')$ be a $2r$-uniform hypergraph that majorizes H_i as in Proposition 8.2.3, which exists since any d-hypergraph is a $2r$-hypergraph. Then, for each $e \in E(H_i)$, there is a unique set $f(e) \subseteq [(2r-1)n]$ such that $e \cup f(e) \in E(H_i')$. It follows that

$$p_{H_i}(x) = \sum_{e \in E(H_i)} \prod_{i \in e} x_i = \sum_{e \in E(H_i)} \prod_{i \in e} x_i \prod_{j \in f(e)} 1 = p_{H_i'}((x, \mathbf{1})),$$

where $\mathbf{1} \in \mathbb{R}^{(2r-1)n}$ is the all-ones vector. Hence, if we let $\psi' : \mathbb{R}^{2rn} \to \mathbb{R}^k$ be the polynomial map whose coefficients are given by $p_{H_i'}$, then

$$w\Big(\psi(\{0,1\}^n)\Big) \leq w\Big(\psi'(\{0,1\}^{2rn})\Big).$$

Since the dependence of our claimed bound on the Gaussian width is polynomial in n, the extra vertices will result in an extra factor depending only on d. It thus suffices to prove the theorem for the case where $H_1, \dots, H_k$ are $2r$-uniform.

Observe that since the polynomials ψ_i are multilinear, the Gaussian width is bounded from above by replacing binary vectors with sign vectors. In particular,

$$w\Big(\psi(\{0,1\}^n)\Big) \leq \mathbb{E} \max \Big\{ \sum_{i=1}^k g_i p_{H_i}(x) : \, x \in \{-1,1\}^n \Big\}.$$

Let $m = C_r n^{1-1/r}$ and $N = n^m$ and for each $i \in [k]$, let $A_i \in \mathbb{R}^{N \times N}$ be a matrix for p_{H_i} as in Lemma 8.2.2. Then, for every $x \in \{-1,1\}^n$,

$$\sum_{i=1}^k g_i p_{H_i}(x) = \frac{n}{c_r N} \sum_{i=1}^n g_i \langle A_i x^{\otimes m}, x^{\otimes m} \rangle \leq \frac{n}{c_r} \Big\| \sum_{i=1}^k g_i A_i \Big\|,$$

where in the inequality we used that $x^{\otimes m}$ has Euclidean norm $\sqrt{N}$. Taking expectations, it then follows from Theorem 8.2.1 that the Gaussian width of $\psi(\{0,1\}^n)$ is at most

$$\frac{n}{c_r} \mathbb{E}\Big[\Big\| \sum_{i=1}^k g_i A_i \Big\|\Big] \lesssim \frac{n}{c_r} \sqrt{\log N} \Big(\sum_{i=1}^k \|A_i\|^2 \Big)^{1/2} \lesssim_r nt\sqrt{kn^{1-1/r} \log n},$$

where in the second inequality we used that $\|A_i\| \leq O_r(t)$ for each $i \in [k]$. □

8.3 Proof of the matrix lemma

In this section we prove Lemma 8.2.2. The starting point is a decomposition of a bounded-degree hypergraph into a small number of matchings. For this, we use the following basic result on edge colorings. The *edge chromatic number* of a hypergraph H, denoted by $\chi_E(H)$, is the minimum number of colors needed to color the edges of H such that no two edges which intersect have the same color. Note that $\chi_E(H)$ equals the smallest number of matchings into which $E(H)$ can be partitioned.

Lemma 8.3.1. *Let H be a d-hypergraph. Then,*

$$\Delta(H) \leq \chi_E(H) \leq d(\Delta(H) - 1) + 1.$$

Proof. Clearly $\chi_E(H) \geq \Delta(H)$ since edges containing a maximum degree vertex should get different colors. To prove the upper bound, form a graph G whose vertices are $E(H)$, and add edges between intersecting hypergraph edges. Then $\chi_E(H)$ is equal to the vertex chromatic number of the graph G, which, by Brooks' Theorem, is at most $\Delta(G) + 1$. Since an edge in H can intersect at most $d(\Delta(H) - 1)$ other edges, $\Delta(G) \leq d(\Delta(H) - 1)$. □

To deal with matchings, we introduce the following definitions. Let $\mathcal{M} \subseteq \binom{[n]}{2r}$ be a maximal $2r$-matching of $[n]$. Let $s = 200 \cdot 4^r$. Given a string $x \in \{-1, 1\}^n$ write its m-fold tensor product as

$$x^{\otimes m} = \left(\prod_{i=1}^{m} x_{f(i)} \right)_{f:[m] \to [n]}.$$

Given a mapping $f : [m] \to [n]$ and set $S \in \mathcal{M}$, let

$$\mu_S(f) = \sum_{T \in \binom{S}{r}} \prod_{i \in T} |f^{-1}(i)|.$$

Note that this is a count of the r-subsets $I \subseteq [m]$ such that $|S \cap f(I)| = r$. Denote

$$\phi(f) = \sum_{S \in \mathcal{M}} \mu_S(f).$$

For $\ell \in \mathbb{N}$, say that f is *ℓ-good* if $1 \leq \phi(f) \leq \ell$. Say that $g : [m] \to [n]$ *complements* f if it satisfies the following two criteria:

1. There exists exactly one $I \in \binom{[m]}{r}$ such that $f(I) \cup g(I) \in \mathcal{M}$.

2. For all $i \in [m] \setminus I$, we have $g(i) = f(i)$.

If g complements f then clearly the converse also holds. Say that the complementary pair (f, g) *covers* $S \in \mathcal{M}$ if $f(I) \cup g(I) = S$. Observe that if (f, g) covers S, then for every $x \in \{-1, 1\}^m$, we have

$$(x^{\otimes m})_f (x^{\otimes m})_g = \prod_{i=1}^{m} x_{f(i)} x_{g(i)} = \prod_{j \in S} x_j. \tag{8.4}$$

Define the set of ordered pairs

$$\mathcal{P} = \Big\{ (f, g) : \; f \text{ is } s\text{-good and } g \text{ complements } f \Big\}. \tag{8.5}$$

Proposition 8.3.2. *Let $\mathcal{P}$ be as in* (8.5)*. Then, for every $S \in \mathcal{M}$, the number of pairs $(f, g) \in \mathcal{P}$ that cover S equals $|\mathcal{P}|/|\mathcal{M}|$.*

Proof. Fix distinct sets $S, T \in \mathcal{M}$ and let $\pi \in S_n$ be a permutation such that $\pi(S) = T, \pi(T) = S$ and $\pi(i) = i$ for all $i \notin S \cup T$. Let $\mathcal{P}_S$ be the set of pairs $(f, g) \in \mathcal{P}$ which cover S and define $\mathcal{P}_T$ similarly. We claim that the map $\psi : (f, g) \mapsto (\pi \circ f, \pi \circ g)$

is an injective map from $\mathcal{P}_S$ to $\mathcal{P}_T$. It follows that T is covered by at least as many pairs from $\mathcal{P}$ as S is. Similarly, interchanging S and T, the converse also holds. To prove the claim, note that if (f, g) covers S, then $(\pi \circ f, \pi \circ g)$ covers T. Moreover, $\phi(\pi \circ f) = \phi(f)$ because π maps edges of the matching $\mathcal{M}$ to edges of $\mathcal{M}$. Thus $\psi(\mathcal{P}_S) \subset \mathcal{P}_T$. Finally ψ is injective because if $\pi \circ f = \pi \circ f'$ for some $f, f' : [m] \to [n]$, then $f = f'$. Hence $\mathcal{P}$ covers all $S \in \mathcal{M}$ equally. □

Proposition 8.3.3. *For every $(f, g) \in \mathcal{P}$, we have that g is s^2-good.*

Proof. Let $S \in \mathcal{M}$ and $(f, g) \in \mathcal{P}$ be such that (f, g) covers S. Consider the histograms $F, G : [n] \to \{0, 1, \ldots, m\}$ given by $F(i) = |f^{-1}(i)|$ and $G(i) = |g^{-1}(i)|$ for each $i \in [n]$. Then F and G differ only in S. In particular, there is an r-set $T \subseteq S$ such that $G(i) = F(i) + 1$ for each $i \in T$ and $G(i) = F(i) - 1$ for each $i \in S \setminus T$. Hence,

$$\begin{aligned}
\mu_S(g) &= \sum_{T \in \binom{S}{r}} \prod_{i \in T} G(i) \\
&\leq \sum_{T \in \binom{S}{r}} \prod_{i \in T} \Big(F(i) + 1\Big) \\
&\leq \sum_{T \in \binom{S}{r}} \Big(1 + 2^r \prod_{i \in T} F(i)\Big) \\
&\leq 4^r + 2^r \mu_S(f).
\end{aligned}$$

For all other $S' \in \mathcal{M}$, we have $\mu_{S'}(g) = \mu_{S'}(f)$. Moreover, f must be s-good for (f, g) to belong to $\mathcal{P}$. It follows that

$$\phi(g) = \sum_{S' \in \mathcal{M}} \mu_{S'}(g) \leq 4^r + 2^r \sum_{S' \in \mathcal{M}} \mu_{S'}(f) = 4^r + 2^r \phi(f) \leq s^2,$$

where in the last line we used the choice of $s = 200 \cdot 4^r$. □

Lemma 8.3.4 (Generalized birthday paradox)**.** *For every $r \in \mathbb{N}$ there exists a $C_r \in (0, \infty)$ and an $n_0(r) \in \mathbb{N}$ such that the following holds. Let h be a uniformly distributed random variable over the set of maps from $[m]$ to $[n]$. Then, provided $n \geq n_0(r)$ and $m = C_r n^{1-1/r}$,*

$$\Pr\Big[h \textit{ is } s\textit{-good}\Big] \geq \frac{1}{2}.$$

We postpone the proof of Lemma 8.3.4 to Section 8.4.

Corollary 8.3.5. *Let $\mathcal{P}$ be as in* (8.5) *and let $A : [n]^m \times [n]^m \to \{0, 1\}$ be its incidence matrix, that is $A(f, g) = 1 \iff (f, g) \in \mathcal{P}$. Then, $|\mathcal{P}| \geq \Omega(N)$ and every row and every column of A has at most $s^2(r!)$ ones.*

Proof. The first claim follows from Lemma 8.3.4 and the fact that $|\mathcal{P}|$ is at least the number of s-good mappings. If h is l-good, then there are at most $l(r!)$ mappings from $[m] \to [n]$ that complement h. Hence, every row of A has at most $s(r!)$ ones and by Proposition 8.3.3, every column of A has at most $s^2(r!)$ ones. □

With this, we can now prove Lemma 8.2.2.

of Lemma 8.2.2. Let $t = \Delta(H)$. By Lemma 8.3.1, H can be decomposed into $\chi_E(H) \leq 2rt$ matchings, which we denote by $\mathcal{F}_1, \dots, \mathcal{F}_{\chi_E(H)}$. Complete each $\mathcal{F}_i$ to a maximal family $\mathcal{M}_i$ of disjoint $2r$-subsets of $[n]$ in some arbitrary way. For each $\mathcal{M}_i$, let $\mathcal{P}_i$ be as in (8.5) and let $A_i : [n]^m \times [n]^m \to \{0, 1\}^n$ be its incidence matrix. Set to zero all the entries of A_i that correspond to a pair (f, g) covering a set in $\mathcal{M}_i \smallsetminus \mathcal{F}_i$. Let $B = A_1 + \cdots + A_{\chi_E(H)}$ and $A = (B + B^{\mathsf{T}})$. It follows from (8.4) and Proposition 8.3.2 that for each $x \in \{-1, 1\}^n$, we have

$$\Big\langle \sum_{i=1}^{\chi_E(H)} (A_i + A_i^{\mathsf{T}}) x^{\otimes m}, x^{\otimes m} \Big\rangle = 2 \sum_{i=1}^{\chi_E(H)} \frac{|\mathcal{P}_i|}{|\mathcal{M}_i|} \sum_{S \in \mathcal{F}_i} \prod_{j \in S} x_i. \tag{8.6}$$

Since all $\mathcal{M}_i$ are maximal, they have the same size, as do the $\mathcal{P}_i$. Hence, by Corollary 8.3.5, there exists a constant $c_r \in (0, 1]$ such that the right-hand side of (8.6) equals $(2c_r N/n)p_H(x)$. Let G be the graph with adjacency matrix A, allowing for parallel edges. Then G has degree at most $2ts^2(r!)$. It follows from Lemma 8.3.1 that G can be partitioned into $O_r(t)$ matchings. Since the adjacency matrix of a matching has unit norm, we get that $\|A\| \leq O_r(t)$. □

8.4 Proof of the generalized birthday paradox.

For the proof of Lemma 8.3.4, we use a standard Poisson approximation result for "balls and bins" problems [MU05, Theorem 5.10]. A discrete Poisson random variable Y with expectation μ is nonnegative, integer valued, and has probability density function

$$\Pr[Y = \ell] = \frac{e^{-\mu}\mu^{\ell}}{\ell!}, \qquad \forall \ell = 0, 1, 2, \ldots \tag{8.7}$$

Proposition 8.4.1. *If X, Y are independent Poisson random variables with expectations μ_X, μ_Y, respectively, then $X + Y$ is a Poisson random variable with expectation $\mu_X + \mu_Y$.*

Lemma 8.4.2. *Let h be a uniformly distributed map from $[m]$ to $[n]$. For each $i \in [n]$, let $X_i = |h^{-1}(i)|$ and let $\mathbf{X} = (X_i)_{i\in[n]}$. Let $\mathbf{Y} = (Y_i)_{i\in[n]}$ be a vector of independent Poisson random variables with expectation m/n. Then, for any nonnegative function $\Phi : (\mathbb{N} \cup \{0\})^n \to \mathbb{R}_+$ such that $\mathbb{E}[\Phi(\mathbf{X})]$ decreases or increases monotonically with m, we have*

$$\mathbb{E}[\Phi(\mathbf{X})] \leq 2\mathbb{E}[\Phi(\mathbf{Y})].$$

of Lemma 8.3.4. Let $C_r > 0$ be a parameter depending only on r to be set later. Let $\mu = C_r m/n = C_r n^{-1/r}$ and assume that $n \geq n_0(r) := 4(C_r r)^r$. For h a random map as in Lemma 8.4.2, we begin by lower bounding the probability of the event that $\phi(h) \geq 1$. Recall that this occurs if there exists an $S \in \mathcal{M}$ and an r-subset $T \in \binom{S}{r}$ such that $T \subseteq \mathrm{im}(h)$. Let $\mathbf{X}$ be as in Lemma 8.4.2. Let $\psi : (\mathbb{N} \cup \{0\})^n \to \{0, 1\}$ be the function

$$\psi(x) = \prod_{S \in \mathcal{M}} \prod_{T \in \binom{S}{r}} \Big(1 - \prod_{i \in T} 1_{\geq 1}(x_i)\Big).$$

Then $\psi(\mathbf{X}) = 1$ if $\phi(h) = 0$ and $\psi(\mathbf{X})$ decreases monotonically with m. Hence, for $\mathbf{Y}$ a Poisson random vector as in Lemma 8.4.2, we have

$$\begin{aligned} \Pr[\phi(h) = 0] &= \mathbb{E}[\psi(\mathbf{X})] \\ &\leq 2\mathbb{E}[\psi(\mathbf{Y})] \\ &= 2 \prod_{S \in \mathcal{M}} \mathbb{E}\Big[\prod_{T \in \binom{S}{r}} \Big(1 - \prod_{i \in T} 1_{\geq 1}(Y_i)\Big)\Big], \end{aligned} \tag{8.8}$$

where in the last line we used the fact that since the sets $S \in \mathcal{M}$ are disjoint, the random variables

$$\prod_{T \in \binom{S}{r}} \Big(1 - \prod_{i \in T} 1_{\geq 1}(Y_i)\Big)$$

are independent. The random variables $1_{\geq 1}(Y_i)$, $i \in S$, are independent Bernoullis that are zero with probability $e^{-\mu}$. The expectation in (8.8) equals the probability that these random variables form a string of Hamming weight strictly less than r. Using that $n \geq 4(C_r r)^r$ and the fact that $1 - x \leq \exp(-x) \leq 1 - x + x^2/2$ when $x > 0$, this probability is at most

$$1 - \Pr[\forall i \in T \; 1_{\geq 1}(Y_i) = 1] = 1 - (1 - e^{-\mu})^r \leq 1 - (\mu(1 - \mu/2))^r \leq 1 - \frac{C_r^r}{en} \leq \exp\left(-\frac{C_r^r}{en}\right)$$

where $T \subset S$ is some fixed subset of size r. Hence, since $\mathcal{M}$ is maximal, the above and (8.8) give

$$\Pr[\phi(h) = 0] \leq 2\exp\left(-\frac{C_r^r|\mathcal{M}|}{en}\right) \leq 2\exp\left(-\frac{C_r^r\lfloor n/r \rfloor}{en}\right) \leq 2\exp\left(-\frac{C_r^r}{2er}\right). \tag{8.9}$$

Set $C_r = (6er)^{1/r}$, then the above right-hand side is at most 1/4. Next, we upper bound the probability that $\phi(h) \geq s = 200 \cdot 4^r$. Define $\chi : (\mathbb{N} \cup \{0\})^n \to \mathbb{R}_+$ by

$$\chi(x) = \sum_{S \in \mathcal{M}} \sum_{T \in \binom{S}{r}} \prod_{i \in T} x_i.$$

Then, $\phi(h) = \chi(\mathbf{X})$. Moreover, $\mathbb{E}[\chi(\mathbf{X})]$ increases monotonically with m. It thus follows from Lemma 8.4.2 that

$$\begin{aligned}\mathbb{E}[\phi(h)] \leq 2\mathbb{E}[\chi(\mathbf{Y})] = 2 \sum_{S \in \mathcal{M}} \sum_{T \in \binom{S}{r}} \prod_{i \in T} \mathbb{E}[Y_i] \\ \leq 2|\mathcal{M}| \binom{2r}{r} \left(\frac{m}{n}\right)^r \leq 2 \cdot \frac{n}{r} \cdot 4^r \cdot (6er) n^{-1} \leq 50 \cdot 4^r.\end{aligned}$$

where in the second line we used the fact that the Y_i are independent. By Markov's inequality, $\Pr[\phi(h) > 200 \cdot 4^r] \leq \frac{1}{4}$. With (8.9), we get that h is s-good with probability at least 1/2. □

8.5 Random differences in Szemerédi's Theorem

In this section we prove Theorem 8.1.3. We first consider a slightly different random model where we form a random multiset D_k of size k by repeatedly sampling a uniformly random element from $\mathbb{Z}/N\mathbb{Z}$. We will need the following equivalent formulation of Szemerédi's Theorem due to Varnavides [Var59] (see [Tao07, Theorem 4.8] for this exact formulation).

Proposition 8.5.1. *For every $\ell \in \mathbb{N}, \alpha \in (0, 1]$ there exists $N_1(\ell, \alpha), \varepsilon(\ell, \alpha)$ such that for every $N \geq N_1(\ell, \alpha)$, the following holds. Every subset $A \subseteq \mathbb{Z}/N\mathbb{Z}$ of size at least αN contains an $\varepsilon(\ell, \alpha)$-fraction of all $\ell + 1$ term arithmetic progressions in $\mathbb{Z}/N\mathbb{Z}$, that is,*

$$\mathbb{E}_{x \in \mathbb{Z}/N\mathbb{Z}, y \in \mathbb{Z}/N\mathbb{Z} \setminus \{0\}}[1_A(x) 1_A(x + y) \ldots 1_A(x + \ell y)] \geq \varepsilon(\ell, \alpha).$$

Proposition 8.5.2. *For all $\ell \in \mathbb{N}, \alpha \in (0,1]$ there exists $N_1(\ell,\alpha) \in \mathbb{N}$ such that for every $N > N_1(\ell,\alpha)$ the following holds. Let $k \geq \omega(N^{1-1/\lceil(\ell+1)/2\rceil}\log N)$ and let D be a random multiset of size k obtained by sampling k times independently and uniformly at random from $\mathbb{Z}/N\mathbb{Z} \setminus \{0\}$. Then, with probability $1-o(1)$, every subset $A \subseteq \mathbb{Z}/N\mathbb{Z}$ of size at least αN contains a proper arithmetic progression of length $\ell+1$ with common difference in D.*

Proof. We will arrive at a contradiction assuming that the statement is false. Let $\Gamma = \mathbb{Z}/N\mathbb{Z}$. For $f : \Gamma \to \mathbb{R}$ and $y \in \Gamma \setminus \{0\}$, define

$$\phi_y(f) = \mathbb{E}_{x\in\Gamma}[f(x)f(x+y)\dots f(x+\ell y)],$$

which is a degree $\ell+1$ polynomial over the variables $(f(x))_{x\in\Gamma}$. For a multiset $S \subseteq \Gamma \setminus \{0\}$, define

$$\Lambda_S(f) = \frac{1}{|S|}\sum_{y\in S}\phi_y(f).$$

If $f = 1_A$, then this counts the fraction of proper $(\ell+1)$-term APs with common difference in S that lie completely in A. Note that $\mathbb{E}_D[\Lambda_D(f)] = \Lambda_{\Gamma\setminus\{0\}}(f)$.

Let $N_1(\ell,\alpha)$ and $\varepsilon(\ell,\alpha)$ be as in Proposition 8.5.1. Suppose that with a constant probability, there is a subset $A \subseteq \Gamma$ of size at least αN with no proper $(\ell+1)$-term APs whose common difference lies in D. Then,

$$\Pr_D\left[\inf_{A:|A|\geq\alpha N}\Lambda_D(1_A) = 0\right] = \Omega(1).$$

By Proposition 8.5.1, for every $A \subseteq \Gamma$ of size at least αN, we have that $\Lambda_{\Gamma\setminus\{0\}}(1_A) \geq \varepsilon$. We are going to apply a standard symmetrization trick to establish a connection with

Gaussian width. Let D' be an independent copy of D. Then,

$$\begin{aligned}\varepsilon &\lesssim \mathbb{E}_D\Big[\sup_{A:|A|\geq\alpha N}\Big|\Lambda_D(1_A)-\Lambda_{\Gamma\setminus\{0\}}(1_A)\Big|\Big]\\ &= \mathbb{E}_D\Big[\sup_{A:|A|\geq\alpha N}\Big|\Lambda_D(1_A)-\mathbb{E}_{D'}[\Lambda_{D'}(1_A)]\Big|\Big]\\ &\leq \mathbb{E}_{D,D'}\Big[\sup_{A:|A|\geq\alpha N}\Big|\Lambda_D(1_A)-\Lambda_{D'}(1_A)\Big|\Big]\\ &= \mathbb{E}_{y_1,\ldots,y_k,y_1',\ldots,y_k'\in\Gamma\setminus\{0\}}\Big[\sup_{A:|A|\geq\alpha N}\Big|\frac{1}{k}\sum_{i=1}^{k}\phi_{y_i}(1_A)-\phi_{y_i'}(1_A)\Big|\Big]\end{aligned}$$

Observe that for i.i.d. random $y,y'\in\Gamma\setminus\{0\}$, the random variable $\phi_y(1_A)-\phi_{y'}(1_A)$ is symmetric in the sense that it has the same distribution as its negation. Let $\sigma_1,\ldots,\sigma_k$ be independent uniformly distributed $\{-1,1\}$-valued random variables. Then it follows from the above that

$$\begin{aligned}\varepsilon &\lesssim \mathbb{E}_{y_1,\ldots,y_k\ \ y_1',\ldots,y_k'\in\Gamma\setminus\{0\}}\mathbb{E}_\sigma\Big[\sup_{A:|A|\geq\alpha N}\Big|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\left(\phi_{y_i}(1_A)-\phi_{y_i'}(1_A)\right)\Big|\Big]\\ &\leq 2\mathbb{E}_{y_1,\ldots,y_k\in\Gamma\setminus\{0\}}\mathbb{E}_\sigma\Big[\sup_{A:|A|\geq\alpha N}\Big|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\phi_{y_i}(1_A)\Big|\Big].\end{aligned}$$

Let us fix $y_1,\ldots,y_k\in\Gamma\setminus\{0\}$. Each ϕ_{y_i} can be written as $\phi_{y_i}=N^{-1}p_{H_i}$ (as in (8.3)) where H_i is the hypergraph on Γ whose edges are given by $(\ell+1)$ term arithmetic progressions with common difference y_i. The maximum degree of H_i is $O(\ell)$. This is because each such AP $(x+ty_i)_{0\leq t\leq\ell}$ intersects another AP $(x'+t'y_i)_{0\leq t'\leq\ell}$ iff $x-x'=(t'-t)y_i$; so there are only $O(\ell)$ such x' for a given x. Let $g_1,\ldots,g_k$ be independent $N(0,1)$ random variables. Then we can bound

$$\begin{aligned}\mathbb{E}_\sigma\Big[\sup_{A:|A|\geq\alpha N}\Big|\frac{1}{k}\sum_{i=1}^{k}\sigma_i\phi_{y_i}(1_A)\Big|\Big] &\lesssim \frac{1}{k}\mathbb{E}_g\Big[\sup_{A}\Big|\sum_{i=1}^{k}g_i\phi_{y_i}(1_A)\Big|\Big]\\ &= \frac{1}{Nk}\mathbb{E}_g\Big[\sup_{A}\Big|\sum_{i=1}^{k}g_ip_{H_i}(1_A)\Big|\Big]\\ &\lesssim_\ell \frac{1}{k}\sqrt{kN^{1-1/\lceil(\ell+1)/2\rceil}\log N},\end{aligned}$$

where the last line follows directly from Theorem 8.1.1. Thus we get $k \lesssim_\ell N^{1-1/\lceil(\ell+1)/2\rceil}\log N$ which is a contradiction. □

We will the need following simple fact that conditioning on a high probability event will not change the probability of any event by much.

Lemma 8.5.3. *Let A, E be some events in some probability space. If $\Pr[E] \geq 1-\varepsilon$ then $|\Pr[A|E]-\Pr[A]| \leq 2\varepsilon/(1-\varepsilon)$.*

Proof.

$$\begin{aligned}|\Pr[A|E]-\Pr[A]| &= \left|\frac{\Pr[A\cap E]}{\Pr[E]}-\Pr[A]\right|\\ &= \left|\frac{1}{\Pr[E]}\left(\Pr[A]+\Pr[E]-\Pr[A\cup E]\right)-\Pr[A]\right|\\ &\leq \left|\Pr[A]\left(\frac{1}{\Pr[E]}-1\right)\right| + \left|1-\frac{\Pr[A\cup E]}{\Pr[E]}\right| \leq \frac{2\varepsilon}{1-\varepsilon}.\end{aligned}$$

□

of Theorem 8.1.3. Let D_k be a random subset of $\mathbb{Z}/N\mathbb{Z} \smallsetminus \{0\}$ of size at most k, formed by sampling a uniformly random element from $\mathbb{Z}/N\mathbb{Z}$ for k times. Let $D_p = [\mathbb{Z}/N\mathbb{Z}\backslash\{0\}]_p$ be a random subset of $\mathbb{Z}/N\mathbb{Z}\backslash\{0\}$ formed by including each element with probability p independently. We claim that if D_k is ℓ-intersective with probability $1-o(1)$, then D_p will also be ℓ-intersective with probability $1-o(1)$ when $p = 2k/N$ and $k = \omega_N(1)$.

Let $p = 2k/N$ and $k = \omega_N(1)$. Let E be the event that D_p has size at least k. By the Chernoff bound,

$$1 - \Pr[E] \leq \exp\left(-\mathrm{D}_{\mathrm{KL}}\left(\frac{p}{2}||p\right)N\right) \leq \exp(-\Omega(pN)) = o(1)$$

where D_{KL} is the Kullback-Leibler divergence. By Lemma 8.5.3, conditioning on E changes the probability of D_p being ℓ-intersective by $o(1)$. Conditioned on E, the probability that D_p is ℓ-intersective is at least the probability that D_k is ℓ-intersective. Indeed, both D_p and D_k, after conditioning on a given size reduce to the uniform distribution over all subsets of that size. Proposition 8.5.2 thus implies D_p is ℓ-intersective when $p = \omega(N^{-1/\lceil(\ell+1)/2\rceil}\log N)$. □

8.6 Upper tails for arithmetic progressions in random sets

Here we prove Theorem 8.1.4. Let $\Gamma = \mathbb{Z}/N\mathbb{Z}$. In the following we identify maps from a set S to $\mathbb{R}$ with vectors in $\mathbb{R}^S$. For $f : \Gamma \to \mathbb{R}$, define

$$\Lambda_k(f) = \sum_{a,b\in\Gamma,b\neq 0} f(a)f(a+b)f(a+2b)\cdots f(a+(k-1)b). \tag{8.10}$$

Observe that for a subset $A \subseteq \Gamma$, we have that $\Lambda_k(1_A)$ counts the number of proper k-term arithmetic progressions in A. Moreover, Λ_k is an N-variate polynomial of degree k. Recall that the gradient of a polynomial $p \in \mathbb{R}[x_1, \ldots, x_n]$ is the mapping $\nabla p : \mathbb{R}^n \to \mathbb{R}^n$ whose ith coordinate is given by $(\nabla p)_i = (\partial p/\partial x_i)(x)$. The proof of Theorem 8.1.4 follows from a simple corollary of Theorem 8.1.1 and one of the main results of [BGSZ18]. For the corollary, we consider polynomial mappings given by gradients of polynomials of the form (8.3).

Corollary 8.6.1. *Let n, t, d be positive integers. Let $H = ([n], E)$ be a $(d+1)$-hypergraph such that at most t edges are incident on any given pair of vertices. Then,*

$$\frac{1}{n} w\Big((\nabla p_H)(\{0,1\}^n)\Big) \lesssim_d t n^{1-\frac{1}{2\lceil d/2\rceil}} \sqrt{\log n}.$$

Proof. For each $i \in [n]$ let $H_i = ([n], E_i)$ be the d-hypergraph with edge set

$$E_i = \{e \setminus \{i\} : e \in E(H) \text{ and } i \in e\}.$$

The claim now follows from Theorem 8.1.1 as $p_{H_i} = (\nabla p_H)_i$ each H_i has degree at most t. □

Theorem 8.6.2 (Bhattacharya–Ganguly–Shao–Zhao)**.** *Let $k \geq 3$ be a fixed integer and let σ, τ be positive real numbers such that*

$$\frac{1}{N} w\Big(\nabla \Lambda_k(\{0,1\}^\Gamma)\Big) \lesssim N^{1-\sigma}(\log N)^\tau.$$

Let $p \in (0,1)$ be bounded away from 1 and let $\delta > 0$ be such that $\delta = O(1)$ and

$$\min\{\delta p^k, \delta^2 p\} \gtrsim N^{-\sigma/3}(\log N)^{1+\tau/3}.$$

Then,

$$\log \Pr[\Lambda_k(\Gamma_p) \geq (1+\delta)\mathbb{E}\Lambda_k(\Gamma_p)] = -\Big(1+o(1)\Big)\, \phi_p\Big((1+o(1))\delta\Big). \tag{8.11}$$

Moreover, provided $\delta p^k N^2 \to \infty$ and N is prime, we have

$$\phi_p(\delta) \asymp N \min\{\sqrt{\delta} p^{k/2} \log(1/p), \delta^2 p\}.$$

of Theorem 8.1.4. Let $H = (\Gamma, E)$ be the hypergraph whose edges are the (unordered) proper k-term arithmetic progressions in Γ. Then, accounting for the fact that Λ_k distinguishes between the same progression with step b run forward from a point a or backward from $a + (k-1)b$ and since N is prime, we have $2p_H = \Lambda_k$. We claim that every pair of distinct vertices appears in $O(k^2)$ edges. First note that H is 2-transitive, since for any two pairs of distinct vertices $(a, b), (c, d)$, the affine linear map $x \mapsto c(x-b)/(a-b) + d(x-a)/(b-a)$ sends a to c, b to d and preserves progressions. It follows that every pair of distinct vertices is contained in the same number of edges. Since each edge contains $\binom{k}{2}$ pairs, the claim follows by double-counting. By Corollary 8.6.1, we may thus set $\sigma = 1/(2\lceil (k-1)/2 \rceil)$ and $\tau = 1/2$ in Theorem 8.6.2 and it follows that for constant δ, the estimate (8.11) holds if

$$p^k \gtrsim \min\{\delta p^k, \delta^2 p\} \gtrsim N^{-\frac{1}{6\lceil (k-1)/2 \rceil}} (\log N)^{1+1/6}.$$

Taking kth roots now gives the claim. □

8.7 Proof of Lemma 8.1.5

In this section we give a proof Lemma 8.1.5. As explained in the proof of Theorem 8.1.1, it suffices to prove the statement when the coordinates of ψ are given by p_{H_i} (as in (8.3)) for d-uniform hypergraphs $H_1, \ldots, H_k$. Let Λ_{H_i} be a d-multilinear form such that $p_{H_i}(x) = \Lambda_{H_i}(x, x, \ldots, x)$. Let $g = (g_1, \ldots, g_k)$ be vector of independent standard Gaussians and $\varepsilon = (\varepsilon_1, \ldots, \varepsilon_k)$ be uniformly random in $\{-1, 1\}^k$.

Then,

$$\begin{aligned} w\Big(\psi(\{0,1\}^n)\Big) &= \mathbb{E}_g \sup_{x\in\{0,1\}^n} \left|\sum_{i=1}^k g_i p_{H_i}(x)\right| \\ &= \mathbb{E}_g \sup_{x\in\{0,1\}^n} \left|\sum_{i=1}^k g_i \Lambda_{H_i}(x,\dots,x)\right| \\ &\le \mathbb{E}_g n^{\sum_{i=1}^k 1/r_i} \left\|\sum_{i=1}^k g_i\Lambda_{H_i}\right\| \\ &= n\mathbb{E}_g\mathbb{E}_\varepsilon \left\|\sum_{i=1}^k \varepsilon_i g_i \Lambda_{H_i}\right\|, \end{aligned}$$

where in the last line we used that each g_i is symmetrically distributed, that is, g_i and $-g_i$ have the same distribution. By Jensen's inequality, the above expectation over ε is at most

$$\left(\mathbb{E}_\varepsilon \left\|\sum_{i=1}^k \varepsilon_i g_i \Lambda_{H_i}\right\|^p\right)^{1/p} \le T_p(\mathcal{L}^n_{r_1,\dots,r_s}) \left(\sum_{i-1}^k \|g_i\Lambda_{H_i}\|^p\right)^{1/p},$$

where the inequality follows from the definition of the type-p constant of $\mathcal{L}^n_{r_1,\dots,r_s}$. Hence,

$$\begin{aligned} w\Big(\psi(\{0,1\}^n)\Big) &\le n\mathbb{E}_g\ T_p(\mathcal{L}^n_{r_1,\dots,r_s}) \left(\sum_{i=1}^k \|g_i\Lambda_{H_i}\|^p\right)^{1/p} \\ &\le nT_p(\mathcal{L}^n_{r_1,\dots,r_s})\mathbb{E}_g \|g\|_{\ell_p} \max_i \|\Lambda_{H_i}\| \\ &\le nT_p(\mathcal{L}^n_{r_1,\dots,r_s})\, k^{1/p} \max_i \|\Lambda_{H_i}\|, \end{aligned}$$

where we used the fact that $\mathbb{E}_g \|g\|_{\ell_p} \le (\sum_{i=1}^k \mathbb{E}_{g_i}|g_i|^p)^{1/p} \le k^{1/p}(\mathbb{E}_{g_1}|g_1|^2)^{1/2} = k^{1/p}$. If H_i is a matching hypergraph, using Hölder's inequality, it is easy to see that $\|\Lambda_{H_i}\| \le 1$. If not, by Lemma 8.3.1, we can decompose H_i into $d\Delta(H_i)$ matchings and use triangle inequality to conclude that $\|\Lambda_{H_i}\| \le d\Delta(H_i)$ which gives the desired bound.

Chapter 9

Local codes for distributed storage

9.1 Introduction

The explosion in the volumes of data being stored online means that duplicating or triplicating data is not economically feasible. This has resulted in distributed storage systems employing erasure coding based schemes in order to ensure reliability with low storage overheads. In recent years Local Reconstruction Codes (LRCs) emerged as the codes of choice for many such scenarios and have been implemented in a number of large scale systems e.g., Microsoft Azure [HSX$^+$12] and Hadoop [SAP$^+$13].

Classical erasure correcting codes [MS77] guarantee that data can be recovered if a bounded number of codeword coordinates is erased. However recovering data typically involves accessing all surviving coordinates. By contrast, Local Reconstruction Codes[1] (LRCs) distinguish between the typical case when only a small number of codeword coordinates are erased (e.g., few machines in a data center fail) and a worst case when a larger number of coordinates might be unavailable, and guarantee that in the prior case recovery of individual coordinates can be accomplished in sub-linear time, without having to access all surviving symbols.

[1] The term local reconstruction codes is from [HSX$^+$12]. Essentially the same codes were called locally repairable codes in [PD14] and locally recoverable codes in [TB14]. Thankfully all names above abbreviate to LRCs.

LRCs are systematic linear codes, where encoding is a two stage process. In the first stage, h redundant heavy parity symbols are generated from k data symbols. Each heavy parity is a linear combination of all k data symbols. During the second stage, the $k + h$ symbols are partitioned into $\frac{k+h}{r-a}$ sets of size $r - a$ and each set is extended with a local parity symbols using an MDS code to form a *local group*. Encoding as above ensures that when at most a coordinates are erased, any missing coordinate can be recovered by accessing at most $r - a$ symbols. However, if a larger number of coordinates (that depends on h) is erased; then all missing symbols can be recovered by potentially accessing all remaining symbols.

Our description of LRC codes above is not complete. To specify a concrete code we need to fix coefficients in linear combinations that define h heavy and $\frac{k+h}{r-a} \cdot a$ local parities. Different choices of coefficients could lead to codes with different erasure correcting capabilities. The best we could hope for is to have an optimal choice of coefficients which ensures that our code can correct every pattern of erasures that is correctable for some setting of coefficients. Such codes always exist and are called Maximally Recoverable (MR) [CHL07, HCL07] LRCs.[2] Combinatorially, an (n, r, h, a, q)-LRC is maximally recoverable it if corrects every pattern of erasures that can be obtained by erasing a coordinates in each local group and up to h additional coordinates elsewhere. Explicit constructions of MR LRCs are available (e.g., [CK17]) for all ranges of parameters. Unfortunately, all known constructions require finite fields of very large size.

Encoding a linear code and decoding it from erasures involve matrix vector multiplication and linear equation solving respectively. Both of these require performing numerous finite field arithmetic operations. Having small finite fields results in faster encoding and decoding and thus improves the overall throughput of the system [PGM13, Section 2]. It is also desirable in practice to work over finite fields of

[2] Maximally recoverable LRCs are called Partial MDS (PMDS) in [Bla13, BHH13] and many follow up works.

characteristic 2. Obtaining MR LRCs over finite fields of minimal size is one of the central problems in the area of codes for distributed storage.

9.1.1 State of the art and our results

We now summarize what is known about the minimal field size of maximally recoverable local reconstruction codes with parameters n, r, a and h and first cover the easy cases.

- When $a = 0$, LRCs are equivalent to classical erasure correcting codes. In this case Reed Solomon codes are maximally recoverable, and they have a field size of roughly n, which is known to be optimal up to constant factors [Bal12].
- When $h \leq 1$, there are constructions of maximally recoverable LRCs over fields of size $O(r)$ [BHH13] which is optimal.
- When $r = a + 1$, codes in the local groups are necessarily simple repetition codes. MR LRCs can be obtained by starting with a Reed Solomon code of length n/r and repeating every coordinate r times. Thus the optimal field size is $\Theta(n/r)$.

This leaves us with the main case, when $a \geq 1$, $r \geq a + 2$, and $h \geq 2$. A number of constructions have been obtained [Bla13, BHH13, TPD16, GHJY14, HY16, GHK$^+$17, CK17, BPSY16, GYBS17]. The best constructions for the case of $h = 2$ are from [BPSY16] and require a field of size $O(a \cdot n)$. For most other settings of parameters the best families of MR LRCs are from [GYBS17]. They require fields of size

$$O\left(r \cdot n^{(a+1)h-1}\right) \quad \text{and} \quad O\left(\max\left(n/r, r^{h+a}\right)^{h}\right). \tag{9.1}$$

The first bound is typically better when $r = \Omega(n)$. The second bound is better when $r \ll n$. Both bounds require q to grow rapidly with the codeword length. We will now present our results.

Lower bound. The bounds in (9.1) exhibit code constructions but not any inherent limitations. In particular, up until our work it remained a possibility that codes over fields of size $O(n)$ could exist for all ranges of LRC parameters. We obtain the first superlinear lower bound on the field size of MR LRCs, prior to our work no superlinear lower bounds were known in any setting of parameters.

Theorem 9.1.1. *Let a and h be fixed constants while r may grow with n. Any maximally recoverable (n, r, h, a, q)-LRC with $g = n/r$ local groups must have:*

$$q \geq \Omega_{h,a}\left(n \cdot r^{\alpha}\right) \text{ where } \alpha = \frac{\min\left\{a, h - 2\lceil h/g \rceil\right\}}{\lceil h/g \rceil}. \tag{9.2}$$

The lower bound (9.2) simplifies as follows in some special cases:

- $g \geq h : q \geq \Omega_{h,a}\left(nr^{\min\{a,h-2\}}\right)$
- $g \leq h$, g divides h and $a \leq h - 2h/g : q \geq \Omega_{h,a}\left(n^{1+ah/g}\right)$
- $g \leq h$, g divides h and $a > h - 2h/g : q \geq \Omega_{h,a}\left(n^{g-1}\right)$.

Note that our lower bound is superlinear whenever r is growing with n except when $a = 0$ or $h = 2$ or $g = 2$ or $(g = 3, h = 4, a = 1)$. Even from a practical standpoint, r should be thought of as growing with n (like say $r = \sqrt{n}$). This is because if r is constant, the number of parity checks or redundant symbols $(an/r + h)$ will be linear in n, and applications of codes in distributed storage demand high rate codes.

When $a = 0$, MR LRCs reduce to MDS codes and so there are linear field size constructions (Reed-Solomon codes). When $h = 2$, we obtain a linear field size construction (Theorem 9.4.4). This leaves $g = 2$ and $(g = 3, h = 4, a = 1)$ as the only cases where we don't know if linear field size is enough for MR LRCs.

The parity check view of MR LRCs throws a different light on our lower bound. The parity check matrix of an MR (n, r, h, a, q)-LRC with $g = n/r$ local groups is an

$(ag + h) \times n$ matrix of the following form:

$$H = \left[\begin{array}{c|c|c|c} A_1 & 0 & \cdots & 0 \\ \hline 0 & A_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_g \\ \hline B_1 & B_2 & \cdots & B_g \end{array}\right]. \tag{9.3}$$

Here $A_1, A_2, \cdots, A_g$ are $a \times r$ matrices over $\mathbb{F}_q$, $B_1, B_2, \cdots, B_g$ are $h \times r$ matrices over $\mathbb{F}_q$. The rest of the matrix is filled with zeros. An erasure pattern with $ag+h$ erasures is correctable iff the corresponding minor in H is non-zero. Thinking of the entries of the matrices A_i, B_i as variables, every $(ag + h) \times (ag + h)$ minor of H is either identically zero or a non-zero polynomial in those variables. We call the minors with zero determinant as trivial and the rest as non-trivial. It turns out that the non-trivial minors of H in (9.3) are exactly those which are obtainable by selecting a columns in each local group and h additional columns anywhere. There exists an MR LRC over $\mathbb{F}_q$ with these parameters iff there exists an assignment of $\mathbb{F}_q$ values to these variables which makes all the non-trivial minors non-zero. It is easy to see that if we assign random values from a large enough finite field $\mathbb{F}_q$ (say $q \gg n^{ag+h}$) to the variables, by Schwartz-Zippel lemma, all the non-trivial minors will be non-zero with high probability. But this probabilistic argument can only work for very large fields. Seen this way, it seems very natural to ask what is the smallest field size required to make all the non-trivial minors non-zero given a matrix with some pattern of zeros.

Thus our lower bound shows that one needs super linear size fields to instantiate H to make all non-trivial minors non-zero. This is even more surprising when contrasted with a recent proof of the GM-MDS conjecture by Lovett [Lov18] and independently by Yildiz and Hassibi [YH18]. This states that a $k \times n$ matrix $(k \leq n)$ with some pattern of zeros such that every $k \times k$ minor is non-trivial can be instantiated with a field of size $q \leq n + k - 1$ to make every $k \times k$ minor non-zero.

Upper bounds (Code constructions). MR LRCs that are deployed in practice typically have a small constant number of global parities, typicially $h = 2, 3$ [HSX$^+$12]. Without explicit constructions, one has to search over assignments from a small field to variables in the parity check matrix (9.3) to find an assignment which makes all the non-trivial minors non-zero. This is prohibitively expensive even for small values of n and q that are deployed in practice. Note that for random assignments to work with high probability, the field should be very large. Keeping this in mind, we design explicit MR LRCs over small field size for $h \leq 3$.

- We obtain a family of MR $(n, r, h = 2, a, q)$-LRCs, where $q = O(n)$ for all settings of parameters. Prior to our work the best constructions [BPSY16] required q to be $O(a \cdot n)$ which in general may be up to quadratic in n. If we require that the field has characteristic two, we get such codes with $q = n^{1+o(1)}$.
- We obtain a family of MR $(n, r, h = 3, a, q)$-LRCs, where $q = O(n^3)$ for all settings of parameters. Prior to our work the best constructions (9.1) required q to be up to $n^{\Theta(a)}$ for some regimes. If we require that the field has characteristic two, we can get such codes with $q = n^{3+o(1)}$.
- Given our linear field size construction for $h = 2$ (and since the problem is trivial for $r = 2$), the setting $r = 3, a = 1, h = 3$ is the next smallest regime to investigate regarding the existence of MR LRCs over fields of near-linear size. We construct such MR LRCs with a field size of $n \cdot \exp(O(\sqrt{\log n}))$ by developing a new approach to LRC constructions based on elliptic curves and AP-free sets.

9.1.2 Our techniques

Similar to most earlier works in the area we represent LRC codes via their parity check matrices which look like (9.3). Such matrices H have size $(a \cdot g + h) \times n$ and a simple block structure. Columns are partitioned into r-sized local groups. For each local group there is a corresponding collection of a rows that impose MDS constraints on coordinates in the group, and have no support outside the group. Remaining h rows of H correspond to heavy parity symbols and carry arbitrary values.

To establish our lower bound when $g \geq h$, we start with a parity check matrix of an arbitrary maximally recoverable local reconstruction code. From it, we obtain a family of large mutually disjoint subsets $X_1, \ldots, X_g$ in the projective space $\mathbb{PF}_q^{h-1}$, such that no hyperplane in $\mathbb{PF}_q^{h-1}$ intersects h distinct sets among $X_1, \ldots, X_g$. For example when $a = 1$ and $h \geq 3$, the set X_i is all the pairwise differences of columns of B_i in (9.3) thought of as points in $\mathbb{PF}_q^{h-1}$. We then show that if q is too small, then a random hyperplane will intersect h distinct sets among $X_1, \ldots, X_g$ with positive probability, which gives the required lower bound. When $h > g$, each X_i will be a collection of subspaces in $\mathbb{F}_q^h$ of dimension roughly h/g such that any collection of g subspaces, one from each X_i, will span $\mathbb{F}_q^h$. Again we show that if q is too small, a random $(h-1)$-dimensional subspace will contain a subspace each from X_i with high probability. The proof is more intricate in this case, because we need to carefully calculate how subspaces inside each X_i intersect with each other.

We now explain the main ideas behind our constructions. An LRC is MR if any subset of columns of H (as in (9.3)) that can be obtained by selecting a columns from each local group and then h more has full rank. Suppose all h additional columns are selected from distinct local groups. In this case showing that some $ag+h$ columns are independent easily reduces to showing that a certain $(ah+h) \times (ah+h)$ determinant is non-zero. An important algebraic identity that underlies our constructions for $h = 2$ and $h = 3$ reduces such determinants to much smaller $h \times h$ determinants of determinants in the entries of H. A special case of this identity when $h = 2$ and matrices are Vandermonde type appears in [BPSY16]. In addition to that we utilize various properties of finite fields such as the structure of multiplicative sub-groups and field extensions. In the case of $h = 3$, we deviate from most existing constructions of MR LRCs in that we do not use linearized constraints (x, x^q, x^{q^2}) or Vandermonde constraints (x, x^2, x^3) and instead rely on Cauchy matrices [LN83] to specify heavy parities.

Our construction of MR $(n, r = 3, h = 3, a = 1, q)$-LRCs is technically disjoint from our other results. We observe that in this narrow case, MR LRCs are equivalent to subsets A of the projective plane $\mathbb{PF}_q^2$, where A is partitioned in to triples $A = \sqcup_i \{a_i, b_i, c_i\}$ so that some three elements of A are collinear if and only if they constitute one of the triples $\{a_i, b_i, c_i\}$ in the partition. Moreover, minimizing the field size of maximally recoverable local reconstruction codes is in fact equivalent to maximizing the cardinality of such sets A. By considering all the $q+1$ lines through an arbitrary point of A, it is easy to see that $|A| \leq q+3$. We construct sets A with size $|A| \geq q^{1-o(1)}$. For our construction we start with an elliptic curve E over $\mathbb{F}_q$ such that the group of $\mathbb{F}_q$-rational points, $E(\mathbb{F}_q)$, is a cyclic group of size $\Omega(q)$. We observe that three points of $E(\mathbb{F}_q)$ are collinear if only and only if they sum to zero in the group. We then select a large AP-free set of points of $E(\mathbb{F}_q)$ using the classical construction of Behrend [Beh46] and complete these points to desired triples.

9.1.3 Related work

The first family of codes with locality for applications in storage comes from [HCL07, CHL07]. These papers also introduced the concept of maximal recoverability in a certain restricted setting. The work of [GHSY12] introduced a formal definition of local recovery and focused on codes that guarantee local recovery for a single failure. For this simple setting they were able to show that optimal codes must have a certain natural topology, e.g., codeword coordinates have to be arranged in groups where each group has a local parity. While [GHSY12] focused on systematic codes that provide local recovery for information symbols, [PD14] considered codes that provide locality for all symbols and defined local reconstruction codes. In parallel works maximally recoverable LRCs have been studied in [BHH13, Bla13]. Construction of local reconstruction codes with optimal distance over fields of linear size has been given in [TB14]. (Note that distance optimality is a much weaker property than maximal recoverability, e.g., when $a + h < r$ it only requires all patterns of size $a + h$ to be correctable, while MR property requires lots of very large patterns including some of size $(a + 1)h$ to be correctable.)

Maximal recoverability can be defined with respect to more general topologies then just local reconstruction codes [GHJY14]. The first lower bound for the field size of MR codes in any topology was recently given in [GHK$^+$17]. This line of work was continued in [KLR17] where nearly matching upper and lower bounds were obtained. The topology considered in [GHK$^+$17, KLR17] is a grid-like topology, where codewords form a codimension one subspace of tensor product codes, i.e., codewords are matrices, there is one heavy parity symbol, and each row / column constitutes a local group with one redundant symbol.

Finally, there are few other models of erasure correcting codes that provide efficient recovery in typical failure scenarios. These include regenerating codes [DGW$^+$10, WTB17, YB17, GW16] that optimize bandwidth consumed during repair rather than the number of coordinates (machines) accessed during repair; locally decodable codes [Yek12] that guarantee sub-linear time recovery of information coordinates even when a constant fraction of coordinates are erased; and SD codes [Bla13, BPSY16] that correct a certain subset of failure patterns correctable by MR LRCs.

9.1.4 Organization

In Section 9.2, we setup our notation, give formal definitions of local reconstruction codes and maximal recoverability, and establish some basic facts about MR LRCs. In Section 9.3, we present our main lower bound on the alphabet size. In Section 9.4, we introduce the determinantal identity and use it to give a construction of MR LRCs with two heavy parity symbols over fields of linear size. In Section 9.5, we get explicit MR codes over fields of cubic size. Finally, in Section 9.6, we focus on the narrow case of codes with three heavy parities, one parity per local group, and local groups of size three. We introduce the machinery of elliptic curves and AP free sets and employ it to obtain maximally recoverable codes over fields of nearly linear size. We conclude by listing some open problems in Section 9.7. Appendix contains some missing proofs and proofs of the determinantal identities.

9.2 Preliminaries

We begin by summarizing few standard facts about erasure correcting codes [MS77].

- $[n, k, d]_q$ denotes a linear code (subspace) of dimension k, codeword length n, and Hamming distance d over a field $\mathbb{F}_q$. We often write $[n, k, d]$ or $[n, k]$ instead of $[n, k, d]_q$ when the left out parameters are not important.
- An $[n, k, d]$ code is called Maximum Distance Separable (MDS) if $d = n - k + 1$.
- A linear $[n, k, d]_q$ code C can be specified via its parity check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$, where $C = \{x \in \mathbb{F}_q^n \mid H \cdot x = 0\}$. A code C is MDS iff every $(n - k) \times (n - k)$ minor of H is full rank.
- Let C be an $[n, k]$ code with a parity check matrix $H \in \mathbb{F}^{(n-k)\times n}$. Let E be a subset of the coordinates of C. If coordinates in E are erased; then they can be recovered (corrected) iff the matrix H restricted to coordinates in E has full rank.

We proceed to formally define local reconstruction codes.

Definition 9.2.1. *Let $r \mid n$, $a < r$, and h be integers and q be a prime power. Let $g = \frac{n}{r}$. Assume $h \leq n - ag$ and let $k = n - ga - h$. A linear $[n, k]$ code C over a field $\mathbb{F}_q$ is an (n, r, h, a, q)-LRC if for each $i \in [g]$, restricting C to coordinates in $\{r(i-1)+1, \ldots, ri\}$, yields a maximum distance separable code with parameters $[r, r-a, a+1]$.*

Let $[n] = \{1, \ldots, n\}$. In what follows we refer to subsets $\{r(i-1)+1, \ldots, ri\}$ of the set of code coordinates $[n]$ as local groups. There are g local groups and each such group has size r. It is immediate from the Definition 9.2.1 that every (n, r, h, a, q)-LRC admits a parity check matrix H of the following form

$$H = \left[\begin{array}{c|c|c|c} A_1 & 0 & \cdots & 0 \\ \hline 0 & A_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_g \\ \hline B_1 & B_2 & \cdots & B_g \end{array}\right]. \tag{9.4}$$

Here $A_1, A_2, \cdots, A_g$ are $a \times r$ matrices over $\mathbb{F}_q$, $B_1, B_2, \cdots, B_g$ are $h \times r$ matrices over $\mathbb{F}_q$. The rest of the matrix is filled with zeros. Every matrix $\{A_i\}_{i \in [g]}$ is a parity check matrix of an $[r, r-a, a+1]$ MDS code. The bottom h rows of H serve to increase the code co-dimension from ag to $ag + h$. Conversely, every matrix H as in (9.4), where $\operatorname{rank}(H) = ag + h$, and every $a \times a$ minor in each $\{A_i\}_{i \in [g]}$ has full rank, defines an (n, r, h, a, q)-LRC.

Definition 9.2.2. [3] *Let C be an arbitrary (n, r, h, a, q)-local reconstruction code. We say that C is maximally recoverable if for any set $E \subseteq [n]$, $|E| = ga + h$, where E is obtained by selecting a coordinates from each of g local groups and then h more coordinates arbitrarily; E is correctable by the code C.*

The term maximally recoverable code is justified by the following observation (e.g., [GHJY14]): if an erasure pattern cannot be obtained via the process detailed in the Definition 9.2.2; then it cannot be corrected by any linear code whose parity check matrix has the shape (9.4). Thus MR codes provide the strongest possible reliability guarantees given the locality constraints defining the shape of the parity check matrix.

Existence of MR LRCs can be established non-explicitly [GHJY14] (i.e., by setting the non-zero entries in the matrix (9.4) at random in a large finite field and then analyzing the properties of the resulting code). There are also multiple explicit constructions available [CK17, GHJY14, GYBS17]. The key challenge in this line of work is to determine the minimal size of finite fields where such codes exist. In practice one is naturally mostly interested in fields of characteristic two.

Notation: We use $A \gtrsim B$ to denote $A = \Omega(B)$ and $A \lesssim B$ to denote $A = O(B)$. We use $A = O_\ell(B)$ and $A = \Omega_\ell(B)$ to denote that the hidden constants can depend on some parameter ℓ but independent of other parameters.

9.3 The lower bound

In this Section we prove Theorem 9.1.1 which gives a lower bound on the field size of maximally recoverable local reconstruction codes. We break up the proof of The-

[3] Alternatively, one could define MR LRCs is as follows. Consider a matrix (9.4). Each way of fixing non-zero entries in (9.4) gives rise to (instantiates) a linear code. An instantiation is MR if it corrects all erasure patterns that are correctable for some other instantiation. It can be shown that under such definition and the minor technical assumption of $h \leq \frac{n}{r} \cdot (r-a) - \max\left\{\frac{n}{r}, r-a\right\}$ local codes have to be MDS [GHK$^+$17, Proposition 4] as required in Definition 9.2.1.

orem 9.1.1 into two cases based on $g \geq h$ and $g < h$ and prove the two cases in Corollary 9.3.6 and Proposition 9.3.7 respectively. Though the underlying ideas in the lower bound for both the cases are very similar, the $g \geq h$ case is simpler and conveys all the main conceptual ideas. So we will prove this case first.

9.3.1 Lower bound when $g \geq h$

A code is MR if it corrects every erasure pattern that can be obtained by erasing a symbols per local group, and then h more. Note that if some local group carries at most a erasures; then it can be immediately corrected using only the properties of the local MDS code. Thus we never need to consider erasure patterns spread across more than h groups. Our lower bound does not use all the properties of MR LRCs, but only relies on code's ability to correct all patterns obtained by erasing $a + h$ elements in a single group as well as all patterns obtained by erasing exactly $a + 1$ coordinates in some h local groups. Note that here we use the fact that the number of local groups g is at least h.

The lower bound is obtained by turning a parity check matrix of an MR (n, r, h, a, q)-LRC into a large collection of points (of size $\approx nr^a$ when $a \leq h - 2$) in the projective space $\mathbb{PF}_q^{h-1}$, partitioned into g equal parts $X_1, \ldots, X_g$, such that no hyperplane can intersect h distinct sets in $\{X_j\}_{j\in[g]}$. For example when $a = 1$ and $h \geq 3$, the set X_i is all the pairwise differences of columns of B_i in (9.4) thought of as points in $\mathbb{PF}_q^{h-1}$ and so $|X_i| = \binom{r}{2}$. In Lemma 9.3.1, we prove the size of such a collection can be at most $O(q)$ which implies the required lower bound. We will start by proving Lemma 9.3.1.

Lemma 9.3.1. *Let $X_1, \ldots, X_g \subseteq \mathbb{PF}_q^d$ be mutually disjoint subsets of size t with $g \geq d + 1$. If*

$$q < \left(\frac{g}{d} - 1\right) t - 4 \tag{9.5}$$

then there exists a hyperplane H in $\mathbb{PF}_q^d$ which intersects $d+1$ distinct subsets among $X_1, \cdots, X_g$.

Proof. We will show that a random hyperplane will intersect $d+1$ distinct subsets among $X_1, \ldots, X_g$ with positive probability if $q < \left(\frac{g}{d} - 1\right) t - 4$. Choose a uniformly random hyperplane H in $\mathbb{PF}_q^d$. Fix some $i \in [g]$, we will first lower bound the probability that H intersects X_i. Let the random variable $Z = |H \cap X_i|$. Since a hyperplane contains $|\mathbb{PF}_q^{d-1}|$ points,

$$\mathbb{E}[Z] = \frac{|\mathbb{PF}_q^{d-1}|}{|\mathbb{PF}_q^d|} t.$$

We can also estimate the second moment as follows:

$$\begin{aligned}\mathbb{E}[Z^2] &= \mathbb{E}[Z] + \sum_{p,p' \in X_i, p \neq p'} \Pr[p, p' \in H] \\ &= \mathbb{E}[Z] + t(t-1)\frac{|\mathbb{PF}_q^{d-2}|}{|\mathbb{PF}_q^d|}\end{aligned}$$

where we used the fact that the number of hyperplanes containing two fixed distinct points is $|\mathbb{PF}_q^{d-2}|$. Now we can lower bound $\Pr[Z > 0]$ as:

$$\Pr[Z > 0] \geq \frac{\mathbb{E}[Z]^2}{\mathbb{E}[Z^2]} \geq \frac{t/q}{1 + t/q}(1 - 1/q^d)^2.$$

Since $X_1, \ldots, X_g$ are mutually disjoint subsets of $\mathbb{PF}_q^d$ of size t, $gt \leq |\mathbb{PF}_q^d| \leq (d+1)q^d$. Therefore

$$\Pr[H \cap X_i \neq \phi] = \Pr[Z > 0] \geq \frac{t}{t+q}\left(1 - \frac{2}{q^d}\right) \geq \frac{t}{t+q}\left(1 - \frac{2(d+1)}{gt}\right)$$

By linearity of expectation, a random hyperplane H intersects $\geq g \cdot \frac{t}{t+q}\left(1 - \frac{2(d+1)}{gt}\right)$ sets among $X_1, \ldots, X_g$ in expectation. Therefore if $\frac{gt}{(q+t)}\left(1 - \frac{2(d+1)}{gt}\right) > d$, there exists a hyperplane which intersects $d+1$ distinct subsets among $X_1, \ldots, X_g$. Rearranging this inequality, such a hyperplane exists whenever $q < \left(\frac{g}{d} - 1\right) t - \frac{2(d+1)}{d}$. □

We are now ready to prove the lower bound. We will first prove a lower bound under the assumption that $a+2 \leq h$. Later in Proposition 9.3.5, we generalize our argument to take care of the case when $h < a+2$.

Proposition 9.3.2. *When $a+2 \leq h \leq n/r$, any maximally recoverable (n, r, h, a, q)-local reconstruction code must have*

$$q \geq \left(\frac{n/r}{h-1} - 1\right) \cdot \binom{r}{a+1} - 4 \tag{9.6}$$

Proof. It might be helpful to the reader to think of the $a = 1$ case through out the proof, as things get simpler. When $a = 1$, wlog, one can assume that the entries of the matrices A_i in (9.7) (which will have only one row) are all 1's.

Consider an arbitrary maximally recoverable (n, r, h, a, q)-LRC C with $g = \frac{n}{r}$ local groups. According to the discussion in Section 9.2 the code C admits a parity check matrix of the shape

$$\left[\begin{array}{c|c|c|c} A_1 & 0 & \cdots & 0 \\ \hline 0 & A_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_g \\ \hline B_1 & B_2 & \cdots & B_g \end{array}\right]. \tag{9.7}$$

Here $A_1, A_2, \cdots, A_g$ are $a \times r$ matrices over $\mathbb{F}_q$, $B_1, B_2, \cdots, B_g$ are $h \times r$ matrices over $\mathbb{F}_q$. The rest of the matrix is filled with zeros. Every $a \times a$ minor in each matrix $\{A_i\}_{i \in [g]}$ has full rank. So for every subset $S \subset [r]$ of size $|S| = a+1$, $A_i(S)$ is an $a \times (a+1)$ matrix of full rank. Let $A_i(S)^{\perp} \in \mathbb{F}_q^{a+1}$ be a non-zero vector orthogonal to the row space of $A_i(S)$ i.e. $A_i(S)A_i(S)^{\perp} = 0$. Note that $A_i(S)^{\perp}$ is unique upto scaling. For $i \in [g]$ and each subset $S \subseteq [r]$ of size $|S| = a+1$, define $p_{i,S} \in \mathbb{F}_q^h$ as [4]

$$p_{i,S} = B_i(S) A_i(S)^\perp.$$

The MR property implies that any subset of columns of the parity check matrix (9.7) which can be obtained by picking a columns in each local group and h arbitrary additional columns is full rank. We will use this property to make two claims about the vectors $\{p_{i,S}\}$.

Claim 9.3.3. *For every distinct $\ell_1, \cdots, \ell_h \in [g]$ and subsets $S_1, \cdots, S_h \subseteq [r]$ of size $a+1$ each, the $h \times h$ matrix $[p_{\ell_1,S_1}, \cdots, p_{\ell_h,S_h}]$ is full rank.*

Proof. Consider the following matrix equation:

$$\left[\begin{array}{c|c|c|c} A_{\ell_1}(S_1) & 0 & \cdots & 0 \\ \hline 0 & A_{\ell_2}(S_2) & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_{\ell_h}(S_h) \\ \hline B_{\ell_1}(S_1) & B_{\ell_2}(S_2) & \cdots & B_{\ell_h}(S_h) \end{array}\right] \left[\begin{array}{c|c|c|c} A_{\ell_1}(S_1)^\perp & 0 & \cdots & 0 \\ \hline 0 & A_{\ell_2}(S_2)^\perp & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_{\ell_h}(S_h)^\perp \end{array}\right]$$

$$= \left[\begin{array}{c|c|c|c} 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & 0 \\ \hline p_{\ell_1,S_1} & p_{\ell_2,S_2} & \cdots & p_{\ell_h,S_h} \end{array}\right].$$

Let us denote the matrices which occur in the above equation as M_1, M_2, M_3 respectively so that the above equation becomes $M_1 M_2 = M_3$. By MR property, when we erase the coordinates corresponding to $S_1, \cdots, S_h$ in groups $\ell_1, \cdots, \ell_h$ respectively,

[4] When $a = 1$, one can take $A_i(S)^\perp = \begin{bmatrix} 1 \\ -1 \end{bmatrix}$ and so $p_{i,S} = B_i(j) - B_i(j')$ where $S = \{j, j'\}$; therefore $\{p_{i,S} : |S| = a+1\}$ is just the set of all pairwise differences of columns of B_i.

the resulting erasure pattern is correctable. This implies that M_1 has full rank. Also M_2 has full column rank because its columns are non-zero and have disjoint support. Therefore M_3 should have full rank which implies that $[p_{\ell_1,S_1},\cdots,p_{\ell_h,S_h}]$ is full rank. $\square$

In particular the vectors $p_{i,S}$ are nonzero for every $i\in[g]$ and $S\in\binom{[r]}{a+1}$. We can also conclude that across different local groups, $p_{i,S}$ and $p_{j,T}$ are never multiples of each other when $i\neq j$. In fact, we will now show that even in the same local group, $p_{i,S}$ and $p_{i,T}$ are not multiples of each other unless $S=T$.

Claim 9.3.4. *For every $i\in[g]$, no two vectors in $\{p_{i,S}:S\subseteq\binom{[r]}{a+1}\}$ are multiples of each other.*

Proof. Suppose $p_{i,S}=\lambda\cdot p_{i,T}$ for some distinct sets $S,T\subset[r]$ of size $a+1$ each and some nonzero $\lambda\in\mathbb{F}_q$. So,

$$\begin{bmatrix}A_i(S)\\B_i(S)\end{bmatrix}\cdot A_i(S)^\perp-\lambda\cdot\begin{bmatrix}A_i(T)\\B_i(T)\end{bmatrix}\cdot A_i(T)^\perp=\begin{bmatrix}0\\p_{i,S}\end{bmatrix}-\lambda\cdot\begin{bmatrix}0\\p_{i,T}\end{bmatrix}=0$$

Note that every coordinate of $A_i(S)^\perp$ is non-zero. If not, then it will imply a linear dependency between a columns of $A_i(S)$ whereas we know that every $a\times a$ minor of $A_i(S)$ is non-zero. Thus we have a linear combination of the columns of $\begin{bmatrix}A_i(S\cup T)\\B_i(S\cup T)\end{bmatrix}$ which is zero. Moreover the combination is non-trivial because there is some $j\in S\backslash T$ and the column $A_i(j)$ has a nonzero coefficient. However

$$|S\cup T|\le 2a+2\le a+h. \tag{9.8}$$

By the MR property, any set of columns of the matrix $\begin{bmatrix}A_i\\B_i\end{bmatrix}$ of size at most $a+h$ has to be full rank, as this set can be obtained by selecting (a subset of) a and then h more columns from the matrix (9.7). Thus we arrive at a contradiction that completes the proof of the claim. □

By Claim 9.3.4 and the discussion above the claim, we can think of

$$\left\{p_{i,S} : i \in [g], S \in \binom{[r]}{a+1}\right\}$$

as distinct points in $\mathbb{PF}_q^{h-1}$. For brevity, from here on we assume that $p_{i,S}$ refers to the corresponding point in $\mathbb{PF}_q^{h-1}$. Define sets $X_1, \cdots, X_g \subseteq \mathbb{PF}_q^{h-1}$ as $X_i = \left\{p_{i,S} : S \in \binom{[r]}{a+1}\right\}$, we have $|X_1| = |X_2| = \cdots = |X_g| = \binom{r}{a+1}$ and they are mutually disjoint. Also $g \geq h$ by the hypothesis. By Claim 9.3.3, there is no hyperplane in $\mathbb{PF}_q^{h-1}$ which contains h points from distinct subsets of $X_1, \cdots, X_g$. So applying Lemma 9.3.1,

$$q \geq \left(\frac{g}{h-1} - 1\right) \cdot \binom{r}{a+1} - 4,$$

which concludes the proof. □

In the argument above we used vectors $\{p_{i,S}\}$, where i varies across indices of g local groups and S varies across all $\binom{r}{a+1}$ subsets of $[r]$ of size $a+1$. In the proof we relied on the condition $a+2 \leq h$ to ensure that the union of any two such sets S has size at most $a+h$.

Parikshit Gopalan [Gop17] has observed (and kindly allowed us to include his observation here) that we can generalize Proposition 9.3.2 to the case when $h < a+2$. To do this, in cases when $h < a+2$ we only consider sets S that have size $a+1$ but are constrained to contain the set $\{1, 2, \ldots, a+2-h\}$, as this ensures that pairwise unions still have size at most $a+h$. Clearly, the total number of such sets is $\binom{r-a+h-2}{h-1}$. The rest of the proof remains the same and yields the following

Proposition 9.3.5. *Assume $h < a+2$ and $h \leq n/r$; then any maximally recoverable (n, r, h, a, q)-local reconstruction code must have*

$$q \geq \left(\frac{n/r}{h-1} - 1\right) \cdot \binom{r-a+h-2}{h-1} - 4. \tag{9.9}$$

The following corollary follows immediately from Propositions 9.3.2 and 9.3.5 and presents the asymptotic form of our field size lower bound when $g \geq h$.

Corollary 9.3.6. *Suppose that a and h are arbitrary constants, but r may grow with n. Further suppose that $h \leq n/r$. In every maximally recoverable (n, r, h, a, q)-LRC, we have:*

$$q \geq \Omega_{a,h}\left(n \cdot r^{\min\{a,h-2\}}\right). \tag{9.10}$$

9.3.2 Lower bound when $g \leq h$

In this case, we cannot distribute the h additional erasures among h different local groups. Instead we will look at erasure patterns where either all the extra h erasures occur in the same group or they are spread equally ($\lceil h/g \rceil$ or $\lfloor h/g \rfloor$) in the g local groups. The sets $X_1, \ldots, X_g$ will now be a collection of subspaces of dimension roughly h/g such that no $(h-1)$-dimensional subspace can contain a subspace each from all of $X_1, \ldots, X_g$. To obtain the lower bound, we show that if q is too small, a random $(h-1)$-dimensional subspace will contain a subspace from each of $X_1, \ldots, X_g$ with high probability. The argument is more involved than in the $g \geq h$ case, because the subspaces inside each X_i can intersect non-trivially and the analysis has to account for this carefully. We obtain the following lower bound, the proof of which appears in Section 9.8.

Proposition 9.3.7. *Suppose that a, g, h are fixed constants such that $g \leq h$. In every maximally recoverable (n, r, h, a, q)-LRC with g local groups each of size $r = n/g$, we* *have:*

$$q \geq \Omega_{a,h,g}\left(n^{1+\alpha}\right) \text{ where } \alpha = \frac{\min\{a, h - 2\lceil h/g\rceil\}}{\lceil h/g\rceil}. \tag{9.11}$$

9.4 Maximally recoverable LRCs with $h = 2$

In this section we present our construction of maximally recoverable local reconstruction codes with two heavy parity symbols. Our construction relies on a determinantal identity (Lemma 9.4.1) and properties of $\mathbb{F}_q^*$, the multiplicative group of the field $\mathbb{F}_q$. The following identity conveniently reduces the $(ah+h) \times (ah+h)$ determinants that arise during our analysis into $h \times h$ determinants which are much easier to calculate. We will prove Lemma 9.4.1 in Section 9.9.

Lemma 9.4.1. *Let $C_1, \cdots, C_h$ be $a \times (a+1)$ dimensional matrices and $D_1, \cdots, D_h$ be $h \times (a+1)$ dimensional matrices over a field and let $D_i^{(j)}$ be the j^{th} row of D_i. Then,*

$$\det\left[\begin{array}{c|c|c|c} C_1 & 0 & \cdots & 0 \\ \hline 0 & C_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & C_h \\ \hline D_1 & D_2 & \cdots & D_h \end{array}\right] = (-1)^{\frac{ah(h-1)}{2}} \det\begin{bmatrix} \det\begin{bmatrix} C_1 \\ D_1^{(1)} \end{bmatrix} & \cdots & \det\begin{bmatrix} C_h \\ D_h^{(1)} \end{bmatrix} \\ \vdots & \ddots & \vdots \\ \det\begin{bmatrix} C_1 \\ D_1^{(h)} \end{bmatrix} & \cdots & \det\begin{bmatrix} C_h \\ D_h^{(h)} \end{bmatrix} \end{bmatrix}.$$

Lemma 9.4.2. *Let $r \mid n$, $a < r$ be integers. Let $g = \frac{n}{r}$. Assume that $n - ga - 2$ is positive. Suppose q is a prime power such that there exists a subgroup of $\mathbb{F}_q^*$ of size at least r and with at least n/r cosets; then there exists an explicit maximally recoverable $(n, r, h = 2, a, q)$-local reconstruction code.*

Proof. Let $G \subset \mathbb{F}_q^*$ be the multiplicative subgroup from the statement of the theorem. Let $\alpha_1, \alpha_2, \cdots, \alpha_r \in G$ be distinct elements from G and let $\lambda_1, \lambda_2, \cdots, \lambda_g \in \mathbb{F}_q^*$ be elements from distinct cosets of G. We specify our code via a parity check matrix of the form (9.4). For $i \in [g]$, we choose matrices $\{A_i\}$ and $\{B_i\}$ as:

$$A_i = \begin{bmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_r \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_r^2 \\ \vdots & \vdots & \ddots & \vdots \\ \alpha_1^a & \alpha_2^a & \cdots & \alpha_r^a \end{bmatrix}; \; B_i = \begin{bmatrix} \lambda_i & \lambda_i & \cdots & \lambda_i \\ \alpha_1^{a+1} & \alpha_2^{a+1} & \cdots & \alpha_r^{a+1} \end{bmatrix}. \tag{9.12}$$

Suppose that we have a erasures per local group and two more. We can easily correct the coordinates corresponding to local groups which have at most a erasures in them. This is because every matrix A_i is a Vandermonde matrix and all its $a \times a$ minors are nonzero. Now we are left with two cases:

Case 1: Both the extra erasures occurred in the same local group. Say, the i^{th} local group. In this case, we can correct the erased coordinates because any $(a+2) \times (a+2)$ minor of $\begin{bmatrix} A_i \\ B_i \end{bmatrix}$ (which is a Vandermonde matrix) is non degenerate.

Case 2: The two extra erasures occur in different groups say groups ℓ and ℓ', so we are left with two groups with $a+1$ erasures in each. Let S be the columns erased in group ℓ and let S' be the columns erased in group ℓ'. We want to argue that the following $(2a+2) \times (2a+2)$ submatrix is full rank:

$$M = \left[\begin{array}{c|c} A_\ell(S) & 0 \\ \hline 0 & A_{\ell'}(S') \\ \hline B_\ell(S) & B_{\ell'}(S') \end{array}\right]. \tag{9.13}$$

Let $S = \{\gamma_1, \gamma_2, \cdots, \gamma_{a+1}\}$ and $S' = \{\gamma_1', \gamma_2', \cdots, \gamma_{a+1}'\}$, then by Lemma 9.4.1,

$$\det(M) = 0 \iff \det \begin{bmatrix} \det \begin{bmatrix} A_\ell(S) \\ B_\ell(S)^{(1)} \end{bmatrix} & \det \begin{bmatrix} A_{\ell'}(S') \\ B_{\ell'}(S')^{(1)} \end{bmatrix} \\ \det \begin{bmatrix} A_\ell(S) \\ B_\ell(S)^{(2)} \end{bmatrix} & \det \begin{bmatrix} A_{\ell'}(S') \\ B_{\ell'}(S')^{(2)} \end{bmatrix} \end{bmatrix} = 0$$

$$\iff \det \begin{bmatrix} \det \begin{pmatrix} \gamma_1 & \cdots & \gamma_{a+1} \\ \gamma_1^2 & \cdots & \gamma_{a+1}^2 \\ \vdots & \ddots & \vdots \\ \gamma_1^a & \cdots & \gamma_{a+1}^a \\ \lambda_\ell & \cdots & \lambda_\ell \end{pmatrix} & \det \begin{pmatrix} \gamma_1' & \cdots & \gamma_{a+1}' \\ (\gamma_1')^2 & \cdots & (\gamma_{a+1}')^2 \\ \vdots & \ddots & \vdots \\ (\gamma_1')^a & \cdots & (\gamma_{a+1}')^a \\ \lambda_{\ell'} & \cdots & \lambda_{\ell'} \end{pmatrix} \\ \det \begin{pmatrix} \gamma_1 & \cdots & \gamma_{a+1} \\ \gamma_1^2 & \cdots & \gamma_{a+1}^2 \\ \vdots & \ddots & \vdots \\ \gamma_1^a & \cdots & \gamma_{a+1}^a \\ \gamma_1^{a+1} & \cdots & \gamma_{a+1}^{a+1} \end{pmatrix} & \det \begin{pmatrix} \gamma_1' & \cdots & \gamma_{a+1}' \\ {\gamma_1'}^2 & \cdots & (\gamma_{a+1}')^2 \\ \vdots & \ddots & \vdots \\ {\gamma_1'}^a & \cdots & (\gamma_{a+1}')^a \\ {\gamma_1'}^{a+1} & \cdots & (\gamma_{a+1}')^{a+1} \end{pmatrix} \end{bmatrix} = 0$$

$$\iff \det \begin{bmatrix} \lambda_\ell & \lambda_{\ell'} \\ \prod_{i \in [a+1]} \gamma_i & \prod_{i \in [a+1]} \gamma_i' \end{bmatrix} = 0$$

where we factored out the (nonzero) Vandermonde determinant from each column. Since $\gamma_i, \gamma_i' \in G$ and $\lambda_\ell, \lambda_{\ell'}$ are in different cosets of G, the last determinant is not zero. □

In Lemma 9.4.2, given n and r such that $r \mid n$, we want to find a small field $\mathbb{F}_q$ such that $\mathbb{F}_q^*$ contains a subgroup of size at least r and with at least n/r cosets. For example, if $n+1$ is a prime power, then we can take $q = n+1$. The following lemma shows that one can always find such a field of size $q = O(n)$. We defer the proof to the Appendix.

Lemma 9.4.3. *Let r, n be some positive integers with $r \leq n$. Then there exists a finite field $\mathbb{F}_q$ with $q = O(n)$ such that the multiplicative group $\mathbb{F}_q^*$ contains a subgroup of size at least r and with at least n/r cosets. If additionally we require that the field has characteristic two, then such a field exists with $q = n \cdot \exp(O(\sqrt{\log n}))$.*

Combining Lemma 9.4.3 with Lemma 9.4.2 gives the following theorem.

Theorem 9.4.4. *Let $r \mid n$, $a < r$ be integers. Let $g = \frac{n}{r}$. Assume that $n - ga - 2$ is positive. Then there exists an explicit maximally recoverable $(n, r, h = 2, a, q)$-local reconstruction code with $q = O(n)$. If we require the field to be of characteristic 2, such a code exists with $q \leq n \cdot \exp(O(\sqrt{\log n}))$.*

9.5 Maximally recoverable LRCs with $h = 3$

In this section, we present our construction of maximally recoverable local reconstruction codes with three heavy parity symbols. Our construction extends the ideas in the construction of Section 9.4 using field extensions. In addition to the determinantal identity 9.4.1, we will need the following identity which follows immediately from Lemma 9.9.2.

Lemma 9.5.1. *Let C_1 be an $a \times (a+1)$ matrix, C_2 be an $a \times (a+2)$ matrix, D_1 be a $3 \times (a+1)$ matrix and D_2 be a $3 \times (a+2)$ matrix and let $D_i^{(j)}$ be the j^{th} row of D_i. Then,*

$$\det \left[\begin{array}{c|c} C_1 & 0 \\ \hline 0 & C_2 \\ \hline D_1 & D_2 \end{array}\right] = 0$$

$$\iff \det \begin{bmatrix} C_1 \\ D_1^{(1)} \end{bmatrix} \cdot \det \begin{bmatrix} C_2 \\ D_2^{(2)} \\ D_2^{(3)} \end{bmatrix} - \det \begin{bmatrix} C_1 \\ D_1^{(2)} \end{bmatrix} \cdot \det \begin{bmatrix} C_2 \\ D_2^{(1)} \\ D_2^{(3)} \end{bmatrix} + \det \begin{bmatrix} C_1 \\ D_1^{(3)} \end{bmatrix} \cdot \det \begin{bmatrix} C_2 \\ D_2^{(1)} \\ D_2^{(2)} \end{bmatrix} = 0$$

Our construction is based on Cauchy matrices, so we will also need the the following lemma about the determinants of such matrices.

Lemma 9.5.2. *([LN83]) Let $\alpha_1, \cdots, \alpha_m, \beta_1, \cdots, \beta_m \in \mathbb{F}_q$ be all distinct; then*

$$\det \begin{bmatrix} \frac{1}{\alpha_1-\beta_1} & \frac{1}{\alpha_2-\beta_1} & \cdots & \frac{1}{\alpha_m-\beta_1} \\ \frac{1}{\alpha_1-\beta_2} & \frac{1}{\alpha_2-\beta_2} & \cdots & \frac{1}{\alpha_m-\beta_2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\alpha_1-\beta_m} & \frac{1}{\alpha_2-\beta_m} & \cdots & \frac{1}{\alpha_m-\beta_m} \end{bmatrix} = \frac{\prod_{i>j}(\alpha_i-\alpha_j)(\beta_j-\beta_i)}{\prod_{i,j}(\alpha_i-\beta_j)}$$

Matrices of the above form are called Cauchy matrices. Every minor of a Cauchy matrix is nonzero because the minors themselves look like a Cauchy matrix. We are now ready to present the construction for three global parities.

Lemma 9.5.3. *Let $r \mid n$, $a < r$ be integers. Let $g = \frac{n}{r}$. Assume that $n - ga - 3$ is positive. Suppose $q_0 \geq 2r + 3$ is a prime power such that there exists a subgroup of $\mathbb{F}_{q_0}^*$ of size at least $r + 2$ and with at least n/r cosets. Then there exists an explicit maximally recoverable $(n, r, h = 3, a, q = q_0^3)$-local reconstruction code.*

Proof. Let $G \subset \mathbb{F}_{q_0}^*$ be the multiplicative subgroup from the statement of the theorem. Choose distinct $\beta_{a+1}, \beta_{a+2}, \beta_{a+3} \in \mathbb{F}_{q_0}$ and let

$$\Omega = \left\{ \alpha \in \mathbb{F}_{q_0} : \frac{\alpha - \beta_{a+2}}{\alpha - \beta_{a+3}} \in G \right\}.$$

Clearly $|\Omega| = |G| - 1 \geq r + 1$, so we can choose distinct $\alpha_1, \cdots, \alpha_r \in \Omega \setminus \{\beta_{a+1}\}$. Finally, since $q_0 \geq 2r + 3 \geq r + a + 3$, we can choose distinct $\beta_1, \cdots, \beta_a \in \mathbb{F}_{q_0} \setminus \{\alpha_1, \cdots, \alpha_r, \beta_{a+1}, \beta_{a+2}, \beta_{a+3}\}$. Let $\mu_1, \cdots, \mu_g \in \mathbb{F}_{q_0}$ be elements from distinct cosets of G.

Now let $\mathbb{F}_q$ be a degree 3 extension of $\mathbb{F}_{q_0}$, so we have $q = q_0^3$. As $\mathbb{F}_q$ is a 3-dimensional vector space over $\mathbb{F}_{q_0}$, choose a basis $v_0, v_1, v_2 \in \mathbb{F}_q$ for this space and choose distinct $\gamma_1, \cdots, \gamma_g \in \mathbb{F}_{q_0}$. Define $\lambda_i = v_0 + \gamma_i v_1 + \gamma_i^2 v_2$. Then any three of the elements $\lambda_1, \cdots, \lambda_g \in \mathbb{F}_q$ are linearly independent over $\mathbb{F}_{q_0}$; we call this property 3-wise independence over $\mathbb{F}_{q_0}$. Define the matrices A_i and B_i as follows:

$$A_i = \begin{bmatrix} \frac{1}{\alpha_1-\beta_1} & \cdots & \frac{1}{\alpha_r-\beta_1} \\ \vdots & \ddots & \vdots \\ \frac{1}{\alpha_1-\beta_a} & \cdots & \frac{1}{\alpha_r-\beta_a} \end{bmatrix}; \; B_i = \begin{bmatrix} \frac{\lambda_i}{\alpha_1-\beta_{a+1}} & \cdots & \frac{\lambda_i}{\alpha_r-\beta_{a+1}} \\ \frac{\mu_i}{\alpha_1-\beta_{a+2}} & \cdots & \frac{\mu_i}{\alpha_r-\beta_{a+2}} \\ \frac{1}{\alpha_1-\beta_{a+3}} & \cdots & \frac{1}{\alpha_r-\beta_{a+3}} \end{bmatrix} \tag{9.14}$$

Now we will show that the above construction satisfies the MR property. We have a erasures per local group and 3 more. We can easily correct groups with only a erasures because A_i are Cauchy matrices where every $a \times a$ minor is non-degenerate. So we only need to worry about local groups with more than a erasures. There are three cases.

Case 1: All three extra erasures in the same group.

Say we have $a+3$ erasures in local group i, then we can correct these errors because the matrix $\begin{bmatrix} A_i \\ B_i \end{bmatrix}$ is a Cauchy matrix (except for some scaling factors in the rows), and therefore each of its $(a+3) \times (a+3)$ minors is nonzero by Lemma 9.5.2.

Case 2: The three extra erasures are distributed across two groups.

Suppose the extra erasures occur in groups ℓ, ℓ' with $(a+1)$ erasures in group ℓ corresponding to a subset $S \subseteq [r]$ of its columns and $(a+2)$ erasures in group ℓ' corresponding to a subset $S' \subseteq [r]$ of its columns. To correct these erasures we need to show the following matrix is full rank:

$$\left[\begin{array}{c|c} A_\ell(S) & 0 \\ \hline 0 & A_{\ell'}(S') \\ \hline B_\ell(S) & B_{\ell'}(S') \end{array}\right]. \tag{9.15}$$

By Lemma 9.5.1, the above matrix fails to be full rank iff

$$\det\begin{bmatrix} A_\ell(S) \\ B_\ell(S)^{(1)} \end{bmatrix} \cdot \det\begin{bmatrix} A_{\ell'}(S') \\ B_{\ell'}(S')^{(2)} \\ B_{\ell'}(S')^{(3)} \end{bmatrix} - \det\begin{bmatrix} A_\ell(S) \\ B_\ell(S)^{(2)} \end{bmatrix} \cdot \det\begin{bmatrix} A_{\ell'}(S') \\ B_{\ell'}(S')^{(1)} \\ B_{\ell'}(S')^{(3)} \end{bmatrix}$$
$$+ \det\begin{bmatrix} A_\ell(S) \\ B_\ell(S)^{(3)} \end{bmatrix} \cdot \det\begin{bmatrix} A_{\ell'}(S') \\ B_{\ell'}(S')^{(1)} \\ B_{\ell'}(S')^{(2)} \end{bmatrix} = 0$$

The above determinant is a $\mathbb{F}_q$-linear combination of λ_ℓ and $\lambda_{\ell'}$ and the coefficient of λ_ℓ, which arises from the first term, is nonzero because $\begin{bmatrix} A_\ell \\ B_\ell \end{bmatrix}$ and $\begin{bmatrix} A_{\ell'} \\ B_{\ell'} \end{bmatrix}$ are Cauchy matrices. By 3-wise independence of λ's, this linear combination cannot be zero, and therefore the matrix (9.15) has full rank.

Case 3: The three extra erasures occur in distinct groups.

Suppose the three extra erasures occur in groups $\ell_1, \ell_2, \ell_3 \in [g]$ and let $S_1, S_2, S_3 \subseteq [r]$ be sets of size $a+1$ corresponding to the erasures in the groups ℓ_1, ℓ_2, ℓ_3 respectively. To correct these erasures we need to show the following matrix is full rank:

$$\left[\begin{array}{c|c|c} A_{\ell_1}(S_1) & 0 & 0 \\ \hline 0 & A_{\ell_2}(S_2) & 0 \\ \hline 0 & 0 & A_{\ell_3}(S_3) \\ \hline B_{\ell_1}(S_1) & B_{\ell_2}(S_2) & B_{\ell_3}(S_3) \end{array}\right]$$

By Lemma 9.4.1, if the above matrix is not full rank then

$$\det\begin{bmatrix} \det\begin{bmatrix} A_{\ell_1}(S_1) \\ B^{(1)}_{\ell_1}(S_1)\end{bmatrix} & \det\begin{bmatrix} A_{\ell_2}(S_2) \\ B^{(1)}_{\ell_2}(S_2)\end{bmatrix} & \det\begin{bmatrix} A_{\ell_3}(S_3) \\ B^{(1)}_{\ell_3}(S_3)\end{bmatrix} \\ \det\begin{bmatrix} A_{\ell_1}(S_1) \\ B^{(2)}_{\ell_1}(S_1)\end{bmatrix} & \det\begin{bmatrix} A_{\ell_2}(S_2) \\ B^{(2)}_{\ell_2}(S_2)\end{bmatrix} & \det\begin{bmatrix} A_{\ell_3}(S_3) \\ B^{(2)}_{\ell_3}(S_3)\end{bmatrix} \\ \det\begin{bmatrix} A_{\ell_1}(S_1) \\ B^{(3)}_{\ell_1}(S_1)\end{bmatrix} & \det\begin{bmatrix} A_{\ell_2}(S_2) \\ B^{(3)}_{\ell_2}(S_2)\end{bmatrix} & \det\begin{bmatrix} A_{\ell_3}(S_3) \\ B^{(3)}_{\ell_3}(S_3)\end{bmatrix} \end{bmatrix} = 0.$$

For $k \in \{1,2,3\}$, let $c_k = \prod_{i>j, i,j \in S_k}(\alpha_i - \alpha_j), d = \prod_{i>j, i,j\in[a]}(\beta_j - \beta_i), e_k = \prod_{i\in S_k, j\in[a]}(\alpha_i - \beta_j)$. By Lemma 9.5.2, we can write down explicit expressions for the entries in the above determinant to get:

$$\det\begin{bmatrix} \lambda_{\ell_1}\frac{c_1 d\prod_{i\in[a]}(\beta_i-\beta_{a+1})}{e_1\prod_{i\in S_1}(\alpha_i-\beta_{a+1})} & \lambda_{\ell_2}\frac{c_2 d\prod_{i\in[a]}(\beta_i-\beta_{a+1})}{e_2\prod_{i\in S_2}(\alpha_i-\beta_{a+1})} & \lambda_{\ell_3}\frac{c_3 d\prod_{i\in[a]}(\beta_i-\beta_{a+1})}{e_3\prod_{i\in S_3}(\alpha_i-\beta_{a+1})} \\ \mu_{\ell_1}\frac{c_1 d\prod_{i\in[a]}(\beta_i-\beta_{a+2})}{e_1\prod_{i\in S_1}(\alpha_i-\beta_{a+2})} & \mu_{\ell_2}\frac{c_2 d\prod_{i\in[a]}(\beta_i-\beta_{a+2})}{e_2\prod_{i\in S_2}(\alpha_i-\beta_{a+2})} & \mu_{\ell_3}\frac{c_3 d\prod_{i\in[a]}(\beta_i-\beta_{a+2})}{e_3\prod_{i\in S_3}(\alpha_i-\beta_{a+2})} \\ \frac{c_1 d\prod_{i\in[a]}(\beta_i-\beta_{a+3})}{e_1\prod_{i\in S_1}(\alpha_i-\beta_{a+3})} & \frac{c_2 d\prod_{i\in[a]}(\beta_i-\beta_{a+3})}{e_2\prod_{i\in S_2}(\alpha_i-\beta_{a+3})} & \frac{c_3 d\prod_{i\in[a]}(\beta_i-\beta_{a+3})}{e_3\prod_{i\in S_3}(\alpha_i-\beta_{a+3})} \end{bmatrix} = 0.$$

We can scale rows and columns to conclude that

$$\det\begin{bmatrix} \lambda_{\ell_1}\prod_{i\in S_1}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+1}}\right) & \lambda_{\ell_2}\prod_{i\in S_2}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+1}}\right) & \lambda_{\ell_3}\prod_{i\in S_3}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+1}}\right) \\ \mu_{\ell_1}\prod_{i\in S_1}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+2}}\right) & \mu_{\ell_2}\prod_{i\in S_2}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+2}}\right) & \mu_{\ell_3}\prod_{i\in S_3}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+2}}\right) \\ 1 & 1 & 1 \end{bmatrix} = 0.$$

By the choice of α's, $\prod_{i\in S_j}\left(\frac{\alpha_i-\beta_{a+3}}{\alpha_i-\beta_{a+2}}\right) \in G$ for $j = 1,2,3$. By writing the Laplace expansion of the determinant over the first row, the above determinant is a linear combination in $\lambda_{\ell_1}, \lambda_{\ell_2}, \lambda_{\ell_3}$ with coefficients from $\mathbb{F}_{q_0}$. The coefficients of λ's in this linear combination are nonzero because $\mu_{\ell_1}, \mu_{\ell_2}, \mu_{\ell_3}$ belong to distinct cosets of G in $\mathbb{F}_{q_0}^*$. Because λ's are 3-wise independent over $\mathbb{F}_{q_0}$, we get a contradiction. $\square$

Combining Lemma 9.5.3 with Lemma 9.4.3 gives the following theorem.

Theorem 9.5.4. *Let $r \mid n$, $a < r$ be integers. Let $g = \frac{n}{r} \geq 2$. Assume that $n - ga - 3$ is positive. Then there exists an explicit maximally recoverable $(n, r, h = 3, a, q)$-local reconstruction code with $q = O(n^3)$. If we require the field to be of characteristic 2, such a code exists with $q = n^3 \cdot \exp(O(\sqrt{\log n}))$.*

9.6 Maximally recoverable LRCs from elliptic curves

Our construction of MR $(n, r = 3, h = 3, a = 1, q)$-LRCs is technically disjoint from our results in the previous sections. We observe that in this narrow case, maximally recoverable LRCs are equivalent to families of *matching collinear triples* in the projective plane $\mathbb{PF}_q^2$, i.e., sets of points partitioned into collinear triples, where no three points other than those forming a triple are collinear. In Section 9.6.1 we state the quantitative parameters of such a family A that we can obtain and translate those to parameters of an MR LRC. The goal of Section 9.6.2 is to construct the family A using elliptic curves and 3-AP free sets. In Section 9.6.2 we develop the necessary machinery of elliptic curves, and in Section 9.6.2 we carry out the construction.

9.6.1 LRCs from matching collinear triples

We will reduce the problem of constructing maximally recoverable codes for $h = 3, r = 3, a = 1$ to the problem of constructing matching collinear triples in $\mathbb{PF}_q^2$ which we define below.

Definition 9.6.1. *We say that $A \subset \mathbb{PF}_q^2$ has matching collinear triples if A can be partitioned into triples, $A = \sqcup_{i=1}^m \{a_i, b_i, c_i\}$, such that the only collinear triples in A are $\{a_i, b_i, c_i\}$ for $i \in [m]$.*

What is the largest subset $A \subset \mathbb{PF}_q^2$ with matching collinear triples? If we consider all the $q+1$ lines through some fixed point of A, at most one line can contain two other points of A. All other lines can contain at most one other point of A. So $|A| \leq q+3$. The following lemma shows that we can construct a set A with size $|A| \geq q^{1-o(1)}$. It is an interesting open question if we can get $|A| \geq \Omega(q)$.

Lemma 9.6.2. *For any prime power q, there is an explicit set $A \subset \mathbb{PF}_q^2$ with matching collinear triples of size $|A| \geq q \cdot \exp(-C\sqrt{\log q})$ where $C > 0$ is some absolute constant.*

We will prove Lemma 9.6.2 in Section 9.6.2.

Lemma 9.6.3. *Assume $g \geq 2$. There exists a subset $S \subset \mathbb{PF}_q^2$ that has g matching collinear triples* if and only if *there exists a maximally recoverable $(3g, r = 3, h = 3, a = 1, q)$-local reconstruction code.*

Proof. We first show how to obtain codes from families of collinear triples. Let $S = \cup_{i=1}^g \{a_i, b_i, c_i\}$ be such that the only collinear triples in S are $\{a_i, b_i, c_i\}$ for $i \in [g]$. From now, we will think of elements of S as vectors in $\mathbb{F}_q^3$ such that every triple of points except for the triples $\{a_i, b_i, c_i\}$ are linearly independent. We can scale each vector with nonzero elements in $\mathbb{F}_q$ such that $a_i + b_i + c_i = 0$ in $\mathbb{F}_q^3$ for every $i \in [g]$. For $i \in [g]$, define blocks A_i and B_i of the parity check matrix (9.4) as:

$$A_i = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}; \; B_i = \begin{bmatrix} 0 & -b_i & c_i \end{bmatrix}.$$

We need to correct 1 erasure per group and any 3 extra erasures. We can correct groups with a single erasure because A_i is a simple parity check constraint on all the coordinates of the group. We now have to correct groups with more than one erasure, there are two cases:

Case 1: The three extra erasures are in two groups.

Suppose the two groups are i, j and in group i all the coordinates are erased and in group j the second and third coordinates are erased (the other two cases are similar). To correct these erasures, we have to argue that the following matrix is full rank:

$$\left[\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline 0 & -b_i & c_i & -b_j & c_j \end{array}\right]$$

Subtract the first column in each group from the rest, it is equivalent to the following matrix being full rank:

$$\left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ \hline 0 & -b_i & c_i & -b_j & c_j + b_j \end{array}\right] = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ \hline 0 & -b_i & c_i & -b_j & a_j \end{array}\right]$$

which is true because b_i, c_i, a_j are linearly independent.

Case 2: The three extra erasures are in distinct groups.

Suppose the three groups are i, j, k and in each group the second and third columns are erased (the other cases are similar). To correct these erasures, we have to argue that the following matrix is full rank:

$$\left[\begin{array}{cc|cc|cc} 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ \hline -b_i & c_i & -b_j & c_j & -b_k & c_k \end{array}\right]$$

Subtract the first column in each group from the rest, it is equivalent to the following matrix being full rank:

$$\left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ \hline -b_i & c_i + b_i & -b_j & c_j + b_j & -b_k & c_k + b_k \end{array}\right] = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 0 \\ \hline -b_i & -a_i & -b_j & -a_j & -b_k & -a_k \end{array}\right]$$

which is true because a_i, a_j, a_k are linearly independent.

Reverse connection. We now proceed to show how to obtain a set with matching collinear triples from codes. Given a maximally recoverable $(3g, r = 3, h = 3, a = 1, q)$-local reconstruction code with a parity check matrix (9.4), without loss of generality assume that for all $i \in [g]$,

$$A_i = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix};\ B_i = \begin{bmatrix} v_i^1 & v_i^2 & v_i^3 \end{bmatrix},$$

where $\{v_i^s\}_{s\in[3], i\in[g]} \subseteq \mathbb{F}_q^3$. For each $i \in [g]$, define

$$a_i = v_i^2 - v_i^1 \qquad b_i = v_i^3 - v_i^2 \qquad c_i = v_i^1 - v_i^3.$$

Clearly, for all $i \in [g]$, $a_i + b_i + c_i = 0$. Consider $\{a_i, b_i, c_i\}_{i\in[g]}$ as elements of $\mathbb{PF}_q^2$ and define our family to be $S = \cup_{i=1}^{g}\{a_i, b_i, c_i\}$. It remains to show that all triples of elements of S other than $\{a_i, b_i, c_i\}$ are non-collinear. When all three elements $v_i^\alpha - v_i^\beta, v_j^\gamma - v_j^\delta, v_k^\varepsilon - v_k^\zeta$ belong to different groups this follows from the fact that, as implied by the MR property, the matrix

$$\left[\begin{array}{cc|cc|cc} 1 & 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 1 & 1 \\ \hline v_i^\beta & v_i^\alpha & v_j^\delta & v_j^\gamma & v_k^\zeta & v_k^\varepsilon \end{array}\right] = \left[\begin{array}{cc|cc|cc} 1 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 1 & 0 & 1 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 & 0 \\ \hline v_i^\beta & v_i^\alpha - v_i^\beta & v_j^\delta & v_j^\gamma - v_j^\delta & v_k^\zeta & v_k^\varepsilon - v_k^\zeta \end{array}\right]$$

is full rank. When triples come from two groups, (say, $v_i^\beta - v_i^\alpha, v_i^\gamma - v_i^\alpha, v_j^\delta - v_j^\varepsilon$) this again follows from the MR property, as the matrix

$$\left[\begin{array}{ccc|cc} 1 & 1 & 1 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 1 \\ \hline v_i^\alpha & v_i^\beta & v_i^\gamma & v_j^\varepsilon & v_j^\delta \end{array}\right] = \left[\begin{array}{ccc|cc} 1 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 1 & 0 \\ \hline v_i^\alpha & v_i^\beta - v_i^\alpha & v_i^\gamma - v_i^\alpha & v_j^\varepsilon & v_j^\delta - v_j^\varepsilon \end{array}\right]$$

is also full rank. □

Combining Lemma 9.6.2 and Lemma 9.6.3 along with the fact that all the constructions are explicit gives the following theorem.

Theorem 9.6.4. *For any $n > 3$ which is a multiple of 3 and for any finite field $\mathbb{F}_q$, there exists an explicit maximally recoverable $(n, r = 3, h = 3, a = 1, q)$-local reconstruction code provided that $q \geq \Omega\left(n \cdot \exp\left(C\sqrt{\log n}\right)\right)$ where $C > 0$ is some absolute constant.*

9.6.2 Matching Collinear Triples from AP free sets

In this section, we will prove Lemma 9.6.2 by constructing a large $A \subset \mathbb{PF}_q^2$ with matching collinear triples. The main idea is to reduce the problem to constructing a large subset $A \subset \mathbb{Z}/N\mathbb{Z}$ with *matching tri-sums* where $N = \Omega(q)$. A subset $A \subset \mathbb{Z}/N\mathbb{Z}$ has matching tri-sums if A can partitioned into disjoint triples, $A = \sqcup_i \{a_i, b_i, c_i\}$ such that the only 3 element subsets of A which sum to zero are the triples $\{a_i, b_i, c_i\}$ in the partition. Such sets can be constructed from subsets of $[N]$ without any non-trivial arithmetic progressions. The best known construction of a subset of $[N]$ with no non-trivial three term arithmetic progressions is due to Behrend [Beh46] which was slightly improved in [Elk11]. An explicit construction with similar bounds as [Beh46] was given in [Mos53].

Theorem 9.6.5 ([Beh46, Mos53, Elk11])**.** *For some absolute constant $C > 0$, there exists an explicit $A \subset \{1, 2, \cdots, N\}$ with $|A| \geq N \cdot \exp(-C\sqrt{\log N})$ which doesn't contain any 3 term arithmetic progressions i.e. there doesn't exist distinct $x, y, z \in A$ such that $x + z = 2y$.*

It is also known that any set $A \subset \{1, 2, \cdots, N\}$ with no non-trivial 3 term arithmetic progressions should have size $|A| \lesssim \frac{(\log\log N)^4}{\log N} \cdot N$ [Blo16].

The reduction from matching collinear triples in $\mathbb{F}_q^2$ to subsets of $\mathbb{Z}/N\mathbb{Z}$ with matching tri-sums is simple when q is a prime. In this case we can set $N = q$. Three points $(x_1, y_1), (x_2, y_2), (x_3, y_3) \in \mathbb{F}_q^2$ on the cubic curve $Y = X^3$ are collinear iff $x_1 + x_2 + x_3 = 0$. So we can get a large subset of $\mathbb{PF}_q^2$ with matching collinear triples, from a large subset of $\mathbb{F}_q \cong \mathbb{Z}/q\mathbb{Z}$ with matching tri-sums. And from Theorem 9.6.5, we can get such a set of size $\geq q \cdot \exp(-O(\sqrt{\log q}))$.

When q is not prime, the additive group of $\mathbb{F}_q$ is not cyclic anymore and subsets of $\mathbb{F}_q$ with matching tri-sums are much smaller. For example, if $\mathbb{F}_q$ has characteristic 2, which is the main setting of interest for us, the size of the largest subset of $\mathbb{F}_q$ with matching tri-sums is $\leq q^c$ for some absolute constant $c < 1$ [Kle16]. We will use some results on elliptic curves which are a special kind of cubic curves to make the reduction work over any field.

Elliptic curves

We will give a quick introduction to elliptic curves, please refer to [Sil09, MBG$^+$13] for proofs and formal definitions. Let $\mathbb{K}$ be a finite field and $\overline{\mathbb{K}}$ be its algebraic closure. A

Weierstrass equation defined over $\mathbb{K}$ is homogeneous cubic equation in three variables of the following form:

$$F(X,Y,Z) = Y^2Z + a_1XYZ + a_3YZ^2 - X^3 - a_2X^2Z - a_4XZ^2 - a_6Z^3 = 0$$

where $a_1, a_2, \cdots, a_6 \in \mathbb{K}$. A point $p \in \mathbb{P}\overline{\mathbb{K}}^2$ is called a *singular point* if

$$\frac{\partial F}{\partial X}(p) = \frac{\partial F}{\partial Y}(p) = \frac{\partial F}{\partial Z}(p) = 0.$$

If there are no such points, we call the equation *non-singular*, else we call the equation *singular*. Since the equation is cubic, it can have at most one singular point. There is an explicit polynomial function Δ in variables $a_1, a_2, \cdots, a_6$ and coefficients in $\mathbb{K}$ called the *discriminant*, such that the Weierstrass equation is singular iff $\Delta(a_1, \cdots, a_6) = 0$ (see Section III.1 in [Sil09] for the explicit polynomial). A singular Weierstrass equation[5] E with singularity at $(X,Y,Z) = (0,0,1)$ can be written as:

$$E: \ Y^2Z + a_1XYZ - a_3X^2Z = X^3.$$

We associate with E the set of all points in $\mathbb{P}\overline{\mathbb{K}}^2$ which satisfy the equation E. There is exactly one point in E with Z-coordinate equal to 0, namely $(0:1:0)$, we call this special point *the point at infinity* and denote it by $\mathcal{O}$. The set of non-singular $\mathbb{K}$-rational points of E, denoted by $E_{ns}(\mathbb{K})$ is defined as follows:

$$E_{ns}(\mathbb{K}) = \{(x:y:1) | F(x,y,1) = 0, \ x,y \in \mathbb{K}, (x,y) \neq (0,0)\} \cup \{\mathcal{O}\}.$$

$E_{ns}(\mathbb{K})$ is an Abelian group under a certain addition operation '+', with the point at infinity $\mathcal{O}$ as the group identity. Under this operation, three points $a, b, c \in E_{ns}(\mathbb{K})$

[5] Usually elliptic curves are defined as curves given by non-singular Weierstrass equations. But for our purpose, it is easier to work with singular Weierstrass equations.

satisfy $a+b+c = \mathcal{O}$ iff a, b, c are collinear in $\mathbb{PK}^2$. The following theorem shows that $E_{ns}(\mathbb{K})$ is isomorphic to $\mathbb{K}^*$ when E is of a special form.

Theorem 9.6.6 (Theorem 8.1 in [MBG$^+$13])**.** *Let $E : (Y - \alpha X)(Y - \beta X)Z = X^3$ be a singular Weierstrass equation with $\alpha, \beta \in \mathbb{K}$ and $\alpha \neq \beta$. Then the map $\phi : E_{ns}(\mathbb{K}) \to \mathbb{K}^*$ defined as:*

$$\phi : \mathcal{O} \mapsto 1 \qquad \phi : (x, y, 1) \mapsto \frac{y - \beta x}{y - \alpha x}$$

is a group isomorphism.

Since $\mathbb{K}^*$ is a cyclic group for any finite field $\mathbb{K}$, $E_{ns}(\mathbb{K})$ is isomorphic to $\mathbb{Z}/N\mathbb{Z}$ for $N = |\mathbb{K}| - 1$ when E is a singular Weierstrass equation as in Theorem 9.6.6.

Proof of Lemma 9.6.2

Proof. Let E be a singular Weierstrass equation[6] defined over $\mathbb{F}_q$ as in Theorem 9.6.6. By Theorem 9.6.6, $E_{ns}(\mathbb{F}_q) \cong \mathbb{Z}/N\mathbb{Z}$ where $N = q - 1$. Recall that $a, b, c \in E_{ns}(\mathbb{F}_q)$ satisfy $a + b + c = 0$ in the group iff they are collinear.

Let $B \subset \{1, 2, \cdots, N/20\}$ be an explicit subset of size $|B| \gtrsim N \cdot \exp(-C\sqrt{\log N})$ with no 3-term arithmetic progressions, as guaranteed by Theorem 9.6.5. Now define subsets $A_1, A_2, A_3 \subset \mathbb{Z}/N\mathbb{Z}$ as

$$A_1 = \{x : x \in B\}, A_2 = \left\{\left\lfloor \frac{N}{3} \right\rfloor + x : x \in B\right\}, A_3 = \left\{N - \left\lfloor \frac{N}{3} \right\rfloor - 2x : x \in B\right\}.$$

Clearly, A_1, A_2, A_3 are disjoint. Finally we define $\tilde{A} = A_1 \cup A_2 \cup A_3$. Now we claim that the only triples from $\tilde{A}$ which sum to zero in $\mathbb{Z}/N\mathbb{Z}$ are $\{x, \lfloor N/3 \rfloor + x, N - \lfloor N/3 \rfloor - 2x\}$ for $x \in B$ and these triples form a partition of $\tilde{A}$.

[6] It is not essential to work with singular Weierstrass equations. The proof also works with non-singular elliptic curves as long the group of $\mathbb{K}$-rational points is cyclic or has a large cyclic subgroup.

It is not hard to see that if three distinct elements $a, b, c \in \tilde{A}$ satisfy $a+b+c=0$, then a, b, c should come from 3 different sets A_1, A_2, A_3. So after reordering, we can assume

$$a = x, b = \lfloor N/3 \rfloor + y, c = N - \lfloor N/3 \rfloor - 2z$$

for some $x, y, z \in B$. Thus $a+b+c=0$ implies that $x+y=2z$, which implies that $x=y=z$ since B is free from 3 arithmetic progressions.

Finally let $A \subset \mathbb{PF}_q^2$ be the set of points in $E_{ns}(\mathbb{F}_q)$ which map to the set $\tilde{A} \subset \mathbb{Z}/N\mathbb{Z}$ under the isomorphism $E_{ns}(\mathbb{F}_q) \cong \mathbb{Z}/N\mathbb{Z}$. Now it is easy to see that A has matching collinear triples and we have $|A| \gtrsim q \cdot \exp(-C\sqrt{\log q})$. □

9.7 Open problems

In this work we made progress towards quantifying the minimal size of finite fields required for existence of maximally recoverable local reconstruction codes and obtained both lower and upper bounds. There is a wide array of questions that remain open. Here we highlight some of them:

- Our lower bound (9.2) implies that even in the regime of constant a and h, when $h \geq 3, a \geq 1$ and r grows with n there exist no MR codes over fields of size $O(n)$. It would be of great interest to understand if such codes always exist when all parameters a, h, and r are held constant and only n grows.
- Our lower bound (9.2) is of the form $q = \Omega(nr^{\alpha})$ where $\alpha > 0$ in all parameter ranges except when $a = 0$ or $h = 2$ or $g = 2$ or $(g = 3, h = 4, a = 1)$. When $a = 0$ or $h = 2$, we now know that there are linear field size constructions for any r. Is this also true when $g = 2$?
- In the case of fields of characteristic two, can one reduce the field sizes in Theorems 9.4.4 and 9.5.4 to $O(n)$ and $O(n^3)$ to match the case of prime fields?

- Our Lemma 9.6.3 provides an equivalence between the parameters of families of matching collinear triples in the projective plane and maximally recoverable local reconstruction codes with $r = 3, h = 3$, and $a = 1$. We hope that this reduction will be useful to obtain an $\omega(n)$ lower bound for the alphabet size of MR $(n, r = 3, h = 3, a = 1, q)$-LRCs, or lead to a construction over fields of linear size. It is also very interesting to see if techniques similar to those in Section 9.6.2 can be used to get codes over fields of nearly linear size when $r > 3$ or $a > 1$ or $h > 3$.
- Finally, it is interesting to see if our lower bound in Theorem 9.1.1 can be generalized to the setting of non-linear codes. Basic results about LRCs such as distance vs. redundancy trade-off [GHSY12] have been generalized to non-linear setting in [SAP$^+$13, FY14].

9.8 Proof of Proposition 9.3.7

We will first focus on the case when $a \leq h - 2\lceil h/g \rceil$ and later in Proposition 9.8.4 we will deal with the case $a > h - 2\lceil h/g \rceil$.

Proposition 9.8.1. *Suppose a, g, h be fixed constants such that $g \leq h$ and $a \leq h - 2\lceil h/g \rceil$. Let C be a maximally recoverable (n, r, h, a, q)-LRC where $r = n/g$ is the size of each local group. Then*

$$q \geq \Omega_{a,h,g}(n^{1+a/\lceil h/g \rceil}).$$

Proof. Let $t_1 \geq t_2 \geq \cdots \geq t_g$ be such that $t_i = \lceil h/g \rceil$ or $t_i = \lfloor h/g \rfloor$ and $\sum_{i=1}^{g} t_i = h$. Given a matrix M, we will denote its kernel by $\ker(M) = \{x : Mx = 0\}$ and its image by $\operatorname{Im}(M) = \{y : \exists x \text{ s.t. } Mx = y\}$. We call the subspace spanned by the rows of M as the row space of M and the subspace spanned by the columns of M as the column space of M and their dimensions are both equal to rank(M). Note that $\mathrm{Im}(M)$ is equal to the column space of M and $\ker(M)$ is the orthogonal subspace of the row space of M. $M^{\perp}$ is defined as a matrix with independent columns such that $\mathrm{Im}(M^{\perp}) = \ker(M)$ and so $MM^{\perp} = 0$. Note that $M^{\perp}$ is not unique, any matrix whose columns span $\ker(M)$ can be used as $M^{\perp}$, but the specific choice of $M^{\perp}$ is not important for the proof.

According to the discussion in Section 9.2 the code C admits a parity check matrix of the shape

$$\left[\begin{array}{c|c|c|c} A_1 & 0 & \cdots & 0 \\ \hline 0 & A_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_g \\ \hline B_1 & B_2 & \cdots & B_g \end{array}\right]. \tag{9.16}$$

Here $A_1, A_2, \cdots, A_g$ are $a \times r$ matrices over $\mathbb{F}_q$, $B_1, B_2, \cdots, B_g$ are $h \times r$ matrices over $\mathbb{F}_q$. The rest of the matrix is filled with zeros. Every $a \times a$ minor in each matrix $\{A_i\}_{i\in[g]}$ has full rank. So for every subset $S \subseteq [r]$ of size $|S| = a + t_i$, the matrix $A_i(S)$ is an $a \times (a + t_i)$ matrix of full rank. Let $A_i(S)^{\perp}$ be an $(a + t_i) \times t_i$ matrix of full rank such that $A_i(S)A_i(S)^{\perp} = 0$ (note that $A_i(S)^{\perp}$ is not unique). Now define

$$P_{i,S} = B_i(S)A_i(S)^{\perp}$$

which is a $h \times t_i$ matrix.

Define $p_{i,S}$ as the subspace of $\mathbb{F}_q^h$ spanned by the columns of $P_{i,S}$. The MR property implies that any subset of columns of the parity check matrix (9.16) which can be obtained by picking a columns in each local group and h arbitrary additional columns is full rank. We will use this property to make two claims about the subspaces $\{p_{i,S}\}$.

Claim 9.8.2. *For every subsets $S_1, \cdots, S_g \subseteq [r]$ such that $|S_i| = a + t_i$, the spaces $p_{1,S_1}, \ldots, p_{g,S_g}$ together span the entire space i.e. $p_{1,S_1} \oplus p_{2,S_2} \oplus \cdots \oplus p_{g,S_g} = \mathbb{F}_q^h$.*

Proof. Consider the following matrix equation:

$$\left[\begin{array}{c|c|c|c} A_{\ell_1}(S_1) & 0 & \cdots & 0 \\ \hline 0 & A_{\ell_2}(S_2) & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_{\ell_h}(S_h) \\ \hline B_{\ell_1}(S_1) & B_{\ell_2}(S_2) & \cdots & B_{\ell_h}(S_h) \end{array}\right] \left[\begin{array}{c|c|c|c} A_{\ell_1}(S_1)^{\perp} & 0 & \cdots & 0 \\ \hline 0 & A_{\ell_2}(S_2)^{\perp} & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A_{\ell_h}(S_h)^{\perp} \end{array}\right]$$

$$= \left[\begin{array}{c|c|c|c} 0 & 0 & \cdots & 0 \\ \hline 0 & 0 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & 0 \\ \hline P_{\ell_1,S_1} & P_{\ell_2,S_2} & \cdots & P_{\ell_h,S_h} \end{array}\right].$$

Let us denote the matrices in the above equation by M_1, M_2, M_3 such that the above equation becomes $M_1 M_2 = M_3$. By MR property, when we erase the coordinates corresponding to $S_1, \cdots, S_g$ in groups $1, \cdots, g$ respectively, the resulting erasure pattern is correctable. This implies that the $(ag+h) \times (ag+h)$ matrix M_1 is full rank. Also M_2 has full column rank because of its block structure. So M_3, which is an $(ag+h) \times h$ matrix, should have full column rank. This proves the required statement since $p_{i,S}$ is the column space of $P_{i,S}$. □

The above claim in particular implies that the matrices $P_{i,S}$ have full rank and that $p_{i,S}$ is a t_i-dimensional subspace of $\mathbb{F}_q^h$ for every i and S. The following claim explains for a fixed i, how subspaces $\{p_{i,S} : |S| = a + t_i\}$ intersect with each other.

Claim 9.8.3. *Let $i \in [g]$ and S, T be subsets of $[r]$ of size $a + t_i$ such that $|S \cap T| = \ell$.*

1. *If $\ell \leq a$ then $p_{i,S} \cap p_{i,T} = \phi$.*

2. If $\ell = a + \ell'$ *for* $\ell' \geq 1$ *then* $\dim(p_{i,S} \cap p_{i,T}) = \ell'$.

Proof. Consider the following matrix equation:

$$\left[\begin{array}{c|c} A_i(S) & A_i(T) \\ \hline B_i(S) & B_i(T) \end{array}\right] \left[\begin{array}{c|c} A_i(S)^\perp & 0 \\ \hline 0 & A_i(T)^\perp \end{array}\right] = \left[\begin{array}{c|c} 0 & 0 \\ \hline P_{i,S} & P_{i,T} \end{array}\right].$$

Let us denote the matrices that appear in the above equation to be M_1, M_2, M_3 in that order so that above equation becomes $M_1 M_2 = M_3$. The matrix M_1 is an $(a+h) \times 2(a+t_i)$ matrix of rank $|S \cup T| = 2(a+t_i) - \ell$. This is because any $a+h$ columns of $\begin{bmatrix} A_i \\ B_i \end{bmatrix}$ are linearly independent by MR property and $|S \cup T| \leq 2(a+t_i) \leq a+h$ by the assumption that $a \leq h - 2\lceil h/g \rceil$. Wlog, we can reorder the columns of M_1 such that the first ℓ columns of $\begin{bmatrix} A_i(S) \\ B_i(S) \end{bmatrix}$ and $\begin{bmatrix} A_i(T) \\ B_i(T) \end{bmatrix}$ are identical. M_2 is an $2(a+t_i) \times 2t_i$ matrix of full rank. M_3 is an $(a+h) \times 2t_i$ matrix and $\dim(p_{i,S} \cap p_{i,T}) = 2t_i - \text{rank}(M_3) = \dim(\ker(M_3))$. Since $\ker(M_2) = \phi$,

$$\dim(p_{i,S} \cap p_{i,T}) = \dim(\ker(M_3)) = \dim(\text{Im}(M_2) \cap \ker(M_1)).$$

Case 1: $|S \cap T| = \ell \leq a$

We need to show that $\text{Im}(M_2) \cap \ker(M_1) = \phi$. Suppose there is a non-zero vector in $\text{Im}(M_2) \cap \ker(M_1)$, say β. We completely understand the kernel of M_1, the only linear dependencies of the columns of M_1 occur because of repetitions i.e.

$$\ker(M_1) = \text{span}\{e_1 - e_{a+t_i+1}, \ldots, e_\ell - e_{a+t_i+\ell}\}.$$

So the first half of β is a non-zero vector in $\text{Im}(A_i(S)^\perp) = \ker(A_i(S))$ which is supported on the first ℓ coordinates. But we know that any a columns of $A_i(S)$ are linearly independent and so its kernel cannot contain any non-zero ℓ-sparse vector when $\ell \leq a$, leading to a contradiction.

Case 2: $|S \cap T| = \ell = a + \ell'$

We need to show that $\dim(\mathrm{Im}(M_2) \cap \ker(M_1)) = \ell'$.

- We will first show that $\dim(\mathrm{Im}(M_2) \cap \ker(M_1)) \geq \ell'$.

  We will exhibit ℓ' linearly independent vectors in $\mathrm{Im}(M_2) \cap \ker(M_1)$. The first a columns of $A_i(S)$ are linearly independent. So the next ℓ' columns of $A_i(S)$ can be written as linear combinations of them. This gives ℓ' linearly independent vectors in $\ker(A_i(S)) = \mathrm{Im}(A_i(S)^{\perp})$, call them $\alpha_1, \ldots, \alpha_{\ell'}$. Since the first $a + \ell'$ columns of $A_i(S)$ and $A_i(T)$ are the same, the vectors $\alpha_1, \ldots, \alpha_{\ell'}$ are also in $\ker(A_i(T)) = \mathrm{Im}(A_i(T)^{\perp})$. Thus the vectors $\begin{bmatrix} \alpha_1 \\ -\alpha_1 \end{bmatrix}, \cdots, \begin{bmatrix} \alpha_{\ell'} \\ -\alpha_{\ell'} \end{bmatrix}$ are in the column space of M_2. But since $\alpha_1, \cdots, \alpha_{\ell'}$ are supported on the first $a + \ell'$ coordinates and the first $a + \ell'$ columns of $\begin{bmatrix} A_i(S) \\ B_i(S) \end{bmatrix}$ and $\begin{bmatrix} A_i(T) \\ B_i(T) \end{bmatrix}$ are identical, it is easy to see that $\begin{bmatrix} \alpha_1 \\ -\alpha_1 \end{bmatrix}, \cdots, \begin{bmatrix} \alpha_{\ell'} \\ -\alpha_{\ell'} \end{bmatrix}$ are in the kernel of M_1. Moreover these vectors are linearly independent because $\alpha_1, \cdots, \alpha_{\ell'}$ are linearly independent. This proves that $\dim(\mathrm{Im}(M_2) \cap \ker(M_1)) \geq \ell'$.

- We now show that $\dim(\mathrm{Im}(M_2) \cap \ker(M_1)) \leq \ell'$.

  Suppose $\dim(\mathrm{Im}(M_2) \cap \ker(M_1)) = \ell'' \geq \ell' + 1$. So $\mathrm{Im}(M_2) \cap \ker(M_1)$ contains a non-zero vector, say β, whose first $\ell'' - 1$ coordinates are zero. Since

  $$\beta \in \ker(M_1) = \mathrm{span}\{e_1 - e_{a+t_i+1}, \ldots, e_\ell - e_{a+t_i+\ell}\},$$

  and the first $\ell'' - 1$ coordinates of β are zero,

  $$\beta \in \mathrm{span}\{e_{\ell''} - e_{a+t_i+\ell''}, \ldots, e_\ell - e_{a+t_i+\ell}\}.$$

Since $\beta \in \operatorname{Im}(M_2)$, the first half of β is a non-zero vector in $\operatorname{Im}(A_i(S)^{\perp})$ supported on $\ell - (\ell'' - 1) \leq a$ coordinates. This is a contradiction because any a columns of $A_i(S)$ are linearly independent and thus $\operatorname{Im}(A_i(S)^{\perp}) = \ker(A_i(S))$ cannot contain a non-zero a-sparse vector. □

Now we will show that if $q = o_{a,g,h}(n^{1+a/\lceil h/g \rceil})$ then a random $(h-1)$-dimensional subspace of $\mathbb{F}_q^h$ will contain $p_{1,S_1}, p_{2,S_2}, \ldots, p_{g,S_g}$ for some subsets $S_1, \ldots, S_g \subset [r]$ with $|S_i| = a + t_i$ with high probability, which contradicts Claim 9.8.2. Let f be a uniformly random vector in $\mathbb{F}_q^h$ and let $F = \{x \in F_q^h : \langle x, f \rangle = 0\}$ i.e. the set of vectors orthogonal to f. If $f \neq 0$, then F is a $(h-1)$-dimensional subspace and if $f = 0$ then $F = \mathbb{F}_q^h$. We want to calculate the probability that F contains $p_{1,S_1}, p_{2,S_2}, \ldots, p_{g,S_g}$ for some subsets $S_1, \ldots, S_g$ conditioned on F not being the entire space i.e. $f \neq 0$. Let's ignore the conditioning for now and estimate the required probability.

Fix some $i \in [g]$. Let Z_i be the number of subspaces among $\{p_{i,S} : S \in \binom{[r]}{a+t_i}\}$ which are contained in F. We have $\Pr[Z_i > 0] \geq \mathbb{E}[Z_i]^2/\mathbb{E}[Z_i^2]$. The probability that F contains a fixed $p_{i,S}$ which is a t_i-dimensional subspace is $1/q^{t_i}$. Therefore,

$$\mathbb{E}[Z_i] = \sum_{S \subset [r], |S| = a+t_i} \Pr[p_{i,S} \in F] = \frac{\binom{r}{a+t_i}}{q^{t_i}}.$$

$$\begin{aligned}
\mathbb{E}[Z_i^2] &= \sum_{S,T \in \binom{r}{a+t_i}} \Pr[p_{i,S}, p_{i,T} \in F] \\
&= \sum_{\ell=0}^{a} \sum_{S,T : |S \cap T| = \ell} \Pr[p_{i,S}, p_{i,T} \in F] + \sum_{\ell'=1}^{t_i} \sum_{S,T : |S \cap T| = a+\ell'} \Pr[p_{i,S}, p_{i,T} \in F].
\end{aligned}$$

By Claim 9.8.3, if $|S \cap T| \leq a$, then $p_{i,S} \cap p_{i,T} = \phi$ and so

$$\Pr[p_{i,S}, p_{i,T} \in F] = \frac{1}{q^{2t_i}}.$$

And if $|S \cap T| = a + \ell'$ then $\dim(p_{i,S} \cap p_{i,T}) = \ell'$ and so

$$\Pr[p_{i,S}, p_{i,T} \in F] = \frac{1}{q^{2t_i - \ell'}}.$$

Therefore,

$$\begin{aligned}\mathbb{E}[Z_i^2] =& \sum_{\ell=0}^{a} \binom{r}{a+t_i} \binom{r-(a+t_i)}{a+t_i-\ell} \binom{a+t_i}{\ell} \frac{1}{q^{2t_i}} \\ &+ \sum_{\ell'=0}^{t_i} \binom{r}{a+t_i} \binom{r-(a+t_i)}{t_i-\ell'} \binom{a+t_i}{a+\ell'} \frac{1}{q^{2t_i-\ell'}}.\end{aligned}$$

Therefore,

$$\frac{\mathbb{E}[Z_i^2]}{\mathbb{E}[Z_i]^2} = 1 + \sum_{\ell'=1}^{t_i} (c_{\ell'} + o_{a,g,h}(1)) \frac{q^{\ell'}}{n^{a+\ell'}} + o_{a,g,h}(1)$$

where $c_{\ell'}$ are constants depending only on a, g, h and indepedent of n, q.

When $q = o_{a,g,h}(n^{1+a/t_i})$, which is true since $t_i \le \lceil h/g \rceil$, $E[Z_i^2]/E[Z_i]^2 = 1 + o(1)$ and so $\Pr[Z_i > 0] = 1 - o(1)$. By union bound, $\Pr[\forall i \in [g], Z_i > 0] = 1 - o(1)$. Note that q should grow with n to have enough subspaces for Claim 9.8.3 to hold. Therefore $\Pr[f = 0] = 1/q^h = o(1)$. So

$$\Pr\left[\forall i \in [g], Z_i > 0 \middle| f \neq 0\right] \ge \Pr[\forall i \in [g], Z_i > 0] - \Pr[f = 0] = 1 - o(1)$$

which implies the required contradiction. □

Using the suggestion of Parikshit Gopalan [Gop17], we can generalize Proposition 9.8.1 to the case when $a > h - 2\lceil h/g \rceil$. In this case, we modify the proof of Proposition 9.8.1 where we only consider sets S_i that have size $a + t_i$ but are constrained to contain the set $\{1, 2, \ldots, a + 2t_i - h\}$, as this ensures that pairwise unions still have size at most $a + h$. Clearly, the total number of such sets is $\binom{r-a+h-2t_i}{h-t_i}$. The rest of the proof remains the same and yields the following:

Proposition 9.8.4. *Assume a, h, g are fixed constants such that $a \geq h - 2\lceil h/g \rceil$ and $h \geq g$ then any maximally recoverable (n, r, h, a, q)-local reconstruction code with $g = n/r$ local groups must have*

$$q \geq \Omega_{a,h,g}(n^{h/\lceil h/g \rceil - 1}). \tag{9.17}$$

Proof of Proposition 9.3.7. Follows immediately from Propositions 9.8.1 and 9.8.4. □

9.9 Determinantal identities

For our constructions, we will need some determinantal identities which we prove here. We need the following expansion of determinant of a column partitioned matrix.

Lemma 9.9.1. *For $i \in [\ell]$, let F_i be an $h \times t_i$ matrix with $\sum_{i=1}^{\ell} t_i = h$. Then,*

$$\det[F_1|F_2|\cdots|F_\ell] = \sum_{S_1 \sqcup \cdots \sqcup S_\ell = [h], |S_i| = t_i} \operatorname{sgn}(S_1, \cdots, S_\ell) \prod_{i \in [\ell]} \det F_i^{(S_i)}$$

where $S_1 \sqcup \cdots \sqcup S_\ell$ ranges over partitions of $[h]$ such that $|S_i| = t_i$. Here $\operatorname{sgn}(S_1, \cdots, S_\ell)$ is the sign of the permutation taking $(1, 2, \cdots, h)$ to $(\tilde{S}_1, \tilde{S}_2, \cdots, \tilde{S}_\ell)$ where $\tilde{S}_i$ is the tuple formed by ordering the elements of S_i in increasing order.

Proof. Given distinct integers $a_1, \cdots, a_n$, define $\operatorname{sgn}(a_1, a_2, \cdots, a_n) := (-1)^t$ where t is number of transpositions needed to sort the elements $a_1, a_2, \cdots, a_n$ in increasing order. Thus for a permutation $\pi \in S_h$, $\operatorname{sgn}(\pi) = \operatorname{sgn}(\pi(1), \pi(2), \cdots, \pi(h))$. Let $F = [F_1|F_2|\cdots|F_\ell]$ and for $i \in [\ell]$, let $T_i = \{t_{i-1} + 1, \cdots, t_i\}$ where $t_0 = 0$. We can expand $\det(F)$ as:

$$\begin{aligned}\det(F) &= \sum_{\pi \in S_h} \operatorname{sgn}(\pi) \prod_{i=1}^{h} F_{\pi(i)i} \\ &= \sum_{S_1 \sqcup \cdots \sqcup S_\ell = [h], |S_i| = t_i} \quad \sum_{\pi:\ \pi(T_i) = S_i} \operatorname{sgn}(\pi) \prod_{i=1}^{h} F_{\pi(i)i}\end{aligned}$$

Note that if $\pi(T_i) = S_i$, then for $i \in [\ell]$,

$$\operatorname{sgn}(\pi) = \operatorname{sgn}(\tilde{S}_1, \cdots, \tilde{S}_\ell) \prod_{i=1}^{\ell} \operatorname{sgn}(\pi(t_{i-1}+1), \cdots, \pi(t_i))$$

because we can sort $(\pi(1), \cdots, \pi(h))$ first within each group to get $(\tilde{S}_1, \cdots, \tilde{S}_\ell)$ and then sort it to get $(1, 2, \cdots, h)$. Therefore,

$$\begin{aligned}&\sum_{\pi:\ \pi(T_i)=S_i} \operatorname{sgn}(\pi) \prod_{i=1}^{h} F_{\pi(i)i} \\ &= \sum_{\sigma_1 : T_1 \to S_1, \ldots,\ \sigma_\ell : T_\ell \to S_\ell} \operatorname{sgn}(\tilde{S}_1, \cdots, \tilde{S}_\ell) \prod_{i=1}^{\ell} \left(\operatorname{sgn}(\sigma_i(t_{i-1}+1), \cdots, \sigma_i(t_i)) \prod_{j=t_{i-1}+1}^{t_i} F_{\sigma_i(j)j} \right) \\ &\qquad\qquad \text{(where the summation is over all bijections } \sigma_i : T_i \to S_i\text{)} \\ &= \operatorname{sgn}(\tilde{S}_1, \cdots, \tilde{S}_\ell) \prod_{i=1}^{\ell} \left(\sum_{\sigma_i : T_i \to S_i} \operatorname{sgn}(\sigma_i(t_{i-1}+1), \cdots, \sigma_i(t_i)) \prod_{j=t_{i-1}+1}^{t_i} F_{\sigma_i(j)j} \right) \\ &= \operatorname{sgn}(\tilde{S}_1, \cdots, \tilde{S}_\ell) \prod_{i=1}^{\ell} \det F_i^{(S_i)}.\end{aligned}$$

□

Lemma 9.9.2. *For $i \in [\ell]$, let C_i be an $a \times (a + t_i)$ matrix and D_i be an $h \times (a + t_i)$ matrix for some $t_1 + t_2 + \cdots + t_\ell = h$ where $t_i \geq 1$. Then,*

$$\det \left[\begin{array}{c|c|c|c} C_1 & 0 & \cdots & 0 \\ \hline 0 & C_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & C_\ell \\ \hline D_1 & D_2 & \cdots & D_\ell \end{array}\right]$$

$$= (-1)^{a \sum_{i=1}^{\ell} t_i(\ell - i)} \sum_{\substack{S_1 \sqcup \cdots \sqcup S_\ell = [h] \\ |S_i| = t_i}} \operatorname{sgn}(S_1, \cdots, S_\ell) \prod_{i \in [\ell]} \det \begin{bmatrix} C_i \\ D_i^{(S_i)} \end{bmatrix}$$

where $S_1 \sqcup \cdots \sqcup S_\ell$ ranges over partitions of $[h]$ such that $|S_i| = t_i$ and $\operatorname{sgn}(S_1, \cdots, S_\ell)$ is defined as in Lemma 9.9.1.

Proof. Let

$$F = [F_1 | F_2 | \cdots | F_\ell] = \left[\begin{array}{c|c|c|c} C_1 & 0 & \cdots & 0 \\ \hline 0 & C_2 & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & C_\ell \\ \hline D_1 & D_2 & \cdots & D_\ell \end{array}\right].$$

Let $[p, q]$ be the integers between p and q, i.e., $[p, q] = \{i : p \leq i \leq q\}$. By Lemma 9.9.1,

$$\det F = \det[F_1 | F_2 | \cdots | F_\ell] = \sum_{T_1 \sqcup \cdots \sqcup T_\ell = [a\ell + h], |T_i| = a + t_i} \operatorname{sgn}(T_1, \cdots, T_\ell) \prod_{i \in [\ell]} \det F_i^{(T_i)}$$

Note that the only terms which survive correspond to partitions $T_1 \sqcup T_2 \sqcup \cdots \sqcup T_\ell$ of rows of F such that for every $i \in [\ell]$, T_i contains the rows of C_i (i.e. $[(i-1)a + 1, ia]$).

In the other terms, there exists some $i \in [\ell]$ such that $F_i^{(T_i)}$ contains a zero row and thus $\det F_i^{(T_i)} = 0$. Such partitions are given by $T_i = [(i-1)a+1, ia] \cup S_i$ where $S_1 \sqcup S_2 \cdots \sqcup S_\ell$ is some partition of rows of $[D_1|D_2|\cdots|D_\ell]$ such that $|S_i| = t_i$. So the expansion for $\det F$ can be written as:

$$\begin{aligned}
&\det F \\
&= \sum_{\substack{S_1 \sqcup \cdots \sqcup S_\ell = [a\ell+1, a\ell+h] \\ |S_i| = t_i}} \operatorname{sgn}([1,a] \cup S_1, \cdots, [(\ell-1)a+1, \ell a] \cup S_\ell) \prod_{i \in [\ell]} \det F_i^{([(i-1)a, ia] \cup S_i)} \\
&= (-1)^{a(\sum_{i=1}^{\ell} t_i(\ell - i))} \sum_{\substack{S_1 \sqcup \cdots \sqcup S_\ell = [a\ell+1, a\ell+h] \\ |S_i| = t_i}} \operatorname{sgn}([1, \ell a], S_1, S_2, \cdots S_\ell) \prod_{i \in [\ell]} \det F_i^{([(i-1)a, ia] \cup S_i)} \\
&= (-1)^{a(\sum_{i=1}^{\ell} t_i(\ell - i))} \sum_{S_1 \sqcup \cdots \sqcup S_\ell = [h], |S_i| = t_i} \operatorname{sgn}(S_1, S_2, \cdots S_\ell) \prod_{i \in [\ell]} \det \begin{bmatrix} C_i \\ D_i^{(S_i)} \end{bmatrix}. \qquad \square
\end{aligned}$$

We will now prove Lemma 9.4.1, which was used in our constructions in Sections 9.4 and 9.5.

Proof of Lemma 9.4.1. After applying Lemma 9.9.2, we just need to note that

$$\sum_{S_1 \sqcup \cdots \sqcup S_h = [h], |S_i| = 1} \operatorname{sgn}(S_1, \cdots, S_\ell) \prod_{i \in [\ell]} \det \begin{bmatrix} C_i \\ D_i^{(S_i)} \end{bmatrix} = \sum_{\pi} \operatorname{sgn}(\pi) \prod_{i \in [h]} \det \begin{bmatrix} C_i \\ D_i^{(\pi(i))} \end{bmatrix}$$

where the last summation is over all permutations π of h elements which is the exactly the required determinant. □

9.10 Proof of Lemma 9.4.3

The goal of the section is to prove Lemma 9.4.3 which is restated here for convenience.

Lemma 9.10.1 (Restatement of Lemma 9.4.3)**.** *Let r, n be some positive integers with $r \leq n$. Then there exists a finite field $\mathbb{F}_q$ with $q = O(n)$ such that the multiplicative group $\mathbb{F}_q^*$ contains a subgroup of size at least r and with at least n/r cosets. If additionally we require that the field has characteristic two, then such a field exists with $q = n \cdot \exp(O(\sqrt{\log n}))$.*

We will need some estimates from analytic number theory, we will setup some notation first.

$$\begin{aligned}
&\pi(x; m, a) : \text{number of primes } p \leq x \text{ such that } p \equiv a \mod m\\
&\pi(x, y; m, a) = \pi(y; m, a) - \pi(x; m, a)\\
&\mathrm{Li}(x) = \int_2^x \frac{1}{\ln t} dt\\
&(a, m) : \text{ greatest common divisor of } a \text{ and } m\\
&\phi(m) : \text{ number of integers } 1 \leq a \leq m \text{ s.t. } (a, m) = 1 \text{ (Euler's totient function)}
\end{aligned}$$

By the prime number theorem, the number of primes $\leq x$ is approximately $\mathrm{Li}(x) = \Theta(x/\log x)$. So if the primes are equidistributed among different congruence classes of m with no obvious divisors (i.e. $a \mod m$ where $(a, m) = 1$), then we expect to see approximately $\mathrm{Li}(x)/\phi(m)$ primes in each such congruence class. The following theorem gives an upper bound on the error term in this approximation averaged over $m < \sqrt{x}(\log x)^A$.

Theorem 9.10.2 (Theorem from [BFI86] (Page 250))**.** *Let $a \neq 0, A \geq 0$ be some fixed constants and $x \geq 3$. We then have*

$$\sum_{(m,a)=1;\ m<\sqrt{x}(\log x)^A} \left| \pi(x; m, a) - \frac{\mathrm{Li}(x)}{\phi(m)} \right| \lesssim_{a,A} x \frac{(\log \log x)^B}{(\log x)^3}$$

where B is an absolute constant.

Applying the above theorem with $a = 1, A = 0$ for x and $2x$, and using triangle inequality, we get the following corollary.

Corollary 9.10.3. *For x large enough,*

$$\sum_{m<\sqrt{x}} \left| \pi(x, 2x; m, 1) - \frac{(\mathrm{Li}(2x) - \mathrm{Li}(x))}{\phi(m)} \right| \lesssim x \frac{(\log\log x)^B}{(\log x)^3}$$

where B is an absolute constant.

Lemma 9.10.4. *Let $a \leq b$ be some positive integers. Then there exists $A \geq a$,$B \geq b$ such that $AB + 1$ is a prime and $AB = O(ab)$.*

Proof. If there exists some A such that $a \leq A \leq 2a$ and there is a prime p between $4ab+1$ and $8ab$ which is congruent to $1 \mod A$, then we can take $B = (p-1)/A \geq b$. Suppose this is not true, we will arrive at a contradiction. For every $a \leq m \leq 2a$, we have $\pi(4ab, 8ab; m, 1) = 0$. Applying corollary 9.10.3 with $x = 4ab$, we get

$$\begin{aligned}
ab\frac{(\log\log ab)^B}{(\log ab)^3} &\gtrsim \sum_{m<2\sqrt{ab}} \left| \pi(4ab, 8ab; m, 1) - \frac{(\mathrm{Li}(8ab) - \mathrm{Li}(4ab))}{\phi(m)} \right| \\
&\geq \sum_{a\leq m<2a} \left| \pi(4ab, 8ab; m, 1) - \frac{(\mathrm{Li}(8ab) - \mathrm{Li}(4ab))}{\phi(m)} \right| \\
&= \sum_{a\leq m<2a} \frac{(\mathrm{Li}(8ab) - \mathrm{Li}(4ab))}{\phi(m)} \\
&\geq a\frac{\mathrm{Li}(8ab) - \mathrm{Li}(4ab)}{2a} \gtrsim \frac{ab}{\log(ab)}
\end{aligned}$$

which is a contradiction when ab is large enough. □

In practice, it is desirable to work with fields of characteristic two, the following lemma gives us such fields.

Lemma 9.10.5. *Let a, b be some positive integers and let $n = ab$. Then there exists $A \geq a$, $B \geq b$ such that $q = AB + 1$ is a power of two and $q = n \cdot \exp(O\sqrt{\log n})$.*

Proof. Let m be a positive integer to be chosen later. Let ℓ be an integer such that

$$2^{\ell(2^m-1)} \geq Cn + 1 > 2^{(\ell-1)(2^m-1)}$$

where $C \geq 1$ is some sufficiently large constant to be chosen later and let $x = 2^\ell, q = x^{2^m}$. We will now show that for any $a \leq n$, we can factor $q - 1$ as $A \cdot B$ where $A \geq a$ and $B \geq n/a = b$. We can factor $q - 1 = x^{2^m} - 1$ as:

$$x^{2^m} - 1 = (x-1) \prod_{i \in [m]} (1 + x^{2^{i-1}}).$$

We will rearrange these factors to get the desired factorization of $q - 1$. Let $0 \leq \alpha \leq 2^m - 1$ be such that $x^{\alpha-1} < a \leq x^\alpha$. Expand α into its binary expansion as $\alpha = \sum_{i \in S} 2^i$ where $S \subset \{0, 1, \cdots, m-1\}$. Define $A = \prod_{i \in S}(1 + x^{2^i})$ and define $B = (x^{2^m} - 1)/A$. Clearly $A \geq x^\alpha \geq a$. We can lower bound B as follows:

$$\begin{aligned}
B = \frac{(x^{2^m} - 1)}{\prod_{i \in S}(1 + x^{2^i})} &= \prod_{i \in S}(1 + x^{-2^i})^{-1} \cdot \frac{(x^{2^m} - 1)}{\prod_{i \in S} x^{2^i}} \\
&\geq \exp(-\sum_{j \geq 0} x^{-2^j}) \frac{(x^{2^m} - 1)}{x^\alpha} \geq \exp(-\sum_{j \geq 0} 2^{-2^j}) \frac{(x^{2^m} - 1)}{xa} \\
&\geq \exp(-\sum_{j \geq 0} 2^{-2^j}) \frac{(x^{2^m - 1} - 1)}{a} \geq \exp(-\sum_{j \geq 0} 2^{-2^j}) \frac{Cn}{a} \geq \frac{n}{a}
\end{aligned}$$

when $C = \exp(\sum_{j \geq 0} 2^{-2^j})$. Now we need to bound $q = x^{2^m}$ as a function of n.

$$\begin{aligned}
q = 2^{\ell 2^m} &= 2^{(\ell-1)(2^m-1)} \cdot 2^\ell \cdot 2^{2^m - 1} \\
&\leq (Cn + 1) \cdot 2^\ell \cdot 2^{2^m - 1} \\
&\lesssim n^{1 + 1/(2^m - 1)} \cdot 2^{2^m - 1} \\
&\lesssim n \exp(O(\sqrt{\log n}))
\end{aligned}$$

if we choose m such that $(2^m - 1) = \Theta(\sqrt{\log n})$.

□

We are now ready to prove Lemma 9.4.3.

Proof of Lemma 9.4.3. By Lemma 9.10.4, there exists $A \geq r$ and $B \geq n/r$ such that $q = AB + 1$ is prime and $q = O(n)$. Since $\mathbb{F}_q^*$ is a cyclic group of size $q - 1$ and A divides $q - 1$, there exists a subgroup of $\mathbb{F}_q^*$ of size $A \geq r$ with $B \geq n/r$ cosets. To get a finite field of characteristic two, we use Lemma 9.10.5 instead. □

Bibliography

[Aar18] Scott Aaronson. PDQP/qpoly = ALL. *arXiv preprint arXiv:1805.08577*, 2018.

[AEL95] Noga Alon, Jeff Edmonds, and Michael Luby. Linear time erasure codes with nearly optimal recovery. In *proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 512–519. IEEE Computer Society, 1995.

[ALM$^+$98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[ALRW17] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 47–66. Society for Industrial and Applied Mathematics, 2017.

[Amb97] Andris Ambainis. Upper bound on communication complexity of private information retrieval. In *ICALP*, pages 401–407, 1997.

[AR94] Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Structures & Algorithms*, 5, 1994.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[AS03] Sanjeev Arora and Madhu Sudan. Improved low-degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003.

[Bal12] Simeon Ball. On sets of vectors of a finite vector space in which every subset of basis size is a basis. *Journal of European Mathematical Society*, 14:733–748, 2012.

[BARDW08] Avraham Ben-Aroya, Oded Regev, and Ronald De Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and ldcs. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 477–486. IEEE, 2008.

[BB15] Arnab Bhattacharyya and Abhishek Bhowmick. Using higher-order Fourier analysis over general fields. *Preprint arXiv:1505.00619*, 2015.

[BDG17] Jop Briët, Zeev Dvir, and Sivakanth Gopi. Outlaw distributions and locally decodable codes. In *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, pages 20:1–20:19, 2017.

[BDL13] A. Bhowmick, Z. Dvir, and S. Lovett. New Bounds for Matching Vector Families. In *Proceedings of the 45th Annual ACM Symposium on Symposium on Theory of Computing*, STOC '13, pages 823–832, 2013.

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

[BDSS16] 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.

[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 *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 519–528. ACM, 2011.

[Beh46] Felix A Behrend. On sets of integers which contain no three terms in arithmetical progression. *Proceedings of the National Academy of Sciences*, 32(12):331–332, 1946.

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

[BFI86] Enrico Bombieri, John B Friedlander, and Henryk Iwaniec. Primes in arithmetic progressions to large moduli. *Acta Mathematica*, 156(1):203–251, 1986.

[BFLS91] László Babai, Lance Fortnow, Leonid A. Levin, and Mario Szegedy. Checking computations in polylogarithmic time. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 21–31. ACM Press, 1991.

[BG17a] Arnab Bhattacharyya and Sivakanth Gopi. Lower bounds for constant query affine-invariant LCCs and LTCs. *TOCT*, 9(2):7:1–7:17, 2017. Preliminary version appeared in CCC'16.

[BG17b] Jop Briët and Sivakanth Gopi. Gaussian width bounds with applications to arithmetic progressions in random settings. *CoRR*, abs/1711.05624, 2017. Available at `http://arxiv.org/abs/1711.05624`.

[BG18] Jop Briët and Sivakanth Gopi. Personal communication, 2018.

[BGLZ17] Bhaswar Bhattacharya, Shirshendu Ganguly, Eyal Lubetzky, and Yufei Zhao. Upper tails and independence polynomials in random graphs. *Advances in Mathematics*, 319:313–347, 2017.

[BGSZ18] Bhaswar Bhattacharya, Shirshendu Ganguly, Xuancheng Shao, and Yufei Zhao. Upper tails for arithmetic progressions in a random set. *International Mathematics Research Notices*, 2018. To appear. Available at arXiv preprint: 1605.02994.

[BGT17] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query LCCs over large alphabet. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2017, August 16-18, 2017, Berkeley, CA, USA*, pages 30:1–30:20, 2017.

[BHH13] Mario Blaum, James Lee Hafner, and Steven Hetzler. Partial-MDS codes and their application to RAID type of architectures. *IEEE Transactions on Information Theory*, 59(7):4510–4519, 2013.

[BI01] Amos Beimel and Yuval Ishai. Information-theoretic private information retrieval: A unified construction. In *ICALP*, pages 912–926, 2001.

[BIKR02] Amos Beimel, Yuval Ishai, Eyal Kushilevitz, and Jean-François Raymond. Breaking the $o(n^{1/(2k-1)})$ barrier for information-theoretic private information retrieval. In *FOCS*, pages 261–270, 2002.

[BIW07] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private PIR. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 311–325. Springer, 2007.

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

[BL96] Vitaly Bergelson and Alexander Leibman. Polynomial extensions of van der Waerden's and SzemerédiÃćÂĂÂŹs theorems. *Journal of the American Mathematical Society*, 9(3):725–753, 1996.

[BL15] Abhishek Bhowmick and Shachar Lovett. Bias vs structure of polynomials in large fields, and applications in effective algebraic geometry and coding theory. *Preprint arXiv:1506.02047*, 2015.

[BL18] Abhishek Bhowmick and Shachar Lovett. The list decoding radius for reed muller codes over small fields. *IEEE Transactions on Information Theory*, 2018.

[Bla13] Mario Blaum. Construction of PMDS and SD codes extending raid 5. *arXiv preprint arXiv:1305.0032*, 2013.

[Blo16] Thomas F Bloom. A quantitative improvement for Roth's theorem on arithmetic progressions. *Journal of the London Mathematical Society*, page jdw010, 2016.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[BNR12] Jop Briët, Assaf Naor, and Oded Regev. Locally decodable codes and the failure of cotype for projective tensor products. *Electronic Research Announcements in Mathematical Sciences (ERA-MS)*, 19:120–130, 2012.

[BPSY16] Mario Blaum, James Plank, Moshe Schwartz, and Eitan Yaakobi. Construction of partial MDS and sector-disk codes with two global parity symbols. *IEEE Transactions on Information Theory*, 62(5):2673–2681, 2016.

[BR16] Jop Briët and Shravas Rao. Arithmetic expanders and deviation bounds for random tensors. *arXiv preprint arXiv:1610.03428*, 2016.

[Bri16] Jop Briët. On embeddings of ℓ_1^k from locally decodable codes. *arXiv preprint: arXiv:1611.06385*, 2016.

[BS08] Eli Ben-Sasson and Madhu Sudan. Short PCPs with polylog query complexity. *SIAM Journal on Computing*, 38(2):551–607, 2008.

[BSGK$^+$10] Eli Ben-Sasson, Venkatesan Guruswami, Tali Kaufman, Madhu Sudan, and Michael Viderman. Locally testable codes require redundant testers. *SIAM J. Comput*, 39(7):3230–3247, 2010.

[BSRZS12] Eli Ben-Sasson, Noga Ron-Zewi, and Madhu Sudan. Sparse affine-invariant linear codes are locally testable. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 561–570. IEEE, 2012.

[BSS06] Eli Ben-Sasson and Madhu Sudan. Robust locally testable codes and products of codes. *Random Structures & Algorithms*, 28(4):387–402, 2006.

[BSS11] Eli Ben-Sasson and Madhu Sudan. Limits on the rate of locally testable affine-invariant codes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 412–423. Springer, 2011.

[CC18] L. Cao and Z. Chen. Partitions of the polytope of Doubly Substochastic Matrices. *ArXiv e-prints*, March 2018.

[CD16] Sourav Chatterjee and Amir Dembo. Nonlinear large deviations. *Advances in Mathematics*, 299:396–450, 2016.

[CFL$^+$13] Yeow Meng Chee, Tao Feng, San Ling, Huaxiong Wang, and Liang Feng Zhang. Query-efficient locally decodable codes of subexponential length. *Computational Complexity*, 22(1):159–189, 2013.

[CGKS98] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. *Journal of the ACM*, 45(6):965–981, 1998.

[CHL07] Minghua Chen, Cheng Huang, and Jin Li. On maximally recoverable property for multi-protection group codes. In *IEEE International Symposium on Information Theory (ISIT)*, pages 486–490, 2007.

[Chr11] Michael Christ. On random multilinear operator inequalities. *arXiv preprint: 1108.5655*, 2011.

[CK17] Gokhan Calis and Ozan Koyluoglu. A general construction fo PMDS codes. *IEEE Communications Letters*, 21(3):452–455, 2017.

[COOW12] Amin Coja-Oghlan, Mikael Onsjö, and Osamu Watanabe. Propagation connectivity of random hypergraphs. *The Electronic Journal of Combinatorics*, 19(1):P17, 2012.

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

[CT14] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. *arXiv preprint arXiv:1404.0654*, 2014.

[CZ17] David Conlon and Yufei Zhao. Quasirandom Cayley graphs. *Discrete Analysis*, 6, 2017. Available at arXiv:1603.03025 [math.CO].

[DG16] Zeev Dvir and Sivakanth Gopi. 2-Server PIR with subpolynomial communication. *J. ACM*, 63(4):39:1–39:15, September 2016. Preliminary version appeared in STOC 2015.

[DGW$^+$10] Alexandros G. Dimakis, Brighten Godfrey, Yunnan Wu, Martin J. Wainwright, and Kannan Ramchandran. Network coding for distributed storage systems. *IEEE Transactions on Information Theory*, 56(9):4539–4551, 2010.

[DGY10] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. In *FOCS*, pages 705–714, 2010.

[DH13] Z. Dvir and G. Hu. Matching-vector families and LDCs over large modulo. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (RANDOM-APPROX)*, volume 8096, pages 513–526. Springer Berlin Heidelberg, 2013.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM*, 54(3):12, 2007.

[DJK$^+$02] Amit Deshpande, Rahul Jain, Telikepalli Kavitha, Satyanarayana V Lokam, and Jaikumar Radhakrishnan. Better lower bounds for locally decodable codes. In *Computational Complexity, 2002. Proceedings. 17th IEEE Annual Conference on*, pages 184–193. IEEE, 2002.

[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.

[DSW14a] 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.

[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.

[Dud78] Richard M Dudley. Central limit theorems for empirical measures. *The Annals of Probability*, pages 899–929, 1978.

[Dvi11] Zeev Dvir. On matrix rigidity and locally self-correctable codes. *computational complexity*, 20(2):367–388, 2011.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *STOC*, pages 39–44, 2009.

[Eld16] Ronen Eldan. Gaussian-width gradient complexity, reverse log-Sobolev inequalities and nonlinear large deviations. *arXiv preprint: 1612.04346*, 2016.

[Elk11] Michael Elkin. An improved construction of progression-free sets. *Israel journal of mathematics*, 184(1):93–128, 2011.

[FLW12] Nikos Frantzikinakis, Emmanuel Lesigne, and Mate Wierdl. Random sequences and pointwise convergence of multiple ergodic averages. *Indiana University Mathematics Journal*, pages 585–617, 2012.

[FLW16a] Nikos Frantzikinakis, Emmanuel Lesigne, and Mate Wierdl. Random differences in szemerédi's theorem and related results. *Journal d'Analyse Mathématique*, 130(1):91–133, 2016.

[FLW16b] Nikos Frantzikinakis, Emmanuel Lesigne, and Mate Wierdl. Random differences in Szemerédi's theorem and related results. *Journal d'Analyse Mathématique*, 130(1):91–133, 2016.

[Fox17] Jacob Fox. Personal communication, 2017.

[FS95] Katalin Friedl and Madhu Sudan. Some improvements to total degree tests. In *proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems (ISTCS)*, pages 190–198. IEEE Computer Society, 1995.

[FY14] Michael Forbes and Sergey Yekhanin. On the locality of codeword symbols in non-linear codes. *Discrete mathematics*, 324:78–84, 2014.

[Gas04] William I. Gasarch. A survey on private information retrieval (column: Computational complexity). *Bulletin of the EATCS*, 82:72–107, 2004.

[GGY17] Sivakanth Gopi, Venkatesan Guruswami, and Sergey Yekhanin. On maximally recoverable local reconstruction codes. *CoRR*, abs/1710.10322, 2017. Available at `http://arxiv.org/abs/1710.10322`.

[GHJY14] Parikshit Gopalan, Cheng Huang, Bob Jenkins, and Sergey Yekhanin. Explicit maximally recoverable codes with locality. *IEEE Transactions on Information Theory*, 60(9):5245–5256, 2014.

[GHK$^+$17] Parikshit Gopalan, Guangda Hu, Swastik Kopparty, Shubhangi Saraf, Carol Wang, and Sergey Yekhanin. Maximally recoverable codes for grid-like topologies. In *28th Annual Symposium on Discrete Algorithms (SODA)*, pages 2092–2108, 2017.

[GHSY12] Parikshit Gopalan, Cheng Huang, Huseyin Simitci, and Sergey Yekhanin. On the locality of codeword symbols. *IEEE Trans. Information Theory*, 58(11):6925–6934, 2012.

[GI04] Venkatesan Guruswami and Piotr Indyk. Efficiently decodable codes meeting gilbert-varshamov bound for low rates. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2004, New Orleans, Louisiana, USA, January 11-14, 2004*, pages 756–757. SIAM, 2004.

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

[GKdO$^+$17] Sivakanth Gopi, Swastik Kopparty, Rafael Mendes de Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the gilbert-varshamov bound. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 2073–2091, 2017.

[GKS13] Alan Guo, Swastik Kopparty, and Madhu Sudan. New affine-invariant codes from lifting. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 529–540. ACM, 2013.

[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.

[Gop17] Parikshit Gopalan. Personal communication, 2017.

[Gow01] William T Gowers. A new proof of szemerédi's theorem. *Geometric & Functional Analysis GAFA*, 11(3):465–588, 2001.

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

[Gre06] Ben Green. Montreal lecture notes on quadratic Fourier analysis. *Preprint arXiv:math/0604089*, 2006.

[Gro99] Vince Grolmusz. Superpolynomial size set-systems with restricted intersections mod 6 and explicit ramsey graphs. *Combinatorica*, 20:2000, 1999.

[GS92] Peter Gemmell and Madhu Sudan. Highly resilient correctors for polynomials. *Information Processing Letters*, 43(4):169–174, 28 September 1992.

[GS99] Venkatesan Guruswami and Madhu Sudan. Improved decoding of reed-solomon and algebraic-geometry codes. *IEEE Trans. Information Theory*, 45(6):1757–1767, 1999.

[GS00] Venkatesan Guruswami and Madhu Sudan. List decoding algorithms for certain concatenated codes. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 181–190. ACM, 2000.

[GS01] Venkatesan Guruswami and Madhu Sudan. Extensions to the Johnson bound, 2001.

[GS02] Venkatesan Guruswami and Madhu Sudan. Decoding concatenated codes using soft information. In *IEEE Conference on Computational Complexity*, pages 148–157. IEEE Computer Society, 2002.

[GS06a] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost-linear length. *Journal of the ACM*, 53(4):558 – 655, July 2006.

[GS06b] Oded Goldreich and Madhu Sudan. Locally testable codes and PCPs of almost linear length. *Journal of ACM*, 53(4):558–655, 2006.

[GSVW14] Venkatesan Guruswami, Madhu Sudan, Ameya Velingker, and Carol Wang. Limitations on testable affine-invariant codes in the high-rate regime. In *Proceedings of the twenty-sixth annual ACM-SIAM symposium on Discrete algorithms*, pages 1312–1325. SIAM, 2014.

[Guo13] Alan Xinyu Guo. Some closure features of locally testable affine-invariant properties. Master's thesis, Massachusetts Institute of Technology, 2013.

[Gur06] Venkatesan Guruswami. Algorithmic results in list decoding. *Foundations and Trends in Theoretical Computer Science*, 2(2), 2006.

[GW16] Venkatesan Guruswami and Mary Wootters. Repairing Reed-Solomon codes. In *48th ACM Symposium on Theory of Computing (STOC)*, pages 216–226, 2016.

[GYBS17] Ryan Gabrys, Eitan Yaakobi, Mario Blaum, and Paul Siegel. Construction of partial MDS codes over small finite fields. In *2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1–5, 2017.

[HCL07] Cheng Huang, Minghua Chen, and Jin Li. Pyramid codes: flexible schemes to trade space for access efficiency in reliable data storage systems. In *6th IEEE International Symposium on Network Computing and Applications (NCA 2007)*, pages 79–86, 2007.

[HH11] Barry Hurley and Ted Hurley. Group ring cryptography. *CoRR*, abs/1104.1724, 2011.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc.*, 43:439–561, 2006.

[HOW15] Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Information and Computation*, 243:178–190, 2015.

[HSX$^+$12] Cheng Huang, Huseyin Simitci, Yikang Xu, Aaron Ogus, Brad Calder, Parikshit Gopalan, Jin Li, and Sergey Yekhanin. Erasure coding in Windows Azure Storage. In *USENIX Annual Technical Conference (ATC)*, pages 15–26, 2012.

[HY16] Guangda Hu and Sergey Yekhanin. New constructions of SD and MR codes over small finite fields. In *2016 IEEE International Symposium on Information Theory (ISIT)*, pages 1591–1595, 2016.

[IK04] Yuval Ishai and Eyal Kushilevitz. On the hardness of information-theoretic multiparty computation. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 439–455. Springer, 2004.

[IS10] Toshiya Itoh and Yasuhiro Suzuki. Improved constructions for query-efficient locally decodable codes of subexponential length. *IEICE Transactions*, 93-D(2):263–270, 2010.

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

[KKS13] Delaram Kahrobaei, Charalambos Koupparis, and Vladimir Shpilrain. Public key exchange using matrices over group rings. *Groups-Complexity-Cryptology*, 5(1):97–115, 2013.

[Kle16] Robert Kleinberg. A nearly tight upper bound on tri-colored sum-free sets in characteristic 2. *arXiv preprint arXiv:1605.08416*, 2016.

[KLP67] T. Kasami, S. Lin, and W.W. Peterson. Some results on cyclic codes which are invariant under the affine group and their applications. *Inform. and Comput.*, 11(5–6):475–496, 1967.

[KLR17] Daniel Kane, Shachar Lovett, and Sankeerth Rao. Labeling the complete bipartite graph with no zero cycles. In *58th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017.

[KMRS15] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally-correctable and locally-testable codes with sub-polynomial query complexity. *Electronic Colloquium on Computational Complexity (ECCC)*, 2015.

[KMRS17] Swastik Kopparty, Or Meir, Noga Ron-Zewi, and Shubhangi Saraf. High-rate locally correctable and locally testable codes with sub-polynomial query complexity. *J. ACM*, 64(2):11:1–11:42, 2017.

[KS07] Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *Foundations of Computer Science, 2007. FOCS'07. 48th Annual IEEE Symposium on*, pages 590–600. IEEE, 2007.

[KS08] Tali Kaufman and Madhu Sudan. Algebraic property testing: the role of invariance. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 403–412. ACM, 2008.

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

[KT00] Jonathan Katz and Luca Trevisan. On the efficiency of local decoding procedures for error-correcting codes. In *Proceedings of the 32nd annual ACM symposium on Theory of computing (STOC 2000)*, pages 80–86. ACM Press, 2000.

[KW04] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. of Computer and System Sciences*, 69:395–420, 2004. Preliminary version appeared in STOC'03.

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

[LN83] Rudolf Lidl and Harald Niederreiter. Finite fields. In Gian-Carlo Rota, editor, *Finite Fields*, volume 20 of *Encyclopedia of Mathematics and its Applications*. Addison-Wesley, Reading, Massachusetts, 1983.

[Lov18] Shachar Lovett. A proof of the GM-MDS conjecture. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:47, 2018.

[LT79] Joram Lindenstrauss and Lior Tzafriri. *Classical Banach spaces. II*, volume 97 of *Ergebnisse der Mathematik und ihrer Grenzgebiete [Results in Mathematics and Related Areas]*. Springer-Verlag, Berlin-New York, 1979. Function spaces.

[LT13] Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

[LVW17] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Conditional disclosure of secrets via non-linear reconstruction. In *Annual International Cryptology Conference*, pages 758–790. Springer, 2017.

[LVW18] Tianren Liu, Vinod Vaikuntanathan, and Hoeteck Wee. Towards breaking the exponential barrier for general secret sharing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 567–596. Springer, 2018.

[LZ17] Eyal Lubetzky and Yufei Zhao. On the variational problem for upper tails in sparse random graphs. *Random Structures & Algorithms*, 50(3):420–436, 2017.

[Mau03] Bernard Maurey. Type, cotype and K-convexity. In *Handbook of the geometry of Banach spaces, Vol. 2*, pages 1299–1332. North-Holland, Amsterdam, 2003.

[MBG$^+$13] A.J. Menezes, I.F. Blake, X.H. Gao, R.C. Mullin, S.A. Vanstone, and T. Yaghoobian. *Applications of Finite Fields*. The Springer International Series in Engineering and Computer Science. Springer US, 2013.

[McD84] B. R. McDonald. *Linear Algebra Over Commutative Rings*. Pure and Applied Mathematics #87. Marcel Dekker, New York, 1984.

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

[Mos53] Leo Moser. *On non-averaging sets of integers*. Canadian Mathematical Society, 1953.

[MP73] Bernard Maurey and Gilles Pisier. Caractérisation d'une classe d'espaces de Banach par des propriétés de séries aléatoires vectorielles. *C. R. Acad. Sci. Paris Sér A*, 277:687—690, 1973.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. North Holland, Amsterdam, New York, 1977.

[MT12] 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.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and computing: Randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[MV03] Shachar Mendelson and Roman Vershynin. Entropy and the combinatorial dimension. *Invent. Math.*, 152(1):37–55, 2003.

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

[NSZ18] Martin Nägeld, Benny Sudakov, and Rico Zenklusen. Submodular minimization under congruency constraints. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 849–866. SIAM, 2018.

[OAC$^+$17] Lee Organick, Siena Dumas Ang, Yuan-Jyue Chen, Randolph Lopez, Sergey Yekhanin, Konstantin Makarychev, Miklos Z Racz, Govinda Kamath, Parikshit Gopalan, Bichlien Nguyen, et al. Scaling up dna data storage and random access retrieval. *bioRxiv*, page 114553, 2017.

[Oba02] Kenji Obata. Optimal lower bounds for 2-query locally decodable linear codes. In *International Workshop on Randomization and Approximation Techniques in Computer Science*, pages 39–50. Springer, 2002.

[OSI07] Rafail Ostrovsky and William E Skeith III. A survey of single-database private information retrieval: Techniques and applications. In *Public Key Cryptography–PKC 2007*, pages 393–411. Springer, 2007.

[PD14] Dimitris S Papailiopoulos and Alexandros G Dimakis. Locally repairable codes. *IEEE Transactions on Information Theory*, 60(10):5843–5855, 2014.

[PGM13] J. S. Plank, K. M. Greenan, and E. L. Miller. Screaming fast Galois field arithmetic using Intel SIMD instructions. In *11th Usenix Conference on File and Storage Technologies (FAST)*, pages 299–306, San Jose, February 2013.

[Pis12] Gilles Pisier. 15th workshop on non-commutative harmonic analysis, BĂŹdlewo, Poland, 2012.

[Rag07] Prasad Raghavendra. A note on yekhaninâĂŹs locally decodable codes. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 14, 2007.

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

[RY06] Alexander A. Razborov and Sergey Yekhanin. An $\Omega(n^{1/3})$ lower bound for bilinear group based private information retrieval. In *FOCS*, pages 739–748, 2006.

[Rya02] Raymond A. Ryan. *Introduction to tensor products of Banach spaces*. Springer Monographs in Mathematics. Springer-Verlag London Ltd., London, 2002.

[SAP$^+$13] Maheswaran Sathiamoorthy, Megasthenis Asteris, Dimitris S. Papailiopoulos, Alexandros G. Dimakis, Ramkumar Vadali, Scott Chen, and Dhruba Borthakur. XORing elephants: novel erasure codes for big data. In *Proceedings of VLDB Endowment (PVLDB)*, pages 325–336, 2013.

[Sár78a] András Sárközy. On difference sets of sequences of integers. I. *Acta Mathematica Hungarica*, 31(1-2):125–149, 1978.

[Sár78b] András Sárközy. On difference sets of sequences of integers. III. *Acta Mathematica Hungarica*, 31(3-4):355–386, 1978.

[Sha48] Claude E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

[Sil09] J.H. Silverman. *The Arithmetic of Elliptic Curves*. Graduate Texts in Mathematics. Springer New York, 2009.

[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.

[Sud97] Madhu Sudan. Decoding of reed solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997.

[Sze75] Endre Szemerédi. On sets of integers containing no k elements in arithmetic progression. *Acta Arith*, 27(299-345):21, 1975.

[Tal14a] Michel Talagrand. *Upper and lower bounds for stochastic processes*, volume 60 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer, Heidelberg, 2014. Modern methods and classical problems.

[Tal14b] Michel Talagrand. *Upper and lower bounds for stochastic processes*, volume 60 of *Ergebnisse der Mathematik und ihrer Grenzgebiete. 3. Folge. A Series of Modern Surveys in Mathematics [Results in Mathematics and Related Areas. 3rd Series. A Series of Modern Surveys in Mathematics]*. Springer, Heidelberg, 2014. Modern methods and classical problems.

[Tao07] Terence Tao. The ergodic and combinatorial approaches to Szemerédi's theorem. In *CRM Proc. Lecture Notes*, volume 43, pages 145–193, 2007.

[Tao12] Terence Tao. *Higher order Fourier analysis*, volume 142. American Mathematical Society, 2012.

[Tao14] Terence Tao. Algebraic combinatorial geometry: the polynomial method in arithmetic combinatorics, incidence combinatorics, and number theory. *EMS Surv. Math. Sci.*, 1:1–46, 2014.

[TB98] Gabor Tardos and DA Mix Barrington. A lower bound on the mod 6 degree of the or function. *Computational Complexity*, 7(2):99–108, 1998.

[TB14] Itzhak Tamo and Alexander Barg. A family of optimal locally recoverable codes. *IEEE Transactions on Information Theory*, 60:4661–4676, 2014.

[Tho83] Christian Thommesen. The existence of binary linear concatenated codes with reed - solomon outer codes which asymptotically meet the gilbert- varshamov bound. *IEEE Trans. Information Theory*, 29(6):850–853, 1983.

[TJ74] Nicole Tomczak-Jaegermann. The moduli of smoothness and convexity and the rademacher averages of the trace classes s_{p}(1âĽď p¡âĹđ). *Studia Mathematica*, 50(2):163–182, 1974.

[TPD16] Itzhak Tamo, Dimitris Papailiopoulos, and Alexandros G. Dimakis. Optimal locally repairable codes and connections to matroid theory. *IEEE Transactions on Information Theory*, 62:6661–6671, 2016.

[Tro15] Joel A Tropp. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015.

[TV06] T. Tao and V.H. Vu. *Additive Combinatorics*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2006.

[TW14] Madhur Tulsiani and Julia Wolf. Quadratic Goldreich-Levin theorems. *SIAM Journal on Computing*, 43(2):730–766, 2014.

[TZ12] Terence Tao and Tamar Ziegler. The inverse conjecture for the gowers norm over finite fields in low characteristic. *Annals of Combinatorics*, 16(1):121–188, 2012.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

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

[Var59] Panayiotis Varnavides. On certain sets of positive density. *Journal of the London Mathematical Society*, 1(3):358–360, 1959.

[Vid15] Michael Viderman. Explicit strong LTCs with inverse poly-log rate and constant soundness. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:20, 2015.

[War16] Lutz Warnke. Upper tails for arithmetic progressions in random subsets. *arXiv preprint: 1612.08559*, 2016.

[WdW05a] Stephanie Wehner and Ronald de Wolf. Improved lower bounds for locally decodable codes and private information retrieval. In *ICALP*, pages 1424–1436, 2005.

[WDW05b] 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.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. *Electronic Colloquium on Computational Complexity (ECCC)*, 14(006), 2007.

[Woo08] David Woodruff. Corruption and recovery-efficient locally decodable codes. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 584–595. Springer, 2008.

[Woo12] David Woodruff. A quadratic lower bound for three-query linear locally decodable codes over any field. *J. Comput. Sci. Technol.*, 27(4):678–686, 2012.

[WTB17] Zhiying Wang, Itzhak Tamo, and Jehoshua Bruck. Optimal rebuilding of multiple erasures in MDS codes. *IEEE Transactions on Information Theory*, 63:1084–1101, 2017.

[WY05] David P. Woodruff and Sergey Yekhanin. A geometric approach to information-theoretic private information retrieval. In *IEEE Conference on Computational Complexity*, pages 275–284, 2005.

[WZ12] Trevor Wooley and Tamar Ziegler. Multiple recurrence and convergence along the primes. *American Journal of Mathematics*, 134(6):1705–1732, 2012.

[YB17] Min Ye and Alexander Barg. Explicit constructions of high-rate MDS array codes with optimal repair bandwidth. *IEEE Transactions on Information Theory*, 63:2001–2014, 2017.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1, 2008.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Foundations and Trends® in Theoretical Computer Science*, 6(3):139–255, 2012.

[YH18] Hikmet Yildiz and Babak Hassibi. Optimum linear codes with support constraints over small fields. *CoRR*, abs/1803.03752, 2018.