Copyright
by
Vinayak Mahesh Kumar
2026

The Dissertation Committee for Vinayak Mahesh Kumar
certifies that this is the approved version of the following dissertation:

Linearity, Locality, and Lower Bounds: A Unified Perspective Through Randomness

Committee:

David Zuckerman, Supervisor

Anna Gál

Venkatesan Guruswami

Dana Moshkovitz

Linearity, Locality, and Lower Bounds: A Unified Perspective Through Randomness

by

Vinayak Mahesh Kumar

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2026

Acknowledgments

It has been an immense privilege to be advised by David Zuckerman. At the start of my PhD, I didn't really have any idea what research was, and all I knew was that solving problems was fun. Despite my incompetence, David patiently met with me, guided me through the process of looking for good problems, and gave me nudges in tackling them. His uncanny ability to give seemingly simple suggestions that manifest as big steps has taught me that, a lot of the time, it is about finding the right simple observation and letting the mathematics carry out the rest. Beyond technical support, David has always been a staunch supporter of mine and has consistently believed in me despite the many times I didn't believe in myself. I will always be grateful for that.

I am fortunate to have had many mentors during my undergraduate years. Chris Umans's PowerPoint presentations in Comic Sans with colorful WordArt and lectures interspersed with puzzles and riddles made learning about complexity theory feel like I was back in elementary school. It was those courses that made me certain I would love to work on complexity theory in the long term. I also thank Leonard Schulman for his course on randomized algorithms, for mentoring me in my first project in theoretical computer science, and for giving general advice. His course was extremely challenging but timeless. Even now, I always find myself using technical tricks that I learned from struggling with his problems or referring back to notes from that class. I would also like to thank Venkatesan Guruswami for mentoring me in my first project in pseudorandomness, which has shaped my research trajectory. He had no reason to respond to a cold email from a random student outside of his institution during COVID, and yet he did. This directly influenced my trajectory towards pseudorandomness.

I would also like to thank the many professors who have given me advice and supported me in various ways throughout my PhD. I would like to thank Anna Gál

and Dana Moshkovitz, both of whom have checked in on me, supported my endeavors, and given me technical and personal advice as if I were their student. I would also like to thank various professors from whom I've learned a lot, who have arranged for visits, and with whom I've had many enlightening discussions. These include Sepehr Assadi, Eshan Chattopadhyay, Bobby Kleinberg, Bill Kuszmaul, Chin Ho Lee, Sidhanth Mohanty, Rocco Servedio, and Avishay Tal.

I would like to thank my various coauthors throughout my PhD: Sabee Grewal, Elena Grigorescu, Vishnu Iyer, Siddhartha Jain, Michael Jaber, Robin Kothari, Matt Kovacs-Deak, Peter Manohar, Geoffrey Mon, Luke Schaeffer, Daochen Wang, Michael Whitmeyer and David Zuckerman. Special thanks go to Geoffrey Mon for his collaborations and introducing me to the world of RLDCs, which became a key direction in this thesis.

I am very lucky to have had a large cohort of fellow students and postdocs to be in the trenches with during my time at UT Austin. Thank you to Jeff Champion, Gautam Chandrasekharan, Joshua Cook, Karen Chen, Jesse Goodman, Sabee Grewal, Shivam Gupta, Zeyu Guo, Vishnu Iyer, Michael Jaber, Sid Jain, Gregory Kehne, Lin Lin Lee, Geoffrey Mon, Aditya Parulekar, Advait Parulekar, Rojin Rezvan, Kristin Sheridan, Tal Yankovitz, and Justin Yirka for the countless hours of conversations, debates, dinners, and company. You all have qualities that I admire and strive to cultivate in myself.

I am also lucky to have been able to travel and make friends with many people during my graduate studies outside of the lab. During my visit to Harvard I had the fortune of meeting Prashanth Amireddy, Chin Ho Lee, Louie Putterman, Ted Pyne, Sunny Qin, Jake Ruotolo, Juspreet Singh Sandhu, and Rosie Zhao. In my visits to the Simons Institute, I had a blast meeting John Bostanci, Miranda Christ, Shivam Nadimpalli, Natalie Parham, Shyamal Patel, and Rocco Servedio. Thank you to Meghal Gupta, William He, Hoai-An Nguyen, and Mihir Singhal for all the clowning, and thank you to Kelsey Ball and Pranav Trivedi for the unadulterated

climbing. Thank you to the many other people I've met over the past five years and with whom I've shared heartwarming experiences. I appreciate you all.

Thank you to my friends from before graduate school. Thank you to Erich Liang, Mayank Pandey, Kushal Tirumala, and Daniel Zhou for all the late-night Discord calls, to Rahil Bathwal, Pavan Chitta, Ankush Hommerich-Dutt, Alveera Khan, Daniel Kyme, and Eunice Yoon for the fun trips, and Srimanth Alluri, Jaclyn Thach, and Alex Wang for dealing with me since high school.

The PhD would not have been possible without the support of my family. I learned that, rather than the actual mathematics, the values of hard work and perseverance instilled in me by my parents were perhaps the most important aspects of research. My mother's unwavering support for my work (sometimes manifesting as an inflation of ECCC downloads) and my father's sage words never fail to rejuvenate me. Throughout my life, my sister has been a role model for how best to navigate life, including working hard, having fun, and appeasing the parentals. Thank you three for a lifetime of support.

Abstract

Linearity, Locality, and Lower Bounds: A Unified Perspective Through Randomness

Vinayak Mahesh Kumar, PhD
The University of Texas at Austin, 2026

SUPERVISOR: David Zuckerman

Randomness plays two seemingly opposed roles in computation. On the one hand, it can be harnessed as a resource, allowing simple algorithms and data structures to accomplish tasks that would be impossible deterministically. On the other hand, it can be deployed as an analytic tool, exposing the limits of efficient computation. This dissertation develops both perspectives through three structural themes: linearity, locality, and lower bounds.

Linearity. We first study random linear maps as practical hash functions, asking how closely these highly structured maps reproduce the load balancing of a fully random function. We show that when a random linear map hashes m balls into n bins, the expected order statistics of the bin loads are governed by the thresholds predicted by n independent Poisson random variables of mean m/n. This yields two corollaries:

- When n balls are hashed into n bins, we prove that the expected maximum load of a random binary linear map is at most that of a fully random map plus $3/2 + o(1)$. This settles a longstanding question raised by Alon, Dietzfelbinger, Miltersen, Petrank, and Tardos [ADM+99] in an extremely strong form.

- When $m > n \log n$ balls are hashed into n bins, we show that *all* bins have close to m/n balls in a quantitatively tight sense. This establishes an ℓ_∞-norm analog of the influential Leftover Hash Lemma.

We then turn to random Toeplitz matrices, which form an especially economical linear hash family that requires essentially the minimum randomness needed for universality and is actively deployed in practice. We prove an $O((n \log n)^{1/3})$ bound on their expected maximum load, giving the first improvement over the generic $O(\sqrt{n})$ bound for universal hashing.

Locality. We next investigate error-correcting codes whose message bits can be recovered by randomized algorithms that inspect only a few bits of the received, potentially corrupted, codeword. In particular, we study basic questions about locally decodable codes (LDCs) and their relaxed counterparts (RLDCs).

RLDCs were defined in 2006 [BSGH$^+$06] mainly because of the halted state of affairs regarding optimal constructions of LDCs. The hope was that codes with this relaxed notion would be easier to construct while still having applicability similar to that of LDCs. In this thesis, we finally realize this hope. We construct the first asymptotically good binary linear relaxed locally decodable codes with $O(\log^{69} n)$ queries, which is optimal up to polynomial factors.

We then study the boundary at which relaxed local decoding and correction become genuinely more powerful than their classical counterparts. Under the standard soundness parameters, every linear 3-query relaxed locally decodable or correctable code is already an ordinary 3-query locally decodable or correctable code. In contrast, we construct a 15-query relaxed locally decodable code and a 41-query relaxed locally correctable code that are not locally decodable with any constant number of queries. More generally, we show that *strong soundness* (the condition that the soundness of the decoder is proportional to the number of corruptions in the received word) forces a linear relaxed local code to be an ordinary local code.

Lower Bounds. Finally, we use randomness to reveal limitations of computation. Our first contribution concerns Impagliazzo's celebrated hardcore lemma [Imp95], a key result used in amplifying correlation bounds against circuits. It states that if a function is mildly hard for size-s circuits, there is a constant-density subset of inputs on which the same function is extremely hard for circuits of size $s' \ll s$. A natural question is whether the size degradation from s to s' is necessary or whether a lossless version of the hardcore lemma can be proven. We show this size degradation is unfortunately necessary for almost all parameter regimes. Our proof is surprisingly simple and relies on locally efficient constructions of a basic randomness primitive: 4-wise independent generators.

We also develop a general method for amplifying average-case hardness using *randomness extractors*. This method allows us to make progress on the question of proving correlation bounds against structured $\mathrm{AC}^0[\oplus]$ circuits and low-degree $\mathbb{F}_2$-polynomials. In particular, by composing hard functions with suitable extractors, we give nearly optimal correlation bounds for quasipolynomial-size constant-depth circuits with many symmetric or threshold gates, exponentially improved pseudorandom generators for width-2 branching programs reading many bits at a time, and strong correlation bounds for set-multilinear polynomials.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Randomness can be used to accomplish fundamental tasks in computing that are *impossible* to achieve deterministically. As an example, say we are given query access to a function $f : \{0,1\}^n \to \{0,1\}$ and want to approximate the mean of f up to a ± 0.01 margin. Deterministically, one would need to query at least $0.98 \cdot 2^n$ inputs, since $0.02 \cdot 2^n$ outputs are enough to sway the average by more than twice the allotted margin. However, a simple Chernoff bound shows that by randomly sampling a constant number of points and taking their empirical mean, we can get an accurate estimate of the global mean with probability 0.99. Theoretical computer science is riddled with such seemingly impossible questions that are suddenly rendered tractable with the clever use of randomness.

This thesis focuses on two primitives where randomness is crucial: local codes and hash functions. Among their many other applications, locally decodable codes allow us to store and retrieve data extremely efficiently, even if the medium of storage becomes corrupted over time. Hashing allows us to compress large data, allowing for much more efficient searching and manipulation of data. Such objects, or their variants, are ubiquitous in computing. Due to their widespread use, it is all the more important we have provable guarantees for these tools—especially if used in mission-critical tasks.

Randomness can also be used to show *limitations* of computing. Perhaps the most important question in complexity theory is whether there exist explicit problems that require superpolynomial-size circuits to solve. While this question is extremely difficult, randomness already gives some insight into it. The classical argument of Shannon [Sha49] shows that a *random* function requires exponential circuit size to compute. While this argument does not solve the problem, as a random function is not necessarily explicit (here, explicit means being in NP), it planted the seed to use randomness in circuit lower bound techniques. A lot of our lower bounds

are established by using a randomness tool to easily capture or simplify small-size circuits and then picking a hard function that is "immune" to that tool. This intuition captures the classical techniques of random restrictions and probabilistic polynomials. In this thesis, we establish results regarding average-case circuit lower bounds and hardness amplification, where randomness primitives are crucially used to construct small-size circuits and hard functions.

These uses of randomness may appear to pull in opposite directions: one harnesses randomness to make computation efficient, while the other uses it to expose the limits of efficient computation. Yet both are guided by a common probabilistic viewpoint, and techniques developed in one setting often inform the other. Our thesis studies these roles together and provides a unified view of randomness as both a computational resource and a proof technique for computational lower bounds. We next describe the contributions of this thesis to linearity, locality, and lower bounds.

1.1 Linearity

Consider tossing n balls uniformly and independently at random into n bins. What is the expected number of balls in the heaviest bin (i.e., the expected max load)? A fundamental result shows it is $(1 + o(1))\frac{\log n}{\log \log n}$ (see, e.g., [MU05]). This phenomenon is extremely useful in storage and retrieval.

To illustrate this, let $U \gg n$ be integers, and consider an arbitrary subset $S \subset [U]$ of size n. If we wanted to check whether k arbitrary integers were in S, it seems the best we can do is check and compare each element of S to each of the k integers, taking $O(kn)$ time. Indeed, this is the best possible protocol deterministically. Now let $h : [U] \to [n]$ be a uniformly random function. The fundamental result in the last paragraph implies that for all subsets $S \subset [U]$ of size n, the maximum number of elements in S that h maps to the same element is $O(\log n / \log \log n)$ on average. Consider doing an $O(n)$ preprocessing step of using h to group the elements of S into n buckets. Then, to check if an integer m is in S, we check the $h(m)$-th bucket. Hence,

checking membership of k integers will only take $O(n + k \log n / \log \log n)$ expected time.

Unfortunately, picking a fully random function is a computationally heavy task. Indeed, most functions have maximal circuit complexity. To use this in practice, we need an efficient and easily implementable class of random functions with $O(\log n / \log \log n)$ expected max load. This leads to a natural question: *what is the simplest hash function with optimal expected max load?*

Our thesis investigates the effectiveness of random linear maps as hash functions. These maps are basic objects, and it is fundamental to ask how well a linear map resembles a fully random one. Computationally, linear maps are very simple to compute using basic XOR operations. The structure of linearity is also a desirable property and is useful in incremental cryptography [BGG94]. An even simpler family of linear hash functions consists of random *Toeplitz* matrices. Such matrices are very easy to implement using a linear number of bitwise shifts and XORs, have been used as examples of hash functions achieving optimal time-space tradeoffs [MNT93], and are used in practice as extractors [SFI+11].

1.1.1 Our Contributions

We prove new expected max load upper bounds for random matrices and for random Toeplitz matrices.

Random binary matrices were proposed as a hash function for the first time in Carter and Wegman's 1977 paper on universal hash functions [CW79]. More formally, assume $U = 2^u$ and $n = 2^\ell$ for $u, \ell \in \mathbb{Z}$. We consider the vector spaces $\mathbb{F}_2^u$ and $\mathbb{F}_2^\ell$, which have the same cardinality as $[U]$ and $[n]$, respectively, but additional linear structure. Our hash function is now a uniformly random $\mathbb{F}_2$-*linear* map $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$. For an arbitrary subset of n vectors $S \subset \mathbb{F}_2^u$, is it true that the maximum number of elements in S that h maps to the same element is $O(\log n / \log \log n)$ on average? This question was explicitly posed in 1997 by Alon, Dietzfelbinger, Miltersen, Petrank, and

Tardos [ADM+99].

In Chapter 3, we resolve this question in an extremely strong sense. We show that the expected max load of a random $\mathbb{F}_2$-matrix is at most that of a fully random map *plus* $\frac{3}{2} + o(1)$. Hence, up to a very small constant, linear hashing performs just as well as fully random hashing.

This ends up being a simple corollary of a more general phenomenon we show: when hashing m balls into n bins using a random linear map, the expected load of the r-th heaviest bin is bounded by the typical value of the r-th largest value among n independent Poisson random variables with mean m/n. As another corollary of this general phenomenon, we show that when $m \gg n$, *every* bin has close to m/n balls in a quantitatively tight sense. This establishes an ℓ_∞ version of the celebrated Leftover Hash Lemma [ILL89]. This chapter consists of joint work with Michael Jaber and David Zuckerman [JKZ25] and upcoming work of the author. Portions of this chapter overlap with independent and concurrent work of Bshouty [Bsh26].

Turning to random Toeplitz matrices, we note that, surprisingly, nothing was known about the expected max load of these maps beyond the $O(\sqrt{n})$ upper bound enjoyed by any 2-universal hash function. The problem of analyzing the max load properties of this hash function was also explicitly posed by Alon, Dietzfelbinger, Miltersen, Petrank, and Tardos [ADM+99].

In Chapter 4, we show that the expected max load when hashing n balls into n bins using a random Toeplitz matrix is $O(\sqrt[3]{n \log n})$, representing the first improvement on this quantity over the folklore bound of $O(\sqrt{n})$. This chapter is based on upcoming work of the author.

1.2 Locality

A binary error-correcting code (or simply a code) is a mapping $C : \{0,1\}^k \to \{0,1\}^n$, which is to be thought of as a mapping from messages to codewords. We will

also identify C with the subset of $\{0,1\}^n$ corresponding to C's image if the particular mapping is not important. For efficiency reasons, we want to maximize the *rate*, k/n, which represents how many bits of message information are conveyed per codeword bit. For two strings $x, y \in \{0,1\}^n$, denote the (relative) *distance* between x and y to be $\text{dist}(x,y) \coloneqq \frac{1}{n}\Delta(x,y)$. The distance of a code is $\delta = \delta(C) \coloneqq \min\{\text{dist}(x,y) : x \neq y \in C\}$. The amount of corruption a codeword can handle while remaining unambiguous scales with the distance, so we would like to maximize this quantity.

By the Singleton bound [Sin64], any code with $\Omega(1)$ distance must have $O(1)$ rate. We call codes achieving constant distance and constant rate *asymptotically good* (*good* for short). Explicitly constructing good codes thereby emerges as a natural question. However, this was solved in 1972 by Justesen [Jus72].

There is a third desirable property of codes: locality. Say we receive a word that is a mildly corrupted codeword, and wish to recover a particular bit of the message or codeword. We would like to recover the correct message/codeword bit by probing only a few other bits of the received word. Codes in which message bits can be recovered as such are called locally decodable codes (LDCs), and codes in which codeword bits can be recovered as such are locally correctable codes (LCCs). In addition to being essential in data recovery, LDCs and LCCs have found uses as a mathematical ingredient in other results in algorithms and complexity theory, like sublinear algorithms [RS96, BLR93], probabilistic proofs [ALM+98, AS98], pseudorandomness [STV99, DMOZ22], and more.

To be more formal, a code $C : \{0,1\}^k \to \{0,1\}^n$ is called a (binary) LDC with query complexity q (or q-LDC for short) and radius δ if there exists a randomized algorithm A such that for every $m \in \{0,1\}^k$, $i \in [k]$, and w with $\text{dist}(w, C(m)) \leq \delta$, $A(w,i)$ queries at most q indices of w and outputs m_i with probability $\geq 2/3$. Furthermore, if $w \in C$, A_i certainly outputs c_i. We analogously define q-query locally correctable codes (q-LCCs) with radius δ to be codes which admit an algorithm A, when on inputs $i \in [n]$, and w with $\text{dist}(w, C) \leq \delta$, outputs c_i with probability $\geq 2/3$.

Randomization of A_i is *crucial* here. If $\delta > q/n$, A_i must be random, since replacing the bits in the q fixed indices queried by A_i with those of a different codeword will cause A_i to fail. We say an LDC/LCC is *good* if it has $\Omega(1)$ rate and radius.

What is the smallest query complexity of a good LDC? This fundamental question remains extremely challenging: after decades of work, there is a superpolynomial gap between the best lower bound of $\Omega(\log n)$ [KT00, Woo07] and best upper bound of $2^{O(\sqrt{\log n})}$ [KMRS17].

Due to this, in 2006, researchers introduced a relaxed notion of locality. A *relaxed locally decodable code* (RLDC) has the same definition as an LDC, but now, if $w \notin C$, A_i either outputs c_i or an error ($\perp$) with probability $\geq 2/3$ [BSGH$^+$06, GRR20]. In particular, it need not recover the bit if it detects corruption. We can define *relaxed locally correctable codes* (RLCCs) to be the analogous relaxation of LCCs. This simplified definition still has applications in probabilistically checkable proofs, making it a compelling object.

Once this relaxed notion of LDCs is defined, two natural questions emerge. *What is the smallest query complexity of a good RLDC/RLCC? Are RLDCs/RLCCs truly different from LDCs/LCCs, or is there a reduction between the two objects?* In this thesis, we will investigate both of these questions.

1.2.1 Our Contributions

As we just observed, there is a large gap between query complexity upper and lower bounds for good LDCs. Could the relaxedness of good RLDCs allow us to give constructions more closely matching known lower bounds? This basic question had remained open since the invention of this code. In Chapter 6, we provide the first construction of RLDCs whose query complexity matches known lower bounds up to polynomial factors. Our construction is a surprisingly simple reduction to locally testable codes and relies on a new code-combining operation which we call *nesting*. The work in this chapter is joint with Geoffrey Mon [KM24c].

After RLDCs are defined, it is natural to ask: can RLDCs achieve strictly better parameters than LDCs? One concrete instantiation of such a question is: *Does there exist q such that there exists a q-RLDC whose rate is larger than that of any q-LCC? If so, what is the smallest such q?*

It was shown that RLDCs with query complexity 2 are essentially 2-LCCs [BCG$^+$22]. However, we know that once $q \geq 10^7$, there is a construction of a q-RLDC whose rate is exponentially larger than that of any possible q-LCC [AS21]. This means there exists some $2 \leq q \leq 10^7$ such that any q-RLDC is essentially a q-LDC, but there exists a $(q+1)$-RLDC whose rate is greater than that of any $(q+1)$-LDC. In Chapter 7, we show that 3-RLDCs are as strong as 3-LDCs, and that there exists a 15-RLDC that is not a q-LDC for any constant q. This narrows the range to $3 \leq q \leq 15$. We give a similar result for RLCCs and LCCs, thereby giving exponentially stronger rate lower bounds for 3-RLCCs due to the LCC lower bounds of Kothari and Manohar [KM24a]. We actually uncover a more general phenomenon. Informally, we say that an RLDC or LDC has *strong soundness* if the probability of the decoder detecting an error is proportional to the received word's distance to the code. We show that if a q-RLCC has the strong soundness property, it is essentially a q-LCC. These results extend to the weaker notion of (relaxed) locally *decodable* codes. The work in this chapter is joint with Elena Grigorescu, Peter Manohar, and Geoffrey Mon [GKMM26].

1.3 Lower Bounds

Finding an explicit function f requiring superpolynomial-size circuits to compute is one of the biggest open problems in complexity theory. Unfortunately, the best circuit lower bound we know against an explicit function is a mere $3.1n - o(1)$ [LY22]. Due to this stark state of affairs, researchers have attempted to first prove strong lower bounds against circuits of constant depth, denoted as AC^0. In this setting, complexity theorists have had major success: we can show that even slightly

correlating with the parity function requires exponential size [FSS84, Hås86, Yao85].

In light of this, the natural next step is to add parity as a gate and ask whether we can derive strong lower bounds against these circuits (denoted $\mathrm{AC}^0[\oplus]$). Due to the celebrated technique of Razborov and Smolensky [Raz87, Smo87], we can prove that computing the majority of n bits *exactly* requires exponential-size $\mathrm{AC}^0[\oplus]$ circuits. However, we may not be satisfied with this in practice and request correlation bounds. After all, if there exists a circuit that agrees with a hard function on all but one point, the difference may be impossible to detect in practice. Furthermore, establishing average-case hardness against $\mathrm{AC}^0[\oplus]$ can allow us to create PRGs against this circuit class via the "hardness to randomness" framework introduced by Nisan and Wigderson [NW94], as well as show hardness results against related function classes, like the majority of multiple $\mathrm{AC}^0[\oplus]$ circuits. Unfortunately, we have much weaker size lower bounds on circuits that simply correlate with an explicit function. The best correlation bounds we have are that the majority has $\tilde{O}(1/\sqrt{n})$ correlation with any $\mathrm{AC}^0[\oplus]$ circuit.

To identify the bottleneck towards proving such correlation bounds, let us briefly review the argument of Razborov and Smolensky. The idea is to first show that $\mathrm{AC}^0[\oplus]$ can be computed by low-degree polynomials. This appears to be impossible, as even one AND-gate of fan-in n requires degree n to compute. However, with a crafty use of randomness, they prove that shallow $\mathrm{AC}^0[\oplus]$ can be computed by a *distribution* of low-degree polynomials over $\mathbb{F}_2$ with high probability. They then prove that majority does not correlate with any low-degree $\mathbb{F}_2$-polynomials. These claims together imply correlation bounds. The latter claim is the limiting step: we can only establish $\tilde{O}(1/\sqrt{n})$ correlation bounds against low-degree polynomials, giving us similar correlation bounds against $\mathrm{AC}^0[\oplus]$. It turns out this question about low-degree polynomials is also a stepping stone towards other important questions in complexity theory, like communication complexity and matrix rigidity (see, e.g., [Vio09]).

There are two natural directions to proceed. One is to take a hard function

achieving our best known correlation bounds, and then apply some gadget to create a harder function with stronger correlation bounds. Another is a "bottom-up" approach: start with restricted subclasses of low-degree polynomials or $\mathrm{AC}^0[\oplus]$, prove strong correlation bounds against them, and then gradually remove these restrictions. We attack this problem in both directions.

1.3.1 Our Contributions

With respect to hardness amplification, a classical approach is to establish XOR lemmas. Say we have a function $f : \{0,1\}^n \to \{0,1\}$ that no size-s circuit can compute on more than a .99-fraction of inputs. The intuition is that computing the XOR of k independent instances of f,

$$f^{\oplus k}(x_1, \ldots, x_k) \coloneqq f(x_1) \oplus f(x_2) \oplus \cdots \oplus f(x_k)$$

should be much harder: size-s circuits should not be able to compute it on more than a $\frac{1}{2} + \exp(-\Omega(k))$ proportion of inputs. Yao's celebrated XOR lemma proves such a statement against circuits, with a catch: $f^{\oplus k}$ will be hard for circuits of size $s' = o_k(s)$. This size degradation prevents us from applying this lemma. The bottleneck in the most well-known proof of this lemma is in the application of another hallmark result: Impagliazzo's hardcore lemma. This lemma states that if a function f cannot be computed on more than a .99-fraction of inputs by size-s circuits, there exists a constant-density subset S of inputs (i.e., the "hardcore") for which size-$s\gamma^2$ circuits cannot compute f on more than a $\frac{1}{2}+\gamma$ portion of S. Seeing whether the degradation from s to $s\gamma^2$ is necessary is thus the natural question to ask when investigating XOR lemmas. It is also of independent interest: it is a basic question about the hardcore lemma, and was explicitly asked by Trevisan [Tre09]. There were some *conditional* lower bounds showing the degradation was necessary [LTW11], but Blanc, Hayderi, Koch, and Tan gave the first unconditional proof of the tightness of the γ^2 dependence for a select range of parameters (s, γ). This remarkable result followed from a "fairly involved" technical argument consisting of Fourier analysis, coupling arguments, and

subgaussian concentration. Unfortunately, the range of parameters they handle is quite narrow: for example, their proof breaks down when $\gamma = o(1/\sqrt{n})$, which is precisely the correlation bound regime of interest to us regarding $\mathrm{AC}^0[\oplus]$.

In Chapter 9, we provide a surprisingly simple proof that Impagliazzo's hard-core theorem is tight in all but a vanishingly small window of parameters. In particular, the result extends even to γ exponentially small. Our proof is a result of viewing the problem through the lens of randomness. The crux of the argument uses a basic primitive in pseudorandomness to construct small-size circuit approximators: 4-wise independent generators. This chapter is based on work of the author [Kum26].

In Chapter 10, we proceed with the bottom-up approach to proving correlation bounds against low-degree polynomials and $\mathrm{AC}^0[\oplus]$. For example, we prove exponentially small correlation bounds against set-multilinear polynomials, $\mathrm{AC}^0[\oplus]$ with a limited number of parity gates, and width-2 general branching programs. The assortment of results is proven by using extractors to craft hard functions. Informally, extractors are Boolean functions that, given a structured distribution of entropy k as input, output close to k nearly uniform bits. Intuitively, the idea is that if our hard function uses extractors, then randomly altering the input in a way that simplifies circuits but keeps the extractors unbiased will separate the two. Depending on the model of computation, one will need to use and combine different extractors and use tailored algebraic or combinatorial techniques suited to the model. This chapter is based on work of the author [Kum25].

Part I

Linearity

Chapter 2: Preliminaries

This chapter collects the notation and preliminary facts used throughout the chapters in Part I.

2.1 General Notation

Throughout Part I, log denotes the natural logarithm, and logarithms to base 2 are written explicitly as $\log_2$. For a positive integer n, let $[n] \coloneqq \{1, \ldots, n\}$. For a real number α, denote its fractional part by $\{\alpha\} \coloneqq \alpha - \lfloor \alpha \rfloor$. The finite field of two elements is denoted by $\mathbb{F}_2$. For a finite set S, the notation $s \sim S$ means that s is sampled uniformly at random from S. We will use the inequality $1 + x \leq e^x$ for all x.

For a list of vectors $v_1, \ldots, v_k$, write $v_{\leq i} \coloneqq (v_1, \ldots, v_i)$ and $V_i \coloneqq \operatorname{span}(v_{\leq i})$, with $V_0 \coloneqq \{0\}$.

2.2 Loads

For a function $f : A \to B$, a set $S \subseteq A$, and $y \in B$, define the load of bin y by

$$L_y(S, f) \coloneqq |f^{-1}(y) \cap S|.$$

Let

$$L_{(1)}(S, f) \leq \cdots \leq L_{(|B|)}(S, f)$$

be these loads sorted in nondecreasing order, so $L_{(r)}(S, f)$ is the load of the rth lightest bin. We write

$$M(S, f) \coloneqq L_{(|B|)}(S, f) = \max_{y \in B} |f^{-1}(y) \cap S|$$

for the max load.

2.3 Poisson Random Variables

We denote by $\mathrm{Pois}(\mu)$ a Poisson random variable with mean μ. We use

$$D(\nu, \mu) \coloneqq \nu \log(\nu/\mu) - \nu + \mu$$

for the Kullback–Leibler divergence between $\mathrm{Pois}(\nu)$ and $\mathrm{Pois}(\mu)$. We will use the following basic estimate.

Proposition 2.1. *For $\mu > 0$ and $t > s > 0$,*

$$D(t, \mu) - D(s, \mu) = \int_s^t \log(x/\mu)\, dx \geq (t - s) \log(s/\mu).$$

Proof. Using the integral of $\log x$, we see

$$D(t,\mu) - D(s,\mu) = t\log(t/\mu) - t - s\log(s/\mu) + s = \int_s^t \log(x/\mu)\, dx \geq (t-s)\log(s/\mu).$$

□

2.4 Polynomials over $\mathbb{F}_2$

We will work over the polynomial ring $\mathbb{F}_2[x]$. For $p, q \in \mathbb{F}_2[x]$, denote the degree of p by $\deg p$. We use the convention $\deg 0 = -1$ and recall that

$$\deg(p + q) \leq \max(\deg p, \deg q).$$

We denote by $\mathbb{F}_2[x]_{=d}$ and $\mathbb{F}_2[x]_{<d}$ the sets of $\mathbb{F}_2$-polynomials of degree d and degree less than d, respectively. We also write

$$R_m \coloneqq \mathbb{F}_2[x]/x^m$$

for the truncated polynomial ring of degree less than m.

Definition 2.2. For polynomials $p, q \in \mathbb{F}_2[x]$ with $\deg p = d$ and $\deg q = e$, we can uniquely write $p = aq + b$, where $\deg a = \max(d - e, -1)$ and $\deg b < e$. We denote $a \coloneqq p \operatorname{div} q$ and $b \coloneqq p \pmod q$.

We say $q \mid p$ if and only if $p \pmod q = 0$. A polynomial $p \in \mathbb{F}_2[x]$ is prime if, for all $q, r \in \mathbb{F}_2[x]$, the relation $p \mid qr$ implies $p \mid q$ or $p \mid r$.

We will require the following classic fact from finite-field theory.

Proposition 2.3 (Prime Number Theorem for Polynomials [LN96, Theorem 3.25])**.** *The number of prime polynomials of degree d in $\mathbb{F}_2[x]$ is $(1 \pm o(1))\frac{2^d}{d}$.*

The proof of this fact is much simpler than the standard prime number theorem over $\mathbb{Z}$, which Knudsen used in his proof [Knu19].

2.5 Ultrametrics

An ultrametric is a metric that satisfies a stronger version of the triangle inequality.

Definition 2.4. An *ultrametric* over a set X is a function $d : X \times X \to \mathbb{R}_{\geq 0}$ such that, for all $x, y, z \in X$,

1. $d(x, y) = 0$ if and only if $x = y$;
2. $d(x, y) = d(y, x)$; and
3. $d(x, z) \leq \max(d(x, y), d(y, z))$.

An ordinary metric has the familiar triangle inequality $d(x, z) \leq d(x, y) + d(y, z)$ in place of the final condition. A trivial ultrametric on any set X is given by $d(x, y) = 1$ when $x \neq y$ and $d(x, y) = 0$ otherwise. The ultrametric relevant here is

$$d_{\deg}(p, q) \coloneqq \deg(p + q) + 1$$

on $\mathbb{F}_2[x]$.

Claim 2.5. *$d_{\deg}$ is an ultrametric over $\mathbb{F}_2[x]$.*

Proof. We prove the three properties in succession.

1. $d_{\deg}(p, q) = 0$ if and only if $\deg(p + q) = -1$, which holds if and only if $p = q$.
2. $d_{\deg}(p, q) = \deg(p + q) + 1 = \deg(q + p) + 1 = d_{\deg}(q, p)$.
3. $$\begin{aligned} d_{\deg}(p, r) &= \deg(p + r) + 1 \\ &= \deg((p + q) + (q + r)) + 1 \\ &\leq \max(\deg(p + q), \deg(q + r)) + 1 \\ &= \max(d_{\deg}(p, q), d_{\deg}(q, r)). \end{aligned}$$

□

For $x \in X$ and $r \in \mathbb{R}$, define the closed ball

$$B(x, r) \coloneqq \{y \in X : d(x, y) \leq r\}.$$

We have the following simple geometry of balls in an ultrametric space.

Proposition 2.6 (Ultrametric Balls [Sch85, Proposition 18.5])**.** *Any two balls are either disjoint or one is contained in the other. In particular:*

- *any two distinct balls of the same radius are disjoint; and*
- *every point in a ball is a center of that ball.*

In $(\mathbb{F}_2[x], d_{\deg})$, the closed balls are

$$B(p, d) = \{q \in \mathbb{F}_2[x] : \deg(p + q) < d\}.$$

The intervals over $\mathbb{Z}/p\mathbb{Z}$ considered by Knudsen [Knu19] do not have this simple structure: they may intersect nontrivially or wrap around, and they do not cleanly partition the space.

For an arbitrary set $S \subseteq \mathbb{F}_2[x]$, we will also count pairs of polynomials that are close.

Definition 2.7. For $S \subseteq \mathbb{F}_2[x]$, define

$$N_r(S) \coloneqq |\{(p,q) \in S^2 : p \neq q,\ d_{\deg}(p,q) \leq r\}|$$

to be the number of ordered distinct pairs in S within distance r of each other.

Chapter 3: Random Linear Maps

3.1 Introduction

Consider tossing n balls uniformly and independently into n bins. The maximum number of balls in any bin–the max load–is a well studied quantity critical to randomized algorithm design [MU05, Chapter 5]. A common application arises in hashing with chaining. In this scenario, keys are mapped to addresses via a random hash, and each address holds a linked list of all keys mapping to it. Consequently, retrieving a key might require sweeping through the largest linked list, and so the worst-case retrieval time would be the max-load.

A classical result says that tossing n balls randomly into n bins will have expected max-load $(1+o(1))\frac{\log n}{\log\log n}$, implying that a fully random hash will have only $(1+o(1))\frac{\log n}{\log\log n}$ expected retrieval time. However, placing balls independently and uniformly requires sampling a truly random function, which has near-maximal time and space complexity. This raises a central question:

What is the simplest hash family with optimal expected max-load?

An engineering approach to this question is to use any tools necessary to design hash functions that minimize the time and space complexity, say in the word-RAM model. Such an approach will naturally give low complexity hash functions, but at the cost of a more contrived construction that is hard to implement practically. Nevertheless, this approach has been extremely successful, and arguably started with Wegman and Carter's definition of k-wise independent hash functions [WC81]. One can easily verify that $O(\log n/\log\log n)$-wise independent hash functions have optimal max-load, and only require $O(\log n/\log\log n)$ evaluation time and $O(\log^2 n/\log\log n)$ description size. Recent works greatly optimized the use of k-wise independence, culminating in a construction of Meka, Reingold, Rothblum, and

Rothblum [CRSW13, MRRR14] only needing $O(\log n \log\log n)$ bits to describe and $O((\log\log n)^2)$ time to evaluate. This gets us within a poly$(\log\log n)$ factor of optimal in both parameters. However, these functions rely on concatenating hashes of gradually increasing independence, which will require implementing either prime fields or polynomial rings, and performing $\omega(1)$ multiplications over them. Although possible, it is quite complicated and slow to do in practice.

The approach we adopt in this chapter is to only consider simple and practical hash functions and analyze their properties as well as possible. To this end, Chung, Mitzenmacher, and Vadhan [CMV13] showed that basic universal hash functions (e.g., linear congruence or multiplication schemes) can achieve optimal max-load if the balls are assumed to have high enough entropy, but say nothing about a worst-case choice of balls. In terms of worst-case results, a surprising result of Pătraşcu and Thorup [PT12] shows that tabulation hashing has optimal max-load. This scheme is only 3-wise independent, is simple and practical to use, and has $O(1)$ evaluation time. However, it requires $O(n^\varepsilon)$ bits to describe. Knudsen [Knu19] considers the extremely simple and efficient hash function of $ax+b \pmod p \pmod n$, and is able to show an $\tilde{O}(n^{1/3})$ bound on this function, a polynomial improvement over the $O(\sqrt{n})$ bound for a generic 2-universal hash family.

Linear Hash Functions

In this chapter, we consider an extremely simple hash family proposed in the first paper on universal hashing [CW79]: random matrices over $\mathbb{F}_2$. In particular, let $\ell := \log_2 n$, and say we arrange our n bins into a vector space $\mathbb{F}_2^\ell$. Further, arrange the universe set into a vector space $\mathbb{F}_2^u$. Our hash family is simply the set of linear maps $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$. This family is arguably the easiest to implement over all constructions mentioned thus far. For $u = O(\ell)$, this family only needs to bitwise XOR $O(\log n)$ words together, and takes $O(\log^2 n)$ bits to describe. This family requires no multiplication, is not even 3-wise independent, and can be seen as the simplest version of tabulation hashing, where each lookup table stores only two values.

Furthermore, linear hash functions can batch compute keys that are clustered together, which happens often in practice. This notion of efficiency, dubbed *incremental cryptography*, was first introduced in a pair of papers by Bellare, Goldreich, and Goldwasser [BGG94, BGG95]. The motivation behind incremental cryptography is that data to be hashed often consists of slight modifications of each other, whether it be consecutive frames of video footage or edited versions of files. Ideally, after computing the first hash, future hashes should have computation time proportional to the amount of modification made to this initial key. This is indeed the case for linear maps. Let $h(x) \coloneqq \sum_{i:x_i=1} c_i$ be a linear map whose matrix has columns $c_1, \ldots, c_u$. If x has Hamming weight w, $h(x)$ is simply a sum of w column vectors. Hence, if $x, x^1, \ldots x^{100}$ are keys such that each x^i differs from x in less than $w \ll u$ bits, we can compute all the hashes by computing $h(x)$, $h(x + x^i)$ for all i, and then computing $h(x^i) = h(x) + h(x + x^i)$. This gives a total of $u + 100w$ additions, which is better than the naive way of computing $h(x), h(x^1), \ldots, h(x^{100})$ directly, which can take up to $100u$ additions.

Computational considerations aside, it is a fundamental question to ask how much a random matrix behaves like a random function. From a technical standpoint, dealing with the correlations between linearly dependent balls has proven to be challenging. For example, the mere existence of good linear seeded extractors is a longstanding open problem in pseudorandomness [Woo14, Question 7.6], and highlights our lack of understanding of random linear maps.

The Expected Max Load of Linear Maps

Characterizing the max-load of $\mathcal{H}$, the set of linear maps, has remained elusive. As a preliminary bound, the expected max-load of $\mathcal{H}$ is $O(\sqrt{n})$ since it is universal [CW79]. Indeed, some universal hash functions saturate this bound [ADM+99]. However, Markowsky, Carter, and Wegman [MCW78] showed $\mathcal{H}$ achieves expected max-load $O(n^{1/4})$, significantly outperforming the generic universal bound [ADM+99].

This initiated further study of $\mathcal{H}$'s max-load, with Mehlhorn and Vishkin [MV84] (implicitly) showing a subpolynomial bound of $2^{O(\sqrt{\log n})}$, and later with Alon, Dietzfelbinger, Miltersen, Petrank, and Tardos [ADM+99] giving a bound of $O(\log n \log\log n)$. Since then, progress has largely stalled, except for a note by Babka [Bab18], who observed that an improved choice of parameters in the argument of [ADM+99] yields a bound of $O(\log n)$. Unfortunately, the argument in [ADM+99] has an inherent barrier at $O(\log n)$,[1] leaving open whether $\mathcal{H}$ can match the performance of a fully random map. Indeed, this natural question of whether the maximum load is $\Theta(\log n/\log\log n)$ was explicitly posed in [ADM+99].

Two-Sided Bounds

Consider the regime of hashing $m = \Omega(n \log n)$ balls into n bins. In this setting, a fully random map will place approximately m/n balls in *every* bin. Quantitatively, a Chernoff bound establishes that with high probability, *every* bin will be within $O\left(\sqrt{\frac{m \log n}{n}}\right)$ of m/n. Naturally, we can ask if a linear hash function achieves such concentration. The celebrated Leftover Hash Lemma gives a positive answer in the ℓ_1-metric: the sum of the absolute deviations from each bin's load to m/n is small. Hence, this question is equivalent to asking for an ℓ_∞ variant of the Leftover Hash Lemma. Dhar and Dvir investigated this exact question, and established the first effective bin load concentration.

Theorem 3.1 ([DD22], Theorem 2.4)**.** *Let $u \geq \ell \geq 1$ be integers, $n \coloneqq 2^\ell$, $\varepsilon \in (0,1)$, and $S \subseteq \mathbb{F}_2^u$ a set of cardinality m. Let $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ be a uniformly random linear map. For $m \geq \frac{2^{32}}{\varepsilon^8} n \log^4(m/\varepsilon)$ we have*

$$\Pr_h\left[\forall y \in \mathbb{F}_2^\ell,\ \left||h^{-1}(y) \cap S| - \frac{m}{n}\right| \leq \left(\frac{m}{n}\right)^{7/8} \sqrt{\log\left(\frac{m}{\varepsilon}\right)}\right] \geq 1 - \varepsilon.$$

[1] Applying their argument to a purely random function only yields an $O(\log n)$ expectation bound on the max-load.

While a significant result, it leaves room for quantitative improvement. We would like such two-sided bounds to be in effect for m as small as $\Theta(n \log n)$. Furthermore, we would want a better tail bound dependence and smaller concentration window on the bin loads matching the $O\left(\sqrt{\frac{m \log n}{n}}\right)$ width achieved by fully random maps.

3.1.1 Our Results

We show that for sufficiently large n, a random linear map hashing n balls to n bins will have expected max load at most that of a fully random map *plus* $1.5 + o(1)$. In fact, for infinitely many n, we can show the additive surplus is $0.5 + o(1)$. This resolves the question posed by Alon, Dietzfelbinger, Miltersen, Petrank, and Tardos in an extremely strong sense [ADM+99].

Theorem 3.2. *Let $u \geq \ell \geq 1$ be integers, and $n \coloneqq 2^\ell$. Let $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ be a uniformly random linear map, and let $f : [n] \to [n]$ be a fully random map. There exists $c_\ell \in [0, 1)$ such that for every $S \subseteq \mathbb{F}_2^u$ of cardinality n,*

$$\mathbb{E}\left[\max_{y \in \mathbb{F}_2^\ell} |h^{-1}(y) \cap S|\right] \leq \mathbb{E}\left[\max_{y \in [n]} |f^{-1}(y)|\right] + \frac{1}{2} + c_\ell + o_\ell(1).$$

Furthermore, there is an increasing sequence (ℓ_j) such that $c_{\ell_j} \to 0$.

We note that this theorem is tight in a very strong sense: when $u \geq n$ and $S \subseteq \mathbb{F}_2^u$ consists of n linearly independent vectors, $h|_S$ is a uniformly random map. Therefore, this upper bound is within $1/2 + c_\ell + o(1)$ of the best possible. We accompany this expectation with effective tail bounds on the max load.

Theorem 3.3. *Let $u \geq \ell \geq 1$ be integers, and $n \coloneqq 2^\ell$. Let $S \subseteq \mathbb{F}_2^u$ be of cardinality n, and let $\mathcal{H}$ be the set of all linear maps $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$. Let t be the unique real number such that $(t/e)^t = n$. For any $r \geq 1$,*

$$\Pr_{h \sim \mathcal{H}}\left[\max_{y \in \mathbb{F}_2^\ell} |h^{-1}(y) \cap S| \geq rt\right] \leq \frac{e^2}{(r+1)^2 t^{2-2/r}}.$$

For example, setting $r = 1 + \frac{\log\log\log n}{\log\log n}$ gives us $(1 + o(1))\frac{\log n}{\log\log n}$ max-load with probability $1 - O(\frac{1}{\log\log n})$, and setting $r = 4$ gives us $O(\frac{\log n}{\log\log n})$ max-load with probability $1 - O\left(\frac{1}{\log^{3/2} n}\right)$. It is natural to ask if linear maps can give optimal max-load with probability $1 - 1/n^{\Omega(1)}$, matching the extremely high probability guarantee of a fully random linear map. Unfortunately, we show that this is not true.

Theorem 3.4. *Let $u \geq \ell \geq 1, n \geq 1$ be integers, $n \coloneqq 2^\ell$, and $\mathcal{H}$ the set of linear maps $\mathbb{F}_2^u \to \mathbb{F}_2^\ell$. There exists $S \subseteq \mathbb{F}_2^u$ of size $n \coloneqq 2^\ell$ such that*

$$\Pr_{h\sim\mathcal{H}}[M(S,h) \geq \log n] \geq (\log n)^{-O(\log\log n)}.$$

Therefore, we cannot hope to do much better than Theorem 3.3.

Our proof is arguably more streamlined than [ADM+99], which first proves a coupon collector property, and then cleverly converts it to a max-load bound. In contrast, we directly analyze the max-load by defining a potential function and tracking its growth. We elaborate further in Section 3.1.2.

The versatility of these potential functions allows us to establish optimal expected max load (up to lower-order terms) in the general setting of hashing $m \neq n$ balls into n bins. In fact, using our potential functions, we can get optimal two-sided bounds in the regime of $m > n \log n$.

Theorem 3.5. *Let $u \geq \ell \geq 1$ be integers, $n \coloneqq 2^\ell$, $\varepsilon \in (0,1)$, and $S \subseteq \mathbb{F}_2^u$ a set of cardinality m. Let $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ be a uniformly random linear map. For $m > n \log n$ we have*

$$\Pr_h\left[\forall y \in \mathbb{F}_2^\ell,\ \left||h^{-1}(y) \cap S| - \frac{m}{n}\right| \leq \frac{4}{\varepsilon}\sqrt{\frac{m\log n}{n}}\right] \geq 1 - \varepsilon.$$

It is possible that better tail bounds can be achieved than in this statement: in the fully random case, one can expect polynomially small error probability.

A Broader Phenomenon

The results mentioned on expected max load and bin concentration are actually corollaries of a broader phenomenon that we establish. By Poissonization (see, e.g., [MU05, Theorem 5.10]), it can be shown that when m balls are hashed into n bins, the load distribution can be approximated by n independent Poisson random variables of mean m/n. One can ask to what extent the load distribution when hashing according to a random linear map resembles that of independent Poisson variables.

Although it can be shown that these distributions are not close, certain statistics of these distributions are. We show that the expected load of the rth heaviest bin is bounded by the expected value of the rth largest Poisson random variable among the n independent variables of mean m/n. The formal statement is deferred to Section 3.3. This broader statement has a surprisingly simple proof. In particular, we believe it to be simultaneously much simpler than the expected max load proof of Alon, Dietzfelbinger, Miltersen, Petrank, and Tardos, and the two-sided bound proof of Dhar and Dvir.

3.1.2 Proof Overview

For simplicity, assume $\mathcal{H}$ is a random *surjective* linear map $\mathbb{F}_2^u \to \mathbb{F}_2^\ell$. We would like to directly analyze the load distribution of $\mathcal{H}$, but this is quite complicated to do. Instead, for a set of balls S and hash h, we define a potential function ϕ that measures how "imbalanced" the allocation of S by h is. In particular, we want a potential function $\phi := \phi(S, h)$ such that if one preimage of h contains $\geq t$ elements in S, then $\phi \geq f(t)$ for some function f. The hope is now that analyzing ϕ will be much easier than analyzing the load distribution directly.

We use the potential $\log(\mathbb{E}_y[e^{\lambda|h^{-1}(y)\cap S|}])$, which takes the logarithm of the average of the exponentials of all the bin loads with some parameter $\lambda > 0$. To analyze this potential, we think of hashing our universe, $\mathbb{F}_2^u$, "one kernel vector at a

time." In particular, any $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ can be decomposed into $h = h_1 \circ h_2 \circ \cdots \circ h_{u-\ell}$, where $h_i : \mathbb{F}_2^{u-i+1} \to \mathbb{F}_2^{u-i}$ is a random surjective map. Each h_i is much simpler to analyze, and allows us to show that if $h_{\le i} \coloneqq h_1 \circ \cdots \circ h_i$ and $\phi_i \coloneqq \phi(S, h_{\le i})$, then $\mathbb{E}[\phi_{i+1} | \phi_1, \ldots, \phi_i] \le 2\phi_i$.

This readily gives an expectation bound on $\phi_{u-\ell} = \phi$. By the design of our potential, we can make $f(t)$ a linear function on t. Hence, by setting t to be the max load random variable, we obtain by linearity of expectation that $\mathbb{E}[\phi] \ge f(\mathbb{E}[t])$. Using the expectation bound on ϕ and solving for $\mathbb{E}[t]$ gives us optimal expected max load.

Surprisingly, our potential functions also allow us to establish *lower bounds* on the loads of all bins. By setting $\lambda < 0$, the potential now detects whether some bin is light. Combining this with the max-load analysis allows us to deduce our two-sided bounds on the bin loads.

To the best of our knowledge, our analysis provides the first proof of optimal max-load for a function that is universal, but not 3-wise independent. In fact, our proof technique applies to a broader class of hash functions mapping $[2^u] \to [2^\ell]$. Imagine starting with 2^u bins, where each of the 2^u universe elements is in its own bin. For $u - \ell$ iterations, pseudorandomly pair bins up in a pairwise independent fashion (i.e., for an arbitrary bin, the marginal distribution of the bin's partner is uniform among all bins) and consolidate each pair into one bin. Consequently, each iteration halves the number of bins, and at the end of this process, we will have hashed into n bins. Our techniques show that any such hashing scheme will have optimal expected max-load. The case of surjective linear maps is when in each round, a random vector v is picked, and each bin x is paired with bin $x + v$.

The exponential of our choice of potential function (i.e., $\Phi_i \coloneqq \mathbb{E}_y[e^{\lambda |h_{\le i}^{-1}(y) \cap S|}]$) has indeed reared its head in previous literature studying linear maps. Perhaps the earliest hint of this potential is in [ADM+99], where they use a degenerate case of our potential functions. When proving that $\mathcal{H}$ has the covering property, [ADM+99]

analyzes the decay rate of the density of empty buckets. The density of empty buckets is simply the exponential potential function with base $b = 0$ (with the convention $0^0 = 1$). The earliest explicit use appears in work of Guruswami, Håstad, Sudan, and Zuckerman [GHSZ02]; follow-up work of Li and Wootters [LW21] uses these potential functions to show that random linear codes achieve list-decoding capacity. We have not found any references using the logarithmic potential functions ϕ_i in the setting of linear maps.

3.2 Preparation

In order to prove our results, we first need to set up some tools. First we will reduce the analysis of random linear maps losslessly to that of random *surjective* linear maps. Next, we will define our potential functions and establish important properties about them that we will use.

3.2.1 Removing Surjectivity For Free

When dealing with random linear maps, one may expect that surjectivity can be assumed in exchange for a small loss in parameters, as random linear maps are full-rank with decent probability. However, to prove our extremely tight bounds, we cannot withstand even the smallest parameter degradation. We prove that most natural load quantities for surjective linear maps can be lifted to general linear maps with *absolutely no* loss of parameters. We believe this lemma to be interesting in its own right.

Lemma 3.6. *Let $f : \{0\} \cup [m] \to [0, \infty)$ be an arbitrary function. Assume there exists Q such that for infinitely many domain dimensions $U \geq \ell$, every $S \subseteq \mathbb{F}_2^U$ of size m, and a uniformly random surjective linear map $h : \mathbb{F}_2^U \to \mathbb{F}_2^\ell$, we have*

$$\mathbb{E}[f(L_{(r)}(S, h))] \leq Q.$$

Then for every *$u \geq \ell$, $S \subseteq \mathbb{F}_2^u$ of size m, and a uniformly random linear map*

$h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$,

$$\mathbb{E}[f(L_{(r)}(S,h))] \le Q.$$

This lemma boosts an "infinitely many" bound to a stronger "for all" bound. Also, the generality of f makes this lemma versatile. Taking $f(i) = i$ to be the identity gives the expected max load. Taking $f(i) = \mathbf{1}_{i \ge t}$ gives tail probabilities. Taking $f(i) = i^m$ gives the m-th moment, and $f(i) = e^{\alpha i}$ gives exponential moments. Hence, for most natural quantities involving the max load, it suffices to focus on surjective linear maps.

Proof of Lemma 3.6. Consider arbitrary $u \ge \ell$ and set $S \subseteq \mathbb{F}_2^u$ of size m. There are infinitely many $U \ge u$ satisfying the lemma's assumption (call such U *good*). Consider an arbitrary one and take an arbitrary u-dimensional subspace $V \le \mathbb{F}_2^U$. There is an isomorphism $\iota : \mathbb{F}_2^u \to V$. Denote $T \coloneqq \iota(S)$. Since $T \subseteq V$, we have for any $H : \mathbb{F}_2^U \to \mathbb{F}_2^\ell$ and $y \in \mathbb{F}_2^\ell$ that

$$|(H|_V)^{-1}(y) \cap T| = |H^{-1}(y) \cap V \cap T| = |H^{-1}(y) \cap T|.$$

Therefore, for every r, $L_{(r)}(T, H|_V) = L_{(r)}(T, H)$. For uniformly random H, $H|_V : V \to \mathbb{F}_2^\ell$ will be a uniformly random linear map. Consequently,

$$\mathbb{E}_h[f(L_{(r)}(S,h))] = \mathbb{E}_H[f(L_{(r)}(T, H|_V))] = \mathbb{E}_H[f(L_{(r)}(T,H))].$$

Let $\mathcal{E}$ be the event that H is surjective. When viewed as a $\ell \times U$ random $\mathbb{F}_2$-matrix, this is equal to the event all rows are linearly independent. Hence

$$\Pr[\mathcal{E}] = (1-2^{-U})(1-2^{-U+1})\dots(1-2^{-U+\ell-1}) \ge 1 - \sum_{i=0}^{\ell-1} 2^{-U+i} \ge 1 - 2^{-U+\ell}.$$

Let $F \coloneqq \max_{0 \le i \le m} f(i)$. We use the fact that Q bounds the surjective expectation to deduce

$$\mathbb{E}_h[f(L_{(r)}(S,h))] = \mathbb{E}_H[f(L_{(r)}(T,H))] \le \Pr[\neg\mathcal{E}]\cdot F + \mathbb{E}[f(L_{(r)}(T,H))|\mathcal{E}] \le 2^{-U+\ell}F + Q.$$

Recalling that there are infinitely many good $U \geq u$ satisfying the lemma's assumption, we can take $U \to \infty$ through an increasing sequence of good integers, yielding the desired statement. $\square$

3.2.2 The Exponential Potential Functions

We consider sampling a random linear map one kernel vector at a time and tracking the load with a potential. Consider k vectors $v_{\leq k} \in (\mathbb{F}_2^u)^k$ and a subset $S \subseteq \mathbb{F}_2^u$. Recall $V_i \coloneqq \operatorname{span}(v_{\leq i})$. For each $0 \leq i \leq k$, define

$$S_i(x) \coloneqq |(x + V_i) \cap S|.$$

Note that S_0 is the indicator of S, and $S_i(x)$ is the number of balls in x's bin when $\mathbb{F}_2^u$ is hashed according to the first i kernel vectors, $v_1, \ldots, v_i$.

Now for arbitrary $\lambda \in \mathbb{R}$, define the sequence of exponential potential functions

$$\Phi_i(S; \lambda; v_{\leq i}) \coloneqq \mathbb{E}_{x \sim \mathbb{F}_2^u}[e^{\lambda S_i(x)}],$$

and abbreviate this quantity by Φ_i. These potentials will serve as a tractable proxy for the rth load. We will now establish three important properties of these potentials.

The first lemma shows that when $\lambda \geq 0$, the potential upper bounds the rth largest load, and when $\lambda \leq 0$, the potential lower bounds the rth largest load.

Lemma 3.7. *Let $\ell, 1 \leq r \leq 2^\ell$ be integers, $v_{\leq i}$ be a fixed tuple of linearly independent vectors in $\mathbb{F}_2^u$, and $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ be a linear map with kernel $V_{u-\ell}$.*

- *When $\lambda \geq 0$, $\Phi_{u-\ell} \geq \frac{2^\ell - r + 1}{2^\ell} e^{\lambda L_{(r)}(S,h)}$.*
- *When $\lambda \leq 0$, $\Phi_{u-\ell} \geq \frac{r}{2^\ell} e^{\lambda L_{(r)}(S,h)}$.*

Proof. Denote $k \coloneqq u - \ell$. Notice that since h is linear, the preimages of h partition $\mathbb{F}_2^u$ into the 2^ℓ cosets of the kernel V_k. Therefore, r cosets contain at most $L_{(r)}(S, h)$

elements of S, and $2^\ell - r + 1$ cosets contain at least $L_{(r)}(S,h)$ elements of S. Call the former cosets *light* and the latter cosets *heavy*.

We show the first item. For any vector x in a heavy coset, $x + V_k$ is heavy, and so $S_k(x) = |(x + V_k) \cap S| \geq L_{(r)}(S,h)$. There are $(2^\ell - r + 1)2^k$ such x. Using the fact that $y \mapsto e^{\lambda y}$ is increasing when $\lambda \geq 0$, we can lower bound

$$\Phi_k = \mathbb{E}_{x \sim \mathbb{F}_2^u}[e^{\lambda S_k(x)}] \geq \frac{2^\ell - r + 1}{2^\ell} e^{\lambda L_{(r)}(S,h)}.$$

The second item is analogous. For any vector x in a light coset, $x + V_k$ is light, and so $S_k(x) = |(x + V_k) \cap S| \leq L_{(r)}(S,h)$. There are at least $r2^k$ such x. Using the fact that $y \mapsto e^{\lambda y}$ is decreasing when $\lambda \leq 0$, we can lower bound

$$\Phi_k = \mathbb{E}_{x \sim \mathbb{F}_2^u}[e^{\lambda S_k(x)}] \geq \frac{r}{2^\ell} e^{\lambda L_{(r)}(S,h)}.$$

□

The next lemma gives a growth bound on the potentials. In particular, it shows that in conditional expectation, the potential function at most squares.

Lemma 3.8. *Let $v_{\leq i} \in (\mathbb{F}_2^u)^i$, and let $v_{i+1} \sim \mathbb{F}_2^u \setminus V_i$. We have $\mathbb{E}_{v_{i+1}}[\Phi_{i+1}] \leq \Phi_i^2$.*

Proof. Denote the following quantities:

$$E_i(x) \coloneqq e^{\lambda S_i(x)}$$

$$\Psi_{i+1}(v_{i+1}) \coloneqq \mathbb{E}_x[E_i(x) E_i(x + v_{i+1})].$$

First notice that for $v \notin V_i$, V_i and $V_i + v$ partition V_{i+1}. Therefore, $S_{i+1}(x) = S_i(x) + S_i(x + v)$, and so $E_{i+1}(x) = E_i(x)E_i(x + v)$. Consequently, for $v_{i+1} \sim \mathbb{F}_2^u \setminus V_i$,

$$\mathbb{E}_{v_{i+1}}[\Phi_{i+1}] = \mathbb{E}_{x, v_{i+1}}[E_{i+1}(x)] = \mathbb{E}_{x, v_{i+1}}[E_i(x)E_i(x + v_{i+1})] = \mathbb{E}_{v_{i+1}}[\Psi_{i+1}(v_{i+1})].$$

Hence it suffices to show $\mathbb{E}_{v_{i+1}}[\Psi_{i+1}(v_{i+1})] \leq \Phi_i^2$. To do so, we will first show the average of Ψ_{i+1} over all of $\mathbb{F}_2^u$ is Φ_i^2, and then show that $\Psi_{i+1}(v) \geq \Phi_i^2$ for all $v \in V_i$.

By an averaging argument, this implies the average of Ψ_{i+1} outside V_i is at most Φ_i^2 as desired.

To prove the first claim, note that for uniform and independent $(x, v) \in (\mathbb{F}_2^u)^2$, so is $(x, x+v)$. Hence,

$$\mathbb{E}_v[\Psi_{i+1}(v)] = \mathbb{E}_{x,v}[E_i(x)E_i(x+v)] = \mathbb{E}_{x,v}[E_i(x)E_i(v)] = \mathbb{E}_x[E_i(x)]\mathbb{E}_v[E_i(v)] = \Phi_i^2.$$

We now show the second claim. Since S_i is constant over cosets of V_i, it follows $E_i(x) = E_i(x+v)$ for any $v \in V_i$. Therefore for all $v \in V_i$,

$$\Psi_{i+1}(v) = \mathbb{E}_x[E_i(x)E_i(x+v)] = \mathbb{E}_x[E_i(x)^2] \geq \mathbb{E}_x[E_i(x)]^2 = \Phi_i^2.$$

Both claims are now proven and we conclude. □

The above two lemmas are enough for us to deduce expected max load bounds and the Poisson comparison in its full generality. For technical reasons we will need the following lemma, which is essential to establishing effective tail bounds.

Lemma 3.9. *Let $v_{\leq i} \in (\mathbb{F}_2^u)^i$, and $v_{i+1} \in \mathbb{F}_2^u \setminus V_i$. For any $b \geq 0$ we have $\Phi_{i+1} - 1 \geq 2(\Phi_i - 1)$.*

Proof. First note that V_i and $V_i + v_{i+1}$ partition V_{i+1}. Therefore, $S_{i+1}(x) = S_i(x) + S_i(x+v_{i+1})$. Combining this observation with elementary manipulations and linearity of expectation, we see

$$\begin{aligned}
\Phi_{i+1} - 1 &= \mathbb{E}_x[e^{\lambda(S_i(x)+S_i(x+v_{i+1}))} - 1] \\
&= \mathbb{E}_x[e^{\lambda S_i(x)} - 1] + \mathbb{E}_x[e^{\lambda S_i(x+v_{i+1})} - 1] \\
&\quad + \mathbb{E}_x[e^{\lambda(S_i(x)+S_i(x+v_{i+1}))} - e^{\lambda S_i(x)} - e^{\lambda S_i(x+v_{i+1})} + 1] \\
&= 2\mathbb{E}_x[e^{\lambda S_i(x)} - 1] + \mathbb{E}_x[(e^{\lambda S_i(x)} - 1)(e^{\lambda S_i(x+v_{i+1})} - 1)] \\
&\geq 2(\Phi_i - 1),
\end{aligned}$$

where the inequality follows from the fact for any $b, r, s \geq 0$, $b^r - 1$ and $b^s - 1$ have the same sign.

□

3.3 Poisson Thresholds Govern Loads

Let $m, n \geq r \geq 1$ be integers, and denote $\mu \coloneqq m/n$. We define $T_r^+(m,n)$ and $T_r^-(m,n)$ to be the unique real $t \in [\mu, \infty)$ and $t \in (0, \mu]$, respectively, such that

$$ne^{-\mu}\left(\frac{e\mu}{t}\right)^t = r, \tag{3.1}$$

Note T_r^+ is well defined: at $t = \mu$ the left-hand side equals $n \geq r$, and it decreases continuously to 0 on $[\mu, \infty)$. When $\mu > \log(n/r)$, T_r^- is well defined: as $t \to 0$, the left-hand side approaches $ne^{-\mu} < r$, and it increases continuously to $n \geq r$ on $(0, \mu]$.

Intuitively, the quantity $T_r^+(m,n)$ is the load level at which the standard Poisson Chernoff bound predicts r heavy bins when hashing m balls randomly into n bins. Each bin's load can be approximated by a Poisson random variable with mean $\mu = m/n$. The Chernoff bound optimized for Poisson random variables states that $\Pr[\mathrm{Pois}(\mu) \geq t] \leq e^{-\mu}\left(\frac{e\mu}{t}\right)^t$ [MU05, Theorem 5.4]. Thus, in the fully random balls-into-bins process, the Chernoff proxy for the expected number of bins with load at least t is $ne^{-\mu}\left(\frac{e\mu}{t}\right)^t$. Setting this proxy equal to r gives precisely $t = T_r^+(m,n)$. In a similar manner, $T_r^-(m,n)$ is the load level at which the Poisson Chernoff bound predicts r light bins. By taking natural logarithms, Eq. (3.1) is equivalent to

$$D(t, \mu) = \log(n/r). \tag{3.2}$$

This version will be convenient for certain calculations later, especially since we can use Proposition 2.1.

In summary, $T_r^+(m,n)$ and $T_r^-(m,n)$ are the expectation-threshold estimates for the rth largest and rth smallest loads, respectively, in the fully random hash setting. Our main conceptual contribution is to show that the rth largest and rth smallest loads of linear hashing are bounded by these expectation thresholds as well.

Theorem 3.10. *Let $u \geq \ell \geq 1$, $1 \leq r \leq n \coloneqq 2^\ell$, and $m \geq 1$ be integers, and let $S \subseteq \mathbb{F}_2^u$ be an arbitrary set of cardinality m. For a uniformly random linear $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$,*

$$\mathbb{E}[L_{(n-r+1)}(S,h)] \leq T_r^+(m,n).$$

Proof. Let $k \coloneqq u - \ell$, $t \coloneqq T_r^+(m,n)$, $\mu \coloneqq m/n$, and $\lambda \coloneqq \log(t/\mu)$. Note $\lambda > 0$.

By Lemma 3.6, we can assume without loss of generality that h is a random *surjective* linear map. To pick h, we will pick linearly independent vectors $v_1, \dots v_k \in \mathbb{F}_2^u$ uniformly at random, and then pick a uniformly random h with kernel $V \coloneqq \operatorname{span}(v_{\le k}) =: V_k$. To sample the $v_{\le k}$, we will iteratively pick $v_i \sim \mathbb{F}_2^u \setminus V_{i-1}$ for $i = 1, \dots, k$.

Using the vectors $v_{\le k}$ and set S, define $\{S_i(x) \coloneqq |(x + V_i) \cap S|\}_{0 \le i \le k}$, the potentials $\{\Phi_i \coloneqq \mathbb{E}_{x \sim \mathbb{F}_2^u}[e^{\lambda S_i(x)}]\}_{0 \le i \le k}$, and the log-potentials $\{\phi_i \coloneqq \log \Phi_i\}_{0 \le i \le k}$. We will first show that $\mathbb{E}[\phi_k]$ is bounded. We will next show that $L_{(n-r+1)}$ is bounded by a linear function in ϕ_k. Linearity of expectation will then give us a bound on $\mathbb{E}[L_{(n-r+1)}]$.

As $|S| = m$, we can compute

$$\Phi_0 = \mathbb{E}_x[e^{\lambda S_0(x)}] = \frac{m}{2^u} e^\lambda + \left(1 - \frac{m}{2^u}\right) = 1 + \frac{m(e^\lambda - 1)}{2^u} = 1 + \frac{t - \mu}{2^k}.$$

Therefore,

$$\phi_0 = \log\left(1 + \frac{t-\mu}{2^k}\right) \le \frac{t-\mu}{2^k}.$$

By Jensen's inequality and Lemma 3.8, we deduce for each $0 \le i < k$,

$$\mathbb{E}_{v_{i+1}}[\phi_{i+1} \mid v_{\le i}] = \mathbb{E}_{v_{i+1}}[\log(\Phi_{i+1}) \mid v_{\le i}] \le \log(\mathbb{E}_{v_{i+1}}[\Phi_{i+1} \mid v_{\le i}]) \le \log(\Phi_i^2) = 2\phi_i.$$

Therefore, we can repeatedly apply linearity of expectation and the law of total expectation to deduce

$$\begin{aligned} \mathbb{E}[\phi_k] &\le 2\mathbb{E}[\phi_{k-1}] \\ &\le \cdots \\ &\le 2^k \phi_0. \end{aligned} \tag{3.3}$$

We now bound $L_{(n-r+1)}$ by a linear function in ϕ_k. By Lemma 3.7,

$$\Phi_k \ge \frac{r}{n} e^{\lambda L_{(n-r+1)}(S,h)} \implies \phi_k \ge \lambda L_{(n-r+1)}(S,h) - \log(n/r).$$

We isolate $L_{(n-r+1)}$ in this inequality, take expectations, and use Eq. (3.3) to deduce

$$\begin{aligned}\mathbb{E}[L_{(n-r+1)}(S,h)] &\le \frac{\mathbb{E}[\phi_k] + \log(n/r)}{\lambda} \\ &\le \frac{2^k\phi_0 + \log(n/r)}{\lambda} \\ &\le \frac{t - \mu + \log(n/r)}{\log(t/\mu)} \\ &= t,\end{aligned}$$

where the last equality follows from Eq. (3.2).

□

We now similarly prove the rth smallest load is lower bounded by the analogous quantity for independent Poisson random variables. The proof is nearly identical: we now set $\lambda < 0$ instead.

Theorem 3.11. *Let $u \ge \ell \ge 1$, $1 \le r \le n \coloneqq 2^\ell$, and $m \ge n \log n$ be integers, and let $S \subseteq \mathbb{F}_2^u$ be an arbitrary set of cardinality m. For a uniformly random linear $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$,*

$$\mathbb{E}[L_{(r)}(S,h)] \ge T_r^-(m,n).$$

Proof. Let $k \coloneqq u - \ell$, $t \coloneqq T_r^-(m,n)$, $\mu \coloneqq m/n$, and $\lambda \coloneqq \log(t/\mu)$. Note $\lambda < 0$.

By Lemma 3.6, we can assume without loss of generality that h is a random *surjective* linear map. To pick h, we will pick linearly independent vectors $v_1, \dots v_k \in \mathbb{F}_2^u$ uniformly at random, and then pick a uniformly random h with kernel $V \coloneqq \operatorname{span}(v_{\le k}) =: V_k$. To sample the $v_{\le k}$, we will iteratively pick $v_i \sim \mathbb{F}_2^u \setminus V_{i-1}$ for $i = 1, \dots, k$.

Using the vectors $v_{\le k}$ and set S, define $\{S_i(x) \coloneqq |(x + V_i) \cap S|\}_{0 \le i \le k}$, the potentials $\{\Phi_i \coloneqq \mathbb{E}_{x \sim \mathbb{F}_2^u}[e^{\lambda S_i(x)}]\}_{0 \le i \le k}$, and the log-potentials $\{\phi_i \coloneqq \log \Phi_i\}_{0 \le i \le k}$. We will first show that $\mathbb{E}[\phi_k]$ is bounded. We will next show that $L_{(r)}$ is bounded by a linear function in ϕ_k. Linearity of expectation will then give us a bound on $\mathbb{E}[L_{(r)}]$.

As $|S| = m$, we can compute

$$\Phi_0 = \mathbb{E}_x[e^{\lambda S_0(x)}] = \frac{m}{2^u}e^{\lambda} + \left(1 - \frac{m}{2^u}\right) = 1 + \frac{m(e^{\lambda} - 1)}{2^u} = 1 + \frac{t-\mu}{2^k}.$$

Therefore,

$$\phi_0 = \log\left(1 + \frac{t-\mu}{2^k}\right) \leq \frac{t-\mu}{2^k}.$$

By Jensen's inequality and Lemma 3.8, we deduce for each $0 \leq i < k$,

$$\mathbb{E}_{v_{i+1}}[\phi_{i+1} \mid v_{\leq i}] = \mathbb{E}_{v_{i+1}}[\log(\Phi_{i+1}) \mid v_{\leq i}] \leq \log(\mathbb{E}_{v_{i+1}}[\Phi_{i+1} \mid v_{\leq i}]) \leq \log(\Phi_i^2) = 2\phi_i.$$

Therefore, we can repeatedly apply linearity of expectation and the law of total expectation to deduce

$$\begin{aligned} \mathbb{E}[\phi_k] &\leq 2\mathbb{E}[\phi_{k-1}] \\ &\leq \cdots \\ &\leq 2^k \phi_0. \end{aligned} \tag{3.4}$$

We now bound $L_{(r)}$ by a linear function in ϕ_k. By Lemma 3.7,

$$\Phi_k \geq \frac{r}{n}e^{\lambda L_{(r)}(S,h)} \implies \phi_k \geq \lambda L_{(r)}(S,h) - \log(n/r).$$

We isolate $L_{(r)}$ in this inequality, reversing the inequality because $\lambda < 0$. Taking expectations and using Eq. (3.4), we deduce

$$\begin{aligned} \mathbb{E}[L_{(r)}(S,h)] &\geq \frac{\mathbb{E}[\phi_k] + \log(n/r)}{\lambda} \\ &\geq \frac{2^k\phi_0 + \log(n/r)}{\lambda} \\ &\geq \frac{t - \mu + \log(n/r)}{\log(t/\mu)} \\ &= t, \end{aligned}$$

where the last equality follows from Eq. (3.2).

□

At this point, the reader interested in getting tight explicit asymptotic upper or lower bounds on the expected rth load in any ball/bin regime can simply disregard linear maps and analyze $T_r^+(m,n)$ or $T_r^-(m,n)$ directly using the Lambert W function and its Taylor expansion [CGH+96]. For example, in the canonical setting of $m = n$, $T_1^+(n,n)$ is the real $t \geq 1$ such that $t \log t - t + 1 = \log n$. This solves to $t = \frac{\log n - 1}{W((\log n - 1)/e)}$. Taylor expanding W and gathering the first few terms of the reciprocal series yields

$$t = \frac{\log n}{\log \log n}\left(1 + \frac{\log \log \log n + 1}{\log \log n} + (1 + o(1))\left(\frac{\log \log \log n}{\log \log n}\right)^2\right).$$

Similar calculations can be done for any m and n and to any fixed asymptotic order.

We now establish two corollaries of this Poisson phenomenon: optimal expected max load and two-sided bounds.

3.3.1 Additively Optimal Expected Max Load

In this section, we show that in the canonical setting of n balls into n bins, the expected max load of a random linear map is extremely close to the expected max load of a fully random map. To start, we first bound how close $T_1^+(m,n)$ is to the max load of a fully random map in the sparse regime. As the number of balls decreases with respect to the number of bins, the Chernoff threshold grows with respect to the average load. Letting $\mu \coloneqq m/n$ be the average load, and $t \coloneqq T_1^+(m,n)$ be the Chernoff threshold, we show in the sparse regime where t/μ grows sufficiently fast, t is within $3/2 + o(1)$ of the expected max load of a fully random map. The proof is inspired by the classic max load lower bounds established via the Poisson Approximation theorem [MU05, Lemma 5.12].

Lemma 3.12. *Let n and $m \coloneqq m(n)$ be integers, let $0 < \eta(n) \leq 1$, and set $\mu \coloneqq m/n$ and $t \coloneqq T_1^+(m,n)$. Define $R(m,n) \coloneqq \mathbb{E}_f[\max_{i \in [n]} |f^{-1}(i)|]$, where $f : [m] \to [n]$ is a uniformly random function. If $\lim_{n\to\infty} \frac{(t/\mu)^\eta}{\sqrt{\mu}\log(e+t)} = \infty$, then*

$$R(m,n) \geq \left\lfloor t - \frac{1}{2} - \eta \right\rfloor - o(1).$$

Proof. Let $\delta \coloneqq \frac{1}{2} + \eta$ and set $s \coloneqq \lfloor t - \delta \rfloor$. If $s \leq \mu$, then since $R(m,n) \geq \mu$, it follows that $R(m,n) \geq \mu \geq s$. Henceforth, assume $s > \mu$, which also implies $s \geq 1$.

Let $X_1, \dots X_n$ be independent $\mathrm{Pois}(\mu)$ random variables. By the Poisson approximation theorem for monotonically decreasing events [MU05, Corollary 5.11],

$$\Pr\left[\max_{i\in[n]} |f^{-1}(i)| < s\right] \leq 2\Pr\left[\max_i X_i < s\right] = 2(\Pr[X_1 < s])^n \leq 2e^{-n\Pr[X_1 \geq s]},$$

where the last inequality used $1 - x \leq e^{-x}$. We can crudely bound

$$R(m,n) \geq s \cdot \Pr\left[\max_{i\in[n]} |f^{-1}(i)| \geq s\right] = s - s\Pr\left[\max_{i\in[n]} |f^{-1}(i)| < s\right] > s - 2se^{-n\Pr[X_1\geq s]}.$$

It remains to show $2se^{-n\Pr[X_1\geq s]} = o(1)$. Using Stirling's formula [Rob55], we estimate

$$\Pr[X_1 = s] = e^{-\mu}\frac{\mu^s}{s!} \geq e^{-\mu}\left(\frac{e\mu}{s}\right)^s \frac{1}{e^{1/12s}\sqrt{2\pi s}} = \frac{e^{-D(s,\mu)}}{e^{1/12s}\sqrt{2\pi s}} \geq \frac{e^{-D(s,\mu)}}{2\sqrt{\pi s}}.$$

We now use the definition of t to lower bound

$$n\Pr[X_1 \geq s] \geq e^{D(t,\mu)} \cdot \Pr[X_1 = s] \geq \frac{e^{D(t,\mu)-D(s,\mu)}}{2\sqrt{\pi s}} \geq \frac{e^{\delta\log(s/\mu)}}{2\sqrt{\pi s}} = \frac{1}{2\sqrt{\pi\mu}}\left(\frac{s}{\mu}\right)^\eta,$$

where the last inequality follows from Proposition 2.1. As $s \geq 1$ and $s = \lfloor t - \delta \rfloor$, it follows $s \geq t/4$. Hence, by the limit assumption,

$$n\Pr[X_1 \geq s] \geq \frac{1}{2\sqrt{\pi\mu}}\left(\frac{s}{\mu}\right)^\eta \geq \frac{1}{2\cdot 4^\eta\sqrt{\pi\mu}}\left(\frac{t}{\mu}\right)^\eta = \omega(\log(e+t)).$$

Therefore,

$$R(m,n) \geq s - 2se^{-n\Pr[X_1\geq s]} \geq s - 2te^{-\omega(\log(e+t))} = s - o(1).$$

□

We can now deduce Theorem 3.2.

Proof of Theorem 3.2. Denote $t \coloneqq T_1^+(n,n)$. Let $\eta(n) \coloneqq \min\{1, \sqrt{1/\log t}\} = o(1)$. We see $\frac{t^\eta}{\log(e+t)} = \frac{e^{\sqrt{\log t}}}{\log(e+t)} \to \infty$ as long as $t \to \infty$. Indeed, $D(t,1) = \log n$ implies

$$\log n = t\log(t/e) + 1 < t^2/e + 1 \implies t > \sqrt{e\log(n/e)} = \omega(1).$$

Hence, we can invoke Lemma 3.12 and Theorem 3.10 with $m = n$ and $r = 1$ to get

$$\mathbb{E}[M(S,h)] \le t \le \mathbb{E}[M([n],f)] + t - \left\lfloor t - \frac{1}{2} - \eta \right\rfloor + o(1) = \mathbb{E}[M([n],f)] + \frac{1}{2} + c_\ell + o(1),$$

where $c_\ell \coloneqq \left\{t - \frac{1}{2} - \eta\right\} = \left\{T_1^+(2^\ell, 2^\ell) - \frac{1}{2} - \eta(2^\ell)\right\}$. Denote $t_\ell \coloneqq T_1^+(2^\ell, 2^\ell)$, $\eta_\ell \coloneqq \eta(2^\ell)$, and $d_\ell \coloneqq t_\ell - \frac{1}{2} - \eta_\ell$, whence $c_\ell = \{d_\ell\}$. Notice that to show a subsequence of (c_ℓ) goes to 0, it suffices to show that (d_ℓ) is monotone increasing, $d_\ell \to \infty$, and $d_{\ell+1} - d_\ell \to 0$. With these three claims, for each integer we can take the first ℓ at which d_ℓ crosses that integer; since the increments are $o(1)$, the overshoot, and hence the corresponding fractional parts, approach 0. The first two claims hold because t_ℓ increases monotonically to ∞ and η_ℓ decreases monotonically to 0. For the final claim, we first observe

$$\log 2 = \log(2^{\ell+1}) - \log(2^\ell) = D(t_{\ell+1}, 1) - D(t_\ell, 1) \ge (t_{\ell+1} - t_\ell)\log t_\ell$$

by Proposition 2.1. Therefore,

$$d_{\ell+1} - d_\ell = t_{\ell+1} - t_\ell - \eta_{\ell+1} + \eta_\ell \le \frac{\log 2}{\log t_\ell} + o(1) = o(1)$$

since $t_\ell \to \infty$. This proves all three claims. □

We note that the same $\frac{3}{2} + o(1)$ gap can be derived for small enough m relative to n (as dictated by the limit condition in Lemma 3.12). We do not include the explicit calculations here.

3.3.2 Two-sided Bounds

In this section, we use Theorems 3.10 and 3.11 to prove sharp load concentration in the dense regime when $m \ge n\log n$. First, we will need estimates on $T_1^+(m,n)$ and $T_1^-(m,n)$ in the dense regime.

Proposition 3.13. *Let n and $m \geq n \log n$ be integers, and denote $\mu \coloneqq m/n$. We have*

$$T_1^+(m,n) - T_1^-(m,n) \leq 4\sqrt{\mu \log n}.$$

The leading constant of 4 can surely be improved.

Proof. Let $a \coloneqq T_1^+(m,n) - \mu$ and $b \coloneqq \mu - T_1^-(m,n)$.

$$\begin{aligned}
\log n = D(\mu + a, \mu) - D(\mu, \mu) &= \int_{\mu}^{\mu+a} \log(x/\mu)\, dx \\
&= \mu \int_0^{a/\mu} \log(1+u)\, du \\
&\geq \mu \int_0^{a/\mu} \frac{u}{1+u}\, du \\
&\geq a \cdot \frac{a/\mu}{1 + a/\mu} = \frac{a^2}{\mu + a}.
\end{aligned}$$

Solving this inequality using the quadratic formula and $a > 0$, we obtain

$$a \leq \frac{\log n + \sqrt{\log^2 n + 4\mu \log n}}{2} \leq \frac{\sqrt{\mu \log n} + \sqrt{5\mu \log n}}{2} \leq 2\sqrt{\mu \log n}.$$

Similarly, we can bound b by noting

$$\begin{aligned}
\log n = D(\mu - b, \mu) - D(\mu, \mu) &= \int_{\mu}^{\mu-b} \log(x/\mu)\, dx \\
&= -\mu \int_0^{b/\mu} \log(1-u)\, du \\
&\geq \mu \int_0^{b/\mu} u\, du = \frac{b^2}{2\mu}.
\end{aligned}$$

Thus $b \leq \sqrt{2\mu \log n}$ and

$$T_1^+(m,n) - T_1^-(m,n) = a + b \leq 4\sqrt{\mu \log n}.$$

□

Our two-sided bounds are now one use of Markov's inequality away.

Theorem 3.14. *Let $u \geq \ell \geq 1$, $n \coloneqq 2^\ell$, and $m \geq n \log n$ be integers, and let $S \subseteq \mathbb{F}_2^u$ be an arbitrary set of cardinality m. For a uniformly random linear $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$, we have*

$$\Pr\left[\forall y \in \mathbb{F}_2^\ell,\ \left|L_y(S,h) - \frac{m}{n}\right| \leq \frac{4}{\varepsilon}\sqrt{\frac{m \log n}{n}}\right] \geq 1 - \varepsilon.$$

Proof. Let $\mu \coloneqq m/n$, $t^+ \coloneqq T_1^+(m,n)$, and $t^- \coloneqq T_1^-(m,n)$. Note that $L_{(1)}(S,h) \leq \mu \leq L_{(n)}(S,h)$ for all h. Hence, by Markov's inequality on $\mu - L_{(1)}(S,h)$ and $L_{(n)}(S,h) - \mu$, we obtain

$$\Pr[L_{(n)}(S,h) \geq \mu + \Delta] \leq \frac{t^+ - \mu}{\Delta}, \qquad \Pr[L_{(1)}(S,h) \leq \mu - \Delta] \leq \frac{\mu - t^-}{\Delta}.$$

Therefore,

$$\begin{aligned}
\Pr[\forall y : L_y(S,h) \in [\mu - \Delta, \mu + \Delta]] &= 1 - \Pr\left[\begin{smallmatrix} L_{(n)}(S,h) \geq \mu + \Delta \text{ or} \\ L_{(1)}(S,h) \leq \mu - \Delta \end{smallmatrix}\right] \\
&\geq 1 - \frac{t^+ - \mu}{\Delta} - \frac{\mu - t^-}{\Delta} \\
&= 1 - \frac{t^+ - t^-}{\Delta} \\
&\geq 1 - \frac{4\sqrt{\mu \log n}}{\Delta}.
\end{aligned}$$

Setting $\Delta \coloneqq 4\varepsilon^{-1}\sqrt{\mu \log n}$ yields the desired result.

□

3.4 Tail Bounds

After obtaining optimal expected max load, it is natural to ask how strong of tail bounds we can achieve. Expectation on its own only yields constant probability tail bounds using Markov's inequality. In this section, we show that linear hash maps enjoy sharper inverse-polylogarithmic tail bounds. On the negative side, we also show that they do not have the inverse-polynomial tail bounds that fully random functions possess.

3.4.1 Tail Bounds For Squaring Random Variables

To show our tail bounds, we first show generic tail bounds for random variables whose conditional expectations square and satisfy a strong monotonicity condition.

Theorem 3.15. *Let $X_0 > 1$ be a constant, and let $X_1, \ldots, X_k$ be random variables satisfying $X_{i+1} - 1 \geq 2(X_i - 1)$ and $\mathbb{E}[X_{i+1} | X_{\leq i}] \leq X_i^2$ for all i. For $t > X_0$, we have*

$$\Pr[X_k \geq t^{2^k}] \leq \left(\frac{X_0 - 1}{t - 1}\right)^2.$$

Remark 3.16. This bound is asymptotically tight as $t \to \infty$. Fix $t > X_0$ and define

$$X_1 \coloneqq \begin{cases} t^2, & \text{with probability } \frac{(X_0-1)^2}{t^2 - 2X_0 + 1}, \\ 2X_0 - 1, & \text{otherwise,} \end{cases} \qquad \text{and} \qquad X_{i+1} \coloneqq X_i^2 \quad (i \geq 1).$$

It can be shown that the conditions of Theorem 3.15 are satisfied. Since $2X_0 - 1 < X_0^2 < t^2$, we have $X_k \geq t^{2^k}$ if and only if $X_1 = t^2$. Hence,

$$\begin{aligned} \Pr[X_k \geq t^{2^k}] &= \frac{(X_0 - 1)^2}{t^2 - 2X_0 + 1} \\ &= \frac{(t-1)^2}{t^2 - 2X_0 + 1} \left(\frac{X_0 - 1}{t - 1}\right)^2 \\ &= (1 - o_t(1)) \left(\frac{X_0 - 1}{t - 1}\right)^2. \end{aligned}$$

Proof. Define the function $\psi(u) \coloneqq \min\{1, u^2\}$, and define $Y_i \coloneqq \psi\left(\frac{X_i - 1}{t^{2^i} - 1}\right)$. We will show the following key claim.

Proposition 3.17. *$(Y_i)_{i \geq 0}$ is a supermartingale.*

From this claim we will have that $\mathbb{E}[Y_k] \leq Y_0 \leq \left(\frac{X_0 - 1}{t - 1}\right)^2$, and so by Markov's inequality,

$$\begin{aligned} \Pr[X_k \geq t^{2^k}] &= \Pr\left[\frac{X_k - 1}{t^{2^k} - 1} \geq 1\right] \\ &= \Pr[Y_k \geq 1] \\ &\leq \mathbb{E}[Y_k] \\ &\leq \left(\frac{X_0 - 1}{t - 1}\right)^2. \end{aligned}$$

□

Proof of Proposition 3.17. Fix an i, and condition on values of $X_{\leq i}$. Let $\delta_i \coloneqq X_i - 1$ and $\tau_i \coloneqq t^{2^i} - 1$. We note that $\tau_{i+1} = \tau_i(\tau_i+2)$, $\delta_{i+1} \geq 2\delta_i$, and $\mathbb{E}[\delta_{i+1} \mid \delta_i] \leq \delta_i(\delta_i+2)$. We will prove the following generic bound, from which the proposition will follow.

Claim 3.18. *Let $\tau \geq \delta > 0$ be real numbers. Let d be a random variable such that $d \geq 2\delta$ and $\mathbb{E}[d] \leq \delta(\delta+2)$. Then*

$$\mathbb{E}\left[\psi\left(\frac{d}{\tau(\tau+2)}\right)\right] \leq \psi\left(\frac{\delta}{\tau}\right).$$

Proof of Claim 3.18. Let $T \coloneqq \tau(\tau+2)$. The expectation above is clearly upper bounded by 1, so it suffices to show

$$\mathbb{E}\left[\psi\left(\frac{d}{T}\right)\right] \leq \frac{\delta^2}{\tau^2}.$$

We will upper bound $\psi(d/T)$ on the domain $d \geq 2\delta$ by a linear function, and then apply linearity of expectation. Define

$$\mathcal{L}(x) \coloneqq \psi(2\delta/T) + \frac{\psi(1) - \psi(2\delta/T)}{T - 2\delta}(x - 2\delta)$$

to be the unique line satisfying $\mathcal{L}(2\delta) = \psi(2\delta/T)$ and $\mathcal{L}(T) = \psi(1) = 1$. Note that $T > 2\delta$, so this line is increasing.

We claim that for $x \geq 2\delta$, $\mathcal{L}(x) \geq \psi(x/T)$. When $x > T$, we see $\mathcal{L}(x) \geq \mathcal{L}(T) = 1 = \psi(x/T)$. When $x \in [2\delta, T]$, the function $\psi(x/T)$ is convex, so the secant line $\mathcal{L}$ lies above it. Therefore,

$$\mathbb{E}\left[\psi\left(\frac{d}{T}\right)\right] \leq \mathbb{E}[\mathcal{L}(d)] = \mathcal{L}(\mathbb{E}[d]) \leq \mathcal{L}(\delta(\delta+2)).$$

We now compute

$$\begin{aligned}
\mathcal{L}(\delta(\delta+2)) &= \frac{4\delta^2}{T^2} + \frac{1-4\delta^2/T^2}{T-2\delta}\cdot\delta^2 \\
&= \frac{\delta^2}{\tau^2}\left(\frac{4}{(\tau+2)^2} + \frac{T^2-4\delta^2}{(\tau+2)^2(T-2\delta)}\right) \\
&= \frac{\delta^2}{\tau^2}\cdot\frac{4+T+2\delta}{(\tau+2)^2} \\
&= \frac{\delta^2}{\tau^2}\cdot\frac{\tau^2+2\tau+2\delta+4}{(\tau+2)^2} \\
&\le \frac{\delta^2}{\tau^2}\cdot\frac{\tau^2+4\tau+4}{(\tau+2)^2} \\
&\le \frac{\delta^2}{\tau^2},
\end{aligned}$$

yielding the desired result. □

Applying the claim with $d \leftarrow \delta_{i+1}$, $\delta \leftarrow \delta_i$ and $\tau \leftarrow \tau_i$ shows that $\mathbb{E}\left[\psi\left(\frac{\delta_{i+1}}{\tau_{i+1}}\right)|X_{\le i}\right] \le \psi\left(\frac{\delta_i}{\tau_i}\right)$, which implies the proposition.

□

3.4.2 Applying the Bounds to the Potentials

With Theorem 3.15 in hand, we can now show linear hashing achieves $(1 + o(1))\frac{\log n}{\log\log n}$ max-load with probability $1 - o(1)$.

Theorem 3.19. *Let $u \ge \ell \ge 1$ be integers, and $n \coloneqq 2^\ell$. Let $S \subseteq \mathbb{F}_2^u$ be of cardinality n, and let $\mathcal{H}$ be the set of all linear maps $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$. Let t be the unique real number such that $(t/e)^t = n$. For any $r \ge 1$,*

$$\Pr_{h\sim\mathcal{H}}[M(S,h) \ge rt] \le \frac{e^2}{(r+1)^2 t^{2-2/r}}.$$

Proof. By Lemma 3.6, we can assume h is surjective. Let $k \coloneqq u - \ell$ and $\lambda \coloneqq 1 + \frac{\log t}{r}$. To pick $h \sim \mathcal{H}$, we will pick linearly independent vectors $v_1, \ldots, v_k \in \mathbb{F}_2^u$ uniformly at random, and then pick a uniformly random h with kernel $V \coloneqq \operatorname{span}(v_{\le k})$. In

particular, for $i = 1, \ldots, k$ we will iteratively pick $v_i \sim \mathbb{F}_2^u \setminus V_{i-1}$. Using S and $v_{\leq k}$, define the functions $\{S_i(x) \coloneqq |(x + V_i) \cap S|\}_{0 \leq i \leq k}$ and the potentials $\{\Phi_i \coloneqq \mathbb{E}_{x \sim \mathbb{F}_2^u}[e^{\lambda S_i(x)}]\}_{0 \leq i \leq k}$. If some preimage of h had load at least rt, then $|(x+V) \cap S| \geq rt$ for some shift x. Lemma 3.7 then implies

$$\Phi_k \geq \frac{e^{\lambda rt}}{2^{u-k}} = \frac{e^{rt} t^t}{n} = e^{t(r+1)}.$$

We can easily compute

$$\Phi_0 = \mathbb{E}_x[e^{\lambda S_0(x)}] = 1 - \frac{n}{2^u} + \frac{ne^\lambda}{2^u} = 1 + \frac{e^\lambda - 1}{2^k}.$$

By Lemmas 3.8 and 3.9, the sequence (Φ_i) satisfies the hypotheses of Theorem 3.15. Hence,

$$\begin{aligned}
\Pr[M(S,h) \geq rt] \leq \Pr[\Phi_k \geq e^{t(r+1)}] &\leq \left(\frac{\Phi_0 - 1}{e^{(r+1)t/2^k} - 1} \right)^2 \\
&\leq \left(\frac{e^\lambda / 2^k}{(r+1)t/2^k} \right)^2 && (e^x - 1 \geq x) \\
&= \left(\frac{et^{1/r}}{(r+1)t} \right)^2 \\
&= \frac{e^2}{(r+1)^2 t^{2-2/r}}.
\end{aligned}$$

□

3.4.3 On the Optimality of our Tail Bounds

At this point, the only remaining pseudorandom property we can ask for is optimal tail bounds. A fully random map is able to achieve max-load $(1+o(1))\frac{\log n}{\log \log n}$ with probability $1 - 1/n^{\Omega(1)}$. Can a random linear map achieve such a strong probability guarantee? We show the answer is no, and that a linear map can only guarantee optimal max-load with probability at most $1 - 2^{-O((\log \log n)^2)}$. More generally, we show the following.

Theorem 3.20. *Let $h : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ be a random linear map. There exists $S \subseteq \mathbb{F}_2^u$ of size $n \coloneqq 2^\ell$ such that for every $2 \le r \le \ell$,*

$$\Pr_h[M(S,h) \ge 2^r] \ge 2^{-r(r+1)}.$$

Proof. Let $S \coloneqq \mathbb{F}_2^\ell \times \{0\}^{u-\ell}$. Let $b_1, \dots, b_\ell$ be a basis of S, and consider the matrix H whose rows are $h(b_1), \dots, h(b_\ell)$. On the event that $\operatorname{rank}(H) = \ell - r$, we have $M(S,h) \ge |h^{-1}(0) \cap S| = |\ker(h|_S)| = 2^r$.

For random h, the matrix H is uniformly random. The event $\operatorname{rank}(H) = \ell - r$ occurs when the image of H is a codimension-r subspace of $\mathbb{F}_2^\ell$. The Gaussian binomial coefficient satisfies

$$\binom{\ell}{\ell - r}_2 = \prod_{j=0}^{r-1} \frac{2^\ell - 2^j}{2^r - 2^j} \ge 2^{r(\ell-r)},$$

since each factor is at least $2^{\ell-r}$. For a fixed codimension-r subspace V, the number of linear maps from S onto V is

$$\prod_{j=0}^{\ell-r-1} (2^\ell - 2^j) = 2^{\ell(\ell-r)} \prod_{i=r+1}^{\ell} (1 - 2^{-i}) \ge 2^{\ell(\ell-r)-1}.$$

Since there are 2^{ℓ^2} possible matrices H, it follows that

$$\Pr[\operatorname{rank}(H) = \ell - r] \ge \frac{2^{r(\ell-r)} 2^{\ell(\ell-r)-1}}{2^{\ell^2}} = 2^{-r^2-1} \ge 2^{-r(r+1)}.$$

Thus, $\Pr_h[M(S,h) \ge 2^r] \ge 2^{-r(r+1)}$. □

Chapter 4: Random Toeplitz Maps

4.1 Introduction

A Toeplitz matrix is a matrix whose main diagonals are constant. Random Toeplitz matrices are a classical and practical construction of hash functions. They have been described as the "simplest bit-efficient construction of a universal hash function" [HILL99]. Due to their extreme efficiency, they are implemented in practice as extractors [SFI+11]. Hash functions satisfying linearity are desired objects in cryptography. However, taking a random linear map may require more bits of randomness to sample than wanted. Toeplitz matrices require the minimal amount of random bits to sample, and are thus a go-to linear hash map in this field. Examples include the Goldreich–Levin construction of hard-core predicates [GL89], Håstad, Impagliazzo, Levin, and Luby's constructions of pseudorandom generators [HILL99], Goldreich, Impagliazzo, Levin, Venkatesan, and Zuckerman's privacy amplification of one-way functions [GIL+90], and Krawczyk's construction of message authentication schemes [Kra94]. Toeplitz matrices are among the simplest universal hash functions from a complexity-theoretic standpoint. Mansour, Nisan, and Tiwari [MNT93] prove that Toeplitz matrices achieve optimal time-space tradeoffs for universal hashing.

In this chapter, we study a basic property of random Toeplitz matrices: their expected max-load. When n balls are tossed into n bins via a random Toeplitz matrix, what is the expected number of balls in the heaviest bin? A folklore argument shows that for any 2-universal hash function (and therefore Toeplitz matrices), the expected max-load is $O(\sqrt{n})$, and beyond this no progress has been made. Indeed, this question has been explicitly asked in the literature. The seminal work of Alon, Dietzfelbinger, Miltersen, Petrank, and Tardos [ADM+99], which showed that random linear matrices achieve logarithmic expected max-load, asked about the occupancy properties of other well-known hash families and explicitly mentioned three families as examples: modular arithmetic over $\mathbb{Z}/p\mathbb{Z}$ [CW79, FKS84], integer arithmetic, and

Boolean convolution. In 2016, a beautiful work of Knudsen gave an improved expected max-load bound of $\tilde{O}(n^{1/3})$ for the first two hash families [Knu19]. The third hash family is equivalent to Toeplitz hashing.

A recent result adds motivation to studying Toeplitz matrices. As shown in Theorem 3.2, a random $\mathbb{F}_2$-matrix achieves the optimal expected max-load of $O\left(\frac{\log n}{\log\log n}\right)$. However, when mapping a poly(n)-sized vector space to a size-n space, this construction uses $O(\log^2 n)$ bits of randomness. It is natural to ask whether a derandomized family of matrices exists. Indeed, the result of Theorem 3.2 implies the *existence* of a poly(n)-sized subfamily of matrices that achieves optimal expected max-load, but finding an explicit family using $O(\log n)$ bits of randomness remains open. A very natural and practical candidate for this is Toeplitz matrices.

In this chapter, we prove a cube-root bound on the expected max-load of a random Toeplitz matrix.

Theorem 4.1. *Let $u \geq 2\ell \geq 1$ and $n = 2^\ell$. Let $T \in \mathbb{F}_2^{\ell\times u}$ be a random Toeplitz matrix, and let $S \subseteq \mathbb{F}_2^u$ be an arbitrary set of size n. We have*

$$\mathbb{E}_T\left[\max_{y\in\mathbb{F}_2^\ell} |T^{-1}y \cap S|\right] = O((n\log n)^{1/3}).$$

4.2 Equivalences

Here we will first show the equivalence between Toeplitz hashing, Boolean convolution hashing, and a polynomial multiplication-based hash. In the next section, we will directly analyze the polynomial hash function to get a $\tilde{O}(n^{1/3})$ bound on the expected max-load of Toeplitz hashing.

For $c \in \mathbb{F}_2^{u+\ell-1}$, define the Toeplitz matrix T^c to satisfy $T^c_{ij} = c_{j-i+\ell}$.

$$T^c := \begin{pmatrix} c_\ell & c_{\ell+1} & c_{\ell+2} & \cdots & c_{u+\ell-1} \\ c_{\ell-1} & c_\ell & c_{\ell+1} & \cdots & c_{u+\ell-2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_2 & c_3 & c_4 & \cdots & c_{u+1} \\ c_1 & c_2 & c_3 & \cdots & c_u \end{pmatrix}.$$

For a string $c \in \mathbb{F}_2^{u+\ell-1}$ and $v \in \mathbb{F}_2^u$, define the convolution

$$c \circ v \coloneqq \left(\sum_{j=1}^{u} c_{i+j-1} v_j \right)_{i \in [\ell]} \in \mathbb{F}_2^\ell.$$

Finally, define $p_c = \sum_i c_i x^{i-1} \in \mathbb{F}_2[x]$. For a string s, let $\bar{s}$ be its reverse.

Theorem 4.2. $p_{\overline{T^c v}} = p_{c \circ v} = \left(p_c p_{\bar{v}} \pmod{x^{u+\ell-1}} \right) \operatorname{div} x^{u-1}$

Proof. We first show that $\overline{c \circ v} = T^c v$. Indeed,

$$(T^c v)_i = \sum_{j=1}^{u} T^c_{ij} v_j = \sum_{j=1}^{u} c_{j-i+\ell} v_j = \sum_{j=1}^{u} c_{(\ell+1-i)+j-1} v_j = (c \circ v)_{\ell+1-i}$$

for every $i \in [\ell]$. Hence, $\overline{c \circ v} = T^c v$, and the first equality follows.

Now write

$$p_c p_{\bar{v}} = \left(\sum_{i=1}^{u+\ell-1} c_i x^{i-1} \right) \left(\sum_{j=1}^{u} \bar{v}_j x^{j-1} \right) = \sum_{i=1}^{u+\ell-1} \sum_{j=1}^{u} c_i \bar{v}_j x^{i+j-2}.$$

For $k \in \{0, 1, \ldots, \ell-1\}$, the coefficient of x^{u+k-1} is

$$\sum_{\substack{i \in [u+\ell-1],\, j \in [u] \\ i+j=u+k+1}} c_i \bar{v}_j = \sum_{j=1}^{u} c_{u+k+1-j} \bar{v}_j = \sum_{j=1}^{u} c_{u+k+1-j} v_{u+1-j} = \sum_{j=1}^{u} c_{k+j} v_j = (c \circ v)_{k+1}.$$

Therefore the coefficients of degrees $u-1, u, \ldots, u+\ell-2$ in $p_c p_{\bar{v}}$ are exactly the coordinates of $c \circ v$, which is equivalent to

$$p_{c \circ v} = \left(p_c p_{\bar{v}} \bmod x^{u+\ell-1} \right) \operatorname{div} x^{u-1}.$$

□

This lemma establishes the equivalence of the expected max-load of three hash families.

Corollary 4.3. *Let $S \subseteq \mathbb{F}_2^u$, and set $S' \coloneqq \{p_{\bar{v}} : v \in S\} \subseteq \mathbb{F}_2[x]_{<u}$. Consider the following three hash families.*

1. $H_c^1 : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ *defined by* $H_c^1(v) \coloneqq T^c v$ *for* $c \sim \mathbb{F}_2^{u+\ell-1}$;

2. $H_c^2 : \mathbb{F}_2^u \to \mathbb{F}_2^\ell$ *defined by* $H_c^2(v) \coloneqq c \circ v$ *for* $c \sim \mathbb{F}_2^{u+\ell-1}$;

3. $H_r^3 : R_{u+\ell-1} \to \mathbb{F}_2[x]_{<\ell}$ $H_r^3(q) \coloneqq rq \operatorname{div} x^{u-1}$ *for* $r \sim R_{u+\ell-1}$.

We have

$$\mathbb{E}_c[M(S, H_c^1)] = \mathbb{E}_c[M(S, H_c^2)] = \mathbb{E}_r[M(S', H_r^3)].$$

Proof. For each fixed c, the outputs of H_c^1 and H_c^2 differ only by reversal, so $M(S, H_c^1) = M(S, H_c^2)$.

Next fix c and let $r = p_c$. By Theorem 4.2, $p_{H_c^2(v)} = H_r^3(p_{\bar{v}})$ for every $v \in \mathbb{F}_2^u$. The map $v \mapsto p_{\bar{v}}$ is a bijection from $\mathbb{F}_2^u$ to $\mathbb{F}_2[x]_{<u}$, and the map from $\mathbb{F}_2^\ell$ to $\mathbb{F}_2[x]_{<\ell}$ given by $y \mapsto p_y$ is also a bijection. Hence H_c^2 on S and H_r^3 on S' induce exactly the same multiset of bin loads. Therefore $M(S, H_c^2) = M(S', H_r^3)$, and taking expectations over uniformly random c proves the claim. □

4.3 Main Result

This section will be dedicated to proving the following main result.

Theorem 4.4. *Let* $u \geq 2\ell \geq 1$ *be integers, and* $n \coloneqq 2^\ell$. *Let* $T \in \mathbb{F}_2^{\ell \times u}$ *be a random Toeplitz matrix. Let* $S \subseteq \mathbb{F}_2^u$ *be an arbitrary subset of size* $n \coloneqq 2^\ell$. *We have*

$$\mathbb{E}_T[M(S, T)] = O((n \log n)^{1/3}).$$

To do so, we will need to establish some structural results regarding balls and close-pairs in the ultrametric space $(R_m, d_{\deg})$. But before that, we justify why we are looking at balls in this ultrametric space in the first place.

Lemma 4.5. *Let* $q \in \mathbb{F}_2[x]$ *and* $p \in \mathbb{F}_2[x]$.

$$q = p \operatorname{div} x^r \iff p \in B(qx^r, r).$$

Proof. By the division algorithm, we can express $p = ax^r + b$, where $\deg b < r$. $q = p \operatorname{div} x^r$ if and only if $a = q$. $p \in B(qx^r, r)$ if and only if $r > \deg(p + qx^r) = \deg((a+q)x^r + b)$. Since $\deg b < r$, this is true if and only if $a + q = 0 \iff a = q$.

□

Hence, the set of all polynomials with the same quotient when divided by a monomial is a ball in this ultrametric space. Hence, balls will naturally show up when analyzing the preimages of the hash function H_r^3 in Corollary 4.3. A closer inspection of this function reveals we will need to reason about scaled balls, which we now investigate.

Lemma 4.6. *Let B, B' be balls of radius r, and let $p, q \in R_m^\times$ be coprime of degree $d \leq r$.*

1. *$p^{-1}B$ is a disjoint union of at most 2^d balls of radius $r - d$.*
2. *If $d + r < m$, $p^{-1}B \cap q^{-1}B'$ is either empty or a ball of radius $r - d$.*

Proof. 1. Notice that $p^{-1}B(c, r) = p^{-1}c + p^{-1}B(0, r)$, so we may assume without loss of generality that $B = B(0, r)$. We claim

$$p^{-1}B(0,r) = \bigsqcup_{\substack{b \in R_m \\ \deg b < d}} B(p^{-1}b, r-d).$$

This would imply the desired result as balls of the same radius are either the same or disjoint. To show that the LHS is included in the RHS, consider any polynomial f of degree $< r$. We can write $f = ap + b$, where $\deg a < r - d$ and $\deg b < d$. Then $p^{-1}f = a + p^{-1}b \in B(p^{-1}b, r - d)$, which is one of the balls in the RHS.

For the reverse inclusion, consider $p^{-1}b + a$ for some polynomials $a, b \in R_m$ where $\deg a < r - d$ and $\deg b < d$. Then $p^{-1}b + a = p^{-1}(b + ap) \in p^{-1}B(0, r)$ because $\deg(b + ap) \leq \max(\deg(b), \deg(ap)) < r$.

2. By Item 1 and Proposition 2.6, it follows $p^{-1}B \cap q^{-1}B'$ is a disjoint union of some number of balls of radius $r-d$. We will show that any two elements of $p^{-1}B \cap q^{-1}B'$ have distance at most $r-d$. By Proposition 2.6, this implies all elements are contained in the same ball.

   Say we have two elements $p^{-1}b_1 = q^{-1}b_2$ and $p^{-1}c_1 = q^{-1}c_2$ in this intersection, where $b_1, c_1 \in B$ and $b_2, c_2 \in B'$. Adding both equalities, we see $p^{-1}(b_1+c_1) \equiv q^{-1}(b_2+c_2) \pmod{x^m}$, implying that

$$x^m | q(b_1+c_1) + p(b_2+c_2).$$

   Now since b_1, c_1 are in the same radius-r ball B, and b_2, c_2 are in the same radius-r ball B', $\deg(b_1+c_1) < r$ and $\deg(b_2+c_2) < r$. Hence

$$\deg(q(b_1+c_1)+p(b_2+c_2)) \leq \max(\deg(q(b_1+c_1)), \deg(p(b_2+c_2))) < d+r < m.$$

   Any polynomial divisible by x^m of degree less than m must be 0. Therefore,

$$p(b_2+c_2)+q(b_1+c_1) = 0 \implies p(b_2+c_2) = q(b_1+c_1) \implies p | q(b_1+c_1)$$

   Since p and q are coprime, we have $p | b_1+c_1$, implying that $p^{-1}(b_1+c_1)$ is some polynomial of degree $\deg(b_1+c_1) - \deg(p) < r-d$. Thus, $d_{\deg}(p^{-1}b_1, p^{-1}c_1) \leq r-d$ as desired.

□

We will now use the above lemma to deduce that if a set S has the property that pS heavily intersects a small-radius ball for many low-degree p, then S must have many close pairs.

Lemma 4.7. *Let $m, r \geq d > 1$ be integers such that $r+d < m$. Let $S \subseteq R_m$. Let $P \subseteq R_m^\times$ be a set of irreducible polynomials of degree d such that for each $p \in P$, pS intersects some radius-r ball in at least $M > 2^d$ points. Then $N_{r-d}(S) \geq \frac{(M-2^d)^2}{2^{d+1}}|P|$.*

Proof. For each $p \in P$, there exists a ball B_p of radius r such that $|pS \cap B_p| = |S \cap p^{-1}B_p| \geq M$. By Lemma 4.6, $p^{-1}B_p$ can be partitioned into at most 2^d balls of radius $r-d$. Let $B_p^1, \ldots, B_p^{k_p}$ be the subset of those balls that intersect S, where $k_p \leq 2^d$. Notice that any pair of points of S contained in the same B_p^i has distance at most $r-d$. Therefore, the number of pairs of polynomials in S of distance at most $r-d$ is at least

$$N_{r-d}(S) \geq \sum_{B \in \{B_p^i\}_{p \in P, i \in [k_p]}} |S \cap B|(|S \cap B| - 1) \geq \sum_{B \in \{B_p^i\}} (|S \cap B| - 1)^2.$$

If the balls (B_p^i) were pairwise distinct, each (i,p) would contribute new pairs to the tally and give us a strong lower bound. Although we are not promised this is the case, Item 2 of Lemma 4.6 tells us that there cannot be too much multiplicity present.

For a ball B, let the multiplicity of B, $m(B)$, be the number of (i,p) for which $B = B_p^i$. First note

$$m(B_i^p) = 1 + \sum_{\substack{q \in P \setminus \{p\} \\ j \leq k_q}} 1_{B_p^i = B_q^j}.$$

Now observe that by Lemma 4.6 that for each fixed $p \neq q \in P$, there is at most one pair (i,j) such that $B_p^i = B_q^j$. Thus, for each fixed $p \in P$ we can bound

$$\sum_{1 \leq i \leq k_p} m(B_p^i) = k_p + \sum_{\substack{q \in P \setminus \{p\} \\ j \leq k_q \\ i \leq k_p}} 1_{B_p^i = B_q^j} \leq 2^d + \sum_{q \in P \setminus \{p\}} 1 = 2^d + |P| - 1 \leq 2^{d+1},$$

where we used the fact $k_p \leq 2^d$ and $|P| \leq (1 + o(1))\frac{2^d}{d} < 2^d$ (via Proposition 2.3).

We can now bound

$$\begin{aligned}\sum_{B\in\{B_p^i\}}(|S\cap B|-1)^2 &= \sum_{\substack{p\in P\\ i\le k_p}}\frac{(|S\cap B_p^i|-1)^2}{m(B_p^i)}\\ &\ge \sum_{p\in P}\frac{(\sum_{i\le k_p}(|S\cap B_p^i|-1))^2}{\sum_{i\le k_p}m(B_p^i)}\\ &\ge \sum_{p\in P}\frac{(M-k_p)^2}{2^{d+1}}\\ &\ge \frac{(M-2^d)^2|P|}{2^{d+1}},\end{aligned}$$

where we used Cauchy-Schwarz, the fact $\sum_i |S\cap B_p^i| = |S\cap p^{-1}B_p| \ge M$, and $k_p \le 2^d$. □

We are now ready to show an $\tilde{O}(n^{1/3})$ max-load bound on a polynomial hash function that is not exactly equivalent to Toeplitz hashing, but can be analyzed easily.

Theorem 4.8. *Let $u \ge \ell \ge 1$ be integers, and define $n \coloneqq 2^\ell$. Let $S \subseteq R_{u+\ell-1}$ be a set of 2^ℓ polynomials. For $r \sim R^\times_{u+\ell-1}$, we define $H_r : R_{u+\ell-1} \to \mathbb{F}_2[x]^{<\ell}$ to be*

$$H_r(p) = rp \operatorname{div} x^{u-1}.$$

We have $\mathbb{E}_r[M(S,H_r)] \le C(n\log n)^{1/3}$, where C is independent of u and ℓ.

We note this is not equivalent to Toeplitz hashing via the function H_r^3 from Corollary 4.3 because $r \sim R^\times_{u+\ell-1}$, rather than $r \sim R_{u+\ell-1}$.

Proof. For any $M \le n$, let $\delta_M \coloneqq \Pr[M(S,H_r) \ge M]$. Let $t \sim R^\times_{u+\ell-1}$ be uniformly random. We will need the following claim (recall Definition 2.7).

Claim 4.9. *For any $1 < d \le \log_2(M/4)$,*

$$\frac{\delta_M(M-2^d)^2}{4d} \le \mathbb{E}_{t\sim R^\times_{u+\ell-1}}[N_{u-d}(tS)] \le \frac{2n}{2^d}.$$

Set $d \coloneqq \lfloor \log_2(M/4) \rfloor$. For any $M > (n \log n)^{1/3}$, $1 < d < \log_2 M$. Therefore, Claim 4.9 tells us

$$\delta_M \leq \frac{8nd}{(M - 2^d)^2 2^d}$$

for all $(n \log n)^{1/3} < M \leq n$. We then compute

$$\mathbb{E}[M(S, H_r)] < (n \log n)^{1/3} + 200n \sum_{M > (n \log n)^{1/3}} \frac{\log M}{M^3} \leq C(n \log n)^{1/3}.$$

□

We now prove Claim 4.9.

Proof of Claim 4.9. We first show the upper bound. Fix elements $s_1 \neq s_2$ of S, and let $x^g \coloneqq \gcd(s_1 + s_2, x^{u+\ell-1})$. Hence $s_1 + s_2 = x^g s'$, where $s' \in R^\times_{u+\ell-1}$. We now see

$$\Pr[d_{\deg}(ts_1, ts_2) \leq u - d] = \Pr[d_{\deg}(0, t(s_1 + s_2)) \leq u - d] = \Pr[d_{\deg}(0, x^g \cdot ts') \leq u - d].$$

When $g \geq u - d$, this event is never satisfied. When $g < u - d$, the event is satisfied iff the coefficients of $x^{u-d-g}, \ldots, x^{u+\ell-2-g}$ in the polynomial ts' are zero. As $ts' \sim R^\times_{u+\ell-1}$, this happens with probability $2^{-d-\ell+1}$. Therefore, by linearity of expectation, we have

$$\mathbb{E}_t[N_{u-d}(tS)] \leq n(n-1)2^{-d-\ell+1} \leq 2n/2^d.$$

We now show the lower bound. Let $R \coloneqq \{r \in R^\times_{u+\ell-1} : M(S, H_r) \geq M\}$ be the set of bad randomness. This means for any $p \in t^{-1}R$, $M(tS, H_p) \geq M$. By Lemma 4.5, this implies that $p \cdot (tS)$ intersects some radius-u ball in at least M points. We want to use the primes in the set $t^{-1}R$ to show that many close pairs exist (via Lemma 4.7). Let $P = \{p \in \mathbb{F}_2[x]^{=d} | p \text{ is prime}\}$. For each $p \in t^{-1}R \cap P$, we know M points of $p \cdot tS$ are contained in some ball of radius u. Because $M \leq n$, we have $d \leq \log_2(M/4) \leq \ell - 2$, and so $u + d < u + \ell - 1$. Furthermore, $u \geq \ell > d$. We can

consequently invoke Lemma 4.7 to deduce $N_{u-d}(tS) \geq \frac{(M-2^d)^2}{2^{d+1}}|t^{-1}R \cap P|$. Since R is a set of density δ_M in $R^{\times}_{u+\ell-1}$, it follows $\mathbb{E}_t[|t^{-1}R \cap P|] = \delta_M|P|$, and so

$$\mathbb{E}_t[N_{u-d}(tS)] \geq \frac{(M-2^d)^2}{2^{d+1}}\mathbb{E}_t[|t^{-1}R \cap P|] = \frac{(M-2^d)^2}{2^{d+1}}\mathbb{E}_t[|t^{-1}R \cap P|] \geq \frac{\delta_M(M-2^d)^2}{4d},$$

where the last inequality follows by Proposition 2.3.

□

We are now finally ready to prove the expected max-load bound for H^3_r from Corollary 4.3, which is equivalent to Toeplitz hashing.

Theorem 4.10. *Let $u \geq 2\ell \geq 1$ be integers, $n \coloneqq 2^\ell$. Let $S \subseteq R_{u+\ell-1}$ be a set of 2^ℓ polynomials of degree $< u$. For $r \sim R_{u+\ell-1}$, define $H_r : R_{u+\ell-1} \to \mathbb{F}_2[x]^{<\ell}$ by the map*

$$H_r(p) = rp \operatorname{div} x^{u-1}.$$

We have $\mathbb{E}_r[M(S, H_r)] = O((n \log n)^{1/3})$.

Proof. Let $x^g \coloneqq \gcd(r, x^{u+\ell-1})$. Notice that $\Pr[g = d] = 2^{-(d+1)}$, and so $\Pr[g \geq \ell] \leq \sum_{d \geq \ell} 2^{-(d+1)} = 1/n$. Conditioning on $g = d$ for some $d < \ell$, we see that $r = x^d r'$ where $r' \sim R^{\times}_{u+\ell-d-1}$. We can then write

$$\begin{aligned} & rp \pmod{x^{u+\ell-1}} \operatorname{div} x^{u-1} \\ = {} & x^d r' p \pmod{x^{u+\ell-1}} \operatorname{div} x^{u-1} \\ = {} & r' p \pmod{x^{u+\ell-d-1}} \operatorname{div} x^{u-d-1}. \end{aligned}$$

Hence, if for each $0 \leq d < \ell$ and $r' \sim R^{\times}_{u+\ell-d-1}$ we defined $H^d_{r'} : R_{u+\ell-d-1} \to \mathbb{F}_2[x]^{<\ell}$ by the map $H^d_{r'}(p) \coloneqq r'p \operatorname{div} x^{u-d-1}$, we see that $(H_r(p)|g = d) \sim H^d_{r'}(p)$. Therefore,

if we let C be the implied constant from Theorem 4.8, we observe

$$\begin{aligned}
\mathbb{E}_{r\sim R_{u+\ell-1}}[M(S,H_r)] &= \mathbb{E}_{r\sim R_{u+\ell-1}}[M(S,H_r)|g\geq \ell]\cdot \Pr[g\geq \ell] \\
&\quad + \sum_{d=0}^{\ell-1} \mathbb{E}_{r\sim R_{u+\ell-1}}[M(S,H_r)|g=d]\Pr[g=d] \\
&\leq n\cdot \frac{1}{n} + \sum_{d=0}^{\ell-1} \mathbb{E}_{r'\sim R^{\times}_{u+\ell-d-1}}[M(S,H^d_{r'})]\cdot 2^{-(d+1)} \\
&\leq 1 + C(n\log n)^{1/3}\sum_{d=0}^{\ell-1} 2^{-(d+1)} \\
&\leq 1 + C(n\log n)^{1/3}
\end{aligned}$$

where the penultimate inequality follows from the fact $\mathbb{E}[M(S,H^d_{r'})]\leq C(n\log n)^{1/3}$ for each $0\leq d<\ell$ (note in this step we crucially used the fact that all polynomials in S have degree $<u$, and that when $d<\ell$, $u-d>u-\ell\geq\ell$ to invoke Theorem 4.8). □

The proof of the main theorem now easily follows.

Proof of Theorem 4.4. Theorem 4.10 along with the equivalence from Corollary 4.3 gives the result. □

Part II

Locality

Chapter 5: Preliminaries

This chapter collects the notation, code definitions, and auxiliary facts used throughout the chapters in Part II.

5.1 General Notation

Throughout Part II, log denotes the natural logarithm, and logarithms to base 2 are written explicitly as $\log_2$. Let $\mathbb{N}$ be the positive integers, let $[n] \coloneqq \{1, \dots, n\}$, and let $\binom{[n]}{t}$ denote the collection of subsets of $[n]$ of size exactly t. We write $\mathbb{F}$ for an arbitrary finite field, $\mathbb{F}_2$ for the field of two elements, and $\{0,1\} \coloneqq \{0,1\}$. Binary strings and vectors over $\mathbb{F}_2$ are identified in the standard way.

For a string $x \in \Sigma^n$ and an index set $I \subseteq [n]$, let $x|_I$ denote the restriction of x to the indices in I. For a set $X \subseteq \Sigma^n$, let $X|_I \coloneqq \{x|_I : x \in X\}$. If $0 \in \Sigma$, define

$$\operatorname{supp}(x) \coloneqq \{i \in [n] : x_i \neq 0\}, \qquad X_{\subseteq I} \coloneqq \{x \in X : \operatorname{supp}(x) \subseteq I\}.$$

For strings $x, y \in \Sigma^n$, their absolute and relative Hamming distances are

$$\Delta(x,y) \coloneqq |\{i \in [n] : x_i \neq y_i\}|, \qquad \operatorname{dist}(x,y) \coloneqq \frac{\Delta(x,y)}{n}.$$

For a nonempty set $Y \subseteq \Sigma^n$, define

$$\Delta(x,Y) \coloneqq \min_{y \in Y} \Delta(x,y), \qquad \operatorname{dist}(x,Y) \coloneqq \min_{y \in Y} \operatorname{dist}(x,y).$$

For $x, y \in \mathbb{F}^n$, write $\langle x, y \rangle \coloneqq \sum_{i=1}^{n} x_i y_i$. We say $f(n) \leq \operatorname{poly}(n)$ if a fixed polynomial p satisfies $f(n) \leq p(n)$ for all sufficiently large n, and define $f(n) \geq \operatorname{poly}(n)$ and $f(n) = \operatorname{poly}(n)$ analogously. The notation $\operatorname{polylog} n$ means $\operatorname{poly}(\log n)$. All implicit polynomials are fixed with respect to the parameters under discussion.

5.2 Codes

Definition 5.1 (Code)**.** A code over an alphabet Σ with dimension k, block length n, and distance δ is a subset $C \subseteq \Sigma^n$ of size $|\Sigma|^k$ such that $\operatorname{dist}(c, C \setminus \{c\}) \geq \delta$ for every $c \in C$. The ratio k/n is its rate. A family of codes is *asymptotically good* if its rate and distance are bounded below by positive constants. If $\Sigma = \{0, 1\}$, the code is binary.

We use $C : \Sigma^k \to \Sigma^n$ when emphasizing an encoding map and write $C \coloneqq \operatorname{im}(C)$ for its codebook. When no ambiguity can arise, we freely identify the map with its image, writing either $x \in C$ or $C \subseteq \Sigma^n$.

Remark 5.2. If $C \subseteq \Sigma^n$ has distance δ, $w \in \Sigma^n$, and $c \in C$ satisfies $\operatorname{dist}(w, c) < \delta/2$, then c is the unique closest codeword to w. Indeed, for every $c' \in C \setminus \{c\}$,

$$\operatorname{dist}(c', w) \geq \operatorname{dist}(c', c) - \operatorname{dist}(w, c) > \operatorname{dist}(c, w).$$

Definition 5.3 (Systematic code)**.** A code $C \subseteq \Sigma^n$ is systematic if it has a bijective encoding map $C : \Sigma^k \to C$ such that $C(m)|_{[k]} = m$ for every $m \in \Sigma^k$.

Definition 5.4 (Linear code)**.** A linear code over $\mathbb{F}$ is an injective $\mathbb{F}$-linear map $C : \mathbb{F}^k \to \mathbb{F}^n$, or equivalently its k-dimensional image $C = \operatorname{im}(C) \subseteq \mathbb{F}^n$.

There is a generator matrix $A \in \mathbb{F}^{n \times k}$ such that $C(b) = Ab$ for every $b \in \mathbb{F}^k$, and a parity-check matrix $B \in \mathbb{F}^{(n-k) \times n}$ such that $x \in C$ if and only if $Bx = 0^{n-k}$. A family of linear codes is explicit if a uniform efficient algorithm computes generator matrices or parity-check matrices for the family.

Every linear code is systematic up to a permutation of its coordinates: apply Gaussian elimination to its generator matrix and permute columns until the first k columns form the identity matrix.

Definition 5.5 (Dual code)**.** For a linear code $C \subseteq \mathbb{F}^n$, its dual code is

$$C^{\perp} \coloneqq \{y \in \mathbb{F}^n : \langle x, y \rangle = 0 \text{ for every } x \in C\}.$$

The following fact relates the dual of a restricted code to the dual of the original code.

Fact 5.6. *Let $C \subseteq \mathbb{F}^n$ be a linear code and let $S \subseteq [n]$. Then*

$$(C|_S)^\perp = C^\perp_{\subseteq S},$$

where vectors in $(C|_S)^\perp$ are identified with vectors in $\mathbb{F}^n$ supported on S.

Proof. A vector $z \in (C|_S)^\perp \subseteq \mathbb{F}^S$ satisfies

$$0 = \langle x|_S, z\rangle = \sum_{i \in S} x_i z_i \qquad \text{for every } x \in C.$$

Extending z by zero outside S gives a vector $y \in C^\perp_{\subseteq S}$, and every vector in $C^\perp_{\subseteq S}$ restricts to such a z. □

Fact 5.7. *Let $C : \mathbb{F}^k \to \mathbb{F}^n$ be a linear code. For $j \in [n]$, let v_j be the jth row of its generator matrix, so $C(b)_j = \langle v_j, b\rangle$. Let $Q \subseteq [n]$, let $x^* = C(b^*)$, and let $v \in \mathbb{F}^k$. Every $x = C(b)$ satisfying $x|_Q = x^*|_Q$ also satisfies $\langle v, b\rangle = \langle v, b^*\rangle$ if and only if*

$$v \in \operatorname{span}(\{v_j\}_{j \in Q}).$$

If v is not in this span, then for every $\sigma \in \mathbb{F}$ there is a message $b \in \mathbb{F}^k$ such that $C(b)|_Q = x^*|_Q$ and $\langle v, b\rangle = \sigma$.

5.3 Local Decoding and Correction

Locally decodable codes and locally correctable codes recover a symbol of the message or closest codeword, respectively, using few queries to a noisy word. Their study has roots in program checking [BK95, Lip90, BFLS91, RS96], and they were first formalized by Katz and Trevisan [KT00]; see Yekhanin's survey [Yek12].

Definition 5.8 (Locally decodable code)**.** A code $C : \Sigma^k \to \Sigma^n$ is a (q, δ, c, s)-LDC if there is a randomized decoder Dec such that:

(1) for every oracle word y and index $i \in [k]$, $\mathrm{Dec}^y(i)$ queries at most q coordinates of y;

(2) for every $b \in \Sigma^k$ and $i \in [k]$, $\Pr[\mathrm{Dec}^{C(b)}(i) = b_i] \geq c$; and

(3) for every $b \in \Sigma^k$, $i \in [k]$, and $y \in \Sigma^n$ with $\Delta(y, C(b)) \leq \delta n$,

$$\Pr[\mathrm{Dec}^y(i) \neq b_i] \leq s.$$

Definition 5.9 (Locally correctable code)**.** A code $C : \Sigma^k \to \Sigma^n$ is a (q, δ, c, s)-LCC if there is a randomized corrector Dec such that:

(1) for every oracle word y and index $u \in [n]$, $\mathrm{Dec}^y(u)$ queries at most q coordinates of y;

(2) for every $b \in \Sigma^k$ and $u \in [n]$, $\Pr[\mathrm{Dec}^{C(b)}(u) = C(b)_u] \geq c$; and

(3) for every $b \in \Sigma^k$, $u \in [n]$, and $y \in \Sigma^n$ with $\Delta(y, C(b)) \leq \delta n$,

$$\Pr[\mathrm{Dec}^y(u) \neq C(b)_u] \leq s.$$

We next define smooth decoders and correctors, which need only operate on valid codewords but may not query any coordinate too frequently.

Definition 5.10 (Smooth decoder)**.** An η-smooth q-query decoder for $C : \Sigma^k \to \Sigma^n$ is a randomized decoder Dec with perfect completeness such that, for every $i \in [k]$ and $j \in [n]$,

$$\Pr[\mathrm{Dec}(i) \text{ queries } j] \leq \frac{1}{\eta n}.$$

Definition 5.11 (Smooth corrector)**.** An η-smooth q-query corrector for a code $C \subseteq \Sigma^n$ is a randomized corrector Dec with perfect completeness such that, for every $u, j \in [n]$,

$$\Pr[\mathrm{Dec}(u) \text{ queries } j] \leq \frac{1}{\eta n}.$$

Fact 5.12 (Smooth decoding implies local decoding)**.** *If a code has a q-query η-smooth decoder or corrector, then it is a $(q, \eta\varepsilon, 1, \varepsilon)$-LDC or LCC, respectively.*

Proof. We prove the decoder case; the corrector case is analogous. Let y satisfy $\Delta(y, C(b)) \leq \eta\varepsilon n$ and let S be the set of corrupted coordinates. Smoothness and a union bound give

$$\Pr[\mathrm{Dec}(i) \text{ queries } S] \leq \frac{|S|}{\eta n} \leq \varepsilon.$$

Whenever no coordinate of S is queried, the decoder sees a view consistent with $C(b)$ and outputs b_i by perfect completeness. □

5.4 Relaxed Local Decoding and Correction

Relaxed locally decodable and correctable codes may detect corruption and output $\perp$ instead of returning an incorrect symbol. These notions are closely related to probabilistically checkable proofs and originate with Ben-Sasson, Goldreich, Harsha, Sudan, and Vadhan [BSGH$^+$06]; relaxed local correction was introduced by Gur, Ramnarayan, and Rothblum [GRR20].

Definition 5.13 (Relaxed locally decodable code)**.** A code $C : \Sigma^k \to \Sigma^n$ is a (q, δ, c, s)-RLDC if there is a randomized decoder Dec whose output lies in $\Sigma \cup \{\perp\}$ such that:

(1) for every y and $i \in [k]$, $\mathrm{Dec}^y(i)$ queries at most q coordinates;

(2) for every $b \in \Sigma^k$ and $i \in [k]$, $\Pr[\mathrm{Dec}^{C(b)}(i) = b_i] \geq c$; and

(3) for every $b \in \Sigma^k$, $i \in [k]$, and $y \in \Sigma^n$ with $\Delta(y, C(b)) \leq \delta n$,

$$\Pr[\mathrm{Dec}^y(i) \notin \{b_i, \perp\}] \leq s.$$

Definition 5.14 (Relaxed locally correctable code)**.** A code $C : \Sigma^k \to \Sigma^n$ is a (q, δ, c, s)-RLCC if there is a randomized corrector Dec whose output lies in $\Sigma \cup \{\perp\}$ such that:

(1) for every y and $u \in [n]$, $\mathrm{Dec}^y(u)$ queries at most q coordinates;

(2) for every $b \in \Sigma^k$ and $u \in [n]$, $\Pr[\mathrm{Dec}^{C(b)}(u) = C(b)_u] \geq c$; and

(3) for every $b \in \Sigma^k$, $u \in [n]$, and $y \in \Sigma^n$ with $\Delta(y, C(b)) \leq \delta n$,

$$\Pr[\mathrm{Dec}^y(u) \notin \{C(b)_u, \perp\}] \leq s.$$

When only a radius δ and query complexity q are specified for an RLDC or RLCC, we use the shorthand $(q, \delta, 1, 1/3)$: perfect completeness and soundness success probability at least $2/3$. Goldberg [Gol23a] shows that, for linear RLDCs and RLCCs, this strong definition is essentially equivalent to allowing imperfect completeness and imposing nonadaptivity. All relaxed correctors constructed in Part II are nonadaptive.

An RLCC implies an RLDC with the same radius and query complexity when it is systematic, by correcting the coordinates containing the message. A general RLCC can be made systematic with only a constant-factor loss in rate; see Appendix B of [KM24c]. Moreover, an RLCC with correcting radius δ must have distance at least δ. Thus, an RLCC is asymptotically good when it has constant rate and constant correcting radius.

For perfect-completeness RLDCs and RLCCs, we may assume a canonical decoding rule.

Fact 5.15 (Canonical behavior of a local decoder)**.** *Let $C : \{0,1\}^k \to \{0,1\}^n$ be a $(q, \delta, 1, s)$-RLDC with decoder Dec_1. There is a decoder Dec_2 with the same parameters such that, whenever it queries a set Q and observes $z \in \{0,1\}^Q$, it behaves as follows:*

(1) find $x = C(b)$ such that $x|_Q = z$, and output $\perp$ if none exists;

(2) otherwise, output b_i.

If Dec_1 is nonadaptive, then so is Dec_2. The analogous statement holds for RLCCs.

Proof. Run Dec_1 until all its queries have been made, then apply the stated rule. Perfect completeness guarantees that the local view determines the requested symbol whenever it is consistent with a codeword. Relative to Dec_1, the new decoder can only replace an output symbol by $\perp$, so its relaxed soundness error cannot increase. □

For linear codes, the canonical rule has a useful linear-algebraic form.

Fact 5.16. *Let $C : \mathbb{F}^k \to \mathbb{F}^n$ be a linear $(q, \delta, 1, s)$-RLDC with canonical decoder* Dec. *Suppose* Dec *queries $Q \subseteq [n]$ and observes $z \in \mathbb{F}^Q$. Then it:*

(1) checks whether $z \in C|_Q$, outputting $\perp$ otherwise; and

(2) chooses coefficients $\{\alpha_j\}_{j\in Q} \subseteq \mathbb{F}$ such that $\sum_{j\in Q} \alpha_j x_j = b_i$ for every $x = C(b)$ and outputs $\sum_{j\in Q} \alpha_j z_j$.

Membership in Item (1) can be checked by a homogeneous system $Mz = 0$ with at most $|Q|$ constraints. We call these the testing constraints and call the relation in Item (2) the decoding constraint.

Proof. Item (1) is the canonical consistency check. For Item (2), let $v_j \in \mathbb{F}^k$ be the jth row of the generator matrix. Recovering b_i from $\{x_j\}_{j\in Q}$ is equivalent to $e_i \in \mathrm{span}(\{v_j\}_{j\in Q})$, which must hold by perfect completeness. Finally, $C|_Q$ is a linear subspace, so membership is characterized by a system of homogeneous linear equations. □

5.5 Locally Testable Codes

Locally testable codes have testing algorithms that estimate corruption while making few queries.

Definition 5.17 (Locally testable code)**.** A code $C \subseteq \Sigma^n$ is a locally testable code with testability κ and query complexity q if it has a randomized tester T that makes at most q queries and outputs either $\top$ or $\perp$ such that:

1. for every $c \in C$, $\Pr[T^c = \top] = 1$; and

2. for every $w \in \Sigma^n$,
$$\Pr[T^w = \perp] \geq \kappa \operatorname{dist}(w, C).$$

We refer to T as a C-tester.

Dinur, Evra, Livne, Lubotzky, and Mozes [DEL+22] and Panteleev and Kalachev [PK22] constructed the first locally testable codes with constant rate, distance, and query complexity. The former construction gives explicit linear LTCs with rate arbitrarily close to 1.

Theorem 5.18 ([DEL+22, Theorem 1.1 and Remark 5.3])**.** *For every $R = 1 - \varepsilon \in (0, 1)$, there are a prime power $p = \Theta((1/\varepsilon)^{10})$ and values*
$$\delta \geq \Omega(\varepsilon^3), \qquad \kappa \geq \Omega(\varepsilon^{15}), \qquad q \leq O((1/\varepsilon)^{20})$$
such that, for every $j \in \mathbb{N}$, there is a binary linear LTC LTC_j with rate at least R, minimum distance at least δ, testability at least κ, query complexity at most q, and block length
$$n_j \coloneqq \frac{q}{8}\big(p^{3j} - p^j\big).$$

We will use the following concrete instantiation. Setting $\varepsilon = \Theta(1/\log N)$ in the construction above yields a family whose parameters are polynomial or inverse polynomial in $\log N$ and whose consecutive block lengths differ by a polynomial factor in $\log N$.

Theorem 5.19 ([DEL+22, Theorem 1.1, Lemma 5.1, and Remark 5.3])**.** *For every sufficiently large $N \in \mathbb{N}$, there is an explicit odd prime power $p = \Theta(\log^{10} N)$ and an infinite family of explicit binary linear locally testable codes $\{\mathsf{LTC}_1, \mathsf{LTC}_2, \dots\}$ such that every LTC_j has:*

- *block length $n_j = \Theta((p^{3j} - p^j) \log^{20} N)$;*

- *rate* $1 - 1/(100 \log N)$*;*
- *distance* $\Omega(1/\log^3 N)$*;*
- *testability* $\Omega(1/\log^{15} N)$*; and*
- *query complexity* $O(\log^{20} N)$*.*

Corollary (Corollary 6.8 restated)**.** *For every sufficiently large* $N \in \mathbb{N}$*, there are an integer*

$$n \in [\Omega(N/\log^{30} N), N]$$

and a family of LTCs $\{\mathsf{LTC}_1, \ldots, \mathsf{LTC}_m\}$ *such that:*

- *every* LTC_j *is binary, linear, and explicit;*
- *every* LTC_j *has rate* $R = 1 - \varepsilon_{\mathsf{LTC}} \geq 1 - O(1/\log N)$*, distance* $\delta_{\mathsf{LTC}} \geq \Omega(1/\log^3 N)$*, testability* $\kappa_{\mathsf{LTC}} \geq \Omega(1/\log^{15} N)$*, and query complexity* $q_{\mathsf{LTC}} \leq O(\log^{20} N)$*;*
- *the block lengths satisfy* $n_1 \leq O(\log^{50} N)$*,* $n_m = n$*, and* $n_{j+1}/n_j = \Theta(\log^{30} N)$*; and*
- $m = O(\log N/\log\log N)$*.*

Proof. Let p and $\{\mathsf{LTC}_1, \mathsf{LTC}_2, \ldots\}$ be given by Theorem 5.19, let m be the largest integer such that $n_m \leq N$, and set $n \coloneqq n_m$. The rate, distance, testability, and query bounds follow immediately, and

$$n_1 \leq O(p^3 \log^{20} N) = O(\log^{50} N).$$

For every $j \geq 1$,

$$\frac{n_{j+1}}{n_j} = \Theta\left(\frac{p^{3(j+1)} - p^{j+1}}{p^{3j} - p^j}\right) = \Theta(p^3) = \Theta(\log^{30} N).$$

Thus $n \le N \le O(p^3 n)$, so $n \in [\Omega(N/\log^{30} N), N]$. Finally,

$$N \ge n_m \ge \prod_{j=1}^{m-1} \frac{n_{j+1}}{n_j} \ge \Omega(\log^{30} N)^{m-1},$$

which gives $m = O(\log N / \log\log N)$. □

5.6 Soundness Amplification

We record the elementary amplification observation used later in Part II.

Observation 5.20 (Observation 7.11)**.** Let $C : \{0,1\}^k \to \{0,1\}^n$ be a $(q, \delta, 1, \varepsilon)$-RLDC with a possibly adaptive decoder. For every integer $t \ge 1$, C is a $(qt, \delta, 1, \varepsilon^t)$-RLDC. The same statement holds for RLCCs.

Proof. Run the original decoder independently t times. Output $\sigma \in \{0,1\}$ if every invocation outputs σ, and output $\perp$ otherwise. Perfect completeness is preserved, and an incorrect symbol is returned only if all t independent invocations return that symbol, which occurs with probability at most ε^t. □

5.7 Linearity Testing

We recall the standard linearity test over $\mathbb{F}_2$.

Fact 5.21 (Linearity Test [BLR93, BCH+95])**.** *Let $G : \mathbb{F}_2^n \to \mathbb{F}_2$ be arbitrary, and let* $\mathrm{dist}(G, \mathrm{LIN})$ *be its minimum relative Hamming distance from a linear function $F : \mathbb{F}_2^n \to \mathbb{F}_2$. Then*

$$\Pr_{x,y \sim \mathbb{F}_2^n}[G(x) + G(y) + G(x+y) = 0] \le 1 - \mathrm{dist}(G, \mathrm{LIN}).$$

Thus, if the test passes with probability at least $1 - \varepsilon$, then $\mathrm{dist}(G, \mathrm{LIN}) \le \varepsilon$.

We will also use self-correction of functions close to linear.

Fact 5.22 (Self-correction of near-linear functions)**.** *Let $G : \mathbb{F}_2^n \to \mathbb{F}_2$ be arbitrary and let $F : \mathbb{F}_2^n \to \mathbb{F}_2$ be linear. For every $x \in \mathbb{F}_2^n$,*

$$\Pr_{y \sim \mathbb{F}_2^n}[G(x+y) + G(y) = F(x)] \geq 1 - 2\,\mathrm{dist}(F, G).$$

Chapter 6: Relaxed Local Correctability from Local Testing

6.1 Introduction

Locally correctable codes (LCCs) and locally decodable codes (LDCs) are error correcting codes that allow any bit of the original codeword or message, respectively, to be recovered using very few queries to a noisy codeword with bounded corruption. This is a natural and useful property, but unfortunately little is known about the best possible parameter tradeoffs between the rate and query complexity. In the asymptotically good (constant rate and distance) regime, Katz and Trevisan [KT00] show that any LDC (and any linear LCC) with block length n must make $\tilde{\Omega}(\log n)$ queries. However, the most query-efficient constant-rate LCCs and LDCs, constructed by Kopparty, Meir, Ron-Zewi, and Saraf [KMRS17], require $2^{\tilde{O}(\sqrt{\log n})}$. Whether the true optimal query complexity is polylogarithmic or not is a longstanding open problem. A Reed–Muller code with appropriate parameters brings us tantalizingly close: such a code is locally correctable with polylogarithmic query complexity, but with slowly vanishing rate $1/n^{\varepsilon}$ (e.g. [Yek12, Section 2.3]).

Ben-Sasson, Goldreich, Harsha, Sudan, and Vadhan [BSGH$^+$06] and Gur, Ramnarayan, and Rothblum [GRR20] introduced the notions of *relaxed* locally decodable codes (RLDCs) and *relaxed* locally correctable codes (RLCCs), respectively. These codes admit local decoders or correctors that, upon receiving a word within some radius of the code, will either decode/correct, or detect corruption in the input by returning a rejection symbol $\perp$. This definition is motivated by the study of probabilistically checkable proofs (PCPs). A major open question is to construct a PCP with constant query complexity and linear proof length. PCPs naturally need to recover information from only a few queries to the proof, and for this reason, many known PCP constructions use LDCs or LCCs under the hood. However, an LDC decoder must always successfully decode, while a PCP verifier can reject if it

notices any errors while trying to decode from a purported proof. An RLDC is precisely a weakened LDC with this ability to reject, and so one can imagine RLDCs or RLCCs potentially giving improved PCPs (although relaxed local decodability or correctability alone does not suffice; see Remark 6.14).

In contrast with LDCs and LCCs, we know much more about what the optimal RLDC and RLCC parameters should be. For asymptotically good RLDCs and RLCCs[1] in particular, the gap between lower and upper bounds is smaller but still significant: the best lower bound is $\tilde{\Omega}(\sqrt{\log n})$ due to Gur and Lachish [GL21], while the best upper bound, due to Cohen and Yankovitz [CY22], is $(\log n)^{O(\log\log\log n)}$. In this chapter, we improve the upper bound by constructing asymptotically good RLCCs with polylogarithmic query complexity.

Theorem (informal, see Corollary 6.12)**.** *For infinitely many positive n and any constant $R \in (0,1)$, there exist explicit binary linear RLCCs (and thus RLDCs) of block length n, rate R, constant correcting (or decoding) radius, and query complexity $O(\log^{69} n)$.*

We make no effort to optimize the exponent, instead striving for a simpler exposition. The related and well-studied notion of locally testable codes (LTCs), where errors can be detected with few queries, proves to be key: we can build a relaxed locally correctable code from any family of high-rate linear locally testable codes. Then, by leveraging the LTCs of Dinur, Evra, Livne, Lubotzky, and Mozes [DEL$^+$22], we construct explicit RLCCs with constant rate, constant distance, and polylogarithmic query complexity. We also get nonexplicit RLCCs with polylogarithmic query complexity that approach the Gilbert–Varshamov bound, which is the best known general tradeoff between rate and distance for which binary codes exist. The last known RLCCs to approach the Gilbert–Varshamov bound are LCCs by Gopi, Kopparty,

[1]The lower bounds are for RLDCs, and RLCCs can be transformed into (and are "stronger" than) RLDCs; see Appendix B of [KM24c]

Oliveira, Ron-Zewi, and Saraf [GKO+18] which require polynomially many (i.e., n^{ε}) queries. They construct both LTCs and LCCs approaching the Gilbert–Varshamov bound, and their former result works for any high-rate LTC, yielding codes that we utilize in Corollary 6.13.[2]

6.1.1 Techniques

To construct an asymptotically good RLCC with an arbitrarily large block length n, we start with a code with extremely small block length (say, polylogarithmic in n). Such a code is trivially an RLCC with polylogarithmic query complexity, because it has a corrector which reads all $\operatorname{polylog} n$ symbols of the input. Then, we apply a transformation that uses this small RLCC to build an RLCC with slightly larger block length. We can repeat this transformation, building RLCCs with successively larger block length, until we reach our target block length n. At each of these steps, we need to maintain the distance and correcting radius of our code while minimizing the gain in query complexity and loss in rate, so that the final code will be asymptotically good. Indeed, such an approach has been used to construct asymptotically good LTCs and LCCs [Mei08, KMRS17], and is the basis for both prior asymptotically good RLCC constructions [GRR20, CY22].

6.1.1.1 Prior constructions

Gur, Ramnarayan, and Rothblum [GRR20] use the tensor product in this framework to construct RLCCs. Starting from a RLCC with tiny block length, they build successively larger RLCCs by taking the tensor product of the code with itself. If $C \subseteq \mathbb{F}^n$ is a linear code, then the tensor product code $C \otimes C \subseteq \mathbb{F}^{n \times n}$ can be viewed as the set of matrices where every row and every column is a codeword of C. They show that if C is an RLCC, $C \otimes C$ is also an RLCC because it has a local corrector that

[2] In contrast, their LCC construction uses additional properties satisfied by Reed–Solomon and Reed–Muller codes, so their transformation would not immediately improve the rate-distance tradeoff of arbitrary high-rate LCCs or RLCCs.

calls the corrector for C as a subroutine on relevant overlapping rows and columns of the input; the corrector for $C \otimes C$ recurses on the smaller code's corrector polylog n times. The tensor product step squares the block length each time, so $\Theta(\log \log n)$ iterations are required to construct an RLCC with block length n. Each tensor product also squares the distance and correcting radius of the code, so a distance amplification step is needed to maintain the initial distance. Each iteration of tensor product and distance amplification adds a polylog n factor to the query complexity, so after $O(\log \log n)$ iterations the total queries required is $(\log n)^{O(\log \log n)}$.

Each tensor product step incurs a polylog n factor in the query complexity because of its rate deterioration. If C has rate $R = 1 - \varepsilon$, then $C \otimes C$ has rate $R^2 \approx 1 - 2\varepsilon$; the loss in rate has doubled after one iteration, so this loss will grow exponentially in the number of iterations. Therefore, for the final RLCC to have constant rate after $\Theta(\log \log n)$ tensoring steps, the initial RLCC must have high rate, at least $1 - 1/\log n$, which limits the distance of the initial RLCC. This initial distance causes the polylog n multiplicative overhead in query complexity, because at each iteration, the number of recursive calls to the previous code's corrector is polynomial in the reciprocal of its distance.[3] To address this, Cohen and Yankovitz [CY22] give an alternative to tensoring called *row-evasive squaring*, which incurs an *additive* loss in rate rather than a multiplicative one. The starting code can now have lower rate, like $1 - 1/\log \log n$, and thus higher distance, which improves the query overhead per iteration (which is still a multiplicative factor) to poly$(\log \log n)$. This yields a total query complexity of $(\log \log n)^{O(\log \log n)} = (\log n)^{O(\log \log \log n)}$.

6.1.1.2 Our construction

To build asymptotically good RLCCs with polylog n query complexity, we develop a new operation for boosting the block length of an RLCC. In contrast to

[3] While the distance of this intermediate code could be amplified to a constant in order to reduce the number of recursive calls needed in the current iteration, over many iterations this additional distance amplification would bring the rate to zero.

tensoring and row-evasive squaring, our operation augments the block length of an RLCC by a very modest polylog n factor, but incurs only an *additive* cost in *both* rate and query complexity. In addition, our operation preserves the distance and correcting radius, so no extra distance amplification step is needed. Each step of our operation solely requires polylog n additional queries, so although we need more iterations (up to $m = O(\log n)$ times to reach block length n) than the tensoring or row-evasive squaring approaches, our final query complexity will be $m \cdot \text{polylog}\, n = \text{polylog}\, n$.

Intuitively, tensoring and row-evasive squaring are operations that work by intertwining the structure of many copies of a smaller RLCC, so that the local corrector of the smaller RLCC can be used to cross-check overlapping portions of the input against each other. This necessitates multiple recursive calls to a smaller code's corrector at each step of the construction, which causes the final query complexity to grow exponentially in the number of iterations. Instead of cross-checking smaller RLCCs, our boosting operation enlists the outside help of a locally testable code to add structure. We will use the LTC's self-contained testing algorithm to ensure that it is safe to recurse on the smaller RLCC's corrector exactly once. This "tail recursion" is why our total query complexity grows linearly in the number of iterations, rather than exponentially.

Our boosting step is called *nesting*. Say we have an RLCC $C \subseteq \Sigma^n$ with correcting radius δ and a locally testable code $\mathsf{LTC} \subseteq \Sigma^N$ with distance 2δ where (for simplicity here) n divides N. Then, the code formed by *nesting* C *in* LTC is defined to be the intersection of LTC with the N/n-fold Cartesian product of C with itself:

$$\mathsf{LTC} \Cap C \coloneqq \mathsf{LTC} \cap C^{N/n}.$$

In other words, each codeword in $\mathsf{LTC} \Cap C$ is the concatenation of N/n codewords $c_1, \cdots, c_{N/n} \in C$ such that $(c_1, \ldots, c_{N/n}) \in \mathsf{LTC}$. We claim that this code is an RLCC with correcting radius δ. That is, this code has a local algorithm that, given any input w that satisfies $\text{dist}(w, \mathsf{LTC} \Cap C) \leq \delta$, either corrects or detects corruption.

To see why, suppose $w \in \Sigma^N$ is very close to the nearest codeword $c \in \mathsf{LTC} \Cap C$, i.e., $\mathrm{dist}(w, \mathsf{LTC} \Cap C) \leq \delta n/N$. Then we can hope to correct any index of w by resorting to the local corrector of C. To correct w_i, we consider the unique interval $I \coloneqq \{kn+1, \ldots, kn+n\}$ containing i. We know by construction that $c|_I$ is a codeword of C. Furthermore, we know that w differs from c in at most δn indices, so $\mathrm{dist}(w|_I, c|_I) \leq \delta$ which is within the correcting radius of C. Hence, we can "zoom-in" and use the corrector for C to correct any symbol of $w|_I$, including w_i.

Otherwise, if $\delta n/N < \mathrm{dist}(w, \mathsf{LTC} \Cap C) \leq \delta$, we can hope to detect (rather than correct) corruption using the local tester of LTC. A locally testable code, by definition, has a local testing algorithm T that rejects its input w with probability proportional to $\mathrm{dist}(w, \mathsf{LTC})$.[4] Because $\mathsf{LTC} \Cap C \subseteq \mathsf{LTC}$ and LTC has distance 2δ, we know that $\mathrm{dist}(w, \mathsf{LTC}) = \mathrm{dist}(w, \mathsf{LTC} \Cap C) > \delta n/N$, and so T will reject w with probability $\Omega(\delta n/N)$. Thus, $O(N/\delta n)$ repetitions (hiding some minor factors) of the local tester suffice to detect corruption with constant probability.

Therefore, by combining the corrector for C and tester for LTC, we can handle both cases. Run both the corrector and the tester; if the tester finds corruption, we can return the reject symbol $\perp$, and otherwise we are likely in the small distance case and can return the output of the corrector. This shows that $\mathsf{LTC} \Cap C$ is an RLCC which inherits the larger block length of LTC while requiring only roughly $O(N/\delta n)$ more queries than C.

Furthermore, we can show that nesting does not destroy the rate. If LTC and C are linear, and if LTC has rate $1 - \varepsilon_{\mathsf{LTC}}$ while C has rate $1 - \varepsilon$, then by counting the number of linear constraints, $\mathsf{LTC} \Cap C$ has rate at least $1 - \varepsilon - \varepsilon_{\mathsf{LTC}}$. If we hope to repeatedly apply nesting with larger and larger LTCs (block length growing by a small factor) to build an RLCC with arbitrarily large block length n, then we will

[4] Here, for simplicity, we pretend that the proportionality between the rejection probability and distance is constant. In reality, it will be closely related to the other parameters of the LTC, and contributes a polylogarithmic factor to the query complexity of our construction.

need to apply nesting $m = O(\log n)$ times. Assuming that all of the LTCs have rate $1 - \varepsilon_{\mathsf{LTC}}$, the RLCC will have rate of $1 - O(\log n) \cdot \varepsilon_{\mathsf{LTC}}$ which we need to be $\Omega(1)$ in order to be asymptotically good. This forces us to use LTCs with rate at least $1 - O(1/\log n)$.

Fortunately, we can use the construction of Dinur, Evra, Livne, Lubotzky, and Mozes [DEL$^+$22] to get the high-rate linear LTCs we need. For any sufficiently large choice of n, there is an infinite sequence of explicit LTCs with rate at least $1 - O(1/\log n)$ and with appropriate local testing parameters so that the additive query overhead of each nesting step is polylog n as desired. By iteratively nesting these LTCs, we can build an RLCC of block length n with constant rate and polylogarithmic query complexity.

One may notice that since the LTCs have rate $1 - o(1)$, they must also have $o(1)$ distance, because of the Singleton bound which states that the sum of a code's rate and distance cannot exceed 1. The correcting radius of our RLCC thus will also be $o(1)$, and so our code is not yet asymptotically good. This is remedied by nesting in one last LTC with constant rate and distance. This boosts the correcting radius to be constant, and the rate of this last LTC dominates the rate of the final RLCC, which at last is asymptotically good.

6.1.2 Related work

6.1.2.1 Prior constructions and lower bounds

The two main parameter regimes for RLDCs and RLCCs are the constant query regime (optimizing block length given the dimension k and query complexity q) and the asymptotically good regime (optimizing query complexity when $k/n = \Theta(1)$ and $\delta = \Theta(1)$). In the constant-query regime, the best known block length is $n = O(k^{1+O(1/q)})$ [AS21], following a line of work [BSGH$^+$06, GRR20, CGS22]. Interestingly, this matches (up to a constant factor in q) the block length lower bound for full-fledged LDCs [KT00, Woo07].

Table 6.1 summarizes the historic state of the art query complexity for asymptotically good RLCCs. This table does not include Gopi, Kopparty, Oliveira, Ron-Zewi, and Saraf [GKO+18] who optimize rate (approaching the Gilbert–Varshamov bound) instead of queries.

Gur and Lachish [GL21] establish lower bounds for RLDCs, including a $\tilde{\Omega}(\sqrt{\log n})$ query lower bound for asymptotically good RLDCs with nonadaptivity. Dall'Agnol, Gur, and Lachish [DGL21] extend the lower bound to adaptive decoders, and Goldreich [Gol23b] provides an alternative and simpler proof, which in some cases is stronger. Block, Blocki, Cheng, Grigorescu, Li, Zheng, and Zhu [BBC+23] prove an exponential block length lower bound for 2-query RLDCs, asymptotically matching the exponential block length lower bound for 2-query LDCs established by Kerenidis and de Wolf [KW03].

6.1.2.2 Alternative error models

LDCs, LCCs, and their relaxed counterparts have been studied in other error models, distinct from the Hamming worst-case error setting studied in this chapter. These codes have been studied in the insertion-deletion error model, where a limited number of symbols can be added or removed (rather than simply flipped) anywhere in the codeword [OP15, BBG+20, CLZ20, BBC+23]. In addition, both the Hamming

Table 6.1: Best known query complexity for good RLCCs with block length n.

Technique	Query complexity	Due to
low-degree polynomials	n^{ε}	[BFLS91, RS96]
multiplicity codes	n^{ε}	[KSY14]
lifted Reed–Solomon codes	n^{ε}	[GKS13]
expander graphs	n^{ε}	[HOW15]
distance amplification	$2^{\tilde{O}(\sqrt{\log n})}$	[KMRS17]
repeated tensoring	$(\log n)^{O(\log \log n)}$	[GRR20]
row-evasive squaring	$(\log n)^{O(\log \log \log n)}$	[CY22]
nested LTCs	$\log^{O(1)} n$	this work

and insertion-deletion models have been studied in the computationally bounded setting, where the adversary choosing the location of bit flips or insertions/deletions has limited resources. Then, cryptographic assumptions can be used to construct LDCs and LCCs [OPS07, HO08, HOSW11, BKZ20, BB21, ABB22] as well as their relaxed counterparts [BGGZ21, BB23].

In particular, the latter two works use these assumptions to (among other results) construct asymptotically good RLDCs and RLCCs in the computationally bounded Hamming model with polylogarithmic queries. We achieve this in the general setting.

6.1.2.3 Subsequent work

Soon after we posted the preprint for this work, Cohen and Yankovitz [CY23] improved our technique and lowered the query complexity for asymptotically good RLCCs from our $\log^{69} n$ to $(\log n)^{2+o(1)}$. They observe that the nesting operation can boost an RLCC using a code that is locally testable only on inputs with bounded corruption, rather than a full-fledged locally testable code which must work on all inputs. Thus, they can use a family of expander codes, which satisfy this weaker property, to simplify the RLCC construction and improve the query complexity.

6.2 Construction

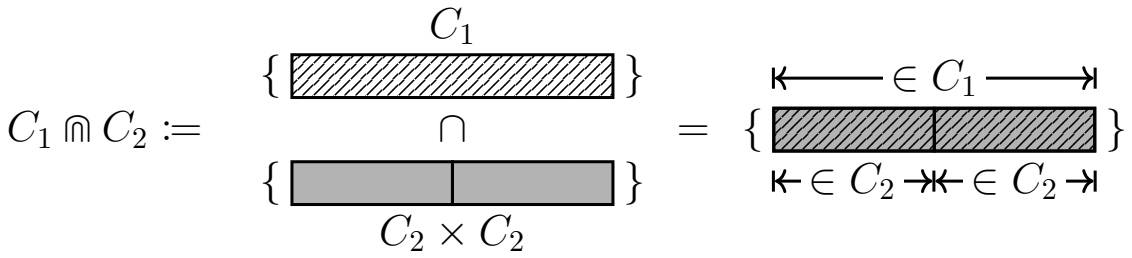
6.2.1 Boosting RLCC block length

The building block of our construction is the *nesting* operation $\Cap$ which combines two codes.

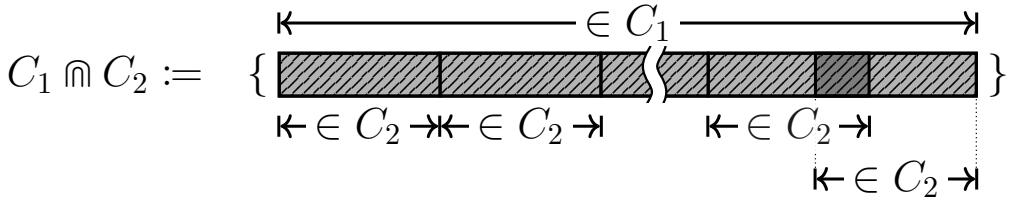
Definition 6.1. Let $C_1 \subseteq \Sigma^N$ and $C_2 \subseteq \Sigma^n$ with $n \leq N$. Define

$$C_1 \Cap C_2 \coloneqq C_1 \cap (C_2^{\lfloor N/n \rfloor} \times \Sigma^{N - \lfloor N/n \rfloor \cdot n}) \cap (\Sigma^{N-n} \times C_2).$$

We say that $C_1 \Cap C_2$ is the code formed by *nesting* C_2 *in* C_1.



(a) When n divides N, the nesting operation intersects C_1 with the Cartesian product $C_2^{N/n}$. Then, $C_1 \Cap C_2$ is a subset of C_1 and every $i \in [N]$ is in some length-n interval I such that restricting $C_1 \Cap C_2$ to I gives a subset of C_2. In this example with $N = 2n$, every $c \in C_1 \Cap C_2$ is a codeword of C_1 and the two halves of c are codewords of C_2.



(b) When n does not divide N, the nesting operation intersects C_1 with length-padded $C_2^{\lfloor N/n \rfloor}$ (so that intersection is well-defined) and then adds constraints so that the length-n suffix is in C_2. This maintains the same properties as above: $C_1 \Cap C_2 \subseteq C_1$ and every $i \in [N]$ is in some length-n interval I such that $(C_1 \Cap C_2)|_I \subseteq C_2$.

Figure 6.1: Two cases of nesting.

That is, $C_1 \Cap C_2$ is the subset of codewords c in C_1 such that $c|_{\{1,\dots,n\}} \in C_2$, $c|_{\{n+1,\dots,2n\}} \in C_2$, and so on. If n does not divide N, then we also require the last n symbols of c to form a codeword in C_2, i.e., $c|_{\{N-n+1,\dots,N\}} \in C_2$. Because $C_1 \Cap C_2$ is a subset of C_1, it must have distance at least that of C_1. In addition, when both codes are linear, we can bound the rate of $C_1 \Cap C_2$. Nesting C_2 in C_1 incurs an additive loss in rate:

Lemma 6.2. *Let $n \leq N$. Let $C_1 \subseteq \Sigma^N$ be a linear code with rate $1 - \varepsilon_1$, and let $C_2 \subseteq \Sigma^n$ be a linear code with rate $1 - \varepsilon_2$. Then, $C_1 \Cap C_2 \subseteq \Sigma^N$ is a linear code with rate at least*

$$1 - \varepsilon_1 - \left(\frac{n}{N} \cdot \left\lceil \frac{N}{n} \right\rceil\right)\varepsilon_2.$$

Proof. Using the rank-nullity theorem, the total number of linear constraints defining $C_1 \Cap C_2$ is

$$\leq \varepsilon_1 N + \varepsilon_2 n \cdot \left\lceil \frac{N}{n} \right\rceil = \left(\varepsilon_1 + \frac{n}{N} \cdot \left\lceil \frac{N}{n} \right\rceil \cdot \varepsilon_2\right) \cdot N$$

which then implies that the rate of $C_1 \Cap C_2$ is

$$\geq 1 - \varepsilon_1 - \left(\frac{n}{N} \cdot \left\lceil \frac{N}{n} \right\rceil\right)\varepsilon_2. \qquad \square$$

Remark 6.3. If n divides N, then $C_1 \Cap C_2 = C_1 \cap C_2^{N/n}$, with rate at least $1 - \varepsilon_1 - \varepsilon_2$. When n does not divide N, the rate of $C_1 \Cap C_2$ is at least $1 - \varepsilon_1 - (1 + n/N)\varepsilon_2$.

Next, we prove that nesting preserves relaxed local correctability when performed with an LTC. Using this, we can lift a smaller RLCC to a larger block length.

Lemma 6.4. *Let $\mathsf{LTC} \subseteq \Sigma^{n_{\mathsf{LTC}}}$ be an LTC with query complexity q_{LTC}, distance δ_{LTC}, and testability κ_{LTC}. Let $C \subseteq \Sigma^n$ be an RLCC with query complexity q and correcting radius δ where $n \leq n_{\mathsf{LTC}}$. Then, $\mathsf{LTC} \Cap C \subseteq \Sigma^{n_{\mathsf{LTC}}}$ is an RLCC with correcting radius $\delta_{\mathsf{LTC}}/2$ and query complexity*

$$q + O\left(\frac{q_{\mathsf{LTC}} n_{\mathsf{LTC}}}{\delta \kappa_{\mathsf{LTC}} n}\right).$$

Proof. Given the LTC-tester T and the C-corrector M_C, we can give a corrector $M^w(i)$ for $\mathsf{LTC} \Cap C$:

1. Let I be an interval of $[n_{\mathsf{LTC}}]$ of size n, defined as follows:

$$I := \begin{cases} \{\lceil i/n \rceil \cdot n - (n-1), \ldots, \lceil i/n \rceil \cdot n\} & \text{if } i < n_{\mathsf{LTC}} - n \\ \{n_{\mathsf{LTC}} - (n-1), \ldots, n_{\mathsf{LTC}}\} & \text{if } i \geq n_{\mathsf{LTC}} - n. \end{cases}$$

   In particular, I is a sub-interval that contains i and satisfies $c|_I \in C$ for all $\forall c \in \mathsf{LTC} \Cap C$.

2. Run $M_C^{w|_I}(i^*)$, where $i^* := i + 1 - \min I$. We choose i^* such that $(w|_I)_{i^*} = w_i$.

3. For $t := O(n_{\mathsf{LTC}}/\delta\kappa_{\mathsf{LTC}}n)$ iterations, run T^w.

4. Output $\perp$ if any run of the LTC-tester in step 3 returns $\perp$; otherwise output the result of step 2.

The query complexity of step 2 is q, while the query complexity of step 3 is $q_{\mathsf{LTC}} \cdot O(n_{\mathsf{LTC}}/\delta\kappa_{\mathsf{LTC}}n)$.

Next, we can show that M satisfies the perfect completeness condition of the RLCC definition. Since $w \in \mathsf{LTC} \Cap C \subseteq \mathsf{LTC}$, every repetition of the LTC-tester in step 3 will return $\top$ (accept), so for all i, $M^w(i)$ will return the result of step 2. By definition, if $w \in \mathsf{LTC} \Cap C$, then $w|_I \in C$. Therefore, the C-corrector in step 2 will return $(w|_I)_{i^*} = w_i$ with certainty, and so will $M^w(i)$, as needed.

Finally, we show soundness. Let $0 < \operatorname{dist}(w, \mathsf{LTC} \Cap C) < \delta_{\mathsf{LTC}}/2$, and let $c \in \mathsf{LTC} \Cap C$ be the unique closest codeword to w. Note that $c \in \mathsf{LTC}$ and $c|_I \in C$. Then, for all i, we need to show $\Pr[M^w(i) \in \{c_i, \perp\}] \geq 2/3$.

- First, assume $\operatorname{dist}(w, c) = \operatorname{dist}(w, \mathsf{LTC} \Cap C) \geq \delta n/2n_{\mathsf{LTC}}$. Because $\operatorname{dist}(w, c) < \delta_{\mathsf{LTC}}/2$, c is the unique codeword in LTC which is closest to w by Remark 5.2.

Hence, one run of T^w will return $\perp$ with probability at least $\kappa_{\mathsf{LTC}}\delta n/2n_{\mathsf{LTC}}$. Therefore, step 3 returns $\perp$ with probability

$$\geq 1 - \left(1 - \frac{\delta\kappa_{\mathsf{LTC}}n}{2n_{\mathsf{LTC}}}\right)^t \geq 1 - \exp\left(-\frac{t\delta\kappa_{\mathsf{LTC}}n}{2n_{\mathsf{LTC}}}\right) \geq 2/3,$$

for our choice of t. Thus, $\forall i.\, \Pr[M^w(i) = \perp] \geq 2/3$ which satisfies soundness.

- Now assume $\text{dist}(w, c) < \delta n/2n_{\mathsf{LTC}}$. From here we can bound the distance between the substrings $w|_I$ and $c|_I$; the distance of the substrings is maximized if all of the symbols that differ between w and c lie in the interval I. Consequently, $\text{dist}(w|_I, c|_I) < \delta/2$, and $c|_I$ is the unique codeword in C closest to $w|_I$ (see Remark 5.2).

  Applying the soundness condition of the C-corrector,

$$\Pr[M_C^{w|_I}(i^*) \in \{\underbrace{(c|_I)_{i^*}}_{c_i}, \perp\}] \geq 2/3.$$

  $M^w(i)$ will return either the output of $M_C^{w|_I}(i^*)$, or $\perp$ if some iteration of step 3 rejects. Hence,

$$\Pr[M^w(i) \in \{c_i, \perp\}] \geq \Pr[M_C^{w|_I}(i^*) \in \{c_i, \perp\}] \geq 2/3.$$

This shows that the soundness condition is satisfied in all cases. □

Remark 6.5. This proof can be modified to show the nested RLCC has correcting radius $(1 - \varepsilon)\delta_{\mathsf{LTC}}$ and query complexity

$$q + O_\varepsilon\left(\frac{q_{\mathsf{LTC}}n_{\mathsf{LTC}}}{\delta\kappa_{\mathsf{LTC}}n}\right).$$

Even if there are multiple codewords c with $\text{dist}(w, c) \leq (1 - \varepsilon)\delta_{\mathsf{LTC}}$, none can have distance less than $\varepsilon\delta_{\mathsf{LTC}}$ by the triangle inequality. Hence, repetitions of the tester will still detect corruptions.

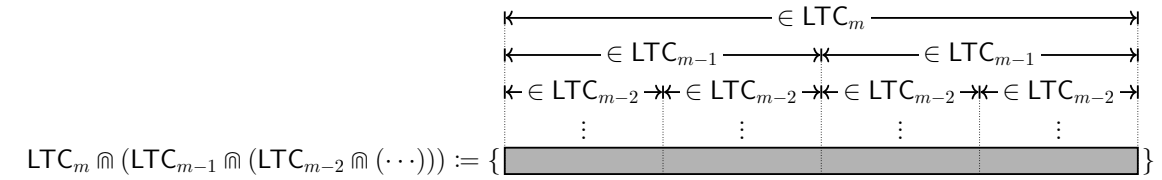


Figure 6.2: Iterated nesting produces a code with a recursive tree structure. In this example, successive block lengths double, producing a binary tree. The whole codeword is in LTC_m, both halves of the codeword are in LTC_{m-1}, all four quarters of the codeword are in LTC_{m-2}, and so on.

In summary, nesting an RLCC in an LTC yields a code which inherits the best properties of both: the resulting code inherits distance and block length from the larger LTC, as well as the relaxed local correctability of the smaller RLCC. Therefore, we can build an RLCC with arbitrarily large block length by iteratively nesting the code in a series of LTCs with larger and larger block length.

Proposition 6.6. *Let $\mathsf{LTC}_1, \ldots, \mathsf{LTC}_m$ be a sequence of linear LTCs over a finite field alphabet $\mathbb{F}$ with block lengths $n_1, \ldots, n_m$, which satisfy the following properties:*

- *$n_j \leq n_{j+1}$ for all j,*
- *every LTC_j has rate at least $1 - \varepsilon_{\mathsf{LTC}}$ and distance at least δ_{LTC}, and*
- *every LTC_j has a local tester with query complexity at most q_{LTC} and testability at least κ_{LTC}.*

Then, $C \coloneqq \mathsf{LTC}_m \Cap (\mathsf{LTC}_{m-1} \Cap \ldots (\mathsf{LTC}_2 \Cap \mathsf{LTC}_1) \ldots)$ is a linear RLCC which satisfies the following properties:

- *alphabet $\mathbb{F}$ and block length n_m,*
- *rate at least*
$$1 - \varepsilon_{\mathsf{LTC}} \left(1 + \sum_{j=2}^{m} \prod_{j'=2}^{j} \frac{n_{j'-1}}{n_{j'}} \left\lceil \frac{n_{j'}}{n_{j'-1}} \right\rceil \right),$$
- *correcting radius at least $\delta_{\mathsf{LTC}}/2$, and*

- *query complexity at most*

$$n_1 + \sum_{j=2}^{m} O\left(\frac{q_{\mathsf{LTC}} n_j}{\delta_{\mathsf{LTC}} \kappa_{\mathsf{LTC}} n_{j-1}}\right).$$

Remark 6.7. If n_j divides n_{j+1} for all j, then the rate lower bound simplifies to $1 - m\varepsilon_{\mathsf{LTC}}$. Even when n_j does not divide n_{j+1}, with appropriate parameter choices we can still simplify the rate to $1 - O(m\varepsilon_{\mathsf{LTC}})$. Therefore, we can view each nesting step as reducing the rate of C by $O(\varepsilon_{\mathsf{LTC}})$, so that each step incurs an additive (rather than multiplicative) cost in both rate and query complexity.

Proof. The alphabet and block length of C follows from the definition of nesting. Next, we show the rate. Let $\mathsf{LTC}_j \Cap \cdots (\mathsf{LTC}_2 \Cap \mathsf{LTC}_1)$ have rate $1 - \varepsilon_j$. We know $\varepsilon_1 \leq \varepsilon_{\mathsf{LTC}}$ by definition, and Lemma 6.2 tells us that for $j \geq 2$,

$$\varepsilon_j \leq \varepsilon_{\mathsf{LTC}} + \frac{n_{j-1}}{n_j} \left\lceil \frac{n_j}{n_{j-1}} \right\rceil \cdot \varepsilon_{j-1}.$$

Hence by induction, ε_m is upper bounded by the series

$$\varepsilon_m \leq \varepsilon_{\mathsf{LTC}} \cdot \left(1 + \sum_{j=2}^{m} \prod_{j'=2}^{j} \frac{n_{j'-1}}{n_{j'}} \left\lceil \frac{n_{j'}}{n_{j'-1}} \right\rceil\right),$$

which gives the desired lower bound on the rate of C.

Finally, we show that C is an RLCC. LTC_1 can be viewed as an RLCC with correcting radius $\delta_{\mathsf{LTC}}/2$ because it has a trivial correcting algorithm that reads the entire input and checks whether it is in LTC_1. This corrector has query complexity n_1. Then by applying Lemma 6.4, $\mathsf{LTC}_2 \Cap \mathsf{LTC}_1$ has a corrector with correcting radius $\delta_{\mathsf{LTC}}/2$ and query complexity

$$\leq n_1 + O\left(\frac{q_{\mathsf{LTC}} n_2}{\delta_{\mathsf{LTC}} \kappa_{\mathsf{LTC}} n_1}\right).$$

By applying this lemma $m - 2$ more times with $\mathsf{LTC}_3, \ldots, \mathsf{LTC}_m$, we can conclude that there is a C-corrector with correcting radius $\delta_{\mathsf{LTC}}/2$ and total query complexity

$$\leq n_1 + \sum_{j=2}^{m} O\left(\frac{q_{\mathsf{LTC}} n_j}{\delta_{\mathsf{LTC}} \kappa_{\mathsf{LTC}} n_{j-1}}\right) \square$$

6.2.2 Instantiating the nesting framework

We now have all of the tools we need to construct asymptotically good RLCCs, as long as we pick a suitable sequence of LTCs. If the block length of each successive LTC grows by at least a constant factor, then $m \leq O(\log n)$. Thus, using Remark 6.7 as guidance, we need a family of explicit LTCs with rate at least $1 - O(1/\log n)$ in order for Proposition 6.6 to yield an RLCC with constant rate. This means these LTCs will have subconstant distance (which we will patch later) due to the Singleton bound. We will use the following instantiation of Theorem 5.18:

Corollary 6.8 ([DEL+22], see Section 5.5)**.** *For every sufficiently large $N \in \mathbb{N}$, there exists an integer $n \in [\Omega(N/\log^{30} N), N]$ and a family of LTCs $\{\mathsf{LTC}_1, \ldots, \mathsf{LTC}_m\}$ such that*

- *Each LTC_j is binary, linear, and explicit.*
- *Each LTC_j has:*
  - *rate $R = 1 - \varepsilon_{\mathsf{LTC}} \geq 1 - O(1/\log N)$,*
  - *distance $\delta_{\mathsf{LTC}} \geq \Omega(1/\log^3 N)$,*
  - *testability $\kappa_{\mathsf{LTC}} \geq \Omega(1/\log^{15} N)$, and*
  - *query complexity $q_{\mathsf{LTC}} \leq O(\log^{20} N)$.*
- *Each LTC_j has block length n_j, such that $n_1 \leq O(\log^{50} N)$, $n_m = n$, and $\forall j.\, n_{j+1}/n_j = \Theta(\log^{30} N)$.*
- *The number of codes is $m = O(\log N / \log\log N)$.*

Remark 6.9. In this concrete family of LTCs, the block length of each successive LTC increases by a polylogarithmic factor, instead of the constant factor we previously imagined using. Thus, $m = O(\log n/\log\log n) = o(\log n)$. The $n_j/n_{j-1} = \operatorname{polylog} n$ factor also appears in the query complexity, which is acceptable. One could choose

the LTC rate to be $1 - O(\log\log n/\log n)$ instead in order to marginally improve parameters such as δ_{LTC}, κ_{LTC}, and q_{LTC} which appear in the query complexity, while worsening the rate.

With this family of LTCs, we can construct high rate RLCCs with polylogarithmic query complexity, albeit with subconstant correcting radius.

Theorem 6.10. *For sufficiently large $N \in \mathbb{N}$, there is an explicit binary linear RLCC with block length $n \in [\Omega(N/\log^{30} N), N]$, rate $1 - O(1/\log\log N)$, correcting radius $\Omega(1/\log^3 N)$, and query complexity $O(\log^{69} N/\log\log N)$.*

Proof. We can use the parameter N with Corollary 6.8 to get a family of LTCs to plug into Proposition 6.6. Formally, our code is $C \coloneqq \mathsf{LTC}_m \Cap (\mathsf{LTC}_{m-1} \Cap \dots (\mathsf{LTC}_2 \Cap \mathsf{LTC}_1)\dots)$, with $\{\mathsf{LTC}_i\}_{i\in[m]}$ having parameters as defined in Corollary 6.8. We find that by Proposition 6.6, our code C has block length $n \in [\Omega(N/\log^{30} N), N]$ and rate

$$
\begin{aligned}
&\geq 1 - \varepsilon_{\mathsf{LTC}}\left(1 + \sum_{j=2}^{m}\prod_{j'=2}^{j}\left(\frac{n_{j'-1}}{n_{j'}}\left\lceil\frac{n_{j'}}{n_{j'-1}}\right\rceil\right)\right)\\
&\geq 1 - \varepsilon_{\mathsf{LTC}}\left(1 + \sum_{j=2}^{m}\prod_{j'=2}^{j}\left(1 + \frac{n_{j'-1}}{n_{j'}}\right)\right)\\
&\geq 1 - \varepsilon_{\mathsf{LTC}}\sum_{j=1}^{m}\big(1 + O\big(1/\log^{30} N\big)\big)^{j-1}\\
&\geq 1 - \varepsilon_{\mathsf{LTC}} m \cdot \exp\big(O\big(m/\log^{30} N\big)\big)\\
&= 1 - (1 + o(1))m\varepsilon_{\mathsf{LTC}}\\
&= 1 - O\left(\frac{1}{\log\log N}\right).
\end{aligned}
$$

In addition, Proposition 6.6 tells us C is an RLCC with radius $\delta_{\mathsf{LTC}}/2 = \Omega(1/\log^3 N)$

and query complexity

$$\begin{aligned}
&\leq n_1 + \sum_{i=2}^{m} O\left(\frac{q_{\mathsf{LTC}} n_i}{\delta_{\mathsf{LTC}} \kappa_{\mathsf{LTC}} n_{i-1}}\right) \\
&\leq O(\log^{50} N) + \sum_{i=2}^{m} O\left(\log^{20+30+3+15} N\right) \\
&\leq O\left(\frac{\log^{69} N}{\log\log N}\right).
\end{aligned}$$

□

C is an RLCC with the desired rate and query complexity, but it is not yet asymptotically good since its correcting radius is determined by the distance of the LTC family from Corollary 6.8. This distance is subconstant since all of the LTCs must have rate $1 - O(1/m)$. Thankfully, we can boost the correcting radius of our RLCC with one final nesting step using an LTC with constant distance and rate. The increase in query complexity is negligible, and the rate of the last LTC dominates the rate of the resulting RLCC.

Corollary 6.11. *Let* LTC *be a binary linear LTC with sufficiently large block length* N*, and with rate* R*, distance* δ_{LTC}*, testability* κ*, and query complexity* q*. Then, there exists a binary linear RLCC* C *with block length* N*, rate* $R - O(1/\log\log N)$*, correcting radius* $\delta_{\mathsf{LTC}}/2$*, and query complexity*

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

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

Proof. We pick sufficiently large N such that we can use Theorem 6.10 to craft RLCC C' with block length $n \in [\Omega(N/\log^{30} N), N]$, rate $1 - O(1/\log\log N)$, correcting radius $\delta = \Omega(1/\log^3 N)$, and query complexity $O(\log^{69} N/\log\log N)$. We now construct

$$C \coloneqq \mathsf{LTC} \Cap C'.$$

By construction, C is explicit and has block length N. By Lemma 6.2 (in particular, Remark 6.3) on LTC and C', the overall rate of C is at least $R - (1 + n/N) \cdot$

$O(1/\log\log N) = R - O(1/\log\log N)$. Applying Lemma 6.4 to LTC and C', the correcting radius of C is at least $\delta_{\mathsf{LTC}}/2$ and the query complexity of C is

$$O\left(\frac{qN}{\delta\kappa n}\right) + O\left(\frac{\log^{69} n}{\log\log n}\right) \leq O\left(\frac{q}{\kappa} \cdot \log^{33} N + \frac{\log^{69} n}{\log\log n}\right). \square$$

This corollary states that for any binary linear LTC, there is a binary linear RLCC with nearly the same rate and distance, and with the same (up to polylogarithmic factors) query complexity, and so all that remains is to pick an LTC achieving the desired rate-distance tradeoff.

6.3 Final construction

Our main results follow from choosing an appropriate LTC for Corollary 6.11. First, we choose an explicit LTC in order to get an explicit RLCC, which for $\varepsilon = \Omega(1)$ is asymptotically good.

Corollary 6.12 (Explicit RLCCs)**.** *For any rate $R = 1 - \varepsilon \in (0, 1)$ and for infinitely many n, there is an explicit binary linear RLCC with block length n, rate $R - O(1/\log\log n)$, correcting radius $\Omega(\varepsilon^3)$, and query complexity*

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

Proof. We can instantiate Corollary 6.11 using the LTCs of Theorem 5.18. $\square$

Next, we can optimize the rate-distance tradeoff by utilizing an appropriate nonexplicit LTC. Let $H(\cdot)$ be the binary entropy function. The Gilbert-Varshamov bound shows a random binary code of rate R will have distance δ, where $R + H(\delta) = 1 - o(1)$. This is the best rate-distance tradeoff known to be attainable by binary codes. We show that we can get arbitrarily close to this tradeoff, *even* if we need relaxed local correctability.

Corollary 6.13 (Gilbert–Varshamov bound RLCCs)**.** *For any $R, \delta, \varepsilon \in (0,1)$ such that*

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

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

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

Proof. Dinur, Evra, Livne, Lubotzky, and Mozes [DEL$^+$22] construct explicit LTCs with rate arbitrarily close to 1, which implies the existence of infinitely many nonexplicit LTCs that approach the Gilbert–Varshamov bound (using ideas from [KMRS17, GKO$^+$18]; see [DEL$^+$22, Corollary 1.2]). These LTCs can have any rate R and distance δ such that $R + H(\delta) = 1 - \varepsilon$, in which case the testability is $\kappa \geq \operatorname{poly}(\varepsilon)$ and the query complexity is $q_{\mathsf{LTC}} \leq \operatorname{poly}(1/\varepsilon)$. We can plug these parameters into Corollary 6.11 to yield RLCCs. Because the rate of the RLCC approaches the rate of the LTC, and the distance of the RLCC is at least the distance of the LTC, we can say that the RLCC also approaches the Gilbert–Varshamov bound. □

Remark 6.14. While we used locally testable codes throughout our construction, our RLCCs are *not* necessarily locally testable. The codes we construct are subsets of LTCs (some codewords are deleted at each nesting step as we amass additional constraints), and we do not know whether we can locally distinguish codewords that are retained from those that are removed. Because the LTC soundness property must hold for all inputs (unlike RLCC soundness which promises that the input has bounded distance from the code), this distinguishing ability seems necessary to showing that our RLCCs are also LTCs. This is important because both properties hand-in-hand are morally necessary (but not sufficient) to construct a PCP (as mentioned in Section 6.1): local testability checks that the input lies within the correcting radius of some valid proof, so that we can use relaxed local correctability to extract information

from the input. The best known asymptotically good RLCCs which are also LTCs are the codes constructed by Gur, Ramnarayan, and Rothblum [GRR20].

Chapter 7: Relaxed vs. Full Local Decodability with Few Queries: Equivalence and Separations for Linear Codes

7.1 Introduction

A binary locally decodable code (LDC) $C\colon \{0,1\}^k \to \{0,1\}^n$ is an error-correcting code that admits a local decoding algorithm with the following guarantee: when given access to a corrupted codeword y obtained by corrupting $x = C(b)$ in a constant fraction of coordinates, the local decoder is able to recover any bit b_i of the chosen message b with good probability while only reading a small (say, constant) number of bits from y. Although they were first formally defined in [KT00], locally decodable codes were implicitly used in the proof of the PCP theorem [ALM+98, AS98], and have since found numerous applications to, e.g., private information retrieval, hardness amplification, probabilistically checkable proofs, self-correction, fault-tolerant circuits and data structures (e.g., [BFLS91, LFKN92, BLR93, BK95, CKGS98, IK99, BIW10, CGW13, ALRW17]). We refer the reader to the surveys of [Tre04, Dvi12, Yek12] for more details.

The central question in the study of LDCs is to understand the length n of the best locally decodable code that can tolerate a small constant fraction of errors as a function of k, the length of the message, and q, the number of queries of the decoder. Following [KT00], there has been a long line of work on both constructing and proving lower bounds for locally decodable codes, with a particular focus on the constant query regime (i.e., $q = O(1)$).

A simple observation (see [KT00, Section 3.2]) shows that it is not possible to construct 1-query locally decodable codes. For $q = 2$, the Hadamard code gives a 2-LDC of blocklength $n = 2^k$, and this is optimal up to constant factors in the exponent: the works of [KW03, GKST06] show that $n \geq 2^{\Omega(k)}$ for any 2-LDC. More

generally, the Reed–Muller code — a generalization of the Hadamard code (which are evaluations of linear functions) to polynomials of larger degree — gives a construction of a q-LDC of length $n = 2^{O(k^{1/(q-1)})}$. However, for $q \geq 3$, this can be improved: the matching vector codes of [Yek08, Efr09, DGY11] give constructions of q-LDCs of subexponential, but still superpolynomial, blocklength $n = 2^{k^{o(1)}}$ for any constant $q \geq 3$.[1] But, unlike the case of $q = 2$, we are far from understanding whether these codes are optimal or not: our best lower bound is a polynomial lower bound of $n \geq \tilde{\Omega}(k^{q/(q-2)})$ for any $q \geq 3$ [KW03, AGKM23, HKM$^+$24, BHKL24, JM24].

This gap between subexponential constructions and polynomial lower bounds for LDCs has led to the definition of a class of codes with weaker local decoding properties called relaxed locally decodable codes (RLDCs), for which much better constructions are known. Introduced in [BSGH$^+$06], a relaxed LDC is no longer required to output the correct message bit b_i with good probability, and instead may output a special symbol $\perp$ to signify that the decoder has detected an error.[2] Unlike standard locally decodable codes, for large enough constant q one can construct q-query RLDCs with a polynomial blocklength of $n = k^{1+O(1/q)}$ [BSGH$^+$06, GRR20, CGS22, AS21, Gol24], and the works of [GL21, DGL21, Gol23b] prove a near-matching lower bound of $n \geq k^{1+\Omega(1/q^2)}$. However, somewhat curiously, for the specific case of $q = 2$ the work of [BBC$^+$23] proves an exponential lower bound of $n \geq 2^{\Omega(k)}$. That is, up to constant factors in the exponent, the best 2-RLDC is a 2-LDC, namely the Hadamard code.

To summarize: for RLDCs, there exists a large constant q such that there are q-RLDCs of length $n = \text{poly}(k)$, whereas the best 2-RLDC has length $n = 2^{\Omega(k)}$. The work of [BBC$^+$23] thus raises the following interesting question (mentioned explicitly

[1] More precisely, the length of these codes is $n = 2^{2^{(\log k)^{\varepsilon(q)}}}$ for some constant $\varepsilon(q) \approx 1/\log_2 q < 1$ that depends on q.

[2] To prevent the trivial decoder that always outputs $\perp$ from satisfying the definition, one requires that the decoder outputs the correct bit b_i with good probability when given access to an uncorrupted codeword $x = C(b)$.

in [BBC+23, Section 2]):

Question 7.1 ([BBC+23])**.** *What is the threshold q where the optimal blocklength of a q-RLDC "transitions" from superpolynomial in k to polynomial in k?*

In this chapter, we investigate Question 7.1. However, there is an obvious barrier to answering Question 7.1, coming from our lack of understanding of the analogous question for LDCs. For example, if it is the case that the "transition threshold" is above $q = 3$, then to prove this one would need to prove that there are no 3-RLDCs of polynomial length. In particular, this would also imply that there are no 3-LDCs (unrelaxed) of polynomial length as well. However, the best 3-LDC lower bound is only $n \geq \tilde{\Omega}(k^3)$ [AGKM23], and improving this (or showing that it is tight) is a well-studied and challenging open question.

To avoid such issues, we instead reinterpret the result of [BBC+23] as follows. Not only do they prove that any 2-RLDC has $n \geq 2^{\Omega(k)}$, they in fact prove this result by showing that any 2-RLDC *is* a 2-LDC, and then they apply the exponential lower bound of [KW03, GKST06] for 2-LDCs. Thus, for $q = 2$, RLDCs and LDCs are equivalent. On the other hand, for large constant q there is a large gap between the best constructions of q-RLDCs (which have length $n = k^{1+O(1/q)}$ [BSGH+06, AS21, Gol24]) and q-LDCs (which have length $n = 2^{k^{o(1)}}$ [Yek08, Efr09, DGY11]), giving evidence that for a large enough constant q, RLDCs and LDCs are not equivalent. We thus pose the following question:

Question 7.2. *What is the smallest r where (1) for every $q < r$, every q-RLDC is a q-LDC, and (2) there is an r-RLDC that is* not *an r-LDC?*

Our main results show that the threshold r in Question 7.2 is between 4 and 15 for the case of linear codes. In fact, we show this for a "stronger" version of Question 7.2 where the r-RLDC in Item (2) is not only not an r-LDC, but also not a t-LDC for any constant t.

Locally correctable and relaxed locally correctable codes. The above discussion is for locally decodable codes and their relaxed variant. One may ask the same questions for locally correctable codes (LCCs) and their relaxed variant, as LCCs are a closely related notion to LDCs that are defined in a near-identical way. The difference between an LCC and an LDC is that a locally correctable code requires that the decoder is able to self-correct bits of the *codeword*, whereas LDCs only need to correct bits of the *message*. One can show that any LCC is in fact an LDC as well, so LCCs are a stronger notion.[3] And, similarly to LDCs, one can define a relaxed notion of LCCs where the decoder is allowed to output a special error symbol $\perp$.

For LCCs, the state-of-the-art constructions and lower bounds are quite different compared to LDCs. For $q = 2$, the lower bounds of [KW03, GKST06, DS07] show that the Hadamard code with $n = 2^k$ is an optimal 2-LCC. For $q \geq 3$, however, the best construction of q-LCCs remains the folklore construction from Reed–Muller codes, which achieves a length of $n = 2^{O(k^{1/(q-1)})}$. This is unlike the case of LDCs, where the constructions of subexponential length coming from matching vector codes are much better than Reed–Muller codes when $q \geq 3$.

We also have stronger lower bounds for LCCs as compared to LDCs. Namely, for $q = 3$, the work of [KM24a] and follow-up works of [AG24, Yan24, KM24b] prove exponential lower bounds for 3-LCCs: the current best results are $n \geq 2^{\Omega(\sqrt{k/\log k})}$ for linear codes [AG24] and $n \geq 2^{\Omega(k^{1/5})}$ for nonlinear codes [KM24b]. For comparison, recall that the best 3-LDC lower bound remains $n \geq \tilde{\Omega}(k^3)$. Furthermore, for any odd constant $q \geq 5$, one can show a lower bound of $n \geq \tilde{\Omega}(k^{(q-1)/(q-3)})$ for linear q-LCCs [AG24], which is better than the best q-LDC lower bound by a small polynomial factor in k.

For relaxed LCCs (RLCCs), the current best results are identical to RLDCs.

[3] This is fairly straightforward to show for linear codes, as without loss of generality, by changing bases one can make any linear code systematic, i.e., the first k bits of any codeword $x = C(b)$ is simply the message b. For nonlinear codes, this can also be done with more effort, see [BGT17, Appendix A].

Namely, for large enough constant q, the work of [AS21] constructs q-query RLCCs of length[4] $n = k^{1+O(1/q)}$ and the lower bounds of [GL21, DGL21, Gol23b] again prove a near-matching lower bound of $n \geq k^{1+\Omega(1/q^2)}$. And, for $q = 2$, [BBC+23] proves an exponential lower bound for 2-RLCCs of $n \geq 2^{\Omega(k)}$.[5]

We can thus also ask Question 7.2 for LCCs and RLCCs.

Question 7.3 (Question 7.2 for LCCs)**.** *What is the smallest r where (1) for every $q < r$, every q-RLCC is a q-LCC, and (2) there is an r-RLCC that* not *an r-LCC?*

We also prove results for LCCs/RLCCs similar to the case of LDCs/RLDCs. In particular, we show that the threshold r in Question 7.3 is between 4 and 41 for the case of linear codes.

7.1.1 Our results

Before we state our results, let us formally define LDCs and RLDCs (LCCs and RLCCs are defined analogously, see Definitions 5.9 and 5.14). For two strings $x, y \in \{0,1\}^n$, we let $\Delta(x, y)$ denote the Hamming distance between x and y, i.e., the number of indices $j \in [n]$ where $x_j \neq y_j$.

Definition 7.4 (Binary locally decodable codes; see Definition 5.8)**.** A code $C \colon \{0,1\}^k \to \{0,1\}^n$ is a (q, δ, c, s)-LDC if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ with the following properties. The algorithm $\mathrm{Dec}(\cdot)$ is given oracle access to a string $y \in \{0,1\}^n$, takes an index $i \in [k]$ as input, and outputs a bit in $\{0,1\}$ with the following guarantees:

(1) (q-queries) for any y and i, the algorithm $\mathrm{Dec}^y(i)$ reads at most q indices of y,

(2) (c-completeness) for all $b \in \{0,1\}^k$ and $i \in [k]$, $\Pr[\mathrm{Dec}^{C(b)}(i) = b_i] \geq c$, and

[4]The works of [BSGH+06, Gol24] construct q-RLDCs with the same parameters, but their codes are not RLCCs.

[5]Unlike the case of RLDCs/LDCs, the work of [BBC+23] does not prove a 2-RLCC to 2-LCC reduction. However, since any 2-RLCC is a 2-RLDC, the exponential lower bound still applies.

(3) ((δ, s)-soundness error) for all $b \in \{0,1\}^k$, $i \in [k]$, and all $y \in \{0,1\}^n$ with $\Delta(y, C(b)) \leq \delta n$, $\Pr[\mathrm{Dec}^y(i) \neq b_i] \leq s$.

Definition 7.5 (Binary relaxed locally decodable codes; see Definition 5.13)**.** A code $C\colon \{0,1\}^k \to \{0,1\}^n$ is a (q, δ, c, s)-RLDC if there exists a randomized decoding algorithm $\mathrm{Dec}(\cdot)$ with the following properties. The algorithm $\mathrm{Dec}(\cdot)$ is given oracle access to a string $y \in \{0,1\}^n$, takes an index $i \in [k]$ as input, and outputs either a bit in $\{0,1\}$ or a special symbol $\perp$ with the following guarantees:

(1) (q-queries) for any y and i, the algorithm $\mathrm{Dec}^y(i)$ reads at most q indices of y,

(2) (c-completeness) for all $b \in \{0,1\}^k$ and $i \in [k]$, $\Pr[\mathrm{Dec}^{C(b)}(i) = b_i] \geq c$, and

(3) ((δ, s)-relaxed soundness error) for all $b \in \{0,1\}^k$, $i \in [k]$, and all $y \in \{0,1\}^n$ with $\Delta(y, C(b)) \leq \delta n$, $\Pr[\mathrm{Dec}^y(i) \notin \{b_i, \perp\}] \leq s$.

The standard definition of an RLDC sets parameters as follows: $\delta = O(1)$ (constant fraction of errors), $c = 1$ (perfect completeness), and $s = 1/3$. That is, in the presence of a constant fraction of errors, the RLDC decoder outputs either the correct bit or a special error symbol $\perp$ with probability at least $2/3$. In the standard definition of an LDC, it is also common to set $s = 1/2 - \varepsilon$ for a small constant ε, meaning that in the presence of a constant fraction of errors, the LDC decoder outputs the correct bit with probability at least $1/2 + \varepsilon$.

Our main results. Our first result shows that any linear 3-RLDC with soundness error $1/2 - \eta$ is a 3-LDC, and thus the threshold in Question 7.2 must be at least 4 for linear codes.

Theorem 7.6. *Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(3, \delta, 1, \frac{1}{2} - \eta)$-RLDC with a possibly adaptive decoder. Then, for any $\varepsilon > 0$, C is a linear $(3, 2\eta\delta\varepsilon/3, 1, \varepsilon)$-LDC. In particular, if C is a linear $(3, \Theta(1), 1, \frac{1}{3})$-RLDC, then C is a linear $(3, \Theta(1), 1, \frac{1}{3})$-LDC. Furthermore, the same statement holds for RLCCs/LCCs.*

By combining Theorem 7.6 with the best known lower bounds for linear 3-LDCs [AGKM23, HKM$^+$24, BHKL24, JM24] and linear 3-LCCs [AG24], we obtain the following new lower bounds for 3-RLDCs and 3-RLCCs.

Corollary 7.7. *Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear code. Then, the following hold:*

(1) If C is a $(3, \delta, 1, \frac{1}{2} - \varepsilon)$-RLDC with a possibly adaptive decoder, then $n \geq \Omega_{\delta,\varepsilon}((k/\log k)^3)$.

(2) If C is a $(3, \delta, 1, \frac{1}{2} - \varepsilon)$-RLCC with a possibly adaptive decoder, then $n \geq 2^{\Omega_{\delta,\varepsilon}(\sqrt{k/\log k})}$.

For comparison, the prior best lower bound for 3-RLDCs or 3-RLCCs came from the works [GL21, DGL21, Gol23b], which prove a lower bound of $n \geq k^{1+\Omega(1/q^2)}$ for any q-RLDC/RLCC, where the constant in the $\Omega(\cdot)$ is smaller than $1/10$. Thus, Corollary 7.7 gives a stronger lower bound for 3-RLDCs, and a substantially stronger lower bound for 3-RLCCs.

The lower bounds in Corollary 7.7 require the RLDC/RLCC to have a soundness error of $1/2 - \varepsilon$. However, for RLDCs/RLCCs (unlike LDCs/LCCs), any soundness error of $1 - \varepsilon$ can be amplified to $1/2 - \varepsilon$ via sequential repetition (see Observation 7.11), although this does increase the number of queries in the RLDC by a constant factor. In this way, one could view the requirement on the soundness error in Corollary 7.7 as a mild limitation, although we note that the lower bound for 2-RLDCs in [BBC$^+$23] also has the same requirement.

Theorem 7.6 is a corollary of the following more general statement that we show. There is a soundness threshold $s(q)$, a function of q, such that *any* linear q-RLDC with soundness error $s \leq (1 - \alpha)s(q)$ must be a linear q-LDC. That is, if a linear q-RLDC has soundness error better than $s(q)$, then the "reason" is that the q-RLDC is in fact a q-LDC. For 3-RLDCs, this threshold $s(q)$ is $1/2$, which implies Theorem 7.6.

Theorem 7.8 (Soundness error threshold for q-RLDCs; binary case of Theorem 7.17)**.** *Let $s(q) \coloneqq 2^{-\lfloor q/2 \rfloor}$. Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(q, \delta, 1, s)$-RLDC with a possibly adaptive decoder, where $s \leq (1-\alpha)s(q)$. Then, for any $\varepsilon > 0$, C is a linear $(q, \alpha\delta\varepsilon/q, 1, \varepsilon)$-LDC. Furthermore, the same result holds for RLCCs/LCCs.*

Theorem 7.8 shows that the answer to Question 7.2 depends on the soundness error s that we choose in Definition 7.5.[6] The standard definition of RLDCs typically chooses $s = 1/3$ (see, e.g., [BSGH$^+$06, GL21]), which is a constant smaller than $\frac{1}{2}$. If one wishes to have soundness error strictly less than, say, $s = 1/8$, then Theorem 7.8 implies that any q-RLDC that is additionally not a q-LDC must have $q \geq 8$.

More generally, Theorem 7.8 shows a qualitative difference between LDCs and RLDCs (that are not LDCs): the only way for a linear RLDC to have *strong soundness*, a property satisfied by constant query linear LDCs, is for it to already be an LDC. Strong soundness is a natural property, typically desired in local testing or decoding algorithms, that intuitively says that corrupted codewords y with fewer errors are decoded more successfully. More formally, strong soundness replaces Item (3) in Definitions 7.4 and 7.5 with the following corresponding condition:

Definition 7.9 (Strong soundness for LDCs and RLDCs)**.**

(1) (δ'-strong LDC soundness) for all $b \in \{0,1\}^k$, $i \in [k]$, and all $y \in \{0,1\}^n$, $\Pr[\mathrm{Dec}^y(i) \neq b_i] \leq \Delta(y, C(b))/(\delta' n)$.

(2) (δ'-strong RLDC soundness) for all $b \in \{0,1\}^k$, $i \in [k]$, and all $y \in \{0,1\}^n$, $\Pr[\mathrm{Dec}^y(i) \notin \{b_i, \perp\}] \leq \Delta(y, C(b))/(\delta' n)$.

Definition 7.9 says that the probability that the decoder is incorrect is proportional to the number of errors in y. The connection established in [KT00] between

[6] Technically, to deduce that this depends on the soundness error one also needs to exhibit a q-RLDC (with soundness error $> s(q)$) that is also not a q-LDC. Existing constructions of RLDCs ([BSGH$^+$06, GRR20, CGS22, AS21, Gol24]) should satisfy this condition, and in this chapter we give an RLDC with this property (Theorem 7.14).

linear constant q-LDCs and "smooth decoders" implies that any $(q, \delta, 1, 1/2-\varepsilon)$-LDC (Definition 7.4) satisfies Definition 7.9 for $\delta' = \delta/q$ (see Section 7.7). On the other hand, Theorem 7.8 implies that no such analogous statement can hold for RLDCs unless all RLDCs are LDCs, which is false (see, e.g., Theorem 7.14). Formally, we have the following corollary of Theorem 7.8.

Corollary 7.10 (RLDCs with Strong Soundness are LDCs)**.** *Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear q-RLDC with perfect completeness and δ'-strong soundness (Definition 7.9). Then for any $\varepsilon > 0$, C is a linear $(q, s(q)\varepsilon\delta'/4q, 1, \varepsilon)$-LDC. Furthermore, the same result holds for RLCCs/LCCs.*

At first glance, the exponential dependence on q in $s(q)$ in Theorem 7.8 may appear to be weak. However, a simple sequential repetition of the decoder, combined with a constant query RLDC that is not an q-LDC for any constant q, shows that the exponential dependence is necessary.

Observation 7.11. Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a $(q, \delta, 1, \varepsilon)$-RLDC with a possibly adaptive decoder. Then, for any integer $t \geq 1$, C is a $(qt, \delta, 1, \varepsilon^t)$-RLDC. The same statement holds for RLCCs.

Observation 7.11 implies that if the threshold $s(q)$ in Theorem 7.8 did not decay exponentially with q, we could take any $(q, \delta, 1, \varepsilon)$-RLDC (where q, δ, ε are constant) and then choose t to be a constant so that $\varepsilon^t < s(qt)$ (which exists if $s(qt)$ does not decay exponentially), and this would show that the code is also a constant query LDC. However, as we will show (Theorem 7.14), there is a code that is an $O(1)$-RLDC but not a q-LDC for any constant q. We thus conclude that $s(q)$ must decay exponentially in q.

The proof of Observation 7.11 is simple, and is included in Section 5.6. We prove Theorems 7.6 and 7.8 in Section 7.3.

Extensions of Theorem 7.8. Theorem 7.8 makes two assumptions on the code: it assumes that C is linear, and that it has perfect completeness. As we will discuss in Section 7.2.2, our proof makes heavy use of the linearity of C, and it is not clear how to remove this assumption from Theorem 7.8. However, we can extend Theorem 7.8 to codes with imperfect completeness via our next theorem, which shows how to convert any linear RLDC with imperfect completeness to a linear RLDC with perfect completeness.

Theorem 7.12 (Binary case of Theorem 7.23)**.** *Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(q, \delta, 1-\varepsilon, s)$-RLDC with a possibly adaptive decoder. Then, C is a linear $(q, \delta, 1, s+3\varepsilon)$-RLDC with a nonadaptive decoder. Furthermore, the same result holds for RLCCs/LCCs.*

Combining Theorems 7.8 and 7.12, we obtain the following corollary, which extends Theorem 7.8 to linear RLDCs with imperfect completeness.

Corollary 7.13. *Let $s(q) \coloneqq 2^{-\lfloor q/2 \rfloor}$. Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(q, \delta, 1-\eta, s)$-RLDC with a possibly adaptive decoder, where $s \leq (1-\alpha)s(q) - 3\eta$. Then, for any $\varepsilon > 0$, C is a linear $(q, \alpha\delta\varepsilon/q, 1, \varepsilon)$-LDC. Furthermore, the same result holds for RLCCs/LCCs.*

Theorem 7.12 is a query-preserving version of a recent theorem of [Gol23a], which shows a similar result but requires an extra query. Namely, [Gol23a] shows how to convert any linear q-RLDC with an adaptive decoder and imperfect completeness into a $(q+1)$-RLDC with a nonadaptive decoder and perfect completeness. This extra query made by [Gol23a] is significant for us, as our soundness threshold $s(q) = 2^{-\lfloor q/2 \rfloor}$ satisfies $s(3) = 1/2$ but $s(4) = 1/4$, and so the result of [Gol23a] only allows us to show that a 3-RLDC (RLCC) with imperfect completeness and soundness error $1/4$ (which is smaller than the standard setting of $1/3$) is a 4-LDC (LCC). For the case of RLCCs, this extra query is very significant, as the best lower bound for linear 4-LCCs

is only $n \geq \tilde{\Omega}(k^2)$ [KW03], whereas the best lower bound for (binary) linear 3-LCCs is $n \geq 2^{\Omega(\sqrt{k/\log k})}$ [AG24]. We prove Theorem 7.12 in Section 7.4.

Larger fields. It is straightforward to extend Theorem 7.8 to larger fields $\mathbb{F}$ by replacing the function $s(q)$ with $s_{\mathbb{F}}(q) \coloneqq |\mathbb{F}|^{-\lfloor q/2 \rfloor}$, and we do this when we prove Theorem 7.8 in Section 7.3. In Section 7.6, we extend Theorem 7.8 to larger fields *without* any field-dependent loss in $s(q)$ for the case of $q = 2$.

Constructions of RLDCs/RLCCs with small queries that are not LDCs. To complement Theorem 7.8 and provide a partial answer to Questions 7.2 and 7.3, we give a simple construction of RLDCs and RLCCs with small queries that are not constant query LDCs for any constant. To our knowledge, these are the first RLDC constructions to achieve constant queries for an explicit small constant.

Theorem 7.14 (Constructions of constant query RLDCs/RLCCs that are not LDCs)**.** *For every k, there is a linear code $C\colon \{0,1\}^k \to \{0,1\}^n$ where $n = k^{O(\log\log k)}$ such that C is a $(15, \delta, 1, 1/3)$-RLDC and a $(41, \delta, 1, 1/2 - \varepsilon)$-RLCC for a small constant $\varepsilon > 0$. Furthermore, C is not a $(O(\log k), O(n^{-1/3}), 1, 1/2 - \varepsilon)$-LDC (or LCC) for any $\varepsilon > 0$ (including subconstant ε).*

The blocklength of Theorem 7.14 is worse than that of prior constructions, as it is slightly superpolynomial in k rather than $\mathrm{poly}(k)$. However, the importance of Theorem 7.14 is that (1) the number of queries made by the code is an explicit small constant, and (2) we can prove that the code is not a constant query LDC. Thus, when combined with Corollary 7.13, Theorem 7.14 shows that the threshold in Question 7.2 is somewhere between 4 and 15 for linear RLDCs with perfect completeness and soundness $1/2 - \varepsilon$ for any constant $\varepsilon > 0$: Corollary 7.13 implies that linear 3-RLDCs with soundness $1/2 - \varepsilon$ are 3-LDCs, and Theorem 7.14 gives a 15-RLDC with soundness $1/2 - \varepsilon$ that is not an q-query LDC for any constant q.

While Theorem 7.14 gives a construction of a q-RLDC with an explicit constant $q = 15$, one may wonder what the implicit constant q is in previous works. We

attempted to determine the number of queries q needed for the RLDCs of [BSGH$^+$06, GRR20] to achieve soundness error $< 1/2$, and to the best of our knowledge, this q is at least 10^7 for both constructions. Intuitively, the reason the query complexity is so high is that these constructions make use of "proof composition" style results with robust probabilistically checkable proofs of proximity (PCPPs), and the soundness gap (defined as $1 -$ soundness error) deteriorates multiplicatively with each composition step. This causes the soundness gap to deteriorate quickly even when just a few composition steps are used, and as a result the final soundness gap is smaller than 10^{-7}. This means that the soundness error is at least $1 - 10^{-7}$, and so one must repeat the decoder many times (see Observation 7.11) to amplify the soundness error back down to $1/2$, which makes the query complexity large. In hindsight, the fact that these RLDCs use a large number of queries is perhaps unsurprising given that the constants in their proofs are likely not intentionally optimized.

We prove Theorem 7.14 in Section 7.5.

7.2 Techniques

In this section, we give a proof overview of our main result, Theorem 7.8. We will primarily focus on the case of $q = 2$ and $q = 3$ for RLDCs, though we shall also explain how to generalize the proof to arbitrary q and also to RLCCs. For simplicity, we will assume that the decoder is nonadaptive; Theorem 7.12 handles the case of adaptive decoders.

7.2.1 The proof strategy for Theorem 7.8

Let $C\colon \{0,1\}^k \to \{0,1\}^n$ be a linear $(q, \delta, 1, s)$-RLDC with a nonadaptive decoder $\mathrm{Dec}(\cdot)$ satisfying the properties in Definition 7.5 with $s \leq (1 - \alpha)s(q)$. Our goal is to show that C is a $(q, \delta', 1, s')$-LDC for some δ', s'. To do this, we need to construct a (possibly different) LDC decoder $\mathrm{Dec}'(\cdot)$ using the RLDC decoder $\mathrm{Dec}(\cdot)$. We will do this by "opening up" $\mathrm{Dec}(\cdot)$, i.e., we will crucially *not* use $\mathrm{Dec}(\cdot)$ in a

black-box manner.

LDCs and smooth decoders. The starting point of our proof is the following observation of [KT00].

Observation 7.15. Suppose that C admits a q-query *smooth decoder* $\mathrm{Dec}_{\mathrm{Smooth}}(\cdot)$, which is a nonadaptive local decoder with the following two properties:

(1) (perfect completeness) for every $b \in \{0,1\}^k$ and $i \in [k]$, $\Pr[\mathrm{Dec}^{C(b)}_{\mathrm{Smooth}}(i) = b_i] = 1$, and

(2) (η-smoothness) for every $i \in [k]$ and $j \in [n]$, $\Pr[\mathrm{Dec}_{\mathrm{Smooth}}(i) \text{ queries } j] \leq \frac{1}{\eta n}$.

Then, for any $\varepsilon > 0$, C is a $(q, \eta\varepsilon, 1, \varepsilon)$-LDC.

Indeed, Observation 7.15 follows by a simple union bound, as if there are at most $\eta\varepsilon n$ errors, then the probability that at least one of the q queries made by the decoder is corrupted is at most $\frac{1}{\eta n} \cdot \eta\varepsilon n = \varepsilon$, and if all queries are uncorrupted then the decoder outputs b_i, by perfect completeness. Thus, to prove Theorem 7.8, it suffices to extract a smooth decoder from $\mathrm{Dec}(\cdot)$.

A canonical RLDC decoder. Let us now consider the behavior of $\mathrm{Dec}^y(i)$ for a fixed $i \in [k]$ and some $y \in \{0,1\}^n$. Because $\mathrm{Dec}^y(i)$ is nonadaptive, we can view its operation as a two step process. First, $\mathrm{Dec}^y(i)$ samples a set of queries Q of size $\leq q$ (for simplicity, let us assume that $|Q| = q$) from a query distribution Ω_i over $\binom{[n]}{\leq q}$ that does not depend on y. Then, it reads the values y_j for each $j \in Q$, and outputs the value in $\{0, 1, \perp\}$ of a (possibly randomized) function $f_Q(\{y_j\}_{j \in Q})$ that depends on the set of queries Q and their answers $\{y_j\}_{j \in Q}$. We can now make the following simple observation: since the original decoder $\mathrm{Dec}(\cdot)$ has perfect completeness, there is a canonical choice of the decoding function f_Q for each set $Q \subseteq [n]$. If the values of y on Q read by the decoder are consistent with some codeword $x = C(b)$, i.e., $y|_Q = x|_Q$, then the decoder must output b_i, the i-th bit of the underlying message

in x, by perfect completeness. And, if the values of y on Q are inconsistent with *all* codewords x, then y must have an error in the set Q and so the decoder can safely output $\perp$, as this can only decrease the soundness error of the decoder.

Decomposing the canonical RLDC decoder into smooth and nonsmooth parts. The above observation thus implies that we can view the decoder $\mathrm{Dec}^y(i)$ as being solely determined by the query distribution $\mathcal{Q}_i$. Given a distribution $\mathcal{Q}_i$, we can define the set of "nonsmooth" or "heavy" queries as $H_i \coloneqq \{j \in [n] : \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q] > \frac{q}{\delta n}\}$. Note that for a (δ/q)-smooth decoder, the set H_i is empty. We clearly have $|H_i| \leq \delta n$, as

$$q = \sum_{j \in [n]} \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q] \geq \sum_{j \in H_i} \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q] \geq \frac{q|H_i|}{\delta n}\,.$$

In particular, this means that for any $b \in \{0,1\}^k$, the RLDC decoder $\mathrm{Dec}(i)$ on input i must satisfy $\Pr[\mathrm{Dec}^y(i) \notin \{b_i, \perp\}] \leq s$ for any y where $y_j = C(b)_j$ for all $j \notin H_i$. That is, if we only introduce errors in the "heavy set" H_i, we have introduced a small enough fraction of errors so that the soundness condition of the RLDC decoder still applies.

Call a set $Q \in \binom{[n]}{q}$ "i-smoothable" if one can recover b_i from $x|_{Q \setminus H_i}$ for any codeword $x = C(b)$, and let $p_{i,\mathrm{good}} = \Pr_{Q \leftarrow \mathcal{Q}_i}[Q \text{ is } i\text{-smoothable}]$. Let $\mathcal{Q}_{i,\mathrm{good}}$ be the distribution $\mathcal{Q}_i$ conditioned on the output Q being "i-smoothable", and let $\mathcal{Q}_{i,\mathrm{bad}}$ be $\mathcal{Q}_i$ conditioned on Q being not "i-smoothable". Recall that because there is an optimal canonical decoder, a (nonadaptive) decoder is determined completely by its query distribution. Thus, the set of query distributions $\mathcal{Q}_{i,\mathrm{good}}$ for each $i \in [k]$ defines a decoder that we call $\mathrm{Dec}_{\mathrm{LDC}}$, and similarly $\{\mathcal{Q}_{i,\mathrm{bad}}\}_{i \in [k]}$ defines a decoder $\mathrm{Dec}_{\mathrm{RLDC}}$. Furthermore, for any $y \in \{0,1\}^n$ and any $i \in [k]$, for every output $\sigma \in \{0,1,\perp\}$, it holds that

$$\Pr[\mathrm{Dec}^y(i) = \sigma] = p_{i,\mathrm{good}} \Pr[\mathrm{Dec}^y_{\mathrm{LDC}}(i) = \sigma] + (1 - p_{i,\mathrm{good}}) \Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) = \sigma]\,.$$

In other words, we can view the behavior of $\mathrm{Dec}^y(i)$ as follows: with probability $p_{i,\mathrm{good}}$, it runs $\mathrm{Dec}^y_{\mathrm{LDC}}(i)$, and with probability $1 - p_{i,\mathrm{good}}$ it runs $\mathrm{Dec}^y_{\mathrm{RLDC}}(i)$. The

two decoders $\mathrm{Dec}_{\mathrm{LDC}}$ and $\mathrm{Dec}_{\mathrm{RLDC}}$ have their names chosen to indicate that $\mathrm{Dec}_{\mathrm{LDC}}$ is the "LDC part" or "smooth part" of the original decoder Dec, and $\mathrm{Dec}_{\mathrm{RLDC}}$ is the "pure RLDC part" or "nonsmooth part" of the original decoder.

Showing that the "smooth part" is nontrivial by breaking soundness of the "nonsmooth" part. To finish proving that C is an LDC, it remains to argue that $p_{i,\mathrm{good}} \geq \Omega(1)$ for each $i \in [k]$, i.e., the "smooth part" of the RLDC decoder is nontrivially large. This is because the decoder $\mathrm{Dec}_{\mathrm{LDC}}$ is $(\delta p_{i,\mathrm{good}}/q)$-smooth[7] for each $i \in [k]$, and so by Observation 7.15 it is an LDC decoder that can tolerate $\delta p/q$ errors, where $p = \min_{i\in[k]} p_{i,\mathrm{good}}$. We thus want $p_{i,\mathrm{good}} \geq \Omega(1)$ for each $i \in [k]$, so that $\delta p/q = \Omega(\delta/q) = \Omega(1)$ is at least a constant.

To argue that $p_{i,\mathrm{good}} \geq \Omega(1)$ for each $i \in [k]$, we will fix $i \in [k]$ and we show that any decoder that makes at least one "nonsmooth" query has soundness error at least $s(q) = 2^{-\lfloor q/2 \rfloor}$. More formally, we show that for each $b \in \{0,1\}^k$, there is a $y \in \{0,1\}^n$ where $y_v = C(b)_v$ for $v \notin H_i$ such that $\Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) \notin \{b_i, \perp\}] \geq s(q)$. That is, y agrees with a codeword on the "smooth queries" and only disagrees on some of the "nonsmooth queries", which also implies that $\Delta(y, C(b)) \leq |H_i| \leq \delta n$. Note that by soundness of the original decoder Dec, this implies

$$\begin{aligned}
&(1-\alpha)s(q) \geq s \geq \Pr[\mathrm{Dec}^y(i) \notin \{b_i, \perp\}] \\
&= p_{i,\mathrm{good}} \Pr[\mathrm{Dec}^y_{\mathrm{LDC}}(i) \notin \{b_i, \perp\}] + (1 - p_{i,\mathrm{good}}) \Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) \notin \{b_i, \perp\}] \\
&\geq s(q)(1 - p_{i,\mathrm{good}}) \,,
\end{aligned}$$

and so $p_{i,\mathrm{good}} \geq \alpha$.

We explain how to break soundness of $\mathrm{Dec}_{\mathrm{RLDC}}$ in the next section.

[7]Here, we sample Q from $\mathcal{Q}_{i,\mathrm{good}}$ and only query $Q \setminus H_i$. Note that because Q is i-smoothable, we can still decode b_i from $Q \setminus H_i$, and so we have preserved perfect completeness while making the decoder smooth.

7.2.2 Analyzing the "nonsmooth" linear RLDC decoder

It remains to show that $\mathrm{Dec}_{\mathrm{RLDC}}$ has soundness error at least $s(q) = 2^{-\lfloor q/2 \rfloor}$. As explained above, we will show that for each $b \in \{0,1\}^k$, there exists $y \in \{0,1\}^n$ that agrees with $x = C(b)$ on all indices in $[n] \setminus H_i$ such that $\Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) = 1 - b_i] \geq s(q)$. Because the code is linear, without loss of generality we may assume that $b = 0^k$, so that $x = 0^n$, and our goal is to show the existence of such a $y \in \{0,1\}^n$ where $\Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) = 1] \geq s(q)$.

Fooling a fixed set Q. Let us now explain how to construct such a y. Fix a set Q in the support of the query distribution of $\mathrm{Dec}_{\mathrm{RLDC}}$. Because we have an optimal canonical decoder, when $\mathrm{Dec}_{\mathrm{RLDC}}$ queries Q, it simply checks if $y|_Q = x|_Q$ for some codeword x, and if so, then it must output the bit b_i where $x = C(b)$.

As a first attempt, suppose we choose y so that $y|_{H_i}$ is random and otherwise y agrees with 0^n, the original uncorrupted codeword. Then, with probability at least 2^{-q}, it holds that $y|_Q = x|_Q$ for some codeword $x = C(b)$ with $b_i = 1$. Indeed, because Q is not "i-smoothable", the message bit b_i cannot be recovered from $x|_{Q \setminus H_i}$ for a codeword $x = C(b)$, and so there must exist a codeword $x = C(b)$ with $b_i = 1$ such that $x|_{Q \setminus H_i}$ agrees with the codeword 0^n. And, since $y|_{H_i}$ is random, we have that $y|_{Q \cap H_i} = x|_{Q \cap H_i}$ with probability at least $2^{-|Q \cap H_i|} \geq 2^{-q}$. We note that this observation is sufficient to prove Theorem 7.8 with $s(q) = 2^{-q}$.

To prove Theorem 7.8 with $s(q) = 2^{-\lfloor q/2 \rfloor}$, however, we require a more sophisticated analysis that will end up using the linearity of C quite strongly. For a linear code C, $y|_Q$ agrees with a codeword x on Q if and only if $y|_Q$ satisfies a certain system of linear equations that are the "local parity check constraints on Q". For example, it could be the case that $Q = \{j_1, j_2, j_3\}$ where any codeword x satisfies $x_{j_1} + x_{j_2} + x_{j_3} = 0$, and there are no other local constraints. In this case, if y satisfies $y_{j_1} + y_{j_2} + y_{j_3} = 0$, then there exists some codeword x where $x_{j_1} = y_{j_1}$, $x_{j_2} = y_{j_2}$, and $x_{j_3} = y_{j_3}$.

However, the above observation is not enough for us, as we additionally need to make $\mathrm{Dec}_{\mathrm{RLDC}}$ output 1. To make this happen, we will again make use of the linear structure of C. Because $C\colon \{0,1\}^k \to \{0,1\}^n$ is a linear map, for every $j \in [n]$ there exists $v_j \in \{0,1\}^k$ such that for every $b \in \{0,1\}^k$, $C(b)_j = \langle b, v_j \rangle$; the vector v_j is simply the j-th row of the generator matrix of C. Note that "local constraints" above correspond to linear dependencies among the rows of the generator matrix. That is, $x_{j_1} + x_{j_2} + x_{j_3} = 0$ for all codewords x if and only if $v_{j_1} + v_{j_2} + v_{j_3} = 0^k$.

The fact that $\mathrm{Dec}_{\mathrm{RLDC}}$ has perfect completeness implies that one can recover b_i exactly from $x = C(b)$ restricted to Q. In linear algebraic terms, this means that the i-th standard basis vector e_i is in $\mathrm{span}(\{v_j : j \in Q\})$. We have assumed that Q is not "i-smoothable", meaning that one *cannot* recover b_i from $x|_{Q \setminus H_i}$, as otherwise Q is a set that "belongs to" $\mathrm{Dec}_{\mathrm{LDC}}$. So, $e_i \notin \mathrm{span}(\{v_j : j \in Q \setminus H_i\})$.

Our main technical lemma (Lemma 7.19) shows that if we pick y such that: (1) $y_j = C(0^k)_j = 0$ for $j \notin H_i$, and (2) $y_j = C(b)_j$ for $j \in H_i$ where $b \in \{0,1\}^k$ is *random* with $b_i = 1$, then with probability at least $s(q) = 2^{-\lfloor q/2 \rfloor}$, $y|_Q = x|_Q$ for some codeword $x = C(b')$ where $b'_i = 1$. That is, $y|_Q$ satisfies all local checks in Q and is consistent with a codeword that has i-th message bit equal to 1, and thus when this occurs $\mathrm{Dec}_{\mathrm{RLDC}}$ outputs 1.

Remark 7.16. The intuition for our main technical lemma (and indeed the main intuition behind Theorem 7.8) is as follows. Firstly, in order for the RLDC decoder to be "not LDC-like", it needs to make queries to the nonsmooth set H_i. Secondly, in order for the RLDC decoder to be hard to fool, it must use the queries to $Q \setminus H_i$ to check consistency with the queries $Q \cap H_i$; if there are no consistency checks, then we can freely choose the values of y on $Q \cap H_i$ and easily fool the decoder. Finally, when q is small (say $q = 2$ or 3), the decoder cannot simultaneously make many queries to both H_i and $[n] \setminus H_i$, i.e., one of $Q \cap H_i$ or $Q \setminus H_i$ must be small, and this is an inherent weakness that allows us to fool $\mathrm{Dec}_{\mathrm{RLDC}}$ with reasonable probability.

Casework for $q = 2$. Let us now explain a simple argument to prove our main

technical lemma when $q = 2$. Let $Q = \{j_1, j_2\}$. We split the analysis into 3 cases, depending on $|Q \cap H_i|$, i.e., the number of "nonsmooth queries" made in Q. As we shall see, the case of $q = 2$ is simple and also somewhat degenerate, in that it is not possible to have a nontrivial "local check". In fact, this is the reason that we can show a very good lower bound for 2-RLDCs over large fields (see Section 7.6).

(1) Case 1: $|Q \cap H_i| = 0$. In this case, $Q \cap H_i = \emptyset$, and therefore e_i is in $\text{span}(\{v_j\}_{j \in Q \setminus H_i})$, contradicting the fact that Q is not i-smoothable. Thus, this case cannot occur.

(2) Case 2: $|Q \cap H_i| = 1$. In this case, $Q = \{j_1, j_2\}$ where $j_1 \in H_i$ and $j_2 \notin H_i$. We have that $e_i \notin \text{span}(v_{j_2})$, which implies that $v_{j_2} \neq e_i$. We also have $e_i \in \text{span}(v_{j_1}, v_{j_2})$, which implies that either $e_i = v_{j_1}$, or $e_i = v_{j_1} + v_{j_2}$.

At this point, either there are no local check constraints, or there is a single constraint $v_{j_1} + v_{j_2} = 0^k$. However, the latter leads to a contradiction, as both of the possibilities for e_i are now impossible:

- $v_{j_1} = e_i$ would imply $v_{j_2} = e_i$, and hence $e_i \in \text{span}(v_{j_2})$ (contradiction);
- $v_{j_1} + v_{j_2} = e_i$ would imply $e_i = 0^k$ (contradiction).

Because $j_2 \notin H_i$, we have $y_{j_2} = 0$, and so we fool the decoder if and only if $y_{j_1} = 1$. This clearly happens with probability at least $1/2$, as either $v_{j_1} = e_i$, in which case $y_{j_1} = 1$ with probability 1, or else $v_{j_2} \neq e_i$ and is nonzero, in which case $y_{j_1} = 1$ with probability $1/2$.

(3) Case 3: $|Q \cap H_i| = 2$. In this case, $y|_Q = x|_Q$ for some $x = C(b')$ where $b'_i = 1$, by definition. Thus, the canonical decoder outputs 1 with probability 1.

In all cases, we see that the decoder outputs 1 with probability at least $s(2) = 1/2$ over the random choice of y, as required.

Generalizing to higher q. The above casework proves the desired claim for $q = 2$. One can repeat a similar case-by-case analysis for $q = 3$, but the analysis quickly becomes unwieldy as q increases. For example, when $q = 3$ one can have $Q = \{j_1, j_2, j_3\}$ where $j_1, j_2 \in H_i$, $j_3 \notin H_i$, and $v_{j_1} + v_{j_2} = e_i$ and $v_{j_2} + v_{j_3} = 0$. That is, we can have both a "decoding constraint" that sums to e_i and a "check constraint" that sums to 0. Again, one can verify that our choice of random y will satisfy both these constraints with probability 1/2: this is because the two constraints imply $v_{j_1} + v_{j_2} = e_i$, and this constraint is satisfied with probability 1 (because y is consistent with $x = C(b')$ with $b'_i = 1$ on indices in H_i), and the constraint $v_{j_2} + v_{j_3} = 0$ is satisfied with probability 1/2, independently of the first constraint. We note that it is easy to show how to set the values y_j for $j \in H_i$ to fool the decoder for a particular Q, but the key difficulty is we need to find a single global y that fools an $s(q)$-fraction of the Q's simultaneously.

To find a single global y, we show that if we sample y from the distribution as above, i.e., where $y|_{H_i}$ is chosen to be $C(b)|_{H_i}$ where b is uniformly random with $b_i = 1$, then for any set Q that is not i-smoothable, with probability at least $s(q)$ over the draw of y, y fools the decoder when it queries Q. Recall that y fools the decoder if and only if $y|_Q \in C'|_Q$, where C' is the affine subspace $\{C(b)|_{Q \cap H_i} : b_i = 1\}$. This requires checking that (1) $y|_{Q \cap H_i}$ is in $C'|_{Q \cap H_i}$, (2) $y|_{Q \setminus H_i}$ is in $C'|_{Q \setminus H_i}$, and (3) $y|_{Q \cap H_i}$ is consistent with $0^{Q \setminus H_i}$ and the constraint that $b_i = 1$. Notice that y satisfies all of the type (1) constraints with probability 1, since $y|_{Q \cap H_i}$ is drawn uniformly over that set, and y satisfies all of the type (2) constraints by definition because Q is not i-smoothable. We can view the type (3) constraints as imposing a system of inhomogeneous linear constraints on the vector $y|_{Q \cap H_i}$, so that y fools the decoder if and only if $y|_{Q \cap H_i}$ lies in a particular affine subspace.

Formally, we show that for each Q and each codeword $C(b)$, there is a vector $z \in \{0,1\}^{Q \cap H_i}$ and linear subspaces $\mathcal{V}$ and $\mathcal{W}$ in $\{0,1\}^{Q \cap H_i}$ with $\mathcal{V} \subseteq \mathcal{W}$ and $\dim(\mathcal{W}/\mathcal{V}) \leq \min(|Q \cap H_i|, |Q \setminus H_i|)$ such that $y|_Q$ fools the decoder if and only if $y|_{Q \cap H_i} - z$ lies in $\mathcal{W}^\perp$. Intuitively, the vector z shifts the affine subspaces to be linear

subspaces, and then the subspace $\mathcal{W}$ is the set of "local parity check constraints" on $Q \cap H_i$ satisfied by all codewords $C(b')$ with $b'_i = 0$ that are also 0 on $Q \setminus H_i$ (type (1) and type (3) constraints), and the subspace $\mathcal{V}$ is the set of constraints on $Q \cap H_i$ satisfied by all codewords $C(b')$ with $b'_i = 0$ (type (1) constraints only). The distribution of $y|_{Q \cap H_i} - z$ is then uniform over the larger subspace $\mathcal{V}^\perp$ in $\{0,1\}^{Q \cap H_i}$ that contains $\mathcal{W}^\perp$. Hence, y lies in the desired affine subspace with probability at least $2^{-\dim(\mathcal{W}/\mathcal{V})}$ and this is at least $2^{-\lfloor q/2 \rfloor}$, as one can show that $\dim(\mathcal{W}/\mathcal{V}) \leq \min(|Q \cap H_i|, |Q \setminus H_i|) \leq \lfloor q/2 \rfloor$. This is the "inherent weakness" mentioned in Remark 7.16: one of $|Q \cap H_i|$ or $|Q \setminus H_i|$ must have size $\leq \lfloor q/2 \rfloor$, and they control the dimension of the "extra local checks" that $y|_{Q \cap H_i}$ must satisfy in order to fool the decoder.

The full proof of Theorem 7.8 is in Section 7.3.

Generalizing to RLCCs. The above analysis considers the behavior of $\mathrm{Dec}^y(i)$ on each input i separately, and shows how to convert it to a smooth decoder. Thus, it seamlessly generalizes to RLCCs, as we can consider the behavior of the RLCC decoder $\mathrm{Dec}^y(u)$ on each input u separately as well, and convert $\mathrm{Dec}^y(u)$ to a smooth decoder for each u to obtain an LCC.

7.3 Relaxed Locally Decodable Codes Cannot Have Strong Soundness

In this section, we prove Theorem 7.8, which shows a qualitative difference between linear RLDCs and LDCs: an LDC has strong soundness (soundness error can be made arbitrarily low by adjusting the decoding radius; see Section 7.7), while an RLDC does not unless it is also an LDC. We will in fact prove the following theorem, which is a generalization of Theorem 7.8 to any finite field for the case of nonadaptive decoders. The case of adaptive decoders and imperfect completeness is handled generically by Theorem 7.23, which we prove in Section 7.4.

Theorem 7.17 (Theorem 7.8 for nonadaptive decoders and any field)**.** *Let* $s_{\mathbb{F}}(q) \coloneqq |\mathbb{F}|^{-\lfloor q/2 \rfloor}$. *Let* $C \colon \mathbb{F}^k \to \mathbb{F}^n$ *be a linear* $(q, \delta, 1, s)$*-RLDC with a nonadaptive decoder,*

where $s \leq (1-\alpha)s_{\mathbb{F}}(q)$. Then, for any $\varepsilon > 0$, C is a linear $(q, \alpha\delta\varepsilon/q, 1, \varepsilon)$-LDC. Furthermore, the same result holds for RLCCs/LCCs.

Proof. Let $C\colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear $(q, \delta, 1, s)$-RLDC with $s \leq (1-\alpha)s_{\mathbb{F}}(q)$. We will follow the proof outline from Section 7.2.

Let $\mathrm{Dec}(\cdot)$ be the nonadaptive RLDC decoder of C. Because $\mathrm{Dec}(\cdot)$ is nonadaptive, we may assume (Fact 5.15) that it behaves as follows. For each i, there is a distribution $\mathcal{Q}_i$ over subsets of $[n]$ of size at most q, and the decoder $\mathrm{Dec}^y(i)$ on input i simply samples $Q \leftarrow \mathcal{Q}_i$, reads y_v for each $v \in Q$, and then follows the behavior of the canonical decoder in Fact 5.15.

Defining the smooth decoder $\mathbf{Dec_{LDC}}(\cdot)$. Let $i \in [k]$ be a message index. For each $i \in [k]$, we partition the codeword indices $[n]$ into heavy ("nonsmooth") and light ("smooth") indices as follows.

$$H_i \coloneqq \{j \in [n] : \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q] > q/\delta n\}$$
$$L_i \coloneqq \{j \in [n] : \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q] \leq q/\delta n\}$$

We then have

$$q \geq \mathbb{E}_{Q \leftarrow \mathcal{Q}^{(i)}}\Big[\sum_{j \in [n]} \mathbf{1}(j \in Q)\Big] \geq \sum_{j \in H_i} \mathbb{E}_{Q \leftarrow \mathcal{Q}^{(i)}}[\mathbf{1}(j \in Q)] \geq \frac{q|H_i|}{\delta n}.$$

Hence, $|H_i| \leq \delta n$.

We now split $\mathcal{Q}_i$ into the "smooth part" and the "nonsmooth part". Recall from the definition of a linear code (Definition 5.4) that for each $j \in [n]$, there exists $v_j \in \mathbb{F}^k$ such that for every $b \in \mathbb{F}^k$, $C(b)_j = \langle v_j, b\rangle$. We now define the notion of a "smoothable" query set, which are sets Q where one can recover b_i from only the light indices, i.e., $Q \cap L_i$.

Definition 7.18. For a set Q in the support of $\mathcal{Q}_i$, we call Q *i-smoothable* if $e_i \in \mathrm{span}\{v_j : j \in Q \cap L_i\}$.

Let us now define the following query distribution $\tilde{\mathcal{Q}}_i$. In the distribution $\tilde{\mathcal{Q}}_i$, we sample $Q \leftarrow \mathcal{Q}_i$ conditioned on Q being i-smoothable (we will show that this probability is nonzero, so this is well-defined), and then we output $\tilde{Q} \coloneqq Q \cap L_i$. Note that $\tilde{Q}$ is i-smoothable since Q is.

Given the distribution $\tilde{\mathcal{Q}}_i$ for each $i \in [k]$, we define a decoder $\mathrm{Dec}_{\mathrm{LDC}}(\cdot)$ as follows. On input i, $\mathrm{Dec}^y_{\mathrm{LDC}}(i)$ draws $\tilde{Q} \leftarrow \tilde{\mathcal{Q}}$, and then uses the "decoding constraint" to decode b_i. That is, because $\tilde{Q}$ is i-smoothable, there exists a subset $T \subseteq \tilde{Q}$ such that $\sum_{j \in T} v_j = e_i$, and the decoder outputs $\sum_{j \in T} y_j$.

Arguing smoothness of $\mathrm{Dec}_{\mathrm{LDC}}(\cdot)$. We will now show that $\mathrm{Dec}_{\mathrm{LDC}}(\cdot)$ is $(\alpha\delta/q)$-smooth (Definition 5.10). For each $i \in [k]$, let $p_{i,\mathrm{good}}$ be the probability that $Q \leftarrow \mathcal{Q}_i$ is i-smoothable. Let $\mathrm{Dec}_{\mathrm{RLDC}}(\cdot)$ denote the decoder that (1) samples $Q \leftarrow \mathcal{Q}_i$ conditioned on Q being *not* i-smoothable, and then (2) decodes using the canonical decoder. Observe that we can view the original decoder as simply calling $\mathrm{Dec}_{\mathrm{RLDC}}(\cdot)$ with probability $1 - p_{i,\mathrm{good}}$, where $i \in [k]$ is the input index (and, with probability $p_{i,\mathrm{good}}$, the decoder does something else). We then have that for any $b \in \mathbb{F}^k$ and $y \in \mathbb{F}^n$ with $\Delta(y, C(b)) \le \delta n$ and any $i \in [k]$, it holds that

$$(1-\alpha)s_{\mathbb{F}}(q) \ge s \ge \Pr[\mathrm{Dec}^y(i) = \sigma \in \mathbb{F} \backslash \{b_i\}] \ge (1-p_{i,\mathrm{good}}) \Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) = \sigma \in \mathbb{F} \backslash \{b_i\}]\,. \tag{7.1}$$

Our main technical lemma is the following lemma, which we will use to lower bound the soundness error of $\mathrm{Dec}_{\mathrm{RLDC}}$.

Lemma 7.19 (Fooling a set Q)**.** *Let $C \colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear code. Let $Q = H \cup L$ where H and L are disjoint. Let $v^* \in \mathbb{F}^k$, and suppose that $v^* \notin \mathrm{span}(\{v_j : j \in L\})$. Let $b \in \mathbb{F}^k$, and let $x = C(b)$. Fix $i \in [k]$, and let $\sigma \in \mathbb{F}$.*

Let $b' \leftarrow \mathbb{F}^k$ be sampled uniformly at random with $\langle v^*, b' \rangle = \sigma$, and let $x' = C(b')$. Define $z \in \mathbb{F}^Q$ to be $z_j = x_j$ for $j \in L$ and $z_j = x'_j$ for $j \in H$. Then, with probability at least $|\mathbb{F}|^{-\min(|H|,|L|)}$ over the choice of b', there exists $b'' \in \mathbb{F}^k$ such that $C(b'')|_Q = z$ and $\langle v^*, b'' \rangle = \sigma$.

Lemma 7.19 says the following. Suppose we are given a query set Q and a codeword $x = C(b)$, and we want to corrupt $x|_Q$ by only modifying its values on H_i so that the corrupted version of $x|_Q$ is now equal to $x''|_Q$ for a different codeword x''. Then, if we corrupt $x|_Q$ by replacing $x|_{Q \cap H_i}$ with a uniformly random codeword, then it will satisfy this condition with some nontrivial probability. Furthermore, if we additionally wish the codeword x'' to be equal to $C(b'')$ for some b'' satisfying a particular inhomogeneous linear constraint $\langle v^*, b'' \rangle = \sigma$, then this is possible provided that $v^* \notin \mathrm{span}(\{v_j : j \in Q \cap L_i\})$ and $v^* \in \mathrm{span}(\{v_j : j \in Q\})$.

We postpone the proof of Lemma 7.19 to Section 7.3.1, and for now use it to finish the proof of Theorem 7.17. As we shall shortly see, Lemma 7.19 implies that the soundness error of $\mathrm{Dec}_{\mathrm{RLDC}}$ is at least $s_{\mathbb{F}}(q)$.

Indeed, let $v^* = e_i$, and let Q be any set in the support of $\mathcal{Q}_i$ that is not i-smoothable. Let $b \in \mathbb{F}^k$ and let $x = C(b)$. Suppose that we define (a distribution over) $y \in \mathbb{F}^n$ with $\Delta(y, C(b)) \leq \delta n$ by (1) sampling $b' \leftarrow \mathbb{F}^k$ with $b'_i \neq b_i$, and (2) setting y to be $y|_{H_i} = C(b')|_{H_i}$, and $y|_{L_i} = C(b)|_{L_i}$. We clearly have that $\Delta(y, C(b)) \leq |H_i| \leq \delta n$ holds with probability 1. On the other hand, by Lemma 7.19, with probability at least $|\mathbb{F}|^{-\min(|Q \cap H_i|, |Q \cap L_i|)} \geq |\mathbb{F}|^{-\lfloor q/2 \rfloor} = s_{\mathbb{F}}(q)$, it holds that $y|_Q = x''|_Q$ for some $x'' = C(b'')$ where $b''_i = b'_i \neq b_i$. Because $\mathrm{Dec}_{\mathrm{RLDC}}$ behaves as the canonical decoder, it must therefore output $b''_i \neq b_i$.

The above shows that for any $i \in [k]$, there is a distribution over y with $\Delta(y, C(b)) \leq \delta n$ such that $\mathbb{E}_y[\Pr[\mathrm{Dec}^y_{\mathrm{RLDC}}(i) \neq b_i]] \geq s_{\mathbb{F}}(q)$. Hence, by averaging, there exists y such that this holds. This implies that the soundness error of $\mathrm{Dec}^y_{\mathrm{RLDC}}(i)$ is at least $s_{\mathbb{F}}(q)$, and so by Eq. (7.1), we conclude that $(1-\alpha)s_{\mathbb{F}}(q) \geq (1-p_{i,\mathrm{good}})s_{\mathbb{F}}(q)$, which implies that $p_{i,\mathrm{good}} \geq \alpha$, and this holds for all $i \in [k]$.

With this lower bound on $p_{i,\mathrm{good}}$ in hand, let us now argue smoothness of $\mathrm{Dec}_{\mathrm{LDC}}(\cdot)$. Indeed, for any $j \in [n]$, we have that

$$\Pr[\mathrm{Dec}^y_{\mathrm{LDC}}(i) \text{ queries } j] = \Pr_{\tilde{Q} \leftarrow \tilde{\mathcal{Q}}_i}[j \in \tilde{Q}] = \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q \cap L_i \mid Q \text{ is } i\text{-smoothable}].$$

Notice that this probability is 0 if $j \in H_i$, and otherwise for $j \in L_i$ we have

$$\begin{aligned}
\Pr[\mathrm{Dec}^{y}_{\mathrm{LDC}}(i) \text{ queries } j] &= \Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q \mid Q \text{ is } i\text{-smoothable}] \\
&= \frac{\Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q \wedge Q \text{ is } i\text{-smoothable}]}{\Pr_{Q \leftarrow \mathcal{Q}_i}[Q \text{ is } i\text{-smoothable}]} \\
&= \frac{\Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q \wedge Q \text{ is } i\text{-smoothable}]}{p_{i,\mathrm{good}}} \\
&\leq \frac{\Pr_{Q \leftarrow \mathcal{Q}_i}[j \in Q]}{p_{i,\mathrm{good}}} \\
&\leq \frac{q}{\alpha\delta n},
\end{aligned}$$

where we use that $j \in L_i$. Hence, $\mathrm{Dec}_{\mathrm{LDC}}$ is $(\alpha\delta/q)$-smooth. By Fact 5.12, this implies that C is a $(q, \alpha\delta\varepsilon/q, 1, \varepsilon)$-LDC with decoder $\mathrm{Dec}_{\mathrm{LDC}}$, which proves Theorem 7.17 for the case of RLDCs.

Extension to the RLCC case. Let us now briefly explain how to extend the above proof to the case of RLCCs. As we are proceeding by analyzing the behavior of $\mathrm{Dec}(\cdot)$ on each input, this generalizes seamlessly to RLCCs. The only difference in the proof is that in Lemma 7.19, we take $v^* = v_j$, where v_j is the j-th row of the generator matrix of C and $j \in [n]$ is the input to $\mathrm{Dec}(\cdot)$. That is, v^* is no longer necessarily a standard basis vector e_i. All the remaining steps of the proof are unchanged. □

7.3.1 Proof of Lemma 7.19

In this subsection, we prove Lemma 7.19.

Proof of Lemma 7.19. Let $Q \subseteq [n]$, and let $Q = H \cup L$ where $H \cap L = \emptyset$. Let $b \in \mathbb{F}^k$, and let $x = C(b)$. Fix $i \in [k]$, and let $\sigma \in \mathbb{F}$. Our goal is to show that, if we choose $b' \leftarrow \mathbb{F}^k$ uniformly at random with $\langle v^*, b' \rangle = \sigma$, then with probability at least $|\mathbb{F}|^{-\min(|H|,|L|)}$, the string $z \in \mathbb{F}^Q$ defined as $z_j = x_j$ for $j \in L$ and $z_j = x'_j$ for $j \in H$ is consistent with some other codeword $C(b'')$ that also satisfies $\langle v^*, b'' \rangle = \sigma$.

Because C is linear, without loss of generality we may assume that $b = 0^k$, so

that $z_j = 0$ for all $j \in L$. We wish to argue that $\Pr_z[\exists b'' \text{ s.t. } z = C(b'')|_Q \wedge \langle v^*, b'' \rangle = \sigma] \geq |\mathbb{F}|^{-\min(|H|,|L|)}$, when z is drawn from the distribution defined in Lemma 7.19.

Let us first argue that such a z exists. This implies that the probability is at least $|\mathbb{F}|^{-|Q|}$, which is already sufficient to prove Theorem 7.17 with $s_{\mathbb{F}}(q) = |\mathbb{F}|^{-q}$.

Indeed, because $v^* \notin \text{span}\{v_j : j \in L\}$ and there is a codeword that is identically 0 on all $j \in L$, it follows by Fact 5.7 that such a z exists. Let $z^* = C(b^*)|_Q$ be any such solution, with corresponding codeword b^*.

Because there exists such a b^*, by the linearity of C, it suffices to show that $\Pr_{z \leftarrow \mathcal{D}}[\exists b'' \text{ s.t. } z = C(b'')|_Q \wedge \langle v^*, b'' \rangle = 0] \geq |\mathbb{F}|^{-\min(|H|,|L|)}$, where $\mathcal{D}$ is the distribution over z given by (1) sampling $b' \leftarrow \mathbb{F}^k$ uniformly at random with $\langle v^*, b' \rangle = 0$, and (2) outputting $z \in \mathbb{F}^Q$ where $z_j = 0$ for $j \in L$ and $z_j = C(b')_j$ for $j \in H$. This is because any z satisfies the above condition if and only if $z + z^*$ satisfies the conditions in Lemma 7.19.

Let us now define the following vector spaces: $\mathcal{V}_0 = \{C(b') : \langle v^*, b' \rangle = 0\}$, $\mathcal{W}_0 = \{C(b') : \langle v^*, b' \rangle = 0, C(b')_j = 0 \ \forall j \in L\}$. Note that $\mathcal{W}_0 \subseteq \mathcal{V}_0$, and these are both linear subspaces. We let $\mathcal{V}_0^\perp$ and $\mathcal{W}_0^\perp$ denote their corresponding dual subspaces (Definition 5.5).

By Fact 5.6, $(\mathcal{V}_0^\perp)_{\subseteq H}$ are the set of local constraints defining $\mathcal{V}_0|_H$, and similarly for $(\mathcal{W}_0^\perp)_{\subseteq H}$ and $\mathcal{W}_0|_H$. We will refer to these subspaces often, and so we let $\mathcal{V} \coloneqq (\mathcal{V}_0^\perp)_{\subseteq H}$ and $\mathcal{W} \coloneqq (\mathcal{W}_0^\perp)_{\subseteq H}$.

A key fact that we will use is that $((\mathcal{V}_0^\perp)_{\subseteq Q})|_H = \mathcal{W}$. This is immediate by definition, as $(\mathcal{V}_0^\perp)_{\subseteq Q}$ is the set of local constraints that defines $\mathcal{V}_0|_Q$, and when we enforce $C(b')_j = 0$ for all $j \in L$, the restriction of any constraint in $\mathcal{V}_0|_Q$ to the set H is a local constraint in $\mathcal{W}_0$ on the subset $H = Q \setminus L$. For notational convenience, we will let $\mathcal{U} = (\mathcal{V}_0^\perp)_{\subseteq Q}$, so that $\mathcal{U}|_H = \mathcal{W}$, and $\mathcal{U}_{\subseteq H} = \mathcal{V}$.

We now show the following two claims, which together imply Lemma 7.19.

Claim 7.20. $\Pr_{z \leftarrow \mathcal{D}}[\exists b'' \text{ s.t. } z = C(b'')|_Q \wedge \langle v^*, b'' \rangle = 0] \geq |\mathbb{F}|^{-\dim(\mathcal{W}/\mathcal{V})}$.

Claim 7.21. $\dim(\mathcal{W}/\mathcal{V}) \leq \min(|H|, |L|)$.

Indeed, Claims 7.20 and 7.21 imply that $\Pr_{z \leftarrow \mathcal{D}}[\exists b'' \text{ s.t. } z = C(b'')|_Q \wedge \langle v^*, b'' \rangle = 0] \geq |\mathbb{F}|^{-\min(|H|,|L|)}$, which we have already shown suffices to prove Lemma 7.19. $\square$

Proof of Claim 7.20. By definition of the subspace $\mathcal{V}_0$, we have that $w := z|_H$ is uniformly distributed over $\mathcal{V}_0|_H$. By definition of the subspace $\mathcal{V}_0$, we have that z satisfies the desired condition if and only if $w \in \mathcal{W}_0$. Thus, $\Pr_{w \leftarrow \mathcal{V}_0|_H}[w \in \mathcal{W}_0|_H] \geq |\mathbb{F}|^{-t}$ where t is the number of "independent checks" in $\mathcal{W}$ that are not in $\mathcal{V}$. We have that $t = \dim(\mathcal{W}/\mathcal{V})$, which gives us the claim.

More formally, let $r^{(1)}, \ldots, r^{(d)}$ be vectors in $\mathcal{W}$ that are linearly independent in $\mathcal{W}/\mathcal{V}$ (and hence also linearly independent in $\mathcal{W}$). By definition of $\mathcal{W}/\mathcal{V}$, any $r \in \mathcal{W}$ can be expressed as $s + \sum_{j \in T} r^{(j)}$, where $s \in \mathcal{V}$. Now, w satisfies $\langle w, s \rangle = 0$, and hence if $\langle w, r^{(j)} \rangle = 0$ holds for all $j \in [d]$, then $\langle w, r \rangle = 0$ for all $r \in \mathcal{W}$. Finally, observe that because $r^{(1)}, \ldots, r^{(t)}$ are linearly independent in $\mathcal{W}/\mathcal{V}$, the elements $\langle w, r^{(j)} \rangle$ are independent and uniformly random from $\mathbb{F}$ when $w \leftarrow V_0$. Hence, the probability that $\langle w, r^{(j)} \rangle = 0$ for all $j \in [t]$ is at least $|\mathbb{F}|^{-t}$. $\square$

Proof of Claim 7.21. Because $\mathcal{V}, \mathcal{W}$ are subspaces in $\mathbb{F}^H$, it follows that $\dim(\mathcal{W}/\mathcal{V}) \leq \dim(\mathcal{W}) \leq |H|$. Thus, it remains to prove that $\dim(\mathcal{W}/\mathcal{V}) \leq |L|$, which is the nontrivial case.

This proof uses the following key fact that we established earlier: $\mathcal{U}|_H = \mathcal{W}$ and $\mathcal{U}_{\subseteq H} = \mathcal{V}$, where $\mathcal{U} \subseteq \mathbb{F}^Q$ is a subspace.

Suppose that $\dim(\mathcal{W}/\mathcal{V}) \geq |L| + 1$. Let $d = |L| + 1$, and let $r^{(1)}, \ldots, r^{(d)}$ be elements of $\mathcal{W}$ that are linearly independent in $\mathcal{W}/\mathcal{V}$. Because $\mathcal{W} = \mathcal{U}|_H$, for each $j \in [t]$, there exist $s^{(1)}, \ldots, s^{(d)} \in \mathbb{F}^Q$ with $\text{supp}(s^{(j)}) \subseteq L$ such that $r^{(j)} + s^{(j)} \in \mathcal{U}$ and $\text{supp}(r^{(j)} + s^{(j)}) \subseteq H$. Now, because $s^{(1)}, \ldots, s^{(d)}$ are in $\mathbb{F}^Q$ and have support contained in L, they lie in a subspace of dimension at most $|L|$, and since $d = |L| + 1$,

they must be linearly dependent. Hence, there exist $\alpha_1, \dots \alpha_d \in \mathbb{F}$, not all zero, such that $\sum_{j=1}^{d} \alpha_j s^{(j)} = 0$ in $\mathbb{F}^Q$. We then have that

$$\sum_{j=1}^{d} \alpha_j r^{(j)} = \sum_{j=1}^{d} \alpha_j (r^{(j)} + s^{(j)}) - \sum_{j=1}^{d} \alpha_j s^{(j)} = \sum_{j=1}^{d} \alpha_j (r^{(j)} + s^{(j)}) \,.$$

Now, because $\operatorname{supp}(r^{(j)} + s^{(j)}) \subseteq H$ for all j, this implies that $\operatorname{supp}(\sum_{j=1}^{d} \alpha_j (r^{(j)} + s^{(j)})) \subseteq H$ also, and therefore $\sum_{j=1}^{t} \alpha_j (r^{(j)} + s^{(j)})$ is an element of $\mathcal{U}$ whose support is contained in H, i.e., an element of $\mathcal{U}_{\subseteq H} = \mathcal{V}$. Hence, $\sum_{j=1}^{d} \alpha_j (r^{(j)} + s^{(j)}) \in \mathcal{V}$. It thus follows that $\sum_{j=1}^{d} \alpha_j (r^{(j)} + s^{(j)})$ is 0 in $\mathcal{W}/\mathcal{V}$, and hence $\sum_{j=1}^{d} \alpha_j r^{(j)}$ is also 0 in $\mathcal{W}/\mathcal{V}$. Therefore, $r^{(1)}, \dots, r^{(d)}$ are linearly independent in $\mathcal{W}/\mathcal{V}$, which proves the claim. □

7.4 Query-Preserving Goldberg Transformation

Up to this point, we have assumed that our RLDC or RLCC has a local decoder/corrector which has *perfect completeness* (always returns the right answer for a valid codeword) and which is *nonadaptive* (the local view is sampled before any queries have been made). We may assume that such a decoder has a *canonical* behavior, as we showed in Fact 5.15, and we rely on this structure in our proofs. However, what if we begin with an RLDC or RLCC with a local decoder that has imperfect completeness, adaptivity, or both? Goldberg's transformation [Gol23a] shows that such a decoder can be transformed (potentially inefficiently) into a nonadaptive decoder with perfect completeness:

Theorem 7.22 ([Gol23a])**.** *Every linear RLDC or RLCC*[8] *has a nonadaptive decoder with perfect completeness:*

1. *If $C \colon \mathbb{F}^k \to \mathbb{F}^n$ is a linear systematic $(q, \delta, 1 - \varepsilon, s)$-RLDC, then C is also a $(q + 1, \delta, 1, s + \varepsilon)$-RLDC with a nonadaptive decoder.*

[8] [Gol23a] only proves this theorem for binary codes, but the proof easily extends to all finite fields.

2. *If* $C\colon \mathbb{F}^k \to \mathbb{F}^n$ *is a linear* $(q, \delta, 1-\varepsilon, s)$*-RLCC, then* C *is also a* $(q+1, \delta, 1, s+\varepsilon)$*-RLCC with a nonadaptive decoder.*

We can use this theorem to lift our lower bound to general linear RLDCs and RLCCs. However, the extra query (from q to $q+1$) is very costly for our purposes. In particular, it would imply that the lower bound in Corollary 7.7 only applies to *2-RLDCs* with imperfect completeness, rather than 3-RLDCs. As discussed in Section 7.1, this extra query has a substantial impact on our results. To avoid losing this query, we give an modified analysis of the main result of [Gol23a] that does not lose this additional query. In doing so, we will lose slightly in the soundness error. We also do not need C to be systematic, which was required in [Gol23a].

Theorem 7.23 (General form of Theorem 7.12)**.** *If* $C\colon \mathbb{F}^k \to \mathbb{F}^n$ *is a linear* $(q, \delta, 1-\varepsilon, s)$*-RLDC, then* C *is also a* $(q, \delta, 1, s+(2+1/(|\mathbb{F}|-1))\varepsilon)$*-RLDC with a nonadaptive decoder. The same result holds for RLCCs.*

We will rely on the lemmas and proofs from [Gol23a] and point out where we differ. To begin, we can model any adaptive randomized relaxed local decoder[9] $\mathrm{Dec}(i)$ as a distribution over decision trees $\Gamma \leftarrow \mathcal{D}(i)$, where vertices are labeled with query indices from $[n]$ and edges are labeled with the symbols read from those query indices. Then, each leaf ℓ in each tree Γ corresponds to an ordered query tuple Q and a corresponding string $\sigma \in \mathbb{F}^Q$. Every leaf ℓ is labeled with a symbol from $\mathbb{F} \cup \{\perp\}$, which is the value returned by the decoder. With this in mind, the Goldberg transformation is based around relabeling leaves and rerandomizing the input.

7.4.1 Relabeling leaves

We first modify the decoder for each $i \in [k]$ to shift the input by a uniformly chosen codeword $C(\tilde{b})$ (and then subtract $\tilde{b}_i$ from the answer if it is not $\perp$); this gives

[9] This entire section will be written in terms of RLDCs, but the same proof works for RLCCs.

a new decoder Dec_R with the same parameters as the initial decoder. Next, every decision tree leaf which contributes to the completeness error needs to be relabeled. For each $i \in [k]$, these are the leaves in trees $\Gamma \in \mathrm{supp}(\mathcal{D}(i))$ which are reached by some uncorrupted codeword $C(b)$ but which are *not* labeled with b_i. Goldberg shows that changing the label of this leaf to b_i trades completeness error for soundness error, and we can iterate this process to get an adaptive relaxed local decoder with perfect completeness. The input rerandomization is crucial to make this lemma work.

Lemma 7.24 (leaf relabeling, [Gol23a, Claim 18])**.** *Let* $\mathrm{Dec}_R(i)$ *be the rerandomized decoder for index* i*, and let* $\mathrm{Dec}'_R(i)$ *be the same decoder with a single leaf of a single decision tree relabeled as described. Then, if* $\mathrm{Dec}_R(i)$ *has completeness error* ε *and soundness error* s*, and* $\mathrm{Dec}'_R(i)$ *has completeness error* ε' *and soundness error* s'*, then*

$$s' - s \leq \varepsilon - \varepsilon' .$$

There is a critical aspect to this relabeling step that we have so far overlooked. What if, for some $i \in [k]$, there is a leaf ℓ corresponding to queried values (Q, σ) where the i-th message index is *linearly independent* of the indices in Q (Fact 5.7)? This means that the decoder receives absolutely no information about the i-th message index from its queries, *even* if all of the queries match an uncorrupted codeword. Thus, there is no fixed label we can give to this leaf to reduce completeness error. Call these leaves *toxic*.

Definition 7.25 (Toxic leaves)**.** A leaf (Q, σ) of a decision tree Γ in the support of $\mathcal{D}(i)$ is *toxic* if there exist codewords $C(b), C(b')$ such that $C(b)|_Q = C(b')|_Q = \sigma$, but $b_i \neq b'_i$. By Fact 5.7, the local view Q does not give any information on the i-th index — for any local view and any codeword, there are $|\mathbb{F}| - 1$ other codewords, one for each possible symbol, which look identical on the local view yet differ on the i-th message index. On the other hand, if a leaf is non-toxic, then by Fact 5.7 the i-th message index for any codeword is completely determined by its restriction to Q.

7.4.2 Isolating toxic leaves

Goldberg disambiguates toxic leaves by adding one extra query to retrieve the value of the desired symbol (this is also why the RLDC must be systematic, so that b_i is part of $C(b)$). This is the query that we wish to save in Theorem 7.23. To avoid losing this extra query, we will isolate the toxic leaves by first relabeling all non-toxic leaves, so that all of the remaining completeness error is caused only by toxic leaves. Then, the following lemma shows that toxic leaves are rarely chosen:

Lemma 7.26. *Let* $\mathrm{Dec}_R(i)$ *be the rerandomized decoder for index* i *after all non-toxic leaves have been relabeled. Suppose it has completeness error* ε *which is caused only by toxic leaves. Then, for all* $b \in \mathbb{F}^k$*,*

$$\Pr[\mathrm{Dec}_R^{C(b)}(i) \text{ ends on a toxic leaf}] \leq \frac{|\mathbb{F}|}{|\mathbb{F}| - 1} \cdot \varepsilon \,.$$

Proof. Intuitively, the best possible behavior on a toxic leaf is to guess a random symbol from $\mathbb{F}$, in order to minimize the worst-case completeness error across all codewords. The probability of guessing correctly is $1/|\mathbb{F}|$, and so we should get a $|\mathbb{F}|/(|\mathbb{F}| - 1)$ factor relative to ε.

Formally, pick $|\mathbb{F}|$ messages $b^1, \ldots, b^{|\mathbb{F}|}$ where b^j has i-th symbol $j \in \mathbb{F}$. All of these codewords have the same completeness error because $\mathrm{Dec}_R(i)$ rerandomizes the input. We then have

$$\varepsilon = \Pr[\mathrm{Dec}_R^{C(b)}(i) \neq b_i] = \sum_{\text{toxic leaves } \ell} \Pr[\text{leaf } \ell \text{ chosen}] \cdot \Pr[\mathrm{Dec}_R^{C(b)}(i) \neq b_i \mid \text{leaf } \ell \text{ chosen}]$$

Because Dec_R rerandomizes over the input and all of the b^j are codewords, the probability of picking each particular leaf is the same for all of the different messages.

Hence,

$$\begin{aligned}
|\mathbb{F}|\varepsilon &= \sum_j \Pr[\mathrm{Dec}_R^{C(b^j)}(i) \neq b_i^j] \\
&= \sum_{\text{toxic leaves } \ell} \Pr[\text{leaf } \ell \text{ chosen}] \cdot \left(\sum_j \Pr[\mathrm{Dec}_R^{C(b^j)}(i) \neq b_i^j \mid \text{leaf } \ell \text{ chosen}] \right) \\
&= \sum_{\text{toxic leaves } \ell} \Pr[\text{leaf } \ell \text{ chosen}] \cdot \#\{\alpha \in \mathbb{F} : \ell \text{ is } \textit{not} \text{ labeled with } \alpha\} \\
&\geq (|\mathbb{F}| - 1) \cdot \Pr[\text{toxic leaf chosen}] .
\end{aligned}$$

□

We now know how often toxic leaves are selected by the decoder, which will help us remove them from the query distribution later on. For now, however, we can relabel them to make additional queries and achieve perfect completeness. When the decoder ends up on a toxic leaf, run a *global decoding* subroutine which queries the entire input and decodes the i-th message index; this shifts the remaining completeness error to soundness error using Lemma 7.24. We use global decoding to avoid requiring that the RLDC is systematic, which was necessary in the original proof of Goldberg. In summary, we have shown the following transformation:

Proposition 7.27. *Let $C \colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear $(q, \delta, 1 - \varepsilon, s)$-RLDC (or RLCC). Then, C has an adaptive relaxed local decoder (or corrector) with perfect completeness and soundness error at most $s+\varepsilon$. This decoder, when given a valid codeword as input, selects a non-toxic leaf (and makes q queries) with probability $\geq 1 - (|\mathbb{F}|\varepsilon)/(|\mathbb{F}| - 1)$, and selects a toxic leaf (and makes n queries) otherwise.*

7.4.3 Removing adaptivity and pruning toxic leaves

Lastly, Goldberg removes adaptivity by showing that we can evaluate the decision tree distribution on a uniformly random codeword to determine the index set to query. Now that the queries are nonadaptive, we can condition on not picking a toxic leaf, which adds a bit more soundness error.

Lemma 7.28 ([Gol23a, Lemma 20])**.** *If C has an adaptive relaxed local decoder* Dec *with perfect completeness and soundness error s, then C has a* nonadaptive *and* canonical *relaxed local decoder with the same completeness and soundness, by selecting a uniformly random codeword c and simulating* Dec *on c to determine its query set Q.*

Proof of Theorem 7.23. Using Proposition 7.27, we can start with C which is a $(q, \delta, 1-\varepsilon, s)$-RLDC and get a relaxed local decoder with perfect completeness and soundness error $s + \varepsilon$, and which selects a relabeled toxic leaf with probability at most $p = (|\mathbb{F}|\varepsilon)/(|\mathbb{F}| - 1)$. Then, use Lemma 7.28 to get a nonadaptive decoder with the same completeness and soundness. Finally, modify the query distribution to condition on never selecting a toxic leaf; the perfect completeness is unharmed but the soundness error will increase from $s + \varepsilon$ to $(s + \varepsilon)/(1 - p) \leq s + \varepsilon + p$, which is $s + \varepsilon(2|\mathbb{F}| - 1)/(|\mathbb{F}| - 1)$. Now, we are also guaranteed that the decoder makes q queries, which finishes the proof. □

7.5 Constructions of Small Query RLDCs/RLCCs That Are Not LDCs

In this section, we give a simple family of explicit codes that are q-RLDCs/RLCCs where q is a small, explicit constant, and are not q-LDCs for any constant q.

Theorem 7.29 (Formal Theorem 7.14)**.** *There is a linear code $C \colon \mathbb{F}_2^k \to \mathbb{F}_2^N$ where $N = k^{O(\log \log k)}$ such that for some constant $\delta > 0$, C is: (1) a $(15, \delta, 1, 1/3)$-RLDC; (2) a $(41, \delta, 1, 1/2 - \varepsilon)$-RLCC for a small constant $\varepsilon > 0$; (3) a $(58, \delta, 1, 1/3)$-RLCC; (4) not a $(O(\log k), O(N^{-1/3}), 1, 1/2 - \varepsilon)$-LDC for any (including subconstant) $\varepsilon > 0$.*

We will prove Theorem 7.29 in the next 3 subsections. In Section 7.5.1, we will define our code. Then, in Section 7.5.2, we will prove that it is an RLDC, and

in Section 7.5.3, we will prove that is an RLCC. Finally, in Section 7.5.4 we will prove that it is not an LDC.

7.5.1 The construction of the code

In this section, we will define the code C in Theorem 7.29. Let $t \in \mathbb{N}$ and let $\mathbb{F}_{2^t}$ be the finite field with 2^t elements. We recall the following basic facts about finite fields.

Fact 7.30 (Finite field notation)**.** *Let $\mathbb{F}_{2^t}$ be the finite field with 2^t elements. The field $\mathbb{F}_{2^t}$ is an $\mathbb{F}_2$-vector space of dimension t, and therefore there is an $\mathbb{F}_2$-linear isomorphism $\pi\colon \mathbb{F}_{2^t} \to \mathbb{F}_2^t$. The map π depends on the choice of basis for $\mathbb{F}_{2^t}$, which we will view as fixed in advance.*

For any $\alpha \in \mathbb{F}_{2^t}$ and $i \in [t]$, we let $\pi(\alpha)_i$ denote the i-th coordinate of α. Because multiplication by α is an invertible $\mathbb{F}_2$-linear transformation in $\mathbb{F}_{2^t}$, there exists a $t \times t$ matrix $M_\alpha \in \mathbb{F}_2^{t\times t}$ such that for any $\beta \in \mathbb{F}_{2^t}$, $\pi^{-1}(M_\alpha \pi(\beta)) = \alpha\beta$. In particular, for each $\alpha \in \mathbb{F}_{2^t}$ and $i \in [t]$, there exists $v \in \mathbb{F}_2^t$ such that for any $\beta \in \mathbb{F}_{2^t}$, $\langle v, \pi(\beta)\rangle = \pi(\alpha\beta)_i$.

The code C is defined formally via its encoding map. We will first define the linear subspace of codewords, and then explain how to define the encoding map.

Let $n, d \in \mathbb{N}$ be parameters with $d < 2^t$, and let $k = \binom{n}{\leq d} \coloneqq \sum_{i=0}^{d} \binom{n}{i}$. Let $f\colon \mathbb{F}_{2^t}^n \to \mathbb{F}_{2^t}$ be a polynomial of degree at most d in n variables $x_1, \ldots, x_n$. For each line L in $\mathbb{F}_{2^t}^n$, let S_L be an arbitrary (ordered) subset of L of size $d+1$. For a collection of field elements $(\alpha_1, \ldots, \alpha_{d+1}) \in \mathbb{F}_{2^t}^{d+1}$, we let $\mathrm{Had}(\alpha_1, \ldots, \alpha_{d+1})$ be the encoding of these field elements using the Hadamard code over $\mathbb{F}_2$ and the map π. That is, $\mathrm{Had}(\alpha_1, \ldots, \alpha_{d+1})$ is a vector of length $2^{t(d+1)}$, where entries are indexed by $v = (v_1, \ldots, v_{d+1}) \in \mathbb{F}_2^{t(d+1)}$, and the v-th entry is $\sum_{i=1}^{d+1} \langle v_i, \pi(\alpha_i)\rangle$.

With the above setup, we can now specify the set of codewords. For each polynomial $f\colon \mathbb{F}_{2^t}^n \to \mathbb{F}_{2^t}$ of degree at most d, we obtain a codeword by encoding the

function f via the bit-wise concatenation of $\mathrm{Had}(f(S_L))$ for each line L. That is, the codeword corresponding to f has, for each line L, a block of $2^{t(d+1)}$ bits that is $\mathrm{Had}(f(S_L))$, so that we can view the codeword as a collection $\{\mathrm{Had}(f(S_L))\}_L$ indexed by all lines L in $\mathbb{F}_{2^t}^n$.

Setting parameters. The set of codewords is clearly an $\mathbb{F}_2$-linear subspace, and it has dimension tk, where $k = \binom{n}{\leq d}$. Moreover, it is a linear subspace in $\mathbb{F}_2^{\#L \cdot 2^{t(d+1)}}$, where $\#L = 2^{tn}(2^{tn}-1)/(2^t(2^t-1))$ is the number of lines in $\mathbb{F}_{2^t}^n$. To minimize the blocklength as a function of k, we take $d = O(n)$ (which forces $t = \Theta(\log d) = \Theta(\log n)$, as we need $2^t > d$), so that $k = 2^{O(n)}$ and the blocklength is $2^{O(n \log n)}$. Hence, the blocklength is $k^{O(\log \log k)}$.

Choosing an encoding map. To finish defining the code C, we need to specify the encoding map. To do this, we first choose an arbitrary $\mathbb{F}_{2^t}$-linear map A from $\mathbb{F}_{2^t}^k$ to the set of degree $\leq d$ polynomials f with the property that there are elements $x^{(1)}, \ldots, x^{(k)} \in \mathbb{F}_{2^t}$ such that if $f = Ab$ for $b \in \mathbb{F}_{2^t}$, then $f(x^{(i)}) = b_i$ for all i. We can then extend A to a map from $\mathbb{F}_2^{tk}$ to the set of degree $\leq d$ polynomials by splitting the input into k blocks of size t, applying the linear map π^{-1} (Fact 7.30) on each block, applying A to the output, and then applying π to each field element in the evaluation table of the resulting polynomial f.

As we will show, our RLDC decoder, when given access to a corrupted version of the codeword corresponding to the polynomial f, will be able to recover $\pi(\alpha f(x))_i$ for every $\alpha \in \mathbb{F}_{2^t}$, $x \in \mathbb{F}_{2^t}^n$, and any $i \in [t]$. As a result, the specific choice of encoding map A specified above is not important.

Comparison to [AS21]. Our construction shares some similarities to the work of [AS21]. Similar to our construction, they first use a Reed–Muller code to encode a message as a polynomial f, and then they encode each "local view" of f in some way. Their "local views", however, are a special set of planes $\mathcal{P}$ that have "directions" in $\mathbb{H}^n$, where $\mathbb{H}$ is a subfield of $\mathbb{F}$. They then encode $f|_{\mathcal{P}}$ for each plane $\mathcal{P}$ using a canonical correctable Probabilistically Checkable Proof of Proximity (ccPCPP). Their

decoder then takes certain random walk of length $[\mathbb{F} : \mathbb{H}] + 1$ on the special planes $\mathcal{P}$, and uses the ccPCPPs to decode values of f and do consistency checks.

In contrast, we encode f by encoding its "local view" $f|_L$ for every line L using the Hadamard code. As we shall see, our RLDC decoder takes a random walk of length 2 on the lines, and our RLCC decoder takes a random walk of length 3.

An observation about C. Finally, we make some observations about C, which will be useful in the proofs.

Observation 7.31. Fix a line L. For any $z \in L$, any $\alpha \in \mathbb{F}_{2^t}$, and any $i \in [t]$, there exists a point $v^{L,z,\alpha,i} \in \mathbb{F}_2^{t(d+1)}$ such that $\mathrm{Had}(f(S_L))(v^{L,z,\alpha,i}) = \pi(\alpha f(z))_i$. That is, for any $i \in [t]$, one can recover the i-th bit of $\alpha \cdot f(z)$ using the Hadamard encoding of (only) $d+1$ points S_L on the line L.

The point $v^{L,z,\alpha,i}$ should be interpreted as: to recover the i-th bit of $\alpha f(z)$ using the Hadamard encoding of $f|_L$, we query the Hadamard encoding of $f|_L$ at the point $v^{L,z,\alpha,i}$.

Proof. Let $z_1, \ldots, z_{d+1}$ be the points in S_L. Because f is a degree d polynomial and $d < 2^t = |\mathbb{F}_{2^t}|$, by polynomial interpolation, there exist coefficients $\alpha_1, \ldots, \alpha_{d+1} \in \mathbb{F}_{2^t}$ such that $f(z) = \sum_{j=1}^{d+1} \alpha_j f(z_j)$. Therefore, $\pi(\alpha f(z))_i = \sum_{j=1}^{d+1} \pi(\alpha\alpha_j f(z_j))_i$. For each j, let v^j be the i-th row of $M_{\alpha\alpha_j}$, the matrix defined in Fact 7.30. The vector v^j then has the property that $\langle v^j, \pi(\beta)\rangle = \pi(\alpha\alpha_j\beta)_i$ for all $\beta \in \mathbb{F}_{2^t}$, and therefore $\pi(\alpha\alpha_j f(z_j))_i = \langle v^j, \pi(f(z_j))\rangle$. Concatenating the vectors v^j together yields the vector $v^{L,z,\alpha,i}$. □

7.5.2 The RLDC decoder and its analysis

In this subsection, we prove Item (1) in Theorem 7.29, where the code is defined in Section 7.5.1.

Proof of Item (1) in Theorem 7.29. To show that the code is an RLDC, we will analyze the decoder defined below.

Algorithm 7.32.

Given: *A collection of functions $\{G_L\}_L$ where each $G_L \colon \mathbb{F}_2^{t(d+1)} \to \mathbb{F}_2$ is an arbitrary function, along with a point $x^* \in \mathbb{F}_{2^t}^n$, a field element $\alpha^* \in \mathbb{F}_{2^t}$, and an index $i^* \in [t]$. The collection $\{G_L\}_L$ is supposed to be equal to $\{\mathrm{Had}(f(S_L))\}_L$ for some polynomial f of degree $\leq d$.*

Output: *A symbol in $\{0, 1, \perp\}$, hopefully equal to $\pi(\alpha^* f(x^*))_{i^*}$.*

Operation:

(1) Choose a line L^* containing x uniformly at random.

(2) ***Run*** *r_1* ***linearity tests on*** *G_{L^*}**:** *pick random $v^{(1)}, v^{(2)}, v^{(3)} \in \mathbb{F}_2^{t(d+1)}$, and check that $G_{L^*}(v^{(1)}) + G_{L^*}(v^{(2)}) + G_{L^*}(v^{(3)}) = 0$. If the check fails, output $\perp$. Repeat r_1 times.*

(3) ***Run*** *r_2* ***consistency tests:***

(a) Pick $y \in L^*$ uniformly at random, $\alpha \in \mathbb{F}_{2^t}$ uniformly at random, and $i \in [t]$.

(b) ***Locally decode*** *$\pi(\alpha f(y))_i$* ***from*** *G_{L^*}**:** let $v^{L,y,\alpha,i}$ be the vector from Observation 7.31. Let $v^{(4)} \in \mathbb{F}_2^{t(d+1)}$ be chosen uniformly at random. Let $a_{L^*} = G_{L^*}(v^{L^*,y,\alpha,i} + v^{(4)}) + G_{L^*}(v^{(4)})$.*

(c) ***Locally decode*** *$\pi(\alpha f(y))_i$* ***from*** *$G_{L'}$**:** choose L' to be a uniformly random line containing y. Let $v^{L',y,\alpha,i}$ be the vector from Observation 7.31. Let $v^{(5)} \in \mathbb{F}_2^{t(d+1)}$ be chosen uniformly at random. Let $a_{L'} = G_{L'}(v^{L',y,\alpha,i} + v^{(5)}) + G_{L'}(v^{(5)})$.*

(d) ***Consistency check:*** *check that $a_{L^*} = a_{L'}$ and output $\perp$ if the check fails.*

(e) Repeat r_2 times.

(4) ***Locally decode*** *$\pi(\alpha^* f(x^*))_{i^*}$* ***from*** *$G_{L^*}$**:** let v^{L^*,x^*,α^*,i^*} be the vector from Observation 7.31. Let $v^{(6)} \in \mathbb{F}_2^{t(d+1)}$ be chosen uniformly at random. Output $G_{L^*}(v^{L^*,x^*,\alpha^*,i^*} + v^{(6)}) + G_{L^*}(v^{(6)})$.*

In the above algorithm, r_1 and r_2 are positive integers, which we will choose later.

First, we observe that by Observation 7.31, Algorithm 7.32 returns

$\pi(\alpha^* f(x^*))_{i^*}$ for any input (x^*, α^*, i^*) if the collection of functions is indeed $\{\mathrm{Had}(f(S_L))\}_L$ for some polynomial f of degree $\leq d$. We note that, under our choice of encoding map, the message bits correspond to evaluations of f on specific points, and so this allows us to recover any bit of the message. Thus, Algorithm 7.32 has perfect completeness. We also note that Algorithm 7.32 makes $3r_1 + 4r_2 + 2$ queries, though we will explain how to slightly reduce the query complexity later.

We now analyze the soundness error of Algorithm 7.32. For each line L, let H_L be the closest linear function to G_L. Given a linear function H_L, there is a unique univariate degree d polynomial h_L on the line L such that for each $z \in L$, $\alpha \in \mathbb{F}_{2^t}$, and $i \in [t]$, $\pi(\alpha h_L(z))_i = H_L(v^{L,z,\alpha,i})$. The polynomial h_L is simply defined by "extracting" the values of h_L on S_L using Observation 7.31 and then defining h_L on the rest of the line using polynomial interpolation. We let F_L be the linear function $\mathrm{Had}(f(S_L))$, and we define f_L to be the polynomial $f|_L$. We note that the process used to obtain h_L from H_L yields f_L if $H_L = F_L$. For two Boolean functions F_L and G_L, we let $\mathrm{dist}(F_L, G_L)$ denote the (relative) Hamming distance over $\mathbb{F}_2$, and for two polynomials f_L and g_L, we let $\mathrm{dist}(f_L, g_L)$ denote the (relative) Hamming distance over $\mathbb{F}_{2^t}$, i.e., the fraction of $x \in L$ such that $f_L(x) \neq g_L(x)$.

We will split our analysis into the following disjoint cases, and bound the probability that decoder errs in each case.

(1) The random line L^* through x^* is "bad", in that $\mathbb{E}_{L':|L' \cap L^*| \geq 1, x^* \notin L'}[\mathrm{dist}(G_{L'}, F_{L'})] \geq \delta^{2/3}$.

(2) $f_{L^*} \neq h_{L^*}$ and $\mathbb{E}_{L':|L' \cap L^*| \geq 1, x^* \notin L'}[\mathrm{dist}(G_{L'}, F_{L'})] \leq \delta^{2/3}$, but both the repeated linearity test and the repeated consistency test pass.

(3) $f_{L^*} = h_{L^*}$, but the repeated linearity test passes and the output is $1 - \pi(\alpha^* f(x^*))_{i^*}$ (i.e., not in $\{\pi(\alpha^* f(x^*))_{i^*}, \perp\}$).

Analysis of Case 1. Recall that the collection $\{G_L\}_L$ is δ-close to $\{F_L\}_L$. Equivalently, this means that $\mathbb{E}_L[\mathrm{dist}(G_L, F_L)] \leq \delta$. Since L^* is a uniformly random line

passing through x^*, and L' is a uniformly random line passing through a random point $y \neq x^*$ on L^*, it follows that L' is distributed as a uniformly random line. Therefore, $\mathbb{E}_L[\mathbb{E}_{L':|L'\cap L|\geq 1, x^*\notin L'}[\mathrm{dist}(G_L, F_L)]] = \delta$. It follows by Markov's inequality that the probability over L that $\mathbb{E}_{L':|L'\cap L|\geq 1, x^*\notin L'}[\mathrm{dist}(G_L, F_L)] \geq \delta^{2/3}$ is at most $\delta^{1/3}$.

Analysis of Case 2. Let $\varepsilon = \mathrm{dist}(G_{L^*}, H_{L^*})$. Recall by Fact 5.21, the probability that one iteration of the linearity test passes is at most $1 - \mathrm{dist}(G_{L^*}, H_{L^*}) = 1 - \varepsilon$. Hence, the probability that all r_1 repetitions pass is at most $(1-\varepsilon)^{r_1}$.

Let us now analyze one iteration of the consistency test. Call the line L' chosen "bad" if $\mathrm{dist}(G_{L'}, F_{L'}) \geq \delta^{1/3}$. Since $\mathbb{E}_{L':|L'\cap L^*|\geq 1, x^*\notin L'}[\mathrm{dist}(G_{L'}, F_{L'})] \leq \delta^{2/3}$, it follows that the probability that L' is "bad" is at most $\delta^{1/3}$.

Let us proceed assuming that L' is not "bad". Then, since $F_{L'}$ is a linear function, by Fact 5.22 and Observation 7.31, it follows that $a_{L'}$ is equal to $\pi(\alpha f_{L^*}(y))_i$ (which is $\pi(\alpha f(y))_i$) with probability at least $1 - 2\delta^{1/3}$. Similarly, because $\mathrm{dist}(G_{L^*}, H_{L^*}) = \varepsilon$, it follows that a_{L^*} is equal to $\pi(\alpha h_{L^*}(y))_i$ with probability at least $1 - 2\varepsilon$. Moreover, this holds regardless of the choice of α and i.

Finally, because $f_{L^*} \neq h_{L^*}$ and these are different degree d polynomials, the probability that $f_{L^*}(y) \neq h_{L^*}(y)$ is at least $1 - d/n$. And, if such a y is chosen, the probability that $\pi(\alpha f_{L^*}(y))_i \neq \pi(\alpha h_{L^*}(y))_i$ is $1/2$, as $\alpha(f_{L^*}(y) - h_{L^*}(y))$ is a random element of $\mathbb{F}_{2^t}$, and so its i-th bit is 1 with probability $1/2$.

Thus, we can conclude that each round of the consistency test passes with probability at most $\delta^{1/3} + d/n + (1 - d/n)(\frac{1}{2} + \frac{1}{2}(2\delta^{1/3} + 2\varepsilon))$. Since the tests are independent, we conclude that the probability that all rounds of both tests pass is at most $(1-\varepsilon)^{r_1}\left(\delta^{1/3} + d/n + (1-d/n)(\frac{1}{2} + \delta^{1/3} + \varepsilon)\right)^{r_2}$.

Analysis of Case 3. By the previous analysis, the probability that all linearity tests pass is at most $(1-\varepsilon)^{r_1}$, where $\varepsilon = \mathrm{dist}(G_{L^*}, H_{L^*})$. By Fact 5.22, the output is $\pi(h_{L^*}(x^*))_{i^*}$ with probability at least $1 - 2\varepsilon$, which is $\pi(f_{L^*}(x^*))_{i^*} = \pi(f(x^*))_{i^*}$ since $h_{L^*} = f_{L^*}$. Thus, the probability that the decoder outputs an incorrect answer is at

most $(1-\varepsilon)^{r_1} \cdot 2\varepsilon$ in this case.

In total, we conclude that the probability that the output of the decoder is not in $\{\pi(f(x^*))_i, \perp\}$ is at most

$$\eta(\varepsilon) = \max\{\delta^{1/3}, (1-\varepsilon)^{r_1}\left(\delta^{1/3} + d/n + (1-d/n)(1/2+\delta^{1/3}+\varepsilon)\right)^{r_2}, (1-\varepsilon)^{r_1}\cdot 2\varepsilon\},$$

where $\varepsilon = \mathrm{dist}(G_{L^*}, H_{L^*})$. Setting $r_1 = r_2 = 2$, δ and d/n to be sufficiently small constants, and taking the maximum over all ε shows that $\eta \leq \frac{1}{2} - \varepsilon'$ for some constant ε'. In fact, by taking δ and d/n to be sufficiently small constants, we can make $\eta = (3/4)^4 + \varepsilon'' \leq 1/3$ for a small constant ε''.

The query complexity is $3r_1 + 4r_2 + 2 = 16$. Below, we shall explain how to save one query.

Saving a query. Observe that in the above analysis, we analyzed Cases (2) and (3) separately. This allows us to "recycle" our queries across the different cases. Namely, we observe that in Item (3b) in Algorithm 7.32, it holds that $v^{(4)}$ is chosen uniformly at random. Because we do not use the consistency test to analyze Case (3) above, we can reuse this query by taking $v^{(6)} = v^{(4)}$ and the analysis does not change. This allows us to make the query complexity slightly smaller: $3r_1 + 4r_2 + 1 = 15$. □

7.5.3 The RLCC decoder and its analysis

In this subsection, we prove Items (2) and (3) in Theorem 7.29. The key difference between the RLCC decoder and RLDC decoder is that the RLCC decoder may be asked to decode an arbitrary point $v \in \mathbb{F}_2^{t(d+1)}$ on the Hadamard encoding of some line L^*. Unlike in the case of the RLDC decoder, where the point v corresponded to decoding $\pi(f(x))_i$ for some point x and some index i, here the point v might not correspond to any evaluation point of the polynomial f.

Proof of Items (2) and (3) in Theorem 7.29. To show that the code is an RLCC, we will analyze the decoder defined below.

Algorithm 7.33.

Given: *A collection of functions* $\{G_L\}_L$ *where each* $G_L \colon \mathbb{F}_2^{t(d+1)} \to \mathbb{F}_2$ *is an arbitrary function, along with a point* $v^* \in \mathbb{F}_2^{t(d+1)}$ *and a line* L^**. The collection* $\{G_L\}_L$ *is supposed to be equal to* $\{\mathrm{Had}(f(S_L))\}_L$ *for some polynomial* f *of degree* $\leq d$*.*

Output: *A symbol in* $\{0, 1, \perp\}$*, hopefully equal to* $\mathrm{Had}(f(S_{L^*}))(v^*)$*.*

Operation:

(1) ***Run*** r_1 ***linearity tests on*** G_{L^*}***:*** *pick random* $v^{(1)}, v^{(2)}, v^{(3)} \in \mathbb{F}_2^{t(d+1)}$*, and check that* $G_{L^*}(v^{(1)}) + G_{L^*}(v^{(2)}) + G_{L^*}(v^{(3)}) = 0$*. If the check fails, output* $\perp$*. Repeat* r_1 *times.*

(2) ***Run*** r_2 ***consistency tests times:***

(1) Pick $y \in L^*$ *uniformly at random,* $\alpha \in \mathbb{F}_{2^t}$ *uniformly at random, and* $i \in [t]$*.*

(2) ***Locally decode*** $\pi(\alpha f(y))_i$ ***from*** G_{L^*}***:*** *let* $v^{L,y,\alpha,i}$ *be the vector from Observation 7.31. Let* $v^{(4)} \in \mathbb{F}_2^{t(d+1)}$ *be chosen uniformly at random. Let* $a_{L^*} = G_{L^*}(v^{L^*,y,\alpha,i} + v^{(4)}) + G_{L^*}(v^{(4)})$*.*

(3) ***Locally decode*** $\pi(\alpha f(y))_i$ ***using the RLDC decoder:*** *Run the RLDC decoder in Algorithm 7.32 to recover* $\pi(\alpha f(y))_i$*. If the RLDC decoder outputs* $\perp$*, then abort and output* $\perp$*. Else, let* $a_{\mathrm{RLDC}} \in \{0, 1\}$ *be the output of the RLDC decoder.*

(4) ***Consistency check:*** *check that* $a_{L^*} = a_{\mathrm{RLDC}}$ *and output* $\perp$ *if the check fails.*

(5) Repeat r_2 *times.*

(3) ***Locally decode*** $\mathrm{Had}(f(S_{L^*}))(v^*)$ ***from*** G_{L^*}***:*** *Let* $v^{(6)} \in \mathbb{F}_2^{t(d+1)}$ *be chosen uniformly at random. Output* $G_{L^*}(v^* + v^{(6)}) + G_{L^*}(v^{(6)})$*.*

The analysis of Algorithm 7.33 is similar to the analysis of Algorithm 7.32. Below, we will reuse the same notation as done in Section 7.5.2. The fact that the decoder has perfect completeness is straightforward, and so we proceed with analyzing the soundness error.

As before, will split our analysis into disjoint cases, and bound the probability that decoder errs in each case.

(1) $f_{L^*} \neq h_{L^*}$, but both the repeated linearity test and repeated consistency test pass.

(2) $f_{L^*} = h_{L^*}$, but the repeated linearity test passes and the output is $1 - \mathrm{Had}(f(S_{L^*}))(v^*)$ (i.e., not in $\{\mathrm{Had}(f(S_{L^*}))(v^*), \perp\}$).

Analysis of Case 1. The probability that all r_1 repetitions of the linearity pass when $\mathrm{dist}(G_{L^*}, H_{L^*}) = \varepsilon$ is at most $(1-\varepsilon)^{r_1}$ by Fact 5.21.

Let us now analyze one iteration of the consistency test. Because $f_{L^*} \neq h_{L^*}$ and these are different degree d polynomials, the probability that $f_{L^*}(y) \neq h_{L^*}(y)$ is at least $1 - d/n$. Suppose that such a y is chosen. Then, the probability that $\pi(\alpha f_{L^*}(y))_i \neq \pi(\alpha h_{L^*}(y))_i$ is $1/2$, as $\alpha(f_{L^*}(y) - h_{L^*}(y))$ is a random element of $\mathbb{F}_{2^t}$, and so its i-th bit is 1 with probability $1/2$.

Let us assume that y, α and i are "good", meaning that $\pi(\alpha f_{L^*}(y))_i \neq \pi(\alpha h_{L^*}(y))_i$. Because $\mathrm{dist}(G_{L^*}, H_{L^*}) \leq \varepsilon$, it follows that a_{L^*} is equal to $\pi(\alpha h_{L^*}(y))_i$ with probability at least $1 - 2\varepsilon$. Moreover, this holds regardless of the choice of α and i.

We now invoke the soundness of the RLDC decoder. The RLDC decoder either outputs the correct bit $\pi(\alpha f_{L^*}(y))_i$, or else it outputs $\perp$, with probability at least $1 - \eta_{\mathrm{RLDC}}$ for some constant η_{RLDC}. Hence, the probability that the RLDC decoder outputs the wrong bit is at most η_{RLDC}.

Thus, the probability that one round of the consistency test passes is at most $d/n + \frac{1}{2}(1 - d/n) + \frac{1}{2}(1 - d/n)(\eta_{\mathrm{RLDC}} + (1 - \eta_{\mathrm{RLDC}})2\varepsilon)$. Because the consistency test and linearity tests are independent, the total probability of all tests passing is at most

$$(1-\varepsilon)^{r_1} \left(d/n + \frac{1}{2}(1 - d/n)(1 + \eta_{\mathrm{RLDC}} + (1 - \eta_{\mathrm{RLDC}})2\varepsilon) \right)^{r_2}.$$

Analysis of Case 2. The probability that all r_1 repetitions of the linearity pass when $\mathrm{dist}(G_{L^*}, H_{L^*}) = \varepsilon$ is at most $(1-\varepsilon)^{r_1}$ by Fact 5.21. By Fact 5.22, the output is

$h_{L^*}(v^*)$ with probability at least $1-2\varepsilon$, which is $\mathrm{Had}(f(S_{L^*}))(v^*)$ since $h_{L^*} = f_{L^*}$. As these are independent, the probability that the decoder outputs an incorrect answer is at most $(1-\varepsilon)^{r_1} \cdot 2\varepsilon$ in this case.

In total, we conclude that the probability that the output of the decoder is incorrect is at most

$$\eta(\varepsilon) = \max\{(1-\varepsilon)^{r_1} \left(d/n + \frac{1}{2}(1-d/n)(1+\eta_{\mathrm{RLDC}} + (1-\eta_{\mathrm{RLDC}})2\varepsilon) \right)^{r_2}, (1-\varepsilon)^{r_1} \cdot 2\varepsilon\},$$

where $\varepsilon = \mathrm{dist}(G_{L^*}, H_{L^*})$. Recall that $\eta_{\mathrm{RLDC}} < 1/3$ provided that δ is a sufficiently small constant. Hence, setting $r_1 = r_2 = 2$, d/n to be a sufficiently small constant, and maximizing $\eta(\varepsilon)$ over all ε, we see that the maximum is at most $\frac{4}{9} + \varepsilon'' < 1/2$ for some sufficiently small constant ε''.

Setting $r_1 = 2$ and $r_2 = 3$, we get that $\eta \leq (2/3)^3 + \varepsilon'' < 1/3$ for a sufficiently small constant ε''.

The query complexity of the decoder is $3r_1 + (2 + q_{\mathrm{RLDC}})r_2 + 1$ (using the trick to reduce the query complexity by 1 from Section 7.5.2). Since $q_{\mathrm{RLDC}} = 15$, this yields 41 queries for the case of soundness error below 1/2, and 58 queries for the case of soundness error below 1/3. □

7.5.4 The code is not an LDC

In this section, we prove Item (4) in Theorem 7.29, i.e., that the code constructed in Section 7.5.1 is not an LDC.

Proof of Item (4) in Theorem 7.29. Recall that the code is defined as follows. For each degree $\leq d$ polynomial $f\colon \mathbb{F}_{2^t}^n \to \mathbb{F}_{2^t}$ in n variables, we encode f by writing down $\mathrm{Had}(f(S_L))$ for each line L in $\mathbb{F}_{2^t}^n$, where we have set $d = \Theta(n)$, $t = \Theta(\log d) = \Theta(\log n)$, and we have $k = t\binom{n}{\leq d}$. The blocklength of the code is $N = \#L \cdot 2^{t(d+1)}$, where $\#L = 2^{tn}(2^{tn}-1)/(2^t(2^t-1))$ is the number of lines in $\mathbb{F}_{2^t}^n$. We will show that

C is not a $(q, \delta, 1, 1/2 - \varepsilon)$-LDC for any $\varepsilon > 0$ (that need not be constant) for $q = d$ and $\delta = 2^{-n(t-1)}$. Note that $q = d \leq O(\log k)$ and $\delta = 2^{-n(t-1)} \leq O(N^{-1/3})$.

Setup. To show that the code is not a $(d, 2^{-n(t-1)}, 1, 1/2 - \varepsilon)$-LDC for any $\varepsilon > 0$, it suffices to show that for any d-query LDC decoder Dec and any choice of $\alpha^* \in \mathbb{F}_{2^t}$, $x^* \in \mathbb{F}_{2^t}^n$, and $i^* \in [t]$, there is a collection of "local encodings" $\{h_L\}_L$ that is $2^{-n(t-1)}$-close to a codeword $\{\mathrm{Had}(f(S_L))\}_L$ such that $\Pr[\mathrm{Dec}^{\{h_L\}_L}(\alpha^*, x^*, i^*)] = \frac{1}{2}$.

First, we observe that it suffices to argue this in the *erasure* error model, as opposed to the standard Hamming error model. In the erasure model, we corrupt a codeword by replacing a bit in the encoding with an error symbol $\perp$ to signify that the bit has been erased. That is, the outputs of the "local encodings" h_L can only agree with $\mathrm{Had}(f(S_L))$ or be $\perp$. Clearly, if a decoder works in the standard Hamming error model, then it also works in the erasure error model, as we can simulate the Hamming error model by replacing any $\perp$ symbol by a 0. Hence, it suffices to argue that there is no decoder in the erasure error model. In fact, it suffices to argue this even in the following weaker "line erasure error model", where we are only allowed to erase entire lines. That is, for each line L, we either have $h_L \equiv \mathrm{Had}(f(S_L))$, i.e., they are the same function, or $h_L \equiv \perp$, i.e., the entire function outputs $\perp$.

Now that we are working in the line erasure error model, we will strengthen the decoder by allowing it to make "line queries". That is, instead of allowing the decoder to make one query to retrieve an evaluation of h_L for a line L and evaluation point of its choice, we will instead allow the decoder to make "line queries" where the decoder reads the entire evaluation table of a function h_L with one "line query L". Clearly, any q-query decoder in the standard query model can be simulated in the line query model with $\leq q$ queries, as any query to a specific evaluation of a local function h_L can simply be replaced by a query to the entire line L. Notice that since the decoder makes line queries and each local function $\mathrm{Had}(f(S_L))$ is either completely replaced with $\perp$ or is untouched, we may further assume that the decoder receives $f|_L$ when it makes the line query L, as opposed to $\mathrm{Had}(f(S_L))$, as it can simply compute

$\mathrm{Had}(f(S_L))$ from $f|_L$. Thus, we will view the decoder as having "line query" access to the polynomial f, where for every line it either receives $f|_L$ (the entire line) or $\perp$ (if the line has been erased). For a set of *erased* lines $\mathcal{L}$, we will use the notation $\mathrm{Dec}^{f,\mathcal{L}}$ to indicate that the decoder has line query access to the function f on all lines *except* those in $\mathcal{L}$. Namely, if the decoder queries a line L and $L \in \mathcal{L}$, then it receives $\perp$, and otherwise it receives $f|_L$.

Lower bound against "line query" decoders in the erasure model. We will now show the following. For any "line query" decoder Dec making at most d queries and for any $\alpha^* \in \mathbb{F}_{2^t}$, $x^* \in \mathbb{F}_{2^t}^n$, and $i^* \in [t]$, there is a set of δ-fraction of lines $\mathcal{L}$ that can be erased such that $\mathbb{E}_f[\Pr[\mathrm{Dec}^{f,\mathcal{L}}(\alpha^*, x^*, i^*) = \pi(\alpha^* f(x^*))_{i^*}]] = \frac{1}{2}$, where the outer expectation is over the random choice of the polynomial f and the inner probability is over the randomness of the decoder. In particular, this implies that there exists a polynomial f of degree $\leq d$ such that $\Pr[\mathrm{Dec}^{f,\mathcal{L}}(\alpha^*, x^*, i^*) = \pi(\alpha^* f(x^*))_{i^*}]] \leq \frac{1}{2}$, which will finish the proof.

Let us now argue the above claim. First, the set $\mathcal{L}$ of lines will be all lines L with $x^* \in L$, which we denote by $\mathcal{L}_{x^*}$. We observe that the number of such lines is $(2^{tn}-1)/(2^t-1)$, which is a $\delta = 2^{-(t-1)n}$-fraction of all lines. As we permit the decoder to be adaptive, the queries made by the decoder can be described as a depth $d+1$ decision tree (with d layers of internal "decision" or "query" nodes and one layer of leaves or output nodes), where each decision node has a line L that will be queried as well as a child for each possible evaluation table $f|_L$, and the output of the decoder is given by the leaves. Note that we may assume without loss of generality that the decoder makes exactly d queries, so that the tree is a complete tree with depth exactly $d+1$. Furthermore, the decision tree is determined by the input (α^*, x^*, i^*) to the decoder and the choice r of the decoder randomness. That is, for each input (α^*, x^*, i^*) and randomness r, there is a tree $T_{(\alpha^*, x^*, i^*, r)}$ that determines the queries that can be made when the decoder randomness is r. We use the notation $T_{(\alpha^*, x^*, i^*, r)}(f, \mathcal{L})$ to denote the output of the tree $T_{(\alpha^*, x^*, i^*, r)}$ when given access to f

on all lines not in $\mathcal{L}$, which we note is equal to $\mathrm{Dec}^{f,\mathcal{L}}(\alpha^*, x^*, i^*; r)$, the output of the decoder when the randomness is r. We thus have

$$\mathbb{E}_f[\Pr[\mathrm{Dec}^{f,\mathcal{L}_{x^*}}(\alpha^*, x^*, i^*) = \pi(\alpha^* f(x^*))_{i^*}]] = \mathbb{E}_f[\mathbb{E}_r[\mathbf{1}(T_{(\alpha^*,x^*,i^*,r)}(f,\mathcal{L}) = \pi(\alpha^* f(x^*))_{i^*})]]$$
$$= \mathbb{E}_r[\mathbb{E}_f[\mathbf{1}(T_{(\alpha^*,x^*,i^*,r)}(f,\mathcal{L}) = \pi(\alpha^* f(x^*))_{i^*})]]\,,$$

using linearity of expectation, where the expectation over f is over a random degree $\leq d$ polynomial and the expectation over r is over the randomness r of the decoder.

We will now argue that for any decision tree T of depth $d+1$ that does not query lines in $\mathcal{L}_{x^*}$, it holds that $\mathbb{E}_f[\mathbf{1}(T(f) = \pi(\alpha^* f(x^*))_{i^*})] = \frac{1}{2}$. (Note that we can replace $T_{(\alpha^*,x^*,i^*,r)}$ with an equivalent decision tree that never queries lines in $\mathcal{L}_{x^*}$.) This follows from the following observation, which we will prove at the end of the subsection.

Claim 7.34. *Let $x \in \mathbb{F}_{2^t}^n$ and let $L_1, \ldots, L_q$ be lines in $\mathbb{F}_{2^t}^n$ that do not contain x. Then, there is a degree-q polynomial $g\colon \mathbb{F}_{2^t}^n \to \mathbb{F}_{2^t}$ such that $g(y) = 0$ for all $y \in \cup_{i=1}^q L_i$ and $g(x) = 1$.*

For a set of d lines $L_1, \ldots, L_d$, let $g_{x^*,L_1,\ldots,L_d}$ be the polynomial in Claim 7.34 (which may not be unique, but we choose an arbitrary fixed one for each choice of lines). Now, consider the following distribution: (1) sample f uniformly at random, (2) let $L_1^{(f)}, \ldots, L_d^{(f)}$ be the lines queried by $T(f)$ (which are not in $\mathcal{L}_{x^*}$), and (3) output $f + \beta g_{x^*,L_1^{(f)},\ldots,L_d^{(f)}}$ where $\beta \leftarrow \mathbb{F}_{2^t}$ uniformly at random. Note that this is well-defined since $x \notin L_i^{(f)}$ for all $i \in [d]$, as $L_i^{(f)} \notin \mathcal{L}_{x^*}$. We claim that this distribution is simply uniform on degree $\leq d$ polynomials, i.e., it is the same as choosing f uniformly at random. To see this, let $S_f = \{f + \beta g_{x^*,L_1^{(f)},\ldots,L_d^{(f)}}\}_{\beta \in \mathbb{F}_{2^t}}$. Observe that for any $h \in S_f$, we have that $S_h = S_f$. Indeed, this follows because $g_{x^*,L_1^{(f)},\ldots,L_d^{(f)}}(y) = 0$ for all $y \in \cup_{i=1}^d L_i^{(f)}$, and so it follows that h agrees with f on all lines $L_1^{(f)}, \ldots, L_d^{(f)}$, and hence the decision tree must follow the same root-to-leaf path for both h and f. In particular, this also implies that $T(f) = T(h)$ for all $h \in S_f$. It thus follows that the S_f's partition the set of degree $\leq d$ polynomials into sets of size 2^t. Hence, the above

distribution simply first chooses a set S_f in the partition uniformly at random, and then chooses a random polynomial in S_f, which is the same as choosing a uniformly random degree $\leq d$ polynomial.

Now, to finish the argument, we have that

$$\begin{aligned}&\mathbb{E}_f[\mathbf{1}(T(f) = \pi(\alpha^* f(x^*))_{i^*})] = \mathbb{E}_f[\mathbb{E}_{h\sim S_f}[\mathbb{E}_\beta[\mathbf{1}(T(h) = \pi(\alpha^* h(x^*))_{i^*})]]]\\&= \mathbb{E}_f[\mathbb{E}_{h\sim S_f}[\mathbb{E}_\beta[\mathbf{1}(T(f) = \pi(\alpha^* h(x^*))_{i^*})]]] = \frac{1}{2}\,,\end{aligned}$$

where the third equality uses that $T(f) = T(h)$ for all $h \in S_f$ and the fourth inequality uses that $g_{x^*, L_1^{(f)},\ldots,L_d^{(f)}}(x^*) = 1$, so that $\pi(\alpha^* h(x^*))_{i^*})$ is distributed as a uniformly random bit when $h \leftarrow S_f$.

Because the above argument holds for any decision tree T that does not query lines in $\mathcal{L}_{x^*}$, it follows that

$$\mathbb{E}_f[\Pr[\mathrm{Dec}^{f,\mathcal{L}_{x^*}}(\alpha^*, x^*, i^*) = \pi(\alpha^* f(x^*))_{i^*}]] = \mathbb{E}_r[\mathbb{E}_f[\mathbf{1}(T_{(\alpha^*,x^*,i^*,r)}(f, \mathcal{L}_{x^*}) = \pi(\alpha^* f(x^*))_{i^*})]] = \frac{1}{2}\,,$$

as $T(\cdot, \mathcal{L}_{x^*})$ is a decision tree that cannot query lines in $\mathcal{L}_{x^*}$. This shows that the decoder's success probability cannot exceed $1/2$ in expectation over a random codeword, and hence there exists a codeword where the success probability is $\leq 1/2$. This finishes the proof up to the proof of Claim 7.34, which we do below. □

Proof of Claim 7.34. We will first show that for any line L and any point $x \notin L$, there is a degree-1 polynomial $g_{x,L}$ satisfying $g_{x,L}(x) = 1$ and $g_{x,L}(y) = 0$ for all $y \in L$. Let $L = \{a + \lambda b\}$ where $a \in \mathbb{F}_{2^t}^n$, $b \in \mathbb{F}_{2^t}^n \setminus \{0^n\}$, and $\lambda \in \mathbb{F}_{2^t}$. Since $x \notin L$, there exists $v \in \mathbb{F}_{2^t}^n$ such that $\langle x - a, v\rangle \neq 0$ and $\langle b, v\rangle = 0$. Indeed, this follows because $x \notin L$ implies that $x - a$ and b are linearly independent, in which case the subspaces $\mathrm{span}\{b\}$ and $\mathrm{span}\{x - a, b\}$ are distinct subspaces of dimension 1 and 2, respectively. The vector v is simply any vector in $(\mathrm{span}\{b\})^\perp \setminus (\mathrm{span}\{x-a, b\})^\perp$, which is nonempty since the two subspaces are distinct.

With the vector v, we now let $g_{x,L}(z) \coloneqq \langle z-a, v\rangle / \langle x-a, v\rangle$, which is a degree-1 polynomial that is well-defined since $\langle x - a, v\rangle \neq 0$. We have that $g_{x,L}(x) = 1$, and

for any $y \in L$, i.e., $y = a + \lambda b$, we have that $g_{x,L}(a + \lambda b) = \lambda\langle b, v\rangle / \langle x - a, v\rangle = 0$ since $\langle b, v\rangle = 0$.

Finally, to finish the proof, we simply take $g_{x,L_1,\ldots,L_q}$ to be $\prod_{i=1}^{q} g_{x,L_i}$. □

7.6 Lower Bound for Linear 2-RLDCs Over Any Finite Field

In this section, we extend Theorem 7.8 to any finite field $\mathbb{F}$, for the specific case of $q = 2$.

Theorem 7.35. *Let $C\colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear $(2, \delta, 1, s)$-RLDC with a nonadaptive decoder. Then, there exists a linear $(2, \delta/2, 1, s)$-LDC $C'\colon \mathbb{F}^{k'} \to \mathbb{F}^n$ where $k' \geq k - \lfloor 2/\delta \rfloor$ (and analogously for 2-RLCCs implying 2-LCCs). By [DS07, Theorem 1.2], this implies that $n \geq 2^{\Omega_{\delta,s}(k)}$.*

Theorem 7.35 thus extends the exponential lower bound of [BBC+23] for 2-query binary RLDCs to arbitrary finite fields. It also extends the proof ideas to get the corresponding result that 2-RLCCs give 2-LCCs. With Theorem 7.23, we can extend this lower bound to 2-RLDCs and 2-RLCCs with imperfect completeness and adaptive decoders.

Corollary 7.36. *Let $C\colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear $(2, \delta, 1-\varepsilon, s)$-RLDC. Then, there exists a linear $(2, \delta/2, 1, s + (2 + 1/(|\mathbb{F}| - 1))\varepsilon)$-LDC $C'\colon \mathbb{F}^{k'} \to \mathbb{F}^n$ where $k' \geq k - \lfloor 2/\delta \rfloor$ (and analogously for 2-RLCCs implying 2-LCCs). By [DS07, Theorem 1.2], this implies that $n \geq 2^{\Omega_{\delta,s,\varepsilon}(k)}$.*

To prove Theorem 7.35, we adapt the approach of [BBC+23] to show that every linear 2-RLDC (or 2-RLCC) over *any* finite field must also be a 2-LDC (or 2-LCC).[10] We will show that the relaxed local decoder has two "modes," one of which behaves like a non-relaxed local decoder, and the other which necessarily has poor

[10] In fact, assuming linear structure eliminates much of the casework necessary in [BBC+23]. It would be interesting to extend this proof to nonlinear codes over large alphabets.

soundness. Thus, if the 2-RLDC or 2-RLCC has decent soundness, we can eliminate the latter case to build a 2-LDC or 2-LCC.

At a high level, we can use the structure of a linear code from Fact 5.15 to argue that every local view of the decoder either looks like a Hadamard code ("smooth" case) or a repetition code ("nonsmooth" case). Then, we can observe that the latter case must have high soundness error, and so our 2-RLDC must essentially be a Hadamard code, which is a 2-LDC.

7.6.1 Smooth (Hadamard code) vs. nonsmooth (repetition code) cases

Let $\mathrm{Dec}(\cdot)$ be the relaxed local decoder (or corrector) for C satisfying the assumed decoding radius, completeness, and soundness parameters. Assuming that the decoder is nonadaptive, we can assume that $\mathrm{Dec}^y(i)$ behaves as follows:

1. Sample a query pair $(j, \ell) \leftarrow \mathcal{Q}_i$ with a corresponding decoding function $f^i_{j,\ell} \colon \mathbb{F}^2 \to \mathbb{F} \cup \{\perp\}$.
2. Query y at indices j, ℓ and return the evaluation of $f^i_{j,\ell}(y_j, y_\ell)$.

Consider each query pair (j, ℓ) from the support of $\mathcal{Q}_i$.

- Suppose C is an RLDC. Then we say that j is *fixed* by i if there is some $\alpha \in \mathbb{F}$ such that $C(b)_j = \alpha b_i$ for all b.
- If C is an RLCC, then we say that j is *fixed* by i if there is some $\alpha \in \mathbb{F}$ such that $C(b)_j = \alpha C(b)_i$ for all b.

Let $S_i \subseteq [n]$ be the set of codeword indices fixed by i. Note that because the encoding is linear, every $j \in [n]$ is either unfixed, or fixed by exactly one index. Then, we can consider each case:

1. If both j and ℓ are not fixed by i, then the decoding function $f^i_{j,\ell}$ cannot return $\perp$, because $C|_{\{j,\ell\}}$ has dimension 2 and any local view can be completed to a valid codeword due to Fact 5.6. Thus, $f^i_{j,\ell}$ always returns a linear combination of its two inputs, which behaves like the local decoder of a *Hadamard code*. This is the "smooth" case described in Section 7.2.1.

2. If both j and ℓ are fixed by i, then $C|_{\{j,\ell\}}$ has dimension 1. The truth table of the decoding function $f^i_{j,\ell}$ has exactly $|\mathbb{F}|^2 - |\mathbb{F}|$ entries that are $\perp$. Because both entries are multiples of the i-th message symbol, this local view looks like a *repetition code*. This is necessarily the "nonsmooth" case because these codeword indices only contain information on a single message index.

3. If j is fixed by i but ℓ is not fixed by i (or vice versa), then $C|_{\{j,\ell\}}$ has dimension 2 and so $f^i_{j,\ell}$ never returns $\perp$. In addition, by perfect completeness, the decoding function must take the form $f^i_{j,\ell}(y_j, y_\ell) = \alpha^{-1} y_j$, and so it does not depend on the ℓ-th symbol at all. Hence, we can treat (j, ℓ) as if it queried only j, which is a repetition code case.

Thus, without loss of generality, either both queries are fixed by i (and the decoder essentially tests equality between two copies of i) or both are not fixed by i (and the decoder cannot test and must always return a linear combination of the two symbols). We can show that as long as $|S_i|$ is not too large, the repetition code case can be fooled with certainty (compare to the higher query case in Lemma 7.19, where the tests can be more sophisticated). This implies that the decoder has good soundness when conditioned on choosing a Hadamard-like query case, where it never returns $\perp$.

Claim 7.37. *Suppose C is a 2-RLDC satisfying the premise of Theorem 7.35 and suppose $i \in [k]$ such that $|S_i| \leq \delta n/2$. Then, for all y such that $\Delta(y, C(b)) \leq \delta n/2$,*

$$\Pr[\mathrm{Dec}^y(i) = b_i \mid \textit{Hadamard-like query case}] \geq 1 - s\,.$$

Analogously for a 2-RLCC C and for $i \in [n]$ satisfying the same conditions,

$$\Pr[\mathrm{Dec}^y(i) = C(b)_i \mid \textit{Hadamard-like query case}] \geq 1 - s\,.$$

Proof. Let $E \coloneqq j \in S_i \wedge k \in S_i$ be the event of the "repetition code" case. We will bound the behavior of the decoder conditioned on $\neg E$ (i.e., conditioned on the Hadamard case) by tampering with y and introducing errors designed to completely fool the repetition code case. We mildly reduce the decoding radius from δ to $\delta/2$ to give us the breathing room to introduce these errors, which are only used for the analysis.

Begin by finding a codeword c that differs in the desired symbol:

- If C is a 2-RLDC, then let $c \coloneqq C(b + e_i)$, so that it differs in the i-th message symbol.
- If C is a 2-RLCC, let c be any arbitrary codeword such that $c_i \neq C(b)_i$; one must exist unless $C|_{\{i\}} = \{0^n\}$.

Then, let y' be the string formed by taking y and then setting $y'|_{S_i} = c|_{S_i}$, i.e., replace the symbols at S_i with those from c. Using the triangle inequality, $\Delta(y', C(b)) \leq \delta n$, and so y' is subject to the soundness error property of C:

- If C is a 2-RLDC, then

  $$1 - s \leq \Pr[E] \cdot \Pr[\mathrm{Dec}^{y'}(i) \in \{b_i, \perp\} \mid E] + \Pr[\neg E] \cdot \Pr[\mathrm{Dec}^{y'}(i) \in \{b_i, \perp\} \mid \neg E]\,.$$

- If C is a 2-RLCC, then

  $$1 - s \leq \Pr[E] \cdot \Pr[\mathrm{Dec}^{y'}(i) \in \{C(b)_i, \perp\} \mid E] + \Pr[\neg E] \cdot \Pr[\mathrm{Dec}^{y'}(i) \in \{C(b)_i, \perp\} \mid \neg E]\,.$$

However, conditioned on event E, our decoder Dec makes two queries to indices in S_i, and hence sees a local view indistinguishable from c, which we purposely chose

to trick our decoder. By the perfect completeness property, the decoder must return $b_i + 1 \neq b_i$ (or $c_i \neq C(b)_i$) with probability 1. This gives a lower bound on the conditional probability of the decoder returning the right answer:

$$1 - s \leq \Pr[\neg E] \cdot \Pr[\mathrm{Dec}^{y'}(i) \in \{b_i, \perp\} \mid \neg E] \leq \Pr[\mathrm{Dec}^{y}(i) = b_i \mid \neg E]\,,$$

or for a 2-RLCC, $\Pr[\mathrm{Dec}^{y}(i) = C(b)_i \mid \neg E] \geq 1 - s$. We use two key observations here:

1. Recall that $y|_{[n]\setminus S_i} = y'|_{[n]\setminus S_i}$. Conditioned on $\neg E$, both queries are outside of S_i, and since $y'|_{S_i} = y|_{S_i}$, we can replace y' with y on the right hand side.

2. In addition, the decoder cannot return $\perp$ because every possible pair of queried values will agree with some codeword. Hence, $\mathrm{Dec}(\cdot)$ must return b_i (or $C(b)_i$) to satisfy perfect completeness. □

7.6.2 Repetition case is rare

If $|S_i| \leq \delta n/2$ for every $i \in [k]$ (or every $i \in [n]$), then Claim 7.37 would prove that C is a $(2, \delta/2, 1, s)$-LDC (or LCC), because we can simply condition our decoder's query distribution to pick a Hadamard-like local view and get the required soundness error property. This may not be true for every i, but we can use a simple counting argument to show that it holds for nearly all of the message indices i. Thus, C itself is not necessarily a 2-LDC (or 2-LCC), but it does "contain" a 2-LDC (or 2-LCC) with nearly the same parameters.

Claim 7.38. *Let $X = \{i : |S_i| > \delta n/2\}$. Then, $|X| \leq 2/\delta$.*

Proof. Recall that a codeword index j cannot be fixed by two different (message for RLDC or codeword for RLCC) indices. Thus, the sets S_i are mutually disjoint, so

$$\delta |X| n/2 < \sum_{i \in X} |S_i| \leq n \implies |X| < 2/\delta\,.$$

Note that the sets S_i may not be mutually disjoint if the code is nonlinear. Indeed, in this case a codeword index could be fixed by many different message indices, which (for the binary case) requires a careful analysis in [BBC+23]. □

Now, we have the tools to prove Theorem 7.35. We will remove the indices in X from the code, either by fixing them to zero for the RLDC case or by adding linear constraints setting them to zero for the RLCC case. Then, every remaining message or codeword index either satisfies Claim 7.37 or has its value fixed.

Proof of Theorem 7.35. Let $k' \coloneqq k - \lfloor 2/\delta \rfloor$. Let C be the RLDC. Without loss of generality, we may assume that $X = \{k'+1, k'+2, \ldots, k\}$. Let $C' \colon \mathbb{F}^{k'} \to \mathbb{F}^n$ be defined as $C'(b) = C((b, 0^{k-k'}))$, i.e., append $k - k' = |X|$ zeroes to the message and encode using C. Then, every $i \in [k']$ satisfies $|S_i| \leq \delta n/2$, and so Claim 7.37 gives us a local decoder for all $i \in [k']$ which never returns $\perp$. This gives the required $(\delta/2, s)$-soundness error property. The query complexity and perfect completeness properties are inherited from the original decoder, showing that C' is a $(2, \delta/2, 1, s)$-LDC. Finally, we apply the 2-LDC lower bound of [DS07, Theorem 1.2] which gives

$$n \geq 2^{\left(1 - \frac{1}{|\mathbb{F}|} - s\right) \cdot \frac{\delta}{2} \cdot \frac{k'}{4} - 1}.$$

Since $|\mathbb{F}| \geq 2$, this implies

$$n \geq 2^{\frac{(1/2 - s)\delta k'}{8} - 1} = 2^{\Omega_{\delta, s}(k)}.$$

For C which is an RLCC, take an arbitrary parity check matrix $B \in \mathbb{F}^{(n-k) \times n}$ for C, and then add each $e_i \in \mathbb{F}^n$ for $i \in X$ as rows to B. This defines a new code $C' \subseteq C$ with dimension at least k' where the indices at X are always zero, and we can pick an arbitrary basis to get the encoding map $C' \colon \mathbb{F}^{k'} \to \mathbb{F}^n$. If $i \in [n]$ is not in X, then it satisfies Claim 7.37 yielding a local corrector for i that never returns $\perp$; if $i \in [n]$ is in X, we can now use a trivial local corrector that always returns zero. Thus, C' is a $(2, \delta/2, 1, s)$-LCC. □

7.7 Linear LDCs and LCCs Have Strong Soundness

In this section, we show that any linear LDC or LCC satisfying Definitions 5.8 and 5.9 additionally has strong soundness (Definition 7.9). More formally, we will show the following lemma.

Lemma 7.39. *Let $C\colon \mathbb{F}^k \to \mathbb{F}^n$ be a linear (q, δ, c, s)-LDC (LCC) with $s < 1 - \frac{1}{|\mathbb{F}|}$. Then, there is a decoder $\mathrm{Dec}(\cdot)$ satisfying the following conditions:*

(1) (q-queries) for any y and i, the algorithm $\mathrm{Dec}^y(i)$ reads at most q indices of y

(2) (perfect completeness) for all $b \in \mathbb{F}^k$ and $i \in [k]$, $\Pr[\mathrm{Dec}^{C(b)}(i) = b_i] = 1$,

(3) ((δ/q)-strong soundness) for all $b \in \mathbb{F}^k$, $i \in [k]$, and all $y \in \mathbb{F}^n$ with $\Delta(y, C(b)) \leq \delta n/q$, $\Pr[\mathrm{Dec}^y(i) \neq b_i] \leq \frac{q\Delta(y,C(b))}{\delta n}$.

Proof. By [Dvi16a], given any (q, δ, c, s)-LDC, there exist collections $\mathcal{H}_1, \ldots, \mathcal{H}_k$ where (1) each $\mathcal{H}_i$ is a set of $|\mathcal{H}_i| \geq \delta n/q$ q-sparse vectors in $\mathbb{F}^n$ with disjoint support, i.e., any $v, v' \in \mathcal{H}_i$, $\mathrm{supp}(v) \cap \mathrm{supp}(v') = \emptyset$, and (2) for every $b \in \mathbb{F}^k$, $i \in [k]$, and $v \in \mathcal{H}_i$, it holds that $b_i = \sum_{j=1}^n v_j x_j$ where $x = C(b)$. Note that the latter sum is actually over at most q indices, as $|\mathrm{supp}(v)| \leq q$.

Consider the decoder $\mathrm{Dec}^y(i)$ that behaves as follows: on input i, sample $v \leftarrow \mathcal{H}_i$ uniformly at random, read y_j for each $j \in \mathrm{supp}(v)$, and output $\sum_{j \in \mathrm{supp}(v)} v_j y_j$. If $y_j = x_j$ for all $j \in \mathrm{supp}(v)$, where $x = C(b)$, then by the above, the decoder outputs b_i.

Finally, $b \in \mathbb{F}^k$, $x = C(b)$, and let $y \in \mathbb{F}^n$ with $\Delta(y, C(b)) \leq \delta n/q$. Let S denote the set of coordinates $j \in [n]$ where $y_j \neq C(b)_j$. Because $\mathrm{supp}(v)$'s are disjoint for $v \in \mathcal{H}_i$, it follows that

$$\begin{aligned}\Pr_{v \leftarrow \mathcal{H}_i}[\mathrm{supp}(v) \cap S \neq \emptyset] &\leq \mathbb{E}_{v \leftarrow \mathcal{H}_i}[|\mathrm{supp}(v) \cap S|] = \frac{1}{|\mathcal{H}_i|} \sum_{v \in \mathcal{H}_i} |\mathrm{supp}(v) \cap S| \\ &= \frac{|(\cup_{v \in \mathcal{H}_i} \mathrm{supp}(v)) \cap S|}{|\mathcal{H}_i|} \leq \frac{|S|}{|\mathcal{H}_i|}\,.\end{aligned}$$

As $|\mathcal{H}_i| \geq \delta n/q$, it follows that the probability the decoder outputs b_i is at least $1 - \frac{q|S|}{\delta n}$, as required.

To prove the statement for LCCs, we use [Dvi16b] instead of [Dvi16a], and the rest of the proof follows immediately. □

Part III

Lower Bounds

Chapter 8: Preliminaries

This chapter collects the notation, computational models, and pseudorandomness tools used throughout the chapters in Part III.

8.1 General Notation

Throughout Part III, log denotes the natural logarithm, and logarithms to base 2 are written explicitly as $\log_2$. For a positive integer n, let $[n] \coloneqq \{1, \dots, n\}$ and let $\binom{[n]}{s}$ be the collection of subsets of $[n]$ of size s. We write $\{0,1\} \coloneqq \{0,1\}$ and $e(x) \coloneqq (-1)^x$. The notation $x \sim D$ means that x is sampled from a distribution D, while $x \sim S$ means that x is sampled uniformly from a finite set S.

For strings x, y, write $x \parallel y$ for concatenation; $\circ$ is reserved for function composition. For a tuple $x = (x_1, \dots, x_n)$, write $x_{\le k} = (x_1, \dots, x_k)$ and use $x_{<k}$, $x_{\ge k}$, and $x_{>k}$ analogously. For $x, y \in \{0,1\}^n$, their absolute Hamming distance is

$$\Delta(x, y) \coloneqq |\{i \in [n] : x_i \neq y_i\}|.$$

8.2 Input Blocks

Whenever a positive integer b divides n, we canonically partition $X \in \{0,1\}^n$ into b contiguous blocks

$$X = (X_1, \dots, X_b), \qquad X_i \in \{0,1\}^{n/b}.$$

The number of blocks will be clear from context. For another n-bit string Y, the notation Y_i refers to its ith block under the same partition, and $X_{-i} \coloneqq (X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_b)$ omits the ith block.

When convenient, we identify a block $X_i \in \{0,1\}^{n/b}$ with the n-bit string obtained by placing X_i in the ith block and filling every other block with zeroes.

Thus expressions such as $X + Y_i$ are well defined when strings are viewed as vectors over $\mathbb{F}_2$.

For functions $f : (\{0,1\}^m)^b \to \{0,1\}$ and $g : \{0,1\}^{n/b} \to \{0,1\}^m$, define

$$f \circ g^b(X) \coloneqq f(g(X_1), \ldots, g(X_b)).$$

8.3 Finite Fields

We work with finite fields of characteristic 2. An element of $\mathbb{F}_{2^n}$ is identified with an n-bit string. Concretely, choose a degree-n polynomial $E(x)$ irreducible over $\mathbb{F}_2[x]$ and realize

$$\mathbb{F}_{2^n} \cong \mathbb{F}_2[x]/E(x).$$

Every field element has a unique representative $\sum_{i=0}^{n-1} c_i x^i$ of degree less than n, which we identify with $(c_{n-1}, \ldots, c_0) \in \{0,1\}^n$. Shorter strings are padded with leading zeroes. Under this identification, field addition is vector addition over $\mathbb{F}_2$, and multiplication is polynomial multiplication modulo $E(x)$. We write

$$\langle x, y \rangle \coloneqq \sum_{i=1}^{n} x_i y_i$$

for the standard inner product over $\mathbb{F}_2$.

Definition 8.1 (Additive character)**.** A map $\chi : \mathbb{F}_{2^n} \to \mathbb{F}_2$ is an additive character if $\chi(x + y) = \chi(x) + \chi(y)$ for all $x, y \in \mathbb{F}_{2^n}$. It is nontrivial if it is not identically zero.

The values on a basis determine an additive character, so there are 2^n characters. They may be represented as $\chi_c(x) = \langle x, c \rangle$, or, after fixing a nontrivial character χ, as $\chi_c(x) = \chi(cx)$.

8.4 Models of Computation

Definition 8.2 ($\mathbb{F}_2$-polynomial)**.** An $\mathbb{F}_2$-polynomial is a Boolean function of the form

$$p(x) = \sum_{S \subseteq [n]} c_S \prod_{i \in S} x_i, \qquad c_S \in \mathbb{F}_2,$$

where all arithmetic is over $\mathbb{F}_2$.

Definition 8.3 (Set-multilinearity)**.** An $\mathbb{F}_2$-polynomial is set-multilinear over a partition $(X_1, \ldots, X_d)$ if every monomial contains at most one variable from each block X_i.

Definition 8.4 (Junta)**.** A Boolean function $f : \{0,1\}^n \to \{0,1\}$ is a k-junta if there are a set $S \subseteq [n]$ of size k and a function $g : \{0,1\}^k \to \{0,1\}$ such that $f(x) = g(x_S)$. We denote this class by $\mathrm{JUNTA}_{n,k}$ and denote the class of parities of t such juntas by $\mathrm{JUNTA}_{n,k}^{\oplus t}$.

Definition 8.5 (k-party NOF protocol)**.** A function $f : (\{0,1\}^{n/k})^k \to \{0,1\}$ is computed by a k-party number-on-forehead protocol with communication cost c if the players communicate at most c bits, player i's messages depend only on X_{-i} and the public transcript, and the final message equals $f(X)$. We denote this class by Π_k^c.

An unqualified Boolean circuit in Part III has fan-in 2, and its size is its number of gates. For the named unbounded-fan-in classes below, size means the number of wires, including input wires. The class AC_d^0 consists of depth-d circuits over $\{\mathrm{AND}, \mathrm{OR}, \mathrm{NOT}\}$ with unbounded fan-in. A SYM gate computes an arbitrary symmetric function, and a THR gate computes an arbitrary linear threshold function. A subscript on a gate, such as AND_k, specifies its fan-in.

Definition 8.6 ($(d, \mathcal{C})$-tree)**.** For an integer d and computational model $\mathcal{C}$, a function is computable by a $(d, \mathcal{C})$-tree if there is a depth-d decision tree such that the function restricted by every root-to-leaf path belongs to $\mathcal{C}$.

8.5 Probability and Limited Independence

Let U_m denote the uniform distribution over $\{0,1\}^m$. The notation $S \subset_p T$ denotes the random subset of T obtained by including each element independently with probability p.

Definition 8.7 (Coordinate-wise k-wise uniformity)**.** A distribution D over $\{0,1\}^n$ is coordinate-wise k-wise uniform if, for every $T \subseteq [n]$ of size k, the marginal $(x_T)_{x\sim D}$ is uniform over $\{0,1\}^k$. A function $G : S \to \{0,1\}^n$ is a coordinate-wise k-wise uniform generator if $(G(s))_{s\sim S}$ has this distribution.

Definition 8.8 (Block-wise k-wise uniformity)**.** Consider a distribution D over $(\{0,1\}^m)^d$. It is block-wise k-wise uniform if, for every set $\{i_1,\ldots,i_k\} \subseteq [d]$ and every $y_1,\ldots,y_k \in \{0,1\}^m$,

$$\Pr_{X\sim D}[X_{i_j} = y_j \text{ for every } j] = 2^{-km}.$$

When the coordinates or blocks are clear, we omit the qualifier.

Definition 8.9 (Statistical closeness)**.** For distributions D_1, D_2 over a finite domain Ω, write $D_1 \approx_\varepsilon D_2$ if, for every $S \subseteq \Omega$,

$$|\Pr_{x\sim D_1}[x \in S] - \Pr_{x\sim D_2}[x \in S]| \le \varepsilon.$$

A crucial property of coordinate-wise 4-wise uniform strings is square-root anticoncentration with constant probability.

Theorem 8.10 ([Ber97, Corollary 3.1])**.** *Let X be a 4-wise uniform distribution over $\{-1,1\}^n$ and let $v \in \mathbb{R}^n$. Then*

$$\Pr_{x\sim X}\left[\left|\sum_{i=1}^n v_i x_i\right| \ge \sqrt{\frac{\sum_{i=1}^n v_i^2}{3}}\right] \ge \frac{2}{11}.$$

We also need a 4-wise uniform generator whose output bits are locally computable by small circuits.

Theorem 8.11. *There is a coordinate-wise 4-wise uniform generator $G : S \to \{0,1\}^{2^n}$ such that, for every $s \in S$ and $x \in [2^n]$, a circuit C_s of size $O(n^2)$ computes $C_s(x) = G(s)_x$.*

Proof. Let $\iota : \mathbb{F}_{2^n} \to \{0,1\}$ return the first bit of the binary representation of a field element, and define

$$G : \mathbb{F}_{2^n}^4 \to \mathbb{F}_2^{2^n}, \qquad G(s) = \left(\iota \left(\sum_{i=1}^{4} s_i x^{i-1} \right) \right)_{x \in \mathbb{F}_{2^n}}.$$

This is a 4-wise uniform generator by [HH24, Theorem 2.2]. For fixed s, each output bit is the evaluation of a degree-3 polynomial over $\mathbb{F}_{2^n}$ and is computable by a size-$O(n^2)$ circuit using grade-school field multiplication. □

8.6 Random Restrictions

A partial assignment or restriction is a string $\rho \in \{0,1,\star\}^n$, where $\star$ denotes a live coordinate. For restrictions ρ^1, ρ^2, define their composition by

$$(\rho^1 \circ \rho^2)_i = \begin{cases} \rho^1_i, & \rho^1_i \neq \star, \\ \rho^2_i, & \rho^1_i = \star. \end{cases}$$

Thus ρ^1 is applied first and ρ^2 assigns some coordinates that remain live.

A random restriction is a distribution over restrictions. We use $\mathcal{R}_p$ for the distribution that independently assigns each coordinate $\star$ with probability p and assigns each of 0 and 1 with probability $(1-p)/2$. For $f : \{0,1\}^n \to \{0,1\}$, define

$$f|_\rho(x) \coloneqq f(\rho \circ x).$$

8.7 Pseudorandomness

Definition 8.12 (ε-PRG)**.** A polynomial-time computable function $G : \{0,1\}^s \to \{0,1\}^n$ is an ε-pseudorandom generator for a class $\mathcal{F}$ of functions $\{0,1\}^n \to \{0,1\}$ if, for every $f \in \mathcal{F}$,

$$\left| \mathbb{E}_{x \sim U_n}[(-1)^{f(x)}] - \mathbb{E}_{z \sim U_s}[(-1)^{f(G(z))}] \right| \leq \varepsilon.$$

We also say that G ε-fools $\mathcal{F}$, and call s its seed length. We use the generator of [ST22], which ε-fools $\mathbb{F}_2$-polynomials with at most S terms using seed length $2^{O(\sqrt{\log S})} \log(1/\varepsilon)$.

Definition 8.13 (Min-entropy)**.** For a distribution D over $\{0,1\}^n$, its min-entropy is

$$H_\infty(D) \coloneqq -\log_2 \left(\max_{x \in \{0,1\}^n} \Pr_{y \sim D}[y = x] \right).$$

In particular, if every point has probability at most 2^{-k}, then $H_\infty(D) \geq k$.

Definition 8.14 (Seeded extractors)**.** A (k, ε)-seeded extractor is a function $\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that $\mathrm{Ext}(\mathbf{X}, \mathbf{W}) \approx_\varepsilon U_m$ whenever $H_\infty(\mathbf{X}) \geq k$ and $\mathbf{W} \sim U_d$ is independent. It is strong if

$$(\mathrm{Ext}(\mathbf{X}, \mathbf{W}), \mathbf{W}) \approx_\varepsilon (U_m, \mathbf{W}),$$

and linear if $\mathrm{Ext}(\cdot, w)$ is linear over $\mathbb{F}_2$ for every fixed seed w. The Leftover Hash Lemma [ILL89] gives a strong linear extractor

$$\mathrm{Ext} : \{0,1\}^n \times \{0,1\}^{2n} \to \{0,1\}^{k - 2\log_2(1/\varepsilon)}.$$

Definition 8.15 (Oblivious bit-fixing source extractor)**.** An (n, k) oblivious bit-fixing source is a distribution over $\{0,1\}^n$ obtained by fixing $n-k$ coordinates and filling the remaining k coordinates with independent uniform bits. A (k, ε) OBF extractor is a function $\mathrm{Ext} : \{0,1\}^n \to \{0,1\}^m$ whose output on every such source is ε-close to U_m. For $k > \sqrt{n}$, Kamp and Zuckerman [KZ07] construct $(k, 2^{-\Omega(k^2/n)})$ OBF extractors with output length $\Omega(k^2/n)$.

8.8 Correlation and Hardness

For Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$ and a distribution D, we say that f has agreement advantage γ over g with respect to D if

$$\Pr_{x \sim D}[f(x) = g(x)] \geq \frac{1}{2} + \gamma.$$

We say that f γ-approximates g over D if $\Pr_{x \sim D}[f(x) = g(x)] \geq \gamma$. If no distribution is specified, it is uniform.

Definition 8.16 (Correlation)**.** For Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$, define

$$\operatorname{corr}_D(f,g) \coloneqq \left|\mathbb{E}_{x\sim D}(-1)^{f(x)+g(x)}\right|.$$

For a class $\mathcal{C}$, let $\operatorname{corr}_D(f,\mathcal{C}) \coloneqq \max_{g\in\mathcal{C}} \operatorname{corr}_D(f,g)$. We omit the subscript when $D = U_n$.

For a function $f : (\{0,1\}^{n/k})^k \to \{0,1\}$, define its k-party norm by

$$R_k(f) \coloneqq \mathbb{E}_{X_1^{(0)},\dots,X_k^{(0)},X_1^{(1)},\dots,X_k^{(1)}} e\left(\sum_{\delta\in\{0,1\}^k} f(X_1^{(\delta_1)},\dots,X_k^{(\delta_k)})\right),$$

where all $2k$ blocks are sampled independently from $U_{n/k}$.

Theorem 8.17 ([VW07])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be arbitrary and let g be computable by a d-party NOF protocol exchanging c bits. Then*

$$R_d(f) \le \operatorname{corr}(f,g) \le 2^c R_d(f)^{1/2^d}.$$

Finally, we use the Nisan–Wigderson hardness-to-randomness framework in the following form, stated in the survey of Hatami and Hoza [HH24].

Theorem 8.18 ([NW94]; [HH24, Theorem 4.2.2])**.** *Let $f : \{0,1\}^n \to \{0,1\}$. Suppose $h : \{0,1\}^r \to \{0,1\}$ is ε-hard for $f \circ \mathrm{JUNTA}_{r,k}$ with respect to the uniform distribution. Then there is a PRG for f with seed length*

$$s = O\left(n^{1/(k+1)}\frac{r^2}{k}\right)$$

and error εn.

Chapter 9: Most Juntas Saturate the Hardcore Lemma

9.1 Introduction

Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function such that every circuit of size s errs on at least a δ-fraction of inputs. How can we amplify the hardness of this function? One approach is to restrict the domain: given a fixed size-s circuit, we select a subset of inputs of density at least 2δ in which half the points come from the error region and half are correct. Such a set forms a *hardcore set*, because on this region the circuit cannot do better than random guessing. Is it possible that there exists a *single* subset of density 2δ that is simultaneously hard for all size-s circuits? Impagliazzo's hardcore lemma establishes the existence of a $\Omega(\delta)$-density hardcore set for *all* circuits of size $s' \ll s$. The version of this lemma with the smallest size degradation from s to s' is the following.

Theorem 9.1 ([Imp95, KS99, BHK09])**.** *Let $\gamma, \delta \in (0, 1/2)$, $n \in \mathbb{N}$, $n \leq s \leq \frac{2^n}{n}$, and $f : \{0,1\}^n \to \{0,1\}$. Suppose that for all circuits C of size at most s,*

$$\Pr_{x \sim \{0,1\}^n}[C(x) = f(x)] \leq 1 - \delta.$$

Then there exists a subset $H \subset \{0,1\}^n$ of density $\Omega(\delta)$ such that for all circuits C of size $O\left(\frac{s\gamma^2}{\log(1/\delta)}\right)$, we have

$$\Pr_{x \sim H}[C(x) = f(x)] \leq \frac{1}{2} + \gamma.$$

Conceptually, the theorem says that circuit hardness can be explained by a subset of "hard inputs" H on which the function looks random to small circuits.[1]

[1] Holenstein [Hol05] gives a set H of optimal density 2δ, but suffers a larger degradation from s to $O\left(\frac{s\gamma^2}{n}\right)$.

This phenomenon has found applications throughout computer science, including hardness amplification [O'D02, Tre05], pseudorandomness [STV99, CL21], cryptography [Hol05], algorithmic fairness [CDV24], combinatorics [RTTV08], and learning theory [BHK09, KS99].

While the hardcore lemma is a remarkable result, a natural question is whether the size degradation $s \to \frac{s\gamma^2}{\log(1/\delta)}$ is necessary. This is formalized by the following conjecture.

Conjecture 9.2. *For any $\delta \in (0, .49), \gamma \in (0, \frac{1}{2})$, $n \in \mathbb{N}$ large enough, and $\Omega\left(\frac{\log(1/\delta)}{\gamma^2}\right) \leq s \leq O_\delta\left(\frac{2^n}{n}\right)$, there exists an f such that*

- *for all circuits C of size $\leq s$,*

$$\Pr_{x \sim \{0,1\}^n}[C(x) = f(x)] \leq 1 - \delta.$$

- *for every subset $H \subset \{0,1\}^n$ of density $\geq \Omega(\delta)$, there exists a circuit C of size $O\left(\frac{s\gamma^2}{\log(1/\delta)}\right)$ such that*

$$\Pr_{x \sim H}[C(x) = f(x)] \geq \frac{1}{2} + \gamma.$$

Such a degradation was shown to be necessary for a certain class of proofs [LTW11], but an unconditional result remained elusive, as proving this theorem appears to require constructing an explicit, mildly approximable function that demands large circuits to strongly approximate it. This felt tantamount to proving breakthrough circuit lower bounds. In a recent work, Blanc, Hayderi, Koch, and Tan [BHKT24] evaded this barrier by arguing about a nonexplicit function. In fact, the proof of Theorem 9.1 first shows the result for H being a δ-smooth distribution over $\{0,1\}^n$ (i.e., no string has more than $\frac{1}{\delta 2^n}$ probability of being sampled), and then uses the distribution to extract a density-δ subset. [BHKT24] proves a tightness result over the more general δ-smooth distributions.

Theorem 9.3 ([BHKT24, Theorem 2])**.** *Let $\delta \in (0,1), \gamma \in \left[\Omega\left(\frac{1}{\sqrt{n}}\right), \frac{1}{2}\right)$, $n \in \mathbb{N}$ large enough. For $s \in [\Omega(\frac{1}{\gamma^2}), O(\frac{2^{\gamma^2 n}}{\gamma^4 n})]$, there exists $f : \{0,1\}^n \to \{0,1\}$ such that*

- *For all circuits C of size $\leq s$,*

$$\Pr_{x\sim\{0,1\}^n}[C(x) = f(x)] \leq 1 - \delta.$$

- *for all δ-smooth distributions H over $\{0,1\}^n$, there exists a circuit C of size $O(s\gamma^2)$ such that*

$$\Pr_{x\sim H}[C(x) = f(x)] \geq \frac{1}{2} + \delta\gamma.$$

The above is a reparametrized version of what appears in [BHKT24], which includes dependencies on δ from their proof (where δ was assumed to be constant).[2] Hence, in the regime $\delta = \Theta(1)$, $\gamma \geq \Omega(1/\sqrt{n})$ and $s = O(2^{\gamma^2 n}/(\gamma^4 n))$, [BHKT24] shows that the γ^2-factor decay in size is tight. Structurally, their argument is very analytically involved and is in multiple stages. In what follows, we say that f has *agreement advantage* γ over g with respect to H if $\Pr_{x\sim H}[f(x) = g(x)] \geq \frac{1}{2} + \gamma$, and that f γ-approximates g over H if $\Pr_{x\sim H}[f(x) = g(x)] \geq \gamma$.

- Let $k = 1/\gamma^2$. They first prove that for any δ-smooth distribution H, the majority on k bits has agreement advantage $\frac{\delta}{\sqrt{k}}$ over a 1-junta with respect to H, but no $0.01k$-junta has agreement advantage $\frac{1}{4}$ over it under the uniform distribution.[3]

- They then bootstrap this result to show that, for any δ-smooth H, the majority of k random functions on k disjoint $\frac{n}{k}$-bit input blocks has agreement advantage $\frac{\delta}{\sqrt{k}}$ over a size-$O\left(\frac{2^{n/k}}{n/k}\right)$ circuit with respect to H, but requires size $\Omega\left(\frac{k2^{n/k}}{n/k}\right)$ to obtain agreement advantage $\frac{1}{4}$ under the uniform distribution. They accomplish this by introducing an analytic relaxation of junta complexity, using Fourier-analytic noise-stability arguments to equate this relaxation to

[2] [BHKT24, Theorem 2] gives, for any parameters s and γ, a tightness result for a function of input length $n(s,\gamma)$. We have reparametrized to the standard convention of fixing the input length to n, and examining which s, γ are possible (in terms of n).

[3] A k-junta $f : \{0,1\}^n \to \{0,1\}$ is a Boolean function depending only on a subset of $k < n$ input variables. The junta complexity of a function is the smallest k such that the function is a k-junta.

the original measure (up to constant factors), and then using coupling arguments, Fourier-analytic calculations, and subgaussian concentration to analyze the relaxed junta complexity of the random-function ensemble.

We note this junta-to-circuit lifting theorem is interesting in its own right, and towards proving this, they establish a novel direct sum theorem. These techniques are also used in [BHKT24] to tightly characterize the sample-complexity overhead of smooth boosting.

In this chapter, we give a very short proof that a random junta saturates the hardcore lemma in the regime $\delta = \Omega(1)$ over *arbitrary* distributions.

Theorem 9.4. *Let $\delta \in (0, 0.49), \gamma \in (0, \frac{1}{2}), n \in \mathbb{N}$ large enough, and $n \leq s \leq O_\delta\left(\frac{2^n}{n}\right)$. If there exists a constant $\varepsilon > 0$ such that $s \geq 1/\gamma^{2+\varepsilon}$, there exists a function f such that*

- *For all circuits C of size at most s,*

$$\Pr_{x \sim \{0,1\}^n}[C(x) = f(x)] \leq 1 - \delta.$$

- *for all distributions H over $\{0,1\}^n$, there exists a circuit C of size $O(s\gamma^2)$ such that*

$$\Pr_{x \sim H}[C(x) = f(x)] \geq \frac{1}{2} + \gamma.$$

Formally, Theorem 9.3 and Theorem 9.4 are incomparable: Theorem 9.3 holds only for $\gamma \geq \Omega(1/\sqrt{n})$ and $s \in [\Omega(1/\gamma^2), O(2^{\gamma^2 n}/(\gamma^4 n))]$, while Theorem 9.4 holds for any γ and any $s \in [\Omega(1/\gamma^{2+\varepsilon}), O(2^n/n)]$. While incomparable, we note the former region is extremely restrictive, and does not include the common setting of $\gamma = \frac{1}{n}$. Meanwhile, the latter region contains almost all possible (γ, s) pairs. The latter region is short of subsuming the former interval only by an arbitrarily small polynomial factor of $\gamma^{-\varepsilon}$. Removing this ε-slack remains open.

When $s \geq 1/\gamma^{2+\varepsilon}$, Theorem 9.4 improves on Theorem 9.3 in two ways:

- Theorem 9.3 constructs circuits with agreement advantage $\delta\gamma$, while Theorem 9.4 achieves agreement advantage γ without degradation in δ.
- Theorem 9.3 requires H to be δ-smooth, but Theorem 9.4 makes no assumption about H.

These two points allow our theorem to remain meaningful even for $\delta = o(1)$. Interestingly, the second point shows that the $s \to s\gamma^2$ decay is unavoidable *even if we permit H to be of density $o(\delta)$*.

In summary, our Theorem 9.4 is stronger in the regime $s \geq 1/\gamma^{2+\varepsilon}$ or $\gamma = O(1/\sqrt{n})$, but gives inferior bounds when $1/\gamma^2 \leq s \leq (1/\gamma)^{2+\varepsilon}$ and $\gamma = \Omega(1/\sqrt{n})$. We also note that Conjecture 9.2 remains open. In particular, it would be interesting to pin down the optimal dependence of the circuit-size decay on δ.

9.2 Tightness of Impagliazzo's Hardcore Lemma

We will now prove a lemma that shows how to construct small circuit approximators for arbitrary functions using 4-wise uniform generators.

Lemma 9.5. *Let $f : \{0,1\}^n \to \{0,1\}$ be an arbitrary Boolean function, let H be any distribution over $\{0,1\}^n$, and let $\gamma \in (\frac{27}{2^{n/2}}, \frac{1}{2})$. Let $G : \{0,1\}^m \to \{0,1\}^{2^n}$ be a 4-wise uniform generator such that, for each $s \in \{0,1\}^m$, there exists a circuit C_s of size r with $C_s(x) = G(s)_x$. Then there exists a circuit C of size $O\left(\frac{\gamma^2 2^n}{\log(\gamma^2 2^n)} + r\right)$ such that*

$$\Pr_{x \sim H}[C(x) = f(x)] \geq \frac{1}{2} + \gamma.$$

Proof. If $\gamma \geq 1/\sqrt{363}$, then compute f exactly using Lupanov's theorem. This will be a circuit of size $O(2^n/n) = O\left(\frac{\gamma^2 2^n}{\log(\gamma^2 2^n)}\right)$ since $\gamma = \Omega(1)$.

Hence, assume $\gamma < 1/\sqrt{363}$ and define the nonnegative integer $\ell := \lfloor \log_2(1/363\gamma^2) \rfloor$, and note $n - \ell \geq 1$. Let H' be the distribution over $\{0,1\}^{n-\ell}$

defined by the probability mass function

$$H'(c) \coloneqq \Pr_{x \sim H}[x \in c \times \{0,1\}^\ell].$$

This is the induced distribution of H on the subcubes defined by the first $n - \ell$ bits of the input. For each c in the support of H', define the conditional distribution over the subcube $c \times \{0,1\}^\ell$ by the function

$$H_c(y) \coloneqq \Pr_{x \sim H}[x = c \parallel y \mid x \in c \times \{0,1\}^\ell] = \frac{\Pr_{x \sim H}[x = c \parallel y]}{H'(c)}.$$

Let G be the 4-wise uniform generator guaranteed by the hypothesis. For a subcube c, denote the indicator variable

$$I_c(s) \coloneqq \mathbf{1}\left(\left|\sum_{y \in \{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+G(s)_{c\|y}}\right| \geq \sqrt{\frac{\sum_{y \in \{0,1\}^\ell} H_c(y)^2}{3}}\right).$$

By Theorem 8.10, we have that for each c and random s, $\Pr_{s \sim \{0,1\}^m}[I_c(s) = 1] \geq \frac{2}{11}$. Hence,

$$\mathbb{E}_{s \sim \{0,1\}^m}\left[\mathbb{E}_{c \sim H'}[I_c(s)]\right] = \mathbb{E}_s\left[\sum_c H'(c) I_c(s)\right] \geq \frac{2}{11}\sum_c H'(c) = \frac{2}{11},$$

and by an averaging argument there exists a choice of s such that $\Pr_{c \sim H'}[I_c(s) = 1] \geq 2/11$. Fix such an s. Now for any c with $I_c(s) = 1$, we can use Cauchy-Schwarz to lower bound

$$\left|\sum_{y \in \{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+G(s)_{c\|y}}\right| \geq \sqrt{\frac{\sum_{y \in \{0,1\}^\ell} H_c(y)^2}{3}} \geq \sqrt{\frac{(1/2^\ell)\sum_{y \in \{0,1\}^\ell} H_c(y)}{3}} = \sqrt{\frac{1}{2^\ell \cdot 3}}$$

Now define $h : \{0,1\}^{n-\ell} \to \{0,1\}$ by the map

$$h(c) = \mathbf{1}\left(\sum_{y \sim \{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+G(s)_{c\|y}} < 0\right),$$

which encodes whether the subcube is positively or negatively correlated with the 4-wise uniform string. We now set our approximator $g : \{0,1\}^n \to \{0,1\}$ to be $g(c \parallel y) \coloneqq h(c) \oplus G(s)_{c\|y}$.

We can now write the correlation between f and g (with respect to H) as

$$\begin{aligned}
\mathbb{E}_{x\sim H}[(-1)^{f(x)+g(x)}] &= \mathbb{E}_{c\sim H'}\left[\sum_{y\in\{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+g(c\|y)}\right] \\
&= \mathbb{E}_{c\sim H'}\left[(-1)^{h(c)}\sum_{y\in\{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+G(s)_{c\|y}}\right] \\
&= \mathbb{E}_{c\sim H'}\left[\left|\sum_{y\in\{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+G(s)_{c\|y}}\right|\right] \\
&\geq \Pr_{c\sim H'}[I_c(s)]\cdot\mathbb{E}_{c\sim H'}\left[\left|\sum_{y\in\{0,1\}^\ell} H_c(y)(-1)^{f(c\|y)+G(s)_{c\|y}}\right| : I_c(s)=1\right] \\
&\geq \frac{2}{11}\cdot\sqrt{\frac{1}{2^\ell\cdot 3}} \geq 2\gamma.
\end{aligned}$$

Hence,

$$\Pr_{x\sim H}[f(x)=g(x)] = \frac{\mathbb{E}_{x\sim H}[(-1)^{f(x)+g(x)}]+1}{2} \geq \frac{1}{2}+\gamma.$$

Since h depends only on the first $n-\ell$ bits, it can be constructed in circuit size $O\left(\frac{2^{n-\ell}}{n-\ell}\right) = O\left(\frac{\gamma^2 2^n}{\log(\gamma^2 2^n)}\right)$. By construction, $x \to G(s)_x$ can be computed by some size-r circuit C_s. Therefore, g has agreement advantage γ over f and can be computed by a circuit of size $O\left(\frac{\gamma^2 2^n}{\log(\gamma^2 2^n)} + r\right)$, as desired. □

We are now ready to prove Theorem 9.4.

Proof of Theorem 9.4. We will first prove the first item. Pick a function $g : \{0,1\}^k \to \{0,1\}$ uniformly at random, and define $f(x) \coloneqq g(x_{\leq k})$ to be g on the first k bits of the input. By a Chernoff bound, we know that for a fixed circuit $C : \{0,1\}^k \to \{0,1\}$ of size $\leq s$ and $\delta < 0.49$,

$$\Pr_f\left[\Pr_x[g(x)=C(x)] \geq 1-\delta\right] \leq 2^{-\Omega(2^k)}.$$

Set $2^k = \Theta_\delta(s\log s)$. Taking a union bound over all circuits of size s (of which there are $s^{O(s)}$) we note that, with probability at most $s^{O(s)}2^{-\Omega(2^k)} < 1$, g can be $(1-\delta)$-approximated by a size-s circuit. Fix a g that cannot. We now show f cannot be

approximated by any circuit $C : \{0,1\}^n \to \{0,1\}$ of size s. Restricting the last $n-k$ bits of C will result in a circuit on the first k bits of size at most s. Hence, we can deduce

$$\Pr_x[f(x) = C(x)] = \mathbb{E}_{x_{>k}}\left[\Pr_{x_{\leq k}}[g(x_{\leq k}) = C(x_{\leq k}, x_{>k})]\right] < 1 - \delta$$

as desired.

We will now prove this f satisfies the second item. Consider an arbitrary distribution H over $\{0,1\}^n$. The marginal distribution of H on the first k bits will be some distribution H' over $\{0,1\}^k$. As $\gamma > s^{-1/2} > 27 \cdot 2^{-k/2}$ for large enough n, Lemma 9.5 and Theorem 8.11 give a circuit $C : \{0,1\}^k \to \{0,1\}$ of size $O\left(\frac{\gamma^2 2^k}{\log(\gamma^2 2^k)} + k^2\right)$ such that $\Pr_{x\sim H'}[g(x) = C(x)] \geq \frac{1}{2} + \gamma$. Letting $C' : \{0,1\}^n \to \{0,1\}$ be the circuit that applies C to the first k bits, we note that

$$\Pr_{x\sim H}[f(x) = C'(x)] = \Pr_{x\sim H'}[g(x) = C(x)] \geq \frac{1}{2} + \gamma.$$

As C' has the same size as C, and $\frac{\gamma^2 2^k}{\log(\gamma^2 2^k)} + k^2 = O\left(s\gamma^2 \cdot \left(\frac{\log s}{\log(\gamma^2 s)}\right) + \log^2 s\right) = O_\varepsilon(s\gamma^2)$ when $s \geq 1/\gamma^{2+\varepsilon}$, C' is a size-$O_\varepsilon(s\gamma^2)$ circuit with agreement advantage γ over f with respect to H. Consequently, f is a k-junta that cannot be $(1-\delta)$-approximated by a size-s circuit, but, over any H, has agreement advantage γ over a size-$O_\varepsilon(s\gamma^2)$ circuit, as desired. □

Chapter 10: New Pseudorandom Generators and Correlation Bounds Using Extractors

10.1 Introduction

Many central questions in complexity theory revolve around proving limitations of various computational models. For example, there are research programs which seek lower bounds against constant depth circuits, low-degree polynomials over $\mathbb{F}_2$, and perhaps most famously the complexity class P.

Usually, lower bounds against a simple class of n-bit Boolean functions $\mathcal{C}$ are established by demonstrating an explicit function f such that no $g \in \mathcal{C}$ can compute f on every input. This is referred to as *worst-case hardness*. However, we may not be satisfied with this in practice and stipulate that no $g \in \mathcal{C}$ can even *approximate* f. After all, if there exists a g that agrees with f on all but one point, the difference may be impossible to detect in practice. Furthermore, establishing average case hardness against $\mathcal{C}$ can allow us to create PRGs against $\mathcal{C}$ via the "hardness to randomness" framework introduced by Nisan and Wigderson [NW94], as well as show hardness results against related function classes, like the majority of functions in $\mathcal{C}$. The study of correlation bounds captures exactly this *average-case hardness* statement.

To formally define this, let D be a distribution over $\{0,1\}^n$. Define the *correlation* of two Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$ *over* D to be

$$\mathrm{corr}_D(f,g) \coloneqq |\mathbb{E}_{x\sim D}[(-1)^{f(x)+g(x)}]|.$$

We will usually be concerned with $D = U_n$, the uniform distribution, and should be assumed so if no distribution D is specified. Notice that this quantity is a real number in $[0,1]$. For intuition, note that if $f = g$ or $f = 1 - g$, the correlation is 1, whereas if f and g only match on about half the inputs, the correlation becomes small. This fact allows us to observe that correlation is the right notion, as $\mathrm{corr}(f,g)$ being small

implies that g cannot predict f much better than a coin flip. For a function f and a function class $\mathcal{C}$, we define $\mathrm{corr}(f, \mathcal{C}) := \max_{g \in \mathcal{C}} \mathrm{corr}(f, g)$. Hence, the notion of f being average-case hard for $\mathcal{C}$ is captured by $\mathrm{corr}(f, \mathcal{C})$ being small.

In this chapter, we are most interested in the case where $\mathcal{C}$ is the class of low-degree $\mathbb{F}_2[x_1, \ldots, x_n]$ polynomials. Establishing correlation bounds against low-degree $\mathbb{F}_2$ polynomials is an extremely interesting and central question in complexity theory, as it is either necessary or sufficient to understand a plethora of other problems, some of which concern communication protocols, matrix rigidity, and PRGs for circuits. See Viola's survey [Vio22] for a detailed exposition on this rich program.

Unfortunately, there is a "$\log n$-degree barrier" for PRGs and correlation bounds against low-degree polynomials. Current PRGs and correlation bounds are asymptotically tight for constant degree polynomials, but become *trivial* at degree $\log n$ [Vio08]. Getting nontrivial PRGs (or even correlation bounds) against $\log n$-degree polynomials has been a longstanding open problem.

Towards breaking this barrier, researchers have shown strong correlation bounds for *structured* subsets of low-degree $\mathbb{F}_2$-polynomials (such as sparse polynomials [LS11, ST18], tensors [BHH+20], small-read polynomials, and symmetric polynomials [BIJ+21]) with the hope of being able to generalize them. In this chapter, we establish new correlation bounds and PRGs for computation models generalizing some of these polynomials, namely width-2 branching programs reading d bits at a time, AC^0 containing a small number of arbitrary symmetric or linear threshold gates, and set-multilinear polynomials.

Interestingly, all of these correlation bounds are obtained by taking a function hard for a more specific class of polynomials and then *fortifying* it with a well-suited extractor. Although such a strategy is not new and has been used to establish stronger lower bounds for formulas [KRT13, CKK+14], they usually rely on the fact that upon randomly fixing a subset of variables of a formula, there are extremely few possibilities for the resulting function. Our results show that extractor fortification is a much

broader technique that can strengthen lower bounds against function classes even if they do not simplify to a tiny collection of functions under a random restriction. In particular, our correlation bounds demonstrate that extractor fortification can work as long as the function class, after a random restriction, has low communication complexity or good algebraic structure.

Inspired by this, we would like to understand when extractors will strengthen correlation bounds. Due to technical reasons, this seems nontrivial to establish. We explain this technical nuance in Section 10.2.5.

The remainder of this section is devoted to introducing and motivating each computational model studied, surveying prior work in the topic, and stating all key results proven.

10.1.1 Better Bounds and PRGs Against AC^0 with More $\{\mathrm{SYM}, \mathrm{THR}\}$ Gates

Our knowledge of hardness and PRG results for AC^0 is much more developed than that of TC^0. The state-of-the-art PRG for AC^0 is Lyu's construction [Lyu22], which ε-fools polysize AC^0 circuits with seed length $\tilde{O}(\log^{d-1}(n)\log(n/\varepsilon))$, whereas the current best PRG of Hatami, Hoza, Tal, and Tell (2^{-n^δ})-fools size-$O(n^{1+\delta})$ TC^0 circuits with seed length $O(n^{1-\delta})$ [HHTT22]. Due to this stark contrast in parameters, it is natural to gradually work upward from AC^0 by allotting a budget of SYM (calculates an arbitrary symmetric function) or THR (calculates an arbitrary linear threshold function) gates in the circuit. This approach has been investigated for over a decade [Vio07, LS11, ST18], beginning with Viola's work, which generalized and simplified pioneering work on PRGs for low-degree $\mathbb{F}_2$-polynomials by Luby, Velicković, and Wigderson [LVW93]. This context explains why this circuit class is a compelling generalization of sparse polynomials (which can be written as a small-size parity of ANDs). All the mentioned works use the following function introduced by Razborov and Wigderson in 1993 [RW93] (all arithmetic is over $\mathbb{F}_2$).

$$\mathrm{RW}_{m,k,r}(x) = \sum_{i=1}^{m} \prod_{j=1}^{k} \sum_{\ell=1}^{r} x_{ij\ell} \tag{10.1}$$

Most recently, Servedio and Tan [ST18] use $\mathrm{RW}_{m,k,r}$ to uncorrelate against constant-depth size-$n^{O(\log n)}$ AC^0 circuits whose top gate is $\{\mathrm{SYM}, \mathrm{THR}\}$ (denoted $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$). Their explicit bound is

$$\mathrm{corr}\left(\mathrm{RW}_{\sqrt{\frac{n}{\log_2 n}}, \log_2 n, \sqrt{\frac{n}{\log_2 n}}}, \ \{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0\right) \leq 2^{-\Omega(n^{.499})}.$$

By the techniques used in [LS11], this can be translated into correlation bounds against AC^0 circuits with up to $n^{.499}$ SYM gates or $n^{.249}$ THR gates. As can be seen by the repeated occurrences of $n^{.499}$, the strength of the correlation bound dictates how many $\{\mathrm{SYM}, \mathrm{THR}\}$ gates we can afford in our budget.

We show that RW is just one of many functions from a general class of hard functions with small correlation against $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuits. For functions $f : (\{0,1\}^r)^k \to \{0,1\}$ and $g : \{0,1\}^m \to \{0,1\}^r$, denote $f \circ g^k(x_1 \ldots, x_k) \coloneqq f(g(x_1), \ldots g(x_k))$.

Theorem 10.1 (informal)**.** *Let g be computable by a size $n^{O(\log n)}$ $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuit. Let f be average-case hard against multiparty protocols*[1]*, and let* Ext *be a suitable extractor. Then*

$$\mathrm{corr}(f \circ \mathrm{Ext}^{.01 \log_2 n}, g) \leq 2^{-\Omega(n^{.999})}.$$

To our knowledge, this theorem gives the first context in which generically precomposing with an extractor boosts correlation bounds whose proof does not rely on simplification under random restriction (indeed, parity does not simplify under

[1] the formal condition is any function with small "k-party norm" or "cube norm", but this is currently the only technique we know that establishes average case hardness against multiparty protocols.

restriction and is contained in $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$). Previously, extractors have only been used to boost correlation bounds for classes that heavily simplify under random restriction [KRT13, CKK+14, CSS16, KKL17], whether it be a small depth decision tree or a function very close to a constant.[2] Our theorem states that extractors can still boost correlation bounds, even if they were proven using communication complexity or algebraic methods (rather than only random restrictions).

Furthermore, our theorem distills the reason why RW was so effective as a hard function. Quantitatively, we can instantiate the template with a suitable extractor to obtain a new hard function with nearly-optimal correlation bounds.

Due to our strengthened correlation bounds, we can now obtain correlation bounds and PRGs against size-$n^{O(\log n)}$ AC^0 circuits with up to $n^{.999}$ SYM gates or $n^{.499}$ THR gates. Prior to this, no nontrivial correlation bound or PRG was known to handle such large size *and* number of $\{\mathrm{SYM}, \mathrm{THR}\}$ gates ([LS11] could handle the same number of $\{\mathrm{SYM}, \mathrm{THR}\}$ gates but only for $n^{O(\log \log n)}$-size circuits, and [ST18] could handle the same size circuits, but only $n^{.499}$ SYM or $n^{.249}$ THR gates).

Even for $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuits that have only one $\{\mathrm{SYM}, \mathrm{THR}\}$ gate, our correlation bounds yield improved PRGs whose seed length is $2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$, which has a better dependence on ε than in previous work (see Table 1). In fact, since the best correlation bound that one can hope for is $2^{-\Omega(n)}$, this dependence is almost optimal under the Nisan-Wigderson framework, and an alternative approach is needed to reach the optimal dependence of $\log(1/\varepsilon)$. Since any $\log n$-degree $\mathbb{F}_2$ polynomial can be expressed as a $\mathrm{SYM} \circ \mathrm{AND}_{\log_2 n}$ circuit of size $n^{\log n}$, any improvement in the dependence of the seed length on S would give nontrivial PRGs for $\log n$-degree polynomials, a breakthrough result.

[2]There have been uses of extractors as a hard function against classes that do not simplify under restriction, like DNFs of Parities [CS16] and strongly read-once linear branching programs [GPT22, LZ24, CL23]. However, they directly establish a correlation bound against the extractor rather than amplify a weaker hard function by precomposing with an extractor.

	Circuit type	Circuit size S	Correlation bound	PRG seed length
[Vio07]	$\{\text{SYM}, \text{THR}\} \circ \text{AC}^0$	$n^{c \log n}$	$n^{-c_d \log n}$	$2^{O(\sqrt{\log(S/\varepsilon)})}$
[LS11]	$\text{SYM} \circ \text{AC}^0$	$n^{c \log \log n}$	$\exp(-n^{0.999})$	$2^{O\left(\frac{\log S}{\log \log S}\right)} + (\log(1/\varepsilon))^{2.01}$
[LS11]	$\text{THR} \circ \text{AC}^0$	$n^{c \log \log n}$	$\exp(-n^{0.499})$	$2^{O\left(\frac{\log S}{\log \log S}\right)} + (\log(1/\varepsilon))^{4.01}$
[ST18]	$\{\text{SYM}, \text{THR}\} \circ \text{AC}^0_d$	$n^{c \log n}$	$\exp(-\Omega(n^{0.499}))$	$2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{4.01}$
This work	$\{\text{SYM}, \text{THR}\} \circ \text{AC}^0$	$n^{c \log n}$	$\exp(-\Omega(n^{0.999}))$	$2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$

Table 10.1: Correlation bounds against $\{\text{SYM}, \text{THR}\} \circ \text{AC}^0_d$ circuits and the PRGs that follow via the [NW94] framework. In all previous work, the "hard" function used was the RW function, which was first considered by Razborov and Wigderson [RW93]. Our construction uses a better-suited function.

10.1.2 Much Better PRGs Against Width-2 Branching Programs Reading d Bits at a Time

Usually, one constructs PRGs for natural *computational* models, with the idea that we can drastically reduce the randomness we use if the randomized algorithm we are running can be simulated by such a model. Low-degree polynomials form an extremely natural mathematical model with applications to circuit complexity, but some may not view them as a well-grounded computational model and thus may not consider them worth finding a PRG for. However, the work of Bogdanov, Dvir, Verbin, and Yehudayoff [BDVY13] showed the beautiful connection that PRGs for degree d polynomials are also PRGs against a particular model described as *width-2 length-*poly(n) *branching programs which read d bits at a time.*

Definition 10.2 ((d, ℓ, n)-2BP ([BDVY13], adapted))**.** A (d, ℓ, n)-2BP (or more colloquially a width-2 length-ℓ branching program over n bits which reads d bits at a time) is a layered directed acyclic graph, where there are ℓ layers and each layer contains two nodes, which we label by 0 and 1. Each vertex in each layer j is associated with an arbitrary d-bit substring $x|_v$ of the input x. Each node in layer j has 2^d outgoing edges to layer $j+1$ that are labeled by all possible values in $\{0,1\}^d$. On input

x, the computation starts with the first node v_{start} in the first layer, then follows the edge labeled by $x|_{v_{start}}$ onto the second layer, and so on until a node in the last layer is reached. The identity of this last node is the outcome of the computation.

Such branching programs are a well-motivated computation model, which covers computation with only one bit of usable memory, low-degree polynomials, and small width DNFs. The survey of unconditional PRGs by Hatami and Hoza refers to this model as a compelling computational model that places low-degree polynomials in the computational landscape [HH24].

Unfortunately, there is a "$\log n$-degree barrier" for PRGs and correlation bounds against low-degree polynomials. Current PRGs and correlation bounds are asymptotically tight for constant degree polynomials, but become *trivial* at degree $\log n$, as can be seen by the current best known PRG for degree-d polynomials by Viola which has seed length $O(d \log n + d2^d \log(n/\varepsilon))$ [Vio08]. Obtaining nontrivial PRGs (or even correlation bounds) against $\log n$-degree polynomials has been a tantalizing open problem, and thus PRGs for $(d, \mathrm{poly}(n), n)$-2BPs also seemingly appeared to inherit this "$d = \log n$ barrier" due to the reduction result of [BDVY13].

In this chapter, we construct PRGs against $(d, \mathrm{poly}(n), n)$-2BPs with exponentially better seed length, thereby giving nontrivial PRGs even in the regime $d = n^{1-o(1)}$. Define a d-junta to be a function $\phi : \{0,1\}^n \to \{0,1\}$ which is solely dependent on d input bits (i.e. can be written as $\phi'(x_i)_{i \in S}$ for some subset $S \subset [n]$ of size d). To get our shortened seed length, we evade the $\log n$-degree barrier by instead showing the equivalence between PRGs for $(d, \mathrm{poly}(n), n)$-2BPs and PRGs for the XOR of $\mathrm{poly}(n)$ many d-juntas (denoted as $\mathrm{JUNTA}_{n,d}^{\oplus \mathrm{poly}(n)}$). This class is already interesting in its own right, as it can be seen as a generalization of sparse $\mathbb{F}_2$-polynomials and combinatorial checkerboards (defined by Watson [Wat11] and also studied by Gopalan, Meka, Reingold and Zuckerman [GMRZ13]), as well as a specific class of bounded-collusion protocols studied by Chattopadhyay et al. [CGG$^+$20]. However, we are not aware of any literature that studies $\mathrm{JUNTA}_{n,d}^{\oplus m}$ specifically.

Our main technical contribution is the establishment of strong correlation bounds for $\mathrm{JUNTA}_{n,d}^{\oplus \mathrm{poly}(n)}$. In particular, we show the following.

Theorem 10.3. *There exists an explicit function f such that*

$$\mathrm{corr}(f, \mathrm{JUNTA}_{n,d}^{\oplus \mathrm{poly}(n)}) \leq \exp\left(-\frac{n}{d2^{O(\sqrt{\log n})}}\right)$$

By combining this with the "hardness-to-randomness" framework of Nisan and Wigderson [NW94], we construct a PRG of seed length $2^{O(\sqrt{\log n})} d^2 \log^2(1/\varepsilon)$. This is only a quadratic factor away from optimal dependence on d and ε. Improving the dependence on n would be a breakthrough, since if we set $n' = 2^{\sqrt{\log_2 n}}$, a $(d, n, \mathrm{poly}(n))$-2BP can simulate any $\log n'$-degree polynomial over $x_1, \ldots x_{n'}$, and so having seed length $o(n')$ would effectively break the $\log n$-degree barrier for $\mathbb{F}_2$-polynomial PRGs.

Interestingly enough, by combining a "simplification under restriction" approach pioneered by Ajtai and Wigderson [AW85] with a PRG for sparse $\mathbb{F}_2$-polynomials by Servedio and Tan [ST22], we can construct a PRG against $\mathrm{JUNTA}_{n,d}^{\oplus \mathrm{poly}(n)}$, and thus $(d, \mathrm{poly}(n), n)$-2BPs, with seed length $d2^{O(\sqrt{\log(n/\varepsilon)})}$. This gives us an optimal dependence on d, but an exponentially worse dependence on ε. This suggests perhaps that with a combination of these two approaches, one might be able to achieve a seed length of $2^{O(\sqrt{\log n})} d \log(1/\varepsilon)$.

10.1.3 Near-Optimal Bounds Against High Degree Set-Multilinear Polynomials

As explained earlier, a central open question in complexity theory is to establish better-than-$O(1/\sqrt{n})$ nontrivial correlation bounds against $\Omega(\log n)$-degree polynomials. In order to make progress on this question, it is natural to consider structured low-degree $\mathbb{F}_2$-polynomials. This is what the work of Bhrushundi et al. does [BHH+20].

Define a polynomial $p : \{0,1\}^n \to \{0,1\}$ as set-multilinear over a partition $X = (X_1, \ldots, X_d)$ of the input bits if every monomial contains *at most one* variable from each X_i (this is slightly more general than the usual definition of *exactly one*). The work of Bhrushundi et al. [BHH+20] shows that a random degree d set-multilinear tensor has exponentially small correlation against generic degree $d/2$ $\mathbb{F}_2$-polynomials for $d = \Omega(n)$. Toward making this correlation bound explicit, they defined $\mathrm{FFM}(X_1, \ldots, X_d) = \mathrm{lsb}(X_1 \cdot X_2 \cdots X_d)$, where multiplication is done by treating X_i as field elements, and lsb outputs the least significant bit of the string. Bhrushundi et al. were able to give exponentially small correlation bounds against polynomials up to degree $o(n/\log n)$ which are set-multilinear over the fixed partition $(X_1, \ldots, X_d)$. However, this leaves much to be desired. The partition with respect to which the polynomial is set-multilinear over needing to be fixed and dependent on FFM_d feels like an extremely strong and asymmetric condition. Can we uncorrelate against degree $< d$ polynomials set-multilinear over *any* equipartition of X into d parts? Can the parts be unequal? Can we have more than d of them?

We show affirmative answers to all the above questions. If we take $\delta > 0$ to be an arbitrarily small constant, we can obtain exponentially small correlation against degree $< n^\delta$ polynomial for which there exists *some* partition of X into up to $n^{1-\delta}$ (not necessarily equal) parts such that p is set-multilinear over it. Notice that improving $n^{1-\delta}$ parts to n would be a breakthrough, since all polynomials are set-multilinear over the n-partition of $X = (x_1, \ldots, x_n)$.

To do so, we fortify the hard function FFM with an extractor. Let $\mathrm{Ext}(X, W)$ be a strong linear seeded extractor (for each fixing of W, $\mathrm{Ext}(\cdot, W)$ is linear). For some parameter d, define the function

$$\mathrm{ExtFFM}_d(X_1, \ldots, X_d, W) \coloneqq \mathrm{lsb}(\mathrm{Ext}(X_1, W) \cdot \mathrm{Ext}(X_2, W) \cdot \ldots \cdot \mathrm{Ext}(X_d, W)),$$

where multiplication is done over a finite field, and lsb outputs the least significant bit of the string. First note that $\mathrm{ExtFFM}_d(X_1, \ldots, X_d, W)$, for a fixed W, is set-multilinear over $X_1, \ldots, X_d$. Hence, our intuition that set-multilinear polynomials

might correlate the most with the hard function is preserved in ExtFFM as well. Using ExtFFM, we are able to obtain correlation bounds against the more intuitive notion of set-multilinear polynomials, where the structure of the partition does not matter. This gives more leeway since now if we want to implement this approach towards correlation bounds against low-degree polynomials, there is a larger class of set-multilinear polynomials to which we can reduce generic polynomials.

10.2 Technical Overview of the Results

In this section, we give the overview of the proofs of the main results we covered above.

10.2.1 Stronger Correlation Bounds Against $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$

We focus on showing stronger correlation bounds against $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$, since the subsequent arguments turning this into PRGs against AC^0 with a few $\{\mathrm{SYM}, \mathrm{THR}\}$ gates are standard. The blueprint behind this argument follows the "simplification under restrictions" approach of previous works, but most similarly of Tan and Servedio [ST18]. A random restriction is a random partial assignment where for each variable, it is left unfixed (or "alive") with probability p, and is otherwise set to a uniform bit. [ST18] shows that under a random restriction, the hard function $\mathrm{RW}_{m,k,r}$ maintains integrity and uncorrelates with multiparty protocols, while $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ simplifies to a short multiparty protocol. However, the roadblock met in [ST18] that prevents a correlation bound of $2^{-\Omega(n)}$ and only gives one of size $2^{-\Omega(n^{.499})}$ is due to the parameters in $\mathrm{RW}_{m,k,r}$ being in contention with each other. To elucidate, if n is the input size, then we must have $mkr = n$. Via the analysis done in [ST18], the correlation bound ends up being in the form of $2^{-\Omega(m)} + 2^{-\tilde{\Omega}(r)}$, which forces any established correlation bound to be at best $2^{-\Omega(\sqrt{n})}$.

To understand why both conflicting terms appear, we give a quick overview of the argument of [ST18]. First, $\mathrm{RW}_{m,k,r}$ (as defined in Equation (10.1)) can be

thought of as a fortified version of the *generalized inner product*, $\mathrm{GIP}_{m,k}(x_1, \ldots, x_k) \coloneqq \sum_{i=1}^{m} \prod_{j=1}^{k} x_{ij}$, where each variable is now replaced by the parity of r new variables. This is effective against random restrictions, since as long as one of the r copies $x_{ij1}, \ldots, x_{ijr}$ survive the restriction, the corresponding term x_{ij} in GIP will survive. They argue that after applying a random restriction ρ, the $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuit simplifies to a short multiparty protocol, while $\mathrm{RW}_{m,k,r}|_\rho$ is still capable of computing $\mathrm{GIP}_{m/2,k}$ with high probability. Conditioning on this, previous results of Babai, Nisan, and Szegedy [BNS89] show that $\mathrm{GIP}_{m/2,k}$ has $2^{-\Omega(m/2^k)}$ correlation against these multiparty protocols, which explains the emergence of the $2^{-\Omega(m)}$ term in the correlation. Conditioning on $\mathrm{RW}_{m,k,r}|_\rho$ being able to compute $\mathrm{GIP}_{m/2,k}$ introduces an additive error to the correlation corresponding to the probability $\mathrm{RW}_{m,k,r}|_\rho$ *fails* to simplify. [ST18] bounds this by the chance that all r copies of some variable x_{ij} becomes fixed by a restriction, which will be $(1-p)^r \approx \exp(-pr)$, explaining the occurrence of the $2^{-\Omega(r)}$ term in the correlation.

In summary, the argument of [ST18] requires that r be large to strongly fortify the hard function against random restrictions, while m needs to be large to have a stronger correlation bound against multiparty protocols. However, with the constraint $mr \leq n$, we are forced to compromise and reach the setting $m = r \approx \sqrt{n}$.

We now propose an abstraction of the hard function, which naturally yields a stronger correlation bound. If we define $\oplus_{m,r} : (\{0,1\}^r)^m \to \{0,1\}^m$ by

$$\oplus_{m,r}(x_1, \ldots, x_m) \coloneqq \left(\sum_{i=1}^{r} x_{1i}, \ldots, \sum_{i=1}^{r} x_{mi} \right),$$

we observe $\mathrm{RW}_{m,k,r} = \mathrm{GIP}_{m,k} \circ \oplus_{m,r}^{k} (x_1, \ldots, x_k) = \mathrm{GIP}_{m,k}(\oplus_{m,r}(x_1), \ldots, \oplus_{m,r}(x_k))$. The key insight is that our argument can be generalized to not just RW, but any function

$$f \circ \mathrm{Ext}^k \coloneqq f(\mathrm{Ext}(x_1), \ldots, \mathrm{Ext}(x_k))$$

where f is average-case hard for multiparty protocols, and Ext is an *oblivious bit-fixing source extractor* (OBF extractor). Informally, an oblivious bit-fixing source

extractor for min-entropy k is a function Ext such that if $\mathbf{X}$ is uniform over $\{0,1\}^n$ and ρ is a restriction which leaves $\geq k$ bits alive, the output $\mathrm{Ext}(\mathbf{X}|_\rho)$ is close to uniform. Recall our approach first applies a random restriction to simplify our circuit to a small multiparty protocol, which we then deal with using GIP. If the random restriction leaves sufficiently many variables alive with high probability, then $f \circ \mathrm{Ext}^k$ should still behave like f due to Ext being an OBF extractor. Since the circuit is now a multiparty protocol, the average-case hardness of f gives us a correlation bound (the argument is actually not as simple as this: see Section 10.2.5 for an explanation).

Notice in the RW construction and the setting of parameters $m = r \approx \sqrt{n}$, $\oplus_{m,r}$ is an OBF extractor which maps n bits to $\sqrt{n}$ bits. But this means that the input to the outer GIP function will only have $\approx \sqrt{n}$ bits, so the best correlation bound we can hope to achieve is $\exp(-\Omega(\sqrt{n}))$. The restrictions used in the proof leave $n^{.99}$ variables alive with high probability, so intuitively we could hope that all these $n^{.99}$ "bits of randomness" could be preserved for GIP (or in general any f) rather than only $\sqrt{n}$, potentially resulting in an $\exp(-\Omega(n^{.99}))$ correlation bound. We do just this by using a much better OBF extractor of Kamp and Zuckerman [KZ07]. By making this intuition more formal using techniques developed by Viola and Wigderson [VW07], we obtain a $2^{-\Omega(n^{1-O(1)})}$ correlation bound. The idea of replacing parities with better-suited extractors has also appeared in previous work [KRT13, CKK+14, CSS16, KKL17]. However, as discussed earlier, they have only been used to boost lower bound arguments that solely use random restrictions. In this result, we are boosting a lower bound argument incorporating a communication complexity argument, which requires more care (see Section 10.2.5).

10.2.2 PRGs for $\mathrm{JUNTA}^{\oplus t}_{n,d}$ and (d,t,n)-2BPs

Our PRG construction blueprint can be briefly described as follows. We first establish correlation bounds against $\mathrm{JUNTA}^{\oplus t}_{n,d}$. We then put this through the Nisan-Wigderson "hardness vs. randomness" framework to create a PRG against $\mathrm{JUNTA}^{\oplus t}_{n,d}$. We then show that PRGs that fool $\mathrm{JUNTA}^{\oplus t}_{n,d}$ actually fool (d,t,n)-2BPs, making

the $\mathrm{JUNTA}_{n,d}^{\oplus t}$ PRG our final construction. We first discuss why PRGs for $\mathrm{JUNTA}_{n,d}^{\oplus t}$ imply PRGs for (d,t,n)-2BPs, and then discuss the techniques needed to show strong correlation bounds against $\mathrm{JUNTA}_{n,d}^{\oplus t}$.

10.2.2.1 PRGs for $\mathrm{JUNTA}_{n,d}^{\oplus t}$ $\implies$ PRGs for (d,t,n)-2BPs

Adopting the exposition in [HH24], the previous work of [BDVY13] can be outlined as follows. Consider a (d,t,n)-2BP B. Observing that all transition functions in B are d-juntas, one can derive that $B(x) = B'(\phi_1, \ldots, \phi_{2t}(x))$, where B' is a $(1,t,2t)$ branching program. By Fourier expanding B', this can be decomposed as

$$B(x) = \sum_{S \subset [2t]} \widehat{B}(S)(-1)^{\sum_{i \in S} \phi_i(x)}.$$

[BDVY13] shows that $\sum_{S \subset [t]} |\widehat{B}(S)|$ is bounded, so by linearity of expectation and the Triangle Inequality, it suffices to fool the terms $(-1)^{\sum_{i \in S} \phi_i(x)}$. The approach in [BDVY13] makes the observation that each ϕ_i, by virtue of being a d-junta, can be written as a degree d polynomial. Consequently, a PRG for degree d polynomials will fool (d,t,n)-2BPs with seed length $O(d \log n + d2^d \log(n/\varepsilon))$. The issue here is that at $d = \log n$, the seed length becomes trivial.

However, we can notice that the $\mathbb{F}_2$-polynomial $p(x) \coloneqq \sum_{i \in S} \phi_i(x)$ has some additional structure. If $t = \mathrm{poly}(n)$, p is the sum of only a polynomial number of d-juntas. If there was a way to leverage this, and get a better PRG that fools $\mathrm{JUNTA}_{n,d}^{\oplus t}$, then we might hope to get nontrivial PRGs even in the regime $d = \Omega(\log n)$.

This observation already yields nontrivial PRGs for $d = \omega(\log n)$. Servedio and Tan [ST18] provide a PRG fooling $\mathbb{F}_2$-polynomials with S terms with seed length $2^{O(\sqrt{\log S})} \log(1/\varepsilon)$. Since each junta can be written as a polynomial with up to 2^d terms, each $g \in \mathrm{JUNTA}_{n,d}^{\mathrm{poly}(n)}$ can be written as a polynomial with $S = 2^d \,\mathrm{poly}(n)$ terms, yielding a PRG with seed length $(2^{O(\sqrt{d})} + O(\log n)) \log(1/\varepsilon)$. Hence we get nontrivial seed length for $d = o(\log^2 n)$ [3]. However, we proceed alternatively to get

[3] it is actually the case that a PRG from [LVW93] already gets nontrivial seed length in the same

an exponentially better seed length.

10.2.3 The Nisan-Wigderson Framework and Correlation Bounds for $\mathrm{JUNTA}_{n,d}^{\oplus \mathrm{poly}(n)}$

We will once again use $f \circ \mathrm{Ext}^k$ [4] as our hard function to establish exponentially small correlation bounds against the class, and then apply the Nisan-Wigderson [NW94] framework to construct the PRG. The latter portion is straightforward, so we focus on establishing the correlation bounds.

Let $g \in \mathrm{JUNTA}_{n,d}^{\oplus \mathrm{poly}(n)}$. We first show that there exists a subset of variables, S, such that when arbitrarily fixing bits outside of this set, g can be expressed as a sparse $\mathbb{F}_2$ polynomial, while each input block of $f \circ \mathrm{Ext}^k$ heavily intersects S. Hence if we fix $X_{\bar{S}}$ and take the correlation over S, each input block still maintains high min-entropy while g becomes a sparse polynomial, which is a small $\mathrm{SYM} \circ \mathrm{AC}^0$ circuit. Since the hard function is also the same, we can then apply techniques in the previous section to conclude.

10.2.4 Correlation Bounds against Set-Multilinear Polynomials

Recall that [BHH+20] has shown FFM_d uncorrelates against any lower degree polynomial which is set-multilinear over $(X_1, \ldots, X_d)$. The key ingredient behind proving strong correlation bounds against set-multilinear polynomials over arbitrary partitions is to first fortify each input block with extractors, and instead consider ExtFFM_d. This allows us to establish the following structural lemma, which intuitively states that even if you do not start out with a polynomial that is set-multilinear over $(X_1, \ldots, X_d)$, if not too many bits in each input block can be restricted to 1s such that the resulting function is set-multilinear over $(X_1, \ldots, X_d)$ induced by the live variables in each block, exponential correlation bounds can still be obtained.

---

regime, albeit with exponentially worse dependence in ε

[4] we also precompose with parities in the formal argument

Theorem 10.4. *Let $c \geq 1$, let g be a polynomial of degree $< d$, and let $S_1, \ldots, S_d \subseteq [n/d]$ satisfy $|S_i| \geq n/(cd)$ for every $i \in [d]$. Let ρ denote the restriction created by fixing the bits in X_i whose indices lie outside S_i to 1 for each $i \in [d]$. If $g|_\rho(X_1, \ldots, X_d)$ is set-multilinear in $(X_1, \ldots, X_d)$, then*

$$\operatorname{corr}(\mathrm{ExtFFM}_d, g) \leq 2^{-\Omega(n/(cd))}.$$

To explain the proof at a high level, if the sets S_i we leave alive aren't too small, then our strong extractor (conditioned on a good seed) will keep each block $\mathrm{Ext}(X_i, W)$ approximately uniform, and since the restricted function $g|_\rho$ is now set-multilinear over $(X_1, \ldots, X_d)$ we may use a similar approach as [BHH+20] to prove the theorem.

It turns out that through a combinatorial argument, one can show that polynomials which are set-multilinear over a large number of blocks can be turned into polynomials set-multilinear over $(X_1, \ldots, X_d)$ by fixing not too many bits per input block X_i. The correlation bounds then follow from the structural lemma.

10.2.5 A Technical Nuance Regarding Extractor Fortification

Say we are given a class $\mathcal{C}$ that simplifies under restrictions proportionally to how aggressive the restriction is. For example, say with high probability under a mild random restriction ρ_1, $C|_{\rho_1}$ simplifies to a class $\mathcal{C}_1 \subset \mathcal{C}$, and with high probability under a more aggressive random restriction $\rho_1 \circ \rho_2$, $\mathcal{C}|_{\rho_1 \circ \rho_2}$ simplifies to a much smaller class $\mathcal{C}_2 \subset \mathcal{C}_1$. A common technique to prove correlation bounds is to take a hard function $f : \{0,1\}^n \to \{0,1\}$ that remains hard after a random restriction. For example, if we know $\mathbb{E}_{\rho_1} \operatorname{corr}(f|_{\rho_1}, \mathcal{C}_1) = \varepsilon_1$, we can bound

$$\operatorname{corr}(f, \mathcal{C}) \leq \mathbb{E}_{\rho_1} \operatorname{corr}(f|_{\rho_1}, \mathcal{C}|_{\rho_1}) \approx \mathbb{E}_{\rho_1} \operatorname{corr}(f|_{\rho_1}, \mathcal{C}_1) = \varepsilon_1,$$

where $\approx$ hides the probability $\mathcal{C}$ does not simplify. With this, one might hypothesize that if we precompose with an OBF extractor $\mathrm{Ext} : \{0,1\}^m \to \{0,1\}^n$, one must have $\operatorname{corr}(f \circ \mathrm{Ext}, \mathcal{C}) \ll \varepsilon_1$.

In particular, if $\mathcal{C}_2$ is much simpler than $\mathcal{C}_1$, we would expect $\varepsilon_2 := \operatorname{corr}(f|_{\rho_1}, \mathcal{C}_2) \ll \operatorname{corr}(f|_{\rho_1}, \mathcal{C}_1) = \varepsilon_1$. Then, by the extractor property, we would have $f \circ \operatorname{Ext}|_{\rho_2}(U) \approx f(U')$, where U and U' are uniform strings (but could be correlated depending on the live variables of ρ_2). Hence,

$$\operatorname{corr}(f \circ \operatorname{Ext}|_{\rho_2}, \mathcal{C}|_{\rho_2}) = |\mathbb{E}(-1)^{f \circ \operatorname{Ext}|_{\rho_2}(U) + g|_{\rho_2}(U)}| \approx |\mathbb{E}(-1)^{f(U') + g|_{\rho_2}(U)}|. \tag{10.2}$$

If U' was almost always a copy of U (over the randomness of ρ_2), the above would let us deduce $\operatorname{corr}(f \circ \operatorname{Ext}|_{\rho_2}, \mathcal{C}_{\rho_2}) \approx \operatorname{corr}(f, \mathcal{C}_{\rho_2})$. This allows us to deduce a much stronger correlation bound as follows.

$$\begin{aligned}\operatorname{corr}(f \circ \operatorname{Ext}, \mathcal{C}) \leq \mathbb{E}_{\rho_2} \operatorname{corr}(f \circ \operatorname{Ext}|_{\rho_2}, \mathcal{C}|_{\rho_2}) &\approx \mathbb{E}_{\rho_2} \operatorname{corr}(f, \mathcal{C}|_{\rho_2}) \\ &\leq \mathbb{E}_{\rho_1, \rho_2} \operatorname{corr}(f|_{\rho_1}, \mathcal{C}|_{\rho_1 \circ \rho_2}) \\ &\approx \operatorname{corr}(f|_{\rho_1}, \mathcal{C}_2) = \varepsilon_2.\end{aligned}$$

Of course, we cannot assume something so strong. It is possible that (U', U) are adversarially correlated such that the expression in Equation (10.2) is extremely large. In this chapter, we give two ways to circumvent this obstacle, resulting in three new correlation bounds.

Since the correlation between U and U' is the issue, we might hope to bypass this obstacle by bounding the RHS of Equation (10.2) by an expression independent of $g|_\rho(U)$. This is indeed possible for multiparty protocols, as Viola and Wigderson [VW07] showed the correlation of a function f against k-party protocols is bounded by the "k-party" norm of f. Our correlation bounds for $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ crucially rely on this to make the extractor fortification argument work.

Another way one might imagine a fortification lemma might hold is if $\operatorname{Ext}|_\rho$ was "easy to invert". To see what we mean, let us say for simplicity that $\operatorname{Ext}|_\rho$ was a bijection and had inverse Ext^{-1}. Then we could see by a change of variables

$$|\mathbb{E}(-1)^{f \circ \operatorname{Ext}|_\rho(U) + g|_\rho(U)}| \approx |\mathbb{E}(-1)^{f(U) + g|_\rho(\operatorname{Ext}^{-1}(U))}|.$$

Hence if Ext^{-1} was computationally simple, we might be able to reason that $g|_\rho \circ \text{Ext}^{-1}$ is also very simple, and thus has very small correlation against f. This intuition is what drives the analysis in the correlation bounds against set-multilinear polynomials. We use a linear seeded extractor, which makes the maps linear and thus "easy to invert". Furthermore, this inversion reasoning explains why for our correlation bounds involving multiparty protocols, we need to apply an independent extractor to each party's input rather than one extractor for the whole input. If we did the latter, then the inverse Ext^{-1} would add correlation among the k input strings in $g|_\rho \circ \text{Ext}^{-1}$, thereby *increasing* the communication complexity of this function.

For correlation bounds against width-2 branching programs, we use both phenomena, as we precompose with an OBF extractor, and then again with a simple parity extractor.

It is an interesting question to further explore the connection between extractors and correlation bounds.

10.3 Nearly Optimal Correlation Bounds against $\{\text{SYM}, \text{THR}\} \circ \text{AC}^0$

We strictly improve upon the result [ST18] by proving a stronger correlation bound against $\{\text{SYM}, \text{THR}\} \circ \text{AC}^0$ circuits. This immediately gives PRGs against this class with improved seed length via the "hardness vs. randomness" framework [NW94]. All previous work [Vio07, LS11, ST18] looked at the function introduced in [RW93] created by taking the generalized inner product of parities. We present a new function comprised of field multiplication of extractors in order to prove stronger correlation bounds. Let $m := .0005 \log_2 n$ and $k := n/(m+1)$. We now prove the following result:

Theorem 10.5. *Let* $\text{Ext} : \{0,1\}^k \to \{0,1\}^{.2k^{.996}}$ *be a* $(k^{.998}, 2^{-.4k^{.996}})$ *OBF-source extractor (explicit ones exist due to [KZ07]). Let* $f : (\{0,1\}^{.2k^{.996}})^{m+1} \to \{0,1\}$ *be any*

function such that $R_{m+1}(f) \leq 2^{-\Omega(k^{.996})}$. *Define* $f \circ \mathrm{Ext}^{m+1} : (\{0,1\}^k)^{m+1} \to \{0,1\}$ *by*

$$f \circ \mathrm{Ext}^{m+1}(X) \coloneqq f(\mathrm{Ext}(X_1), \ldots, \mathrm{Ext}(X_{m+1})).$$

Let g *be any function implementable by an* $n^{O(\log n)}$*-size* $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ *circuit. Then*

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, g) \leq 2^{-\Omega(n^{.995})}.$$

In particular, by instantiating this template, say, with Ext being the extractor of [KZ07] and f being either GIP [BNS89] or FFM [FG13], we get explicit $f \circ \mathrm{Ext}^{m+1}$. We also note that, by simply adjusting the constants, we can get any $2^{-\Omega(n^{1-\varepsilon})}$ for constant $\varepsilon > 0$. This gives an improvement over the correlation bound of $2^{-\Omega(n^{.499})}$ given in [ST18].

Proof. We follow the same approach as in [ST18]. The uniform distribution can be expressed as applying a random restriction, and then filling in the remaining bits uniformly. For good random restrictions, we argue that g simplifies to a $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AND}_m$ circuit. We then argue that even after the random restriction, $f \circ \mathrm{Ext}^{m+1}$ maintains its structural integrity due to the extractor. We then finish the argument by using Hastad and Goldmann's connection between $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AND}_m$ and NOF protocols, and the fact that f has small correlation with $(m+1)$-party protocols.

The proof for the simplification of g is the same as seen in [ST18] so we merely cite it here. The only change is the tuning of parameters. Here is the lemma restated for our use.

Lemma 10.6. *Let* $g \in \{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0_d$ *with circuit size* $s = n^{\tau \log_2 n}$. *Then for*

$p = \frac{1}{48}(48 \log_2 s)^{-(d-1)}$

$$\begin{aligned}\Pr_{\rho \leftarrow \mathcal{R}_p}[g|\rho \textit{ is not computed by } (.001pk, \{\mathrm{SYM}_{s^2}, \mathrm{THR}_{s^2}\} \circ \mathrm{AND}_{\log_2 s}\})\textit{-tree}] \\ &\leq s \cdot 2^{-.001pk/2^d} \\ &= 2^{-\Omega_d(pk)}\end{aligned}$$

Notice that for constant d this gives a bound of $2^{-\Omega(n/\operatorname{polylog}(n))}$, versus its use in [ST18] in which a $2^{-\Omega(\sqrt{n/\log n})}$ error was gained. We will see later that we can liberally set parameters here because our hard function maintains integrity even after traversing down a path of size $n/\operatorname{polylog}(n)$ (equivalent to randomly fixing $n/\operatorname{polylog}(n)$ bits), whereas the previous GIP function could only withstand $\sqrt{n}$ bits. This is the result of using an OBF extractor with much better parameters than simply taking the XOR of many copies.

The circuits at the leaves of our tree now form a much simpler class, but this class is not simple enough. Our correlation bounds can only handle circuits with fan-in $m = O(\log n)$, but we currently have fan-in $\log_2 s = O(\log^2 n)$. Fix a leaf ℓ of the tree, and let $\{C_1, \ldots, C_{s^2}\}$ be a collection of subsets of $[n]$ where C_i contains the $\leq \log_2 s$ indices of the variables that feed into the ith $\mathrm{AND}_{\log_2 s}$ gate in the bottom layer. We now use the following basic fact, as in [LS11] and [ST18], that there is a large subset of variables that minimally intersects each C_i.

Claim 10.7. *A random $\mathbf{L} \subset_q [n]$ (add each element to $\mathbf{L}$ with probability q) satisfies*

$$\Pr[\exists i \in [s^2] \textit{ such that } |C_i \cap \mathbf{L}| > m] \leq s^2 \binom{w}{m} q^m.$$

Instantiating this claim with our parameter setting of m and s, and setting $q = \Theta(n^{-.001})$ tells us

$$\Pr[\exists i \in [s^2] \text{ such that } |C_i \cap \mathbf{L}| > m] \leq \frac{1}{s}.$$

Hence there exists such an $L = L(\ell)$ such that restricting all bits outside L makes only $\leq m$ variables feed into each AND gate as desired.

To summarize, our restriction ρ is sampled by a distribution D specified by these three steps.

1. We first perform restriction $\mathcal{R}_p$,
2. and then randomly restrict $\leq .001pk$ while walking down the depth-$.001pk$ tree to a leaf ℓ,
3. and then randomly restrict all the variables alive in this leaf that is *not* in the $L(\ell)$ set that we showed existed

At the end of this process, we have by the union bound that with all but $2^{-\Omega(pk)}$ probability, $g|_\rho$ becomes a $\{\mathrm{SYM}_{s^2}, \mathrm{THR}_{s^2}\} \circ \mathrm{AND}_m$ circuit.

We now observe what happens to $f \circ \mathrm{Ext}^{m+1}$ under this restriction ρ. We claim $f \circ \mathrm{Ext}^{m+1}$ retains its structure. Our wish is for at least $k^{.998}$ bits in each block to survive. That way, we will have a high entropy oblivious bit-fixing source fed into each extractor, and the function will be able to continue to strongly uncorrelate with m-party protocols. In Step 1, we draw a restriction from $\mathcal{R}_p$. Notice the live variables are distributed like a set $S \subset_p [n]$. We see that by a simple Chernoff and union bound,

$$\Pr_{\mathbf{S} \leftarrow \mathcal{R}_p}\left[\exists i \in [m+1] \text{ such that } |X_i \cap \mathbf{S}| < \frac{pk}{2}\right] \leq (m+1)2^{-\Omega(pk)}$$

Hence except for probability $m2^{-\Omega(pk)} = 2^{-\Omega(n^{1-o(1)})}$, each block X_i will have $\geq pk/2$ live variables. Conditioned on this, when we follow Step 2 and perform a random walk down the decision tree to a leaf, we will assign at most $.001pk$ bits, so we are guaranteed that each block X_i will contain at least $.499pk$ live variables. Step 3 is to take set $L(\ell)$ and arbitrarily restrict variables outside of it. We showed there

exists an $L(\ell)$ which minimally overlaps with the input variables to the $\mathrm{AND}_{\log_2 s}$ gates, but we want it to simultaneously overlap heavily with each block. That way most of the X_i will stay alive after restricting the bits outside of $L(\ell)$. The existence of such an $L(\ell)$ can be established by "completing the probabilistic method" started a few paragraphs above. Conditioning on good restrictions so far, let Y_i denote the variables that survived in X_i (hence $|Y_i| \geq .499pk$). We see that

$$\Pr_{\mathbf{L} \subset_q [n]} \left[\exists i \in [m+1] \text{ such that } |Y_i \cap \mathbf{L}| < \frac{.499pqk}{2} \right] \leq (m+1)2^{-\Omega(pqk)}.$$

Hence, the probability that $\mathbf{L}$ either intersects some C_i too much or some Y_i too little will happen with probability $\leq \frac{1}{s} + (m+1)2^{-\Omega(pqk)} \ll 1$. Thus there exists an $L(\ell)$ such that restricting all variables outside of it will simultaneously simplify g to a $\{\mathrm{SYM}_{s^2}, \mathrm{THR}_{s^2}\} \circ \mathrm{AND}_m$ and also leave at least $\frac{.499pqk}{2} \geq .249k^{.999}/\operatorname{polylog}(n) \gg k^{.998}$ variables alive. Stringing all three steps together, we know that except with probability $2^{-\Omega(pk)}$, our random restriction ρ reduces g to $\{\mathrm{SYM}_{s^2}, \mathrm{THR}_{s^2}\} \circ \mathrm{AND}_m$, while simultaneously keeping $\geq k^{.998}$ variables in each X_i block alive.

We are now in the final phase of the argument where we now directly bound the correlation against the simplified circuit. We first state the results that will convert our circuits to NOF protocols.

Theorem 10.8 ([HG90])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function computed by a size-s $\mathrm{SYM} \circ \mathrm{AND}_m$ circuit. Then for any partition of the n inputs of f into $m+1$ blocks, there is a deterministic NOF $(m+1)$-party communication protocol that computes f using $O(m \log s)$ bits of communication.*

Theorem 10.9 ([Nis93])**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a Boolean function computed by a $\mathrm{THR} \circ \mathrm{AND}_m$ circuit. Then for any partition of the n inputs of f into $m+1$ blocks, there is a randomized NOF $(m+1)$-party communication protocol that computes f with error γ_{err} using $O(m^3 \log n \log(n/\gamma_{err}))$ bits of communication.*

We now need to show an average-case hardness result for $f \circ \mathrm{Ext}^{m+1}|_\rho$ against NOF protocols. To do so, we will first calculate the k-party norm of $f \circ \mathrm{Ext}^{m+1}|_\rho$.

Lemma 10.10. *Let ρ be a restriction which keeps $\geq k^{.998}$ variables in each X_i alive. Then $R_{m+1}(f \circ \mathrm{Ext}^{m+1}|\rho) \leq R_{m+1}(f) + 4(m+1) \cdot 2^{-.4k^{.996}}$.*

Proof. Now notice that

$$R_{m+1}(f \circ \mathrm{Ext}^{m+1}|_\rho) = \mathbb{E}_{X^{(0)},X^{(1)}} e\left(\sum_{\delta \in \{0,1\}^{m+1}} f(\mathrm{Ext}(X_1^{(\delta_1)}|_\rho), \ldots, \mathrm{Ext}(X_{m+1}^{(\delta_{m+1})}|_\rho))\right) \tag{10.3}$$

By assumption on ρ, each $X_i^{(b)}|_\rho$, for $i \in [m+1]$ and $b \in \{0,1\}$, is an OBF source with min-entropy at least $k^{.998}$. Hence

$$\mathrm{Ext}(X_i^{(b)}|_\rho) \approx_{2^{-.4k^{.996}}} U_{.2k^{.996}}.$$

Since the $2(m+1)$ variables $X_i^{(b)}$ are mutually independent, a hybrid argument gives

$$\left(\mathrm{Ext}(X_i^{(b)}|_\rho)\right)_{i\in[m+1],\, b\in\{0,1\}} \approx_{2(m+1)2^{-.4k^{.996}}} \left(Y_i^{(b)}\right)_{i\in[m+1],\, b\in\{0,1\}},$$

where the $Y_i^{(b)}$ are mutually independent and uniform over $\{0,1\}^{.2k^{.996}}$. Therefore, Eq. (10.3) is at most

$$\begin{aligned}&\mathbb{E}_{(Y_i^{(b)})_{i\in[m+1],\, b\in\{0,1\}}} e\left(\sum_{\delta\in\{0,1\}^{m+1}} f\left(Y_1^{(\delta_1)}, \ldots, Y_{m+1}^{(\delta_{m+1})}\right)\right) + 4(m+1)2^{-.4k^{.996}} \\ &\quad = R_{m+1}(f) + 4(m+1)2^{-.4k^{.996}},\end{aligned}$$

as desired.

□

With this, we can show that $f \circ \mathrm{Ext}^{m+1}|_\rho$ uncorrelates against randomized multiparty protocols.

Theorem 10.11. *Let $g : \{0,1\}^n \to \{0,1\}$ be a Boolean function, and let ρ be a restriction such that $X_i|_\rho$ has $\geq k^{.998}$ live variables for each i, and $g|_\rho$ can be computed by an $(m+1)$-party NOF randomized protocol with $\leq c$ bits and with error γ. Then*

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}|_\rho, g|_\rho) \leq 2\gamma + 2^{c-\Omega(k^{.996}/2^m)}.$$

Proof. Notice that a randomized NOF protocol is simply a distribution over deterministic NOF protocols. Let π be some distribution over protocols such that for each x, $\Pr_{P\sim\pi}[P(X) \neq g|_\rho(X)] \leq \gamma$. We now manipulate the correlation as

$$\begin{aligned}
\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}|_\rho, g|_\rho) &= |\mathbb{E}_X(-1)^{f\circ\mathrm{Ext}^{m+1}|_\rho(X)+g|_\rho(X)}| \\
&\leq |\mathbb{E}_{X,P}[(-1)^{f\circ\mathrm{Ext}^{m+1}|_\rho(X)+P(X)}]| + 2\gamma \\
&= 2^c\mathbb{E}_P[R_{m+1}(f \circ \mathrm{Ext}^{m+1}|_\rho)^{1/2^{m+1}}] + 2\gamma \\
&\leq 2\gamma + 2^c(R_{m+1}(f) + 4(m+1)\cdot 2^{-.4k^{.996}})^{1/2^{m+1}} \\
&\leq 2\gamma + 2^c(R_{m+1}(f) + 2^{-\Omega(k^{.996})})^{1/2^{m+1}} \\
&\leq 2\gamma + 2^{c-\Omega(k^{.996}/2^m)}
\end{aligned}$$

□

We now have all the ingredients to finish. Say ρ is good if ρ keeps $\geq k^{.998}$ variables alive in each block X_i *and* $g|_\rho$ is computable by $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AND}_m$. We have shown for $\rho \sim D$, this doesn't happen only with probability $2^{-\Omega(pk)}$. If $g|_\rho$ has a SYM gate at the top, then Theorem 10.8 says the $\mathrm{SYM} \circ \mathrm{AND}_m$ circuit can be computed by a deterministic NOF protocol over $X_1, \ldots, X_{m+1}$ using $O(m \log s)$ bits. Plugging this in to Theorem 10.11 tells us

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}|_\rho, g|_\rho) \leq 2^{m\log_2 s - \Omega(k^{.996}/2^m)} \leq 2^{-\Omega(n^{.995})}.$$

If the top gate is a THR, use Theorem 10.9 with $\gamma_{err} = 2^{-n^{.997}}$ to get that the circuit is a randomized NOF protocol over $X_1, \ldots, X_{m+1}$ using $O(m^3 \log n \log(n/\gamma_{err})) = O(n^{.995})$ bits. Plugging this into Theorem 10.11 gives us a correlation bound of

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}|_\rho, g|_\rho) \leq 2^{n^{.995} - \Omega(k^{.996}/2^m)} \leq 2^{-\Omega(n^{.996})}.$$

In either case we get the same bound, so we can bound

$$\begin{aligned}\operatorname{corr}(f \circ \mathrm{Ext}^{m+1}, g) &= |\mathbb{E}_{\rho\sim D}\mathbb{E}_X(-1)^{f\circ\mathrm{Ext}^{m+1}|_\rho(X)+g|_\rho(X)}| \\ &\leq 2^{-\Omega(pk)} + \mathbb{E}_{\rho\sim D}[|\mathbb{E}_X(-1)^{f\circ\mathrm{Ext}^{m+1}|_\rho(X)+g|_\rho(X)}||\rho \text{ is good}] \\ &\leq 2^{-\Omega(pk)} + 2^{-\Omega(n^{.995})} \\ &= 2^{-\Omega(n^{.995})}.\end{aligned}$$

The theorem is proved. □

Remark 10.12. We note that the original RW function instantiated with different parameters can also get the same strengthened correlation bound. This requires a more nuanced analysis than present in [ST18], and does not extend to general functions of the form $f \circ \mathrm{Ext}^{m+1}$ as it relies on the specific structure of GIP and $\bigoplus$.

To recap the argument for a size s circuit, we first use the multi-switching lemma to reduce to a depth-2 circuit of fan-in $\log_2 s$. We then restrict more variables so that the fan-in reduces to $\sqrt{\log_2 s}$. We then apply correlation bounds for $\sqrt{\log_2 s}$-party protocols to get an error of $\exp(-n/2^{\sqrt{\log_2 s}})$. If one trusts that this error is the bottleneck in the argument, one can imagine running through the above argument again with $s = n^{\Theta(1)}$ to get a better error.

Corollary 10.13. *Let $g(X)$ be a function implementable by an $s = n^{O(1)}$-size $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuit, and let $m = 2^{\sqrt{\log_2 n}}$. Define $k := n/(m+1)$, and let $\mathrm{Ext} : \{0,1\}^k \to \{0,1\}^{k/2^{O(\sqrt{\log n})}}$ be a $(k/2^{O(\sqrt{\log n})}, 2^{-k/2^{O(\sqrt{\log n})}})$-extractor constructed from [KZ07]. Then*

$$\operatorname{corr}(f \circ \mathrm{Ext}^{m+1}, g) \leq 2^{-(n/2^{O(\sqrt{\log s})})}.$$

This refinement will be useful for our correlation bounds against branching programs in the next section.

Proof sketch. We proceed exactly as in Section 10.3 all the way until Claim 10.7. Here we note that upon setting $s = n^{\Theta(1)}$, we only need to set $q = 2^{-\Omega(\sqrt{\log n})}$ to make

the desired probability at most some constant. We then continue with the rest of the argument exactly as normal to get the desired correlation bound. □

From Theorem 10.5, we derive the following two theorems as well.

Theorem 10.14. *There exists an ε-PRG against size-S $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuits with seed length $s = 2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$.*

Proof. First, notice that it is straightforward to adjust the constants in our argument for Theorem 10.5 so that our correlation is $2^{-\Omega(n^{1-\delta})}$ for an arbitrarily small constant δ. Set

$$r := 2^{O(\sqrt{\log(nS^2)})} + (\log(1/\varepsilon))^{1/(1-\delta)}.$$

Theorem 10.5 then gives us a function $f : \{0,1\}^r \to \{0,1\}$ which obtains correlation

$$\exp(-\Omega(r^{1-\delta})) \leq \exp(-\Omega(\log(1/\varepsilon))) \leq \varepsilon$$

against $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuits of size

$$r^{O(\log r)} = 2^{O(\log^2 r)} \leq nS^2.$$

Since a $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0 \circ \mathrm{JUNTA}_{\log_2 n}$-circuit of size S can be trivially expressed by a $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuit of size nS^2, it follows from Theorem 8.18 that we can construct an εn-PRG against size-S $\{\mathrm{SYM}, \mathrm{THR}\} \circ \mathrm{AC}^0$ circuits with seed length

$$O(r^2/\log n) = 2^{O(\sqrt{\log(nS^2)})} + (\log(1/\varepsilon))^{2/(1-\delta)} = 2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$$

for δ small enough and $S \geq n$. Substituting $\varepsilon \leftarrow \varepsilon/n$ gives the desired result. □

Lemma 10.15. *If g is computable by a $\mathrm{JUNTA}_{n^{.997}} \circ \mathrm{SYM} \circ \mathrm{AC}^0$ circuit where each depth-2 $\mathrm{SYM} \circ \mathrm{AC}^0$ subcircuit is of size $s = n^{O(\log n)}$, then*

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, g) \leq 2^{-\Omega(n^{.997})}.$$

Similarly, if g is computable by a $\mathrm{JUNTA}_u \circ \mathrm{THR} \circ \mathrm{AC}^0$ circuit where each depth-2 $\mathrm{THR} \circ \mathrm{AC}^0$ subcircuit is of size $s = n^{O(\log n)}$, then

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, g) \leq 2^{-\Omega(n^{.997}/u)}.$$

Proof. We essentially prove a variant of 10.5 where the circuit class is now $\text{JUNTA}_u \circ \{\text{SYM}, \text{THR}\} \circ \text{AC}^0$, an arbitrary JUNTA_u with size $s = n^{O(\log n)}$ $\{\text{SYM}, \text{THR}\} \circ \text{AC}^0$ circuits hanging from each of the u input wires. u will end up representing the number of $\{\text{SYM}, \text{THR}\}$ gates we use. As this circuit is now of size us, we see a variant of Lemma 10.6 will be shown.

Lemma 10.16. *Let $g \in \text{JUNTA}_u \circ \{\text{SYM}, \text{THR}\} \circ \text{AC}^0_d$ with circuit size $s = un^{\tau \log_2 n}$. Then for $p = \frac{1}{48}(48 \log_2 s)^{-(d-1)}$*

$$\begin{aligned}\Pr_{\rho \leftarrow \mathcal{R}_p}[g|_\rho \textit{ is not computed by } (.001pk, \text{JUNTA}_u \circ \{\text{SYM}_{s^2}, \text{THR}_{s^2}\} \circ \text{AND}_{\log_2 s}\})\textit{-tree}] \\ &\leq s \cdot 2^{-.001pk/2^d} \\ &= 2^{-\Omega_d(pk)}\end{aligned}$$

The remainder of the proof is identical to that of Theorem 10.5 until the final step involving multiparty protocols. At this stage, with probability at least $1 - 2^{-\Omega(n^{.999})}$, $g|_\rho$ has been simplified to a $\text{JUNTA}_u \circ \{\text{SYM}, \text{THR}\} \circ \text{AND}_m$ circuit, and at least $k^{.998}$ variables per input block remain alive.

Case 1 - SYM **Gate**: By a trivial extension of Theorem 10.8, this circuit can be calculated by an $(m+1)$-party protocol using $u \cdot O(m \log s)$ bits (explicitly stated as Fact B.3 in [ST18]). Therefore, by Theorem 10.11,

$$\text{corr}(f \circ \text{Ext}^{m+1}|_\rho, g|_\rho) \leq 2^{-\Omega(um \log s - k^{.998}/2^m)}.$$

Setting $u = n^{.997}$ gets the desired result.

Case 2 - THR **Gate**: By a trivial extension of Theorem 10.9, this circuit can be calculated by an $(m+1)$-party protocol with error γ using $u \cdot O(m^3 \log n \log(n/\gamma))$ bits (explicitly stated as Theorem 13 in [ST18]). Setting $\gamma = 2^{-\Omega(k^{.998}/u)}$, it follows by Theorem 10.11,

$$\text{corr}(f \circ \text{Ext}^{m+1}|_\rho, g|_\rho) \leq 2\gamma + 2^{-\Omega(um^3 \log n \log(n/\gamma) - k^{.998}/2^m)} = 2^{-\Omega(k^{.998}/u)},$$

yielding the desired result. □

Theorem 10.17. *Let* $\mathrm{AC}^0[\mathrm{G}, t, s]$ *be the class of size-*s AC^0 *where* $\leq t$ *gates are allowed to be of type* G*. We then have*

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, \mathrm{AC}^0[\mathrm{SYM}, n^{.996}, n^{O(\log n)}]) \leq 2^{-\Omega(n^{.996})}$$

and

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, \mathrm{AC}^0[\mathrm{THR}, n^{.49}, n^{O(\log n)}]) \leq 2^{-\Omega(n^{.49})}.$$

Proof. We replicate the proof given in [LS11]. Let $g \in \mathrm{AC}^0[\mathrm{G}, t, s]$. Let $G_1, \ldots, G_t$ be the gates which occur in g, sorted such that it respects the topological order (if gate G_i is contained in the subcircuit with top gate G_j, then we must have $i < j$). WLOG, assume the top gate is G_t. Consider the decision tree T computing g which sequentially queries the value of the circuit C_i defined by the subcircuit with top gate G_i for $i = 1, 2, \ldots, t$ with the following caveat: if C_i contains some gate G_j, we must have already queried that subcircuit, and so replace it with the queried bit. With this caveat, all C_i are $\mathrm{G} \circ \mathrm{AC}^0$ circuits of size at most st. Hence, T is a depth-t decision tree which queries $\mathrm{G} \circ \mathrm{AC}^0$ circuits at each step. Note for a path P of T, notice the predicate $h_P(x)$, which indicates whether an input caused a traversal down path P in T, is a $\mathrm{JUNTA}_t \circ G \circ \mathrm{AC}^0$ function, as it simply checks that all the t circuits queried down that path has the desired output. Using the fact that an input x cannot simultaneously indicate multiple accepting paths, it follows that

$$(-1)^{g(x)} = \sum_{\text{accepting } P} (-1)^{h_P(x)} - (N - 1)$$

where N is the number of accepting paths. Therefore,

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, g) \leq \sum_{\text{accepting } P} \mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, h_P) + N \,\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, 0)$$

Noting that $N \leq 2^t$, and each h_P along with 0 can be computed by $\mathrm{G} \circ \mathrm{AC}^0$ of size $\leq st$, it follows from Lemma 10.15 that for $\mathrm{G} = \mathrm{SYM}$,

$$\mathrm{corr}(f \circ \mathrm{Ext}^{m+1}, \mathrm{AC}^0[\mathrm{SYM}, n^{.996}, n^{O(\log n)}]) \leq 2 \cdot 2^{n^{.996}} \cdot 2^{-\Omega(n^{.997})} \leq 2^{-\Omega(n^{.996})},$$

and if G = THR,

$$\operatorname{corr}(f \circ \operatorname{Ext}^{m+1}, \operatorname{AC}^0[\operatorname{THR}, n^{.49}, n^{O(\log n)}]) \leq 2 \cdot 2^{n^{.49}} \cdot 2^{-\Omega(n^{.997}/n^{.49})} \leq 2^{-\Omega(n^{.49})}.$$

□

Via a straightforward proof, almost exactly the same as that of Theorem 10.14, we can use the Nisan-Wigderson framework to turn the above theorem into new PRGs against AC^0 with a limited number of $\{\operatorname{SYM}, \operatorname{THR}\}$ gates.

Theorem 10.18. *There is an efficient ε-PRG which fools $\operatorname{AC}^0[\operatorname{SYM}, n^{.999}, S]$ with seed length $2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{2.01}$ and an ε-PRG which fools $\operatorname{AC}^0[\operatorname{THR}, n^{.499}, S]$ with seed length $2^{O(\sqrt{\log S})} + (\log(1/\varepsilon))^{4.01}$.*

10.4 PRGs against $(d, \operatorname{poly}(n), n)$-2BPs

In this section, we use fortified hard functions to establish strong correlation bounds against the XOR of juntas, $\operatorname{JUNTA}_{n,d}^{\oplus \operatorname{poly}(n)}$. These are then pushed through the Nisan-Wigderson "hardness vs. randomness" framework to create PRGs which can fool $(d, \operatorname{poly}(n), n)$-2BPs. We first establish the correlation bounds, and then we show that this implies our desired PRG.

10.4.1 Correlation Bounds Against $\operatorname{JUNTA}_{n,d}^{\oplus \operatorname{poly}(n)}$

This subsection is devoted to proving the following result.

Theorem 10.19. *Let $m = d \log_2 n$, let h be the hard function in Corollary 10.13 instantiated on $k \coloneqq n/m$ bits, and let $\oplus_m : \{0,1\}^m \to \{0,1\}$ be the parity function on m bits. We then have*

$$\operatorname{corr}(h \circ \oplus_m^k, \operatorname{JUNTA}_{n,d}^{\oplus n^c}) \leq \exp\left(-\frac{n}{d 2^{O(\sqrt{\log n})}}\right).$$

Proof. Consider arbitrary $g \in \operatorname{JUNTA}_{n,d}^{\oplus n^c}$. We will show that there exists a subset $T \subset [n]$ of variables such that upon fixing all variables outside T, g simplifies to a

sparse polynomial, while at least one input variable in each $\oplus_m$ stays alive. Write $g = \sum_{i=1}^{n^c} \phi_i$, where each ϕ_i is a d-junta. Let $S_i \subset [n]$ be the indices of the variables that ϕ_i depends on. Pick $T \subset_{1/d} [n]$. For a fixed i, we can bound

$$\begin{aligned}
\Pr_T[|T \cap S_i| \geq \ell] &\leq \sum_{\substack{S \subset S_i \\ |S| = \ell}} \Pr_T[S \subset T] \\
&= \binom{d}{\ell} \left(\frac{1}{d}\right)^{\ell} \\
&\leq \exp(-\Omega(\ell \log \ell)) \\
&\leq 0.1 n^{-c}.
\end{aligned}$$

for $\ell = \Theta(\log n)$. Union bounding over all i, it follows that

$$\Pr_{\rho \sim \mathcal{R}_{1/d}} [\exists i, |T \cap S_i| \geq \ell] < 0.1. \tag{10.4}$$

Let $X_1, \ldots, X_k$ be the input blocks of size m feeding into h. We can easily calculate

$$\Pr_T [\exists i, X_i \cap T = \emptyset] \leq k(1 - 1/d)^m \leq k \exp(-m/d) = 1/m = o(1). \tag{10.5}$$

Union bounding Equation (10.4) and Equation (10.5), it follows that there exists a subset $T \subset [n]$ that simultaneously intersects at most ℓ variables alive in each junta ϕ_i, and intersects at least one variable in each X_i. By pruning out elements, we can assume WLOG that there is exactly one variable in each X_i.

Since a function over b bits can be written as an $\mathbb{F}_2$-polynomial with up to 2^b terms, it follows for any restriction ρ with $\rho^{-1}(\star) = T$, $\phi_i|_\rho$ is a polynomial with $2^\ell = n^{\Theta(1)}$ terms. Therefore, $g|_\rho$ is a polynomial with $n^{\Theta(1)}$ terms as well, which can be written as a $n^{\Theta(1)}$-sized PAR $\circ$ AND circuit. Furthermore, we know that $h \circ \oplus_m^k|_\rho$ is equivalent to h up to negations of the inputs. As SYM $\circ$ AC^0 is invariant under

shifts of the input, we can appeal to Corollary 10.13 and observe

$$\begin{aligned}\operatorname{corr}(h \circ \oplus_m^k, g) &= |\mathbb{E}_X (-1)^{h \circ \oplus_m^k(X) + g(X)}| \\ &\leq \mathbb{E}_{X_{\overline{T}}} |\mathbb{E}_{X_T} (-1)^{h \circ \oplus_m^k(X_T, X_{\overline{T}}) + g(X_T, X_{\overline{T}})}| \\ &\leq \exp\left(-(n/d)/2^{O(\sqrt{\log n})}\right).\end{aligned}$$

□

10.4.2 Constructing and Analyzing the PRG

With this correlation bound in hand, we can construct good PRGs against the XOR of juntas using the Nisan-Wigderson framework.

Corollary 10.20. *There is an ε-PRG for $\mathrm{JUNTA}_{n,d}^{\oplus n^{\Theta(1)}}$ with seed length $s = 2^{O(\sqrt{\log n})} d^2 \log^2(1/\varepsilon)$.*

Proof. By Theorem 10.19 we have an explicit function $f : \{0,1\}^r \to \{0,1\}$ such that

$$\operatorname{corr}(f, \mathrm{JUNTA}_{r, d \log_2 n}^{\oplus n^{100}}) \leq \exp\left(-\frac{r}{2^{O(\sqrt{\log n})} d \log_2 n}\right).$$

We can set $r \coloneqq d 2^{\Theta(\sqrt{\log n})} \log(1/\varepsilon)$ so that the above correlation is below ε. Therefore, f is ε-hard for

$$\mathrm{JUNTA}_{r, d \log_2 n}^{\oplus n^{100}} \supset \left(\mathrm{JUNTA}_{n,d}^{\oplus n^{100}}\right) \circ \mathrm{JUNTA}_{r, \log_2 n},$$

since the composition of an a-junta with a b-junta is an ab-junta. By Theorem 8.18 with $k \leftarrow \log_2 n$ and r as defined earlier, it follows there exists an ε-PRG for $\mathrm{JUNTA}_{n,d}^{\oplus n^{\Theta(1)}}$ with seed length

$$O\left(\frac{d^2 2^{\Theta(\sqrt{\log n})} \log^2(1/\varepsilon)}{\log n}\right) = 2^{O(\sqrt{\log n})} \cdot d^2 \log^2(1/\varepsilon)$$

as desired. □

We now show that fooling the parity of juntas actually allows us to fool arbitrary functions of juntas as long as the function has low Fourier L_1 norm.

Theorem 10.21. *Let G be an ε-PRG for $\mathrm{JUNTA}_{n,d}^{\oplus m}$, and let $f : \{0,1\}^m \to \{0,1\}$. Then G is an $\varepsilon \cdot L_1(f)$-PRG for $f \circ \mathrm{JUNTA}_{n,d}$.*

Proof. Let $g \in f \circ \mathrm{JUNTA}_{n,d}$, implying that $g(x) = f(\phi_1(x), \ldots, \phi_m(x))$ for some $\phi_i \in \mathrm{JUNTA}_{n,d}$. By Fourier expanding f, it follows that

$$g(x) = \sum_{S \subset [m]} \hat{f}(S)(-1)^{\sum_{i \in S} \phi_i(x)}.$$

Notice that for each S, $(-1)^{\sum_{i \in S} \phi_i(x)} \in \mathrm{JUNTA}_{n,d}^{\oplus m}$. Therefore if we let D be the pseudorandom distribution over $\{0,1\}^n$ induced by the image of G, it follows

$$\begin{aligned}
|\mathbb{E}_{x \sim D}[g(x)] - \mathbb{E}_{x \sim U_n}[g(x)]| &\leq \sum_{S \subset [m]} |\widehat{f}(S)||\mathbb{E}_{x \sim D}[(-1)^{\sum_{i \in S} \phi_i(x)}] - \mathbb{E}_{x \sim U}[(-1)^{\sum_{i \in S} \phi_i(x)}]| \\
&\leq \varepsilon \sum_{S \subset [m]} |\widehat{f}(S)| \\
&= \varepsilon \cdot L_1(f).
\end{aligned}$$

□

Finally, as an application, we show PRGs against (d, t, n)-2BPs, branching programs over n bits with width 2, length t, and reads d bits at a time. We will use the fact that width-2 branching programs which read one bit at a time have low Fourier L_1 norm (a proof can be found in [HH24]).

Lemma 10.22. *If f is a $(1, t, n)$-2BP, then $L_1(f) \leq (t+1)/2$.*

We now use the fact that a (d, t, n)-2BP can be represented by a normal width-2 branching program acting on juntas to prove that the PRG from Corollary 10.20 fools (d, t, n)-2BPs.

Theorem 10.23. *There exists an ε-PRG for (d, n^c, n)-2BPs with seed length $s = 2^{O(\sqrt{\log n})} \cdot d^2 \log^2(n/\varepsilon)$.*

Proof. Given a (d, n^c, n)-2BP B, we note that at each vertex $v \in [2n^c]$ of B, the transition function is some d-junta ϕ_v which will map the d bits read at that vertex to the next vertex to move to. Now consider the $(1, n^c, 2n^c)$-2BP B' defined with the same vertex set as B, and define the transition function for $v \in [2n^c]$ in B' to read the vth bit of the input, and then map to the node in the next layer labeled by that bit. It is easy to see by construction that $B(x) = B'(\phi_1(x), \ldots, \phi_{2n^c}(x))$, which is a function in $B' \circ \mathrm{JUNTA}_{n,d}$. By Theorem 10.21, this can be ε-fooled by an $(\varepsilon/L_1(B'))$-PRG for $\mathrm{JUNTA}_{n,d}^{\oplus 2n^c}$. Using the L_1 bound from Lemma 10.22 and the construction from Corollary 10.20, we see that such a PRG has seed length $2^{O(\sqrt{\log n})} d^2 \log^2(n/\varepsilon)$. □

10.4.3 PRGs for Branching Programs via Ajtai-Wigderson

Here, we use the Ajtai-Wigderson framework to get a PRG whose dependence on d is optimal, but whose dependence on ε is exponentially worse than in Theorem 10.23. This may serve as proof of concept that we should be able to construct a PRG that is optimal in both parameters.

For strings $x, y \in \{0,1\}^n$, define the restriction $x \circledast y \in \{0, 1, \star\}^n$ to be

$$(x \circledast y)_i = \begin{cases} x_i & y_i = 0 \\ \star & y_i = 1 \end{cases}.$$

We now recall the Ajtai-Wigderson framework.

Theorem 10.24 ([AW85], as stated in [HH24], Theorem 5.3.3)**.** *Let $\mathcal{F}$ and $\mathcal{F}_{simp}$ be classes of functions $f : \{0,1\}^n \to \{0,1\}$. Assume that $\mathcal{F}$ is closed under restrictions. Let Z be a random variable over $\{0,1\}^n$ that can be explicitly sampled using s truly random bits such that*

$$\forall f \in \mathcal{F}, \Pr[f|_{U \circledast Z} \in \mathcal{F}_{simp}] \geq 1 - \delta$$

where $U \in \{0,1\}^n$ is uniform random and independent of Z. If $\mathbb{E}[Z_i] \geq p$ and there is an explicit δ-PRG for $\mathcal{F}_{simp}$ with seed length s', then there is a PRG against $\mathcal{F}$ with seed length $O(p^{-1}\log(n/\delta)(s+s'))$ and error $O(p^{-1}\delta\log(n/\delta))$.

We first show the following pseudorandom simplification lemma.

Lemma 10.25. *For sufficiently large n, $\delta \in (0,1)$, and some setting of $\ell = \Theta(\log(n/\delta))$, the following holds. Let $U \sim \{0,1\}^n$ be uniformly random and let Z be $(\frac{\delta}{2n^c})$-almost ℓ-wise close to $\mathrm{Ber}(1/d)^{\otimes n}$. Then it follows for any $f \in \mathrm{JUNTA}_{n,d}^{\oplus n^c}$ that*

$$\Pr[f|_{U \circledast Z} \in \mathrm{JUNTA}_{n,\ell}^{\oplus n^c}] \geq 1 - \delta.$$

Furthermore, Z can be sampled using $O(\log d \log(n/\delta))$ random bits.

Proof. Write $f = \sum_{i=1}^{n^c} \phi_i$, where each ϕ_i is a d-junta. Let $S_i \subset [n]$ be the indices of the variables that ϕ_i depends on. We will show that with high probability, only $\log(n/\delta)$ variables in each S_i will stay alive after applying $U \circledast Z$. Letting $\tilde{Z} \sim \mathrm{Ber}(1/d)^{\otimes n}$,

$$\begin{aligned}
\Pr_Z[\mathrm{wt}(Z_{S_i}) \geq \ell] &\leq \sum_{\substack{S \subset S_i \\ |S| = \ell}} \Pr_Z[Z_S = 1^S] \\
&\leq \frac{\delta}{2n^c} + \sum_{\substack{S \subset S_i \\ |S| = \ell}} \Pr_{\tilde{Z}}[\tilde{Z}_S = 1^S] \\
&= \frac{\delta}{2n^c} + \binom{d}{\ell}\left(\frac{1}{d}\right)^{\ell} \\
&\leq \frac{\delta}{2n^c} + 2^{-\Omega(\ell \log \ell)} \\
&\leq \frac{\delta}{n^c}
\end{aligned}$$

upon setting $\ell = \Theta(\log(n/\delta))$. Union bounding over all i yields that $\Pr[\forall i, \mathrm{wt}(Z_{S_i}) < \ell] \geq 1 - \delta$. Let $\mathcal{E}$ be the aforementioned event. Since $\mathrm{wt}(Z_{S_i})$ is exactly the number of variables that stay alive in $\phi_i|_{U \circledast Z}$ (regardless of U), $\mathcal{E}$ implies that each $\phi_i|_{U \circledast Z}$ becomes an ℓ-junta, which consequently implies $f|_{U \circledast Z} \in \mathrm{JUNTA}_{n,\ell}^{\oplus n^c}$. Hence

$$\Pr_{U,Z}[f|_{U \circledast Z} \in \mathrm{JUNTA}_{n,\ell}^{\oplus n^c}] \geq \Pr_Z[\forall i, \mathrm{wt}(Z_{S_i}) < \ell] \geq 1 - \delta,$$

as desired.

To construct Z, we can let $(Z^{(i)})_{1\le i\le \log_2 d}$ be independently sampled $\frac{\delta}{2n^c \log_2 d}$-almost ℓ-wise uniform strings, and set $Z = \bigwedge_{1\le i\le \log_2 d} Z^{(i)}$. By a simple hybrid argument, it follows that Z has the desired distribution. Each $Z^{(i)}$ is constructed using $O(\log \ell + \log(2n^c \log d/\delta) + \log\log n) = O(\log(n/\delta))$ random bits, and thus Z is constructed using $O(\log d \log(n/\delta))$ bits. □

Next, we show how we can fool $\mathrm{JUNTA}^{\oplus n^c}_{n,\Theta(\log(n/\varepsilon))}$.

Claim 10.26. *There exists a δ-PRG against $\mathrm{JUNTA}^{\oplus n^c}_{n,\Theta(\log(n/\delta))}$ with seed length $2^{O\left(\sqrt{\log(n/\delta)}\right)}$.*

Proof. First notice that we can write out a j-junta as an $\mathbb{F}_2$-polynomial with 2^j terms. Hence, we can write any $f \in \mathrm{JUNTA}^{\oplus n^c}_{n,\Theta(\log(n/\delta))}$ as an $\mathbb{F}_2$-polynomial with $n^c \cdot 2^{\Theta(\log(n/\delta))} = \mathrm{poly}(n/\delta)$ terms. Using the sparse polynomial PRG of Servedio and Tan [ST22] yields a PRG against this sparse polynomial with seed length $2^{\sqrt{\log_2(n/\delta)}}$ as desired. □

We now use the Ajtai-Wigderson approach to combine Lemma 10.25 and Claim 10.26 into a PRG construction.

Theorem 10.27. *There exists an explicit ε-PRG against $\mathrm{JUNTA}^{\oplus n^c}_{n,d}$ with seed length $2^{O(\sqrt{\log(n/\varepsilon)})}d$.*

Proof. Applying Lemma 10.25 and Claim 10.26 to Theorem 10.24, we get a PRG for $\mathrm{JUNTA}^{\oplus n^c}_{n,d}$ with error $O(\delta d \log(n/\delta))$ and seed length

$$O(d\log(n/\delta)(2^{O(\sqrt{\log(n/\delta)})} + \log d \log(n/\delta))) = 2^{O(\sqrt{\log(n/\varepsilon)})}d,$$

where we used $d \le n$. Setting $\delta = \Theta\left(\frac{\varepsilon}{2d\log(n/\varepsilon)}\right)$ gives us error ε and seed length $2^{O(\sqrt{\log(n/\varepsilon)})}d$. □

10.5 Correlation Bounds Against Set-Multilinear Polynomials

Our correlation bound for set-multilinear polynomials follows from an instantiation of the following theorem.

Theorem 10.28. *Let $d \le n$ be an integer, let $0 < \varepsilon < 1/2$, and set $r := k - 2\log_2(1/\varepsilon)$. Let*

$$\mathrm{Ext} : \{0,1\}^{n/d} \times \{0,1\}^{2n/d} \to \{0,1\}^r$$

be a strong linear seeded (k,ε)-extractor with seed length $2n/d$ obtained from the Left-over Hash Lemma [ILL89], and let χ be a nontrivial additive character of $\mathbb{F}_{2^r}$. Define $\mathrm{ExtFFM}_d : \{0,1\}^{n+2n/d} \to \{0,1\}$ by

$$\mathrm{ExtFFM}_d(X, W) := \chi\left(\prod_{i=1}^{d} \mathrm{Ext}(X_i, W)\right).$$

Let $g : \{0,1\}^{n+2n/d} \to \{0,1\}$ be a degree-$< d$ polynomial, and let $S_1, \dots, S_d \subseteq [n/d]$ be subsets of size at least k. Suppose that every restriction obtained by arbitrarily fixing all bits in W and all bits outside S_i in each X_i makes g set-multilinear in $X_1, \dots, X_d$. Then

$$\mathrm{corr}(\mathrm{ExtFFM}_d, g) \le d\varepsilon + (d-1)\left(\frac{1}{2^k\varepsilon^2} + \varepsilon\right).$$

Proof. For brevity, let $f := \mathrm{ExtFFM}_d$. We first average over restrictions ρ that assign the bits of each X_i outside S_i, then over the seed W, and finally over the live variables in the sets S_i. For a fixed ρ, let W_ρ be the set of seeds w such that

$$\mathrm{Ext}(X_i|_\rho, w) \approx_\varepsilon U_r$$

for every $i \in [d]$. Since Ext is strong, a union bound shows that W_ρ contains all but

a $d\varepsilon$ fraction of the seeds. Consequently,

$$\begin{aligned}\operatorname{corr}(f,g) &= \left|\mathbb{E}_{X,W}(-1)^{f(X,W)+g(X,W)}\right| \\ &\le \mathbb{E}_{W,\rho}\left|\mathbb{E}_{X}(-1)^{f|_\rho(X,W)+g|_\rho(X,W)}\right| \\ &\le d\varepsilon + \mathbb{E}_{\rho}\mathbb{E}_{w\sim W_\rho}\left|\mathbb{E}_{X}(-1)^{f|_\rho(X,w)+g|_\rho(X,w)}\right|. \end{aligned} \tag{10.6}$$

Fix ρ and $w \in W_\rho$, and write

$$f'(X) \coloneqq f|_\rho(X,w) \qquad \text{and} \qquad g'(X) \coloneqq g|_\rho(X,w).$$

The polynomial g' is set-multilinear and has degree less than d, so every monomial omits at least one block. Assigning each monomial to its largest omitted block gives a decomposition

$$g'(X) = \sum_{i=1}^{d} g_i(X_{-i}),$$

where g_i is linear in every block X_j with $j > i$. Let α map a linear form $\ell(X_d) = \sum_{j\in[n/d]} c_j X_{d,j}$ to its coefficient vector $(c_j)_{j\in[n/d]}$. Thus $\langle \alpha(\ell), X_d\rangle = \ell(X_d)$. With $e(x) \coloneqq (-1)^x$, we obtain

$$\begin{aligned}\left|\mathbb{E}_{X}(-1)^{f'(X)+g'(X)}\right| &= \left|\mathbb{E}_{X} e\left(f'(X) + \sum_{i=1}^{d-1} g_i(X_{-i}) + g_d(X_{-d})\right)\right| \\ &\le \mathbb{E}_{X_{[d-1]}}\left|\mathbb{E}_{X_d} e\left(\left\langle \alpha\left(f'(X) + \sum_{i=1}^{d-1} g_i(X_{-i})\right), X_d\right\rangle + g_d(X_{-d})\right)\right| \\ &\le \Pr_{X_{[d-1]}}\left[\alpha\left(f'(X) + \sum_{i=1}^{d-1} g_i(X_{-i})\right) = 0\right]. \end{aligned} \tag{10.7}$$

Here f' and g_i for $i < d$ are linear in X_d, while $g_d(X_{-d})$ is independent of X_d. We now repeatedly use the fact that, for a linear map $h : \mathbb{F}_2^m \to \mathbb{F}_2^k$ and any $a \in \mathbb{F}_2^k$, $\Pr_x[h(x) = a] \le \Pr_x[h(x) = 0]$.

$$
\begin{aligned}
&\Pr_{X_{[d-1]}}\left[\alpha\left(f'(X)+\sum_{i=1}^{d-1}g_i(X_{-i})\right)=0\right]\\
&=\mathbb{E}_{X_{[d-2]}}\Pr_{X_{d-1}}\left[\alpha\left(f'(X)+\sum_{i=1}^{d-2}g_i(X_{-i})\right)=\alpha\left(g_{d-1}(X_{-(d-1)})\right)\right]\\
&\le\Pr_{X_{[d-1]}}\left[\alpha\left(f'(X)+\sum_{i=1}^{d-2}g_i(X_{-i})\right)=0\right]\\
&\le\cdots\le\Pr_{X_{[d-1]}}\left[\alpha(f'(X))=0\right].
\end{aligned}
\tag{10.8}
$$

To analyze this probability, we show a quick lemma.

Lemma 10.29. *For a linear form $\ell(X_d)$, $\alpha(\ell(X_d))=0$ if and only if $\ell(X_d)=0$ for all X_d.*

Proof. If $\alpha(\ell)=0$, then for every X_d,

$$\ell(X_d)=\sum_{j=1}^{n/d}\alpha(\ell)_jX_{d,j}=0.$$

Conversely, if $\alpha(\ell)_i\neq 0$ for some i, then for the ith unit vector e_i,

$$\ell(e_i)=\sum_{j=1}^{n/d}\alpha(\ell)_j(e_i)_j=\alpha(\ell)_i\neq 0.$$

□

By Lemma 10.29,

$$\Pr_{X_{[d-1]}}\left[\alpha(f'(X))=0\right]=\Pr_{X_{[d-1]}}\left[\forall X_d,\ \chi\left(\prod_{i=1}^{d}\operatorname{Ext}(X_i|_\rho,w)\right)=0\right].$$

If $\prod_{i=1}^{d-1}\operatorname{Ext}(X_i|_\rho,w)=0$, then f' is identically zero as a function of X_d. Otherwise, f' has the form $\chi(c\operatorname{Ext}(X_d|_\rho,w))$ for some nonzero $c\in\mathbb{F}_{2^r}$. Exactly half of the elements

$Y \in \mathbb{F}_{2^r}$ satisfy $\chi(cY) = 1$. Since $w \in W_\rho$ and $\varepsilon < 1/2$, there is then some $X_d|_\rho$ for which $\chi(c \operatorname{Ext}(X_d|_\rho, w)) = 1$. Thus

$$\begin{aligned}
\Pr_{X_{[d-1]}}[\alpha(f'(X)) = 0] &\leq \Pr_{X_{[d-1]}}\left[\prod_{i=1}^{d-1} \operatorname{Ext}(X_i|_\rho, w) = 0\right] \\
&\leq \sum_{i=1}^{d-1} \Pr_{X_i}[\operatorname{Ext}(X_i|_\rho, w) = 0] \\
&\leq (d-1)\left(2^{-r} + \varepsilon\right) \\
&= (d-1)\left(\frac{1}{2^k \varepsilon^2} + \varepsilon\right).
\end{aligned}$$

Combining this bound with Eqs. (10.6) to (10.8) gives

$$\operatorname{corr}(\operatorname{ExtFFM}_d, g) \leq d\varepsilon + (d-1)\left(\frac{1}{2^k \varepsilon^2} + \varepsilon\right).$$

□

As a very nice application of this structural theorem, we show that we can achieve exponentially small correlation against $n^{O(1)}$-degree polynomials which are set-multilinear over some partition of the input into up to $n^{1-O(1)}$ parts.

Corollary 10.30. *Let g be a degree $< d$ polynomial which is set-multilinear over an arbitrary partition $(A_1, \ldots, A_c)$ of X into c parts. Then*

$$\operatorname{corr}(\operatorname{ExtFFM}_d, g) \leq 2^{-\Omega(n/cd)}.$$

Proof. For each $i \in [n/d]$, define S_i to be the largest set among $\{X_i \cap A_1, \ldots, X_i \cap A_c\}$ (arbitrarily pick one if there are ties). Notice that the sets $\{X_i \cap A_j\}_{j \in [c]}$ partition X_i, and $|X_i| = n/d$. Therefore, we know that each $|S_i| \geq \frac{n/d}{c} = \frac{n}{cd}$. We now claim that any restriction ρ formed by arbitrarily fixing all the bits in X_i which are outside S_i, for each i, will make $g|_\rho$ set-multilinear over $(X_1, \ldots, X_d)$. Assume for the sake of contradiction there existed some monomial in $g|_\rho(X)$ that contained 2 variables from some X_i. Since $S_i \subset X_i$ and $S_j \cap X_i = \emptyset$ for $j \neq i$, both of these variables had to have come

from S_i. But note that $S_i = X_i \cap A_\ell \subset A_\ell$ for some ℓ, and we know no monomial has 2 terms from the same A_i by our assumption of g. This yields our desired contradiction.

Therefore, we can apply Theorem 10.28 on the sets (S_i) with $k = n/cd$ and $\varepsilon = 2^{-.1n/cd}$ to deduce that

$$\mathrm{corr}(f, g) \leq d2^{-.1n/cd} + (d-1)(2^{-.8n/cd} + 2^{-.1n/cd}) = 2^{-\Omega(n/cd)}.$$

□

We can set $c = n^{1-2\delta}$ and $d = n^\delta$ to get $2^{-\Omega(n^\delta)}$ correlation against degree n^δ polynomials which are set-multilinear over some partition of the input into $n^{1-2\delta}$ blocks, which is interesting as these are exponential correlation bounds over the high degree regime. From the expressions, we note that there is a tradeoff between the strength of the correlation bounds, the maximum degree polynomials we fool, and the number of parts in our set-multilinear partition we can handle. Unfortunately plugging in $c = n$, which would allow g to be any low-degree polynomial, doesn't yield any nontrivial correlation bound.

Works Cited

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

[ADM+99] Noga Alon, Martin Dietzfelbinger, Peter Bro Miltersen, Erez Petrank, and Gábor Tardos. Linear hash functions. *Journal of the ACM*, 46(5):667–683, 1999.

[AG24] Omar Alrabiah and Venkatesan Guruswami. Near-tight bounds for 3-query locally correctable binary linear codes via rainbow cycles. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024.

[AGKM23] Omar Alrabiah, Venkatesan Guruswami, Pravesh K. Kothari, and Peter Manohar. A near-cubic lower bound for 3-query locally decodable codes from semirandom CSP refutation. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1438–1448, New York, NY, USA, 2023. Association for Computing Machinery.

[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 (JACM)*, 45(3):501–555, 1998.

[ALRW17] Alexandr Andoni, Thijs Laarhoven, Ilya P. Razenshteyn, and Erik Waingarten. Optimal hashing-based time-space trade-offs for approximate near neighbors. In *SODA*, pages 47–66, 2017.

[AS98] Sanjeev Arora and Shmuel Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM (JACM)*, 45(1):70–122, 1998.

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

[AW85] Miklos Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant depth circuits. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 11–19, 1985.

[Bab18] Martin Babka. A note on the size of largest bins using placement with linear transformations. arXiv:1810.04161, 2018.

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

[BB23] Alexander R. Block and Jeremiah Blocki. Computationally relaxed locally decodable codes, revisited. In *2023 IEEE International Symposium on Information Theory (ISIT)*, pages 2714–2719, 2023.

[BBC+23] Alexander R. Block, Jeremiah Blocki, Kuan Cheng, Elena Grigorescu, Xin Li, Yu Zheng, and Minshen Zhu. On relaxed locally decodable codes for Hamming and insertion-deletion errors. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference (CCC 2023)*, volume 264 of

Leibniz International Proceedings in Informatics (LIPIcs), pages 14:1–14:25, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

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

[BCG+22] Jeremiah Blocki, Kuan Cheng, Elena Grigorescu, Xin Li, Yu Zheng, and Minshen Zhu. Exponential lower bounds for locally decodable and correctable codes for insertions and deletions. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 739–750, 2022.

[BCH+95] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos A. Kiwi, and Madhu Sudan. Linearity testing in characteristic two. In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, USA, 23-25 October 1995*, pages 432–441. IEEE Computer Society, 1995.

[BDVY13] Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff. Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9(7):283–293, 2013.

[Ber97] Bonnie Berger. The fourth moment method. *SIAM Journal on Computing*, 1997.

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

[BGG94] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. Incremental cryptography: The case of hashing and signing. In *Advances in Cryptology – CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 216–233, 1994.

[BGG95] Mihir Bellare, Oded Goldreich, and Shafi Goldwasser. Incremental cryptography and application to virus protection. In *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing*, pages 45–56, 1995.

[BGGZ21] Jeremiah Blocki, Venkata Gandikota, Elena Grigorescu, and Samson Zhou. Relaxed locally correctable codes in computationally bounded channels. *IEEE Trans. Inform. Theory*, 67(7):4338–4360, 2021.

[BGT17] Arnab Bhattacharyya, Sivakanth Gopi, and Avishay Tal. Lower bounds for 2-query LCCs over large alphabet. In Klaus Jansen, José D. P. Rolim, David P. Williamson, and Santosh S. Vempala, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, volume 81 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:20, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BHH+20] Abhishek Bhrushundi, Prahladh Harsha, Pooya Hatami, Swastik Kopparty, and Mrinal Kumar. On Multilinear Forms: Bias, Correlation, and Tensor Rank. In Jarosław Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and*

Techniques (APPROX/RANDOM 2020), volume 176 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[BHK09] Boaz Barak, Moritz Hardt, and Satyen Kale. The uniform hardcore lemma via approximate bregman projections. In *SODA*, 2009.

[BHKL24] Arpon Basu, Jun-Ting Hsieh, Pravesh K. Kothari, and Andrew D. Lin. Improved lower bounds for all odd-query locally decodable codes. *CoRR*, abs/2411.14361, 2024.

[BHKT24] Guy Blanc, Alexandre Hayderi, Caleb Koch, and Li-Yang Tan. The sample complexity of smooth boosting and the tightness of the hardcore theorem. In *FOCS*, 2024.

[BIJ+21] Jaroslaw Blasiok, Peter Ivanov, Yaonan Jin, Chin Ho Lee, Rocco A. Servedio, and Emanuele Viola. Fourier growth of structured f2-polynomials and applications. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 53:1–53:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[BIW10] Omer Barkol, Yuval Ishai, and Enav Weinreb. On locally decodable codes, self-correctable codes, and t-private PIR. *Algorithmica*, 58(4):831–859, 2010.

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

[BKZ20] Jeremiah Blocki, Shubhang Kulkarni, and Samson Zhou. On locally decodable codes in resource bounded channels. In Yael Tauman

Kalai, Adam D. Smith, and Daniel Wichs, editors, *1st Conference on Information-Theoretic Cryptography (ITC 2020)*, volume 163 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:23, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[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.

[BNS89] L. Babai, N. Nisan, and M. Szegedy. Multiparty protocols and logspace-hard pseudorandom sequences. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 1–11, New York, NY, USA, 1989. Association for Computing Machinery.

[BSGH+06] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. Robust PCPs of proximity, shorter PCPs, and applications to coding. *SIAM J. Comput*, 36(4):889–974, January 2006. Publisher: Society for Industrial and Applied Mathematics.

[Bsh26] Nader H. Bshouty. A note on second-order expected maximum-load bounds for binary linear hashing, 2026.

[CDV24] Sílvia Casacuberta, Cynthia Dwork, and Salil Vadhan. Complexity-theoretic implications of multicalibration. In *STOC*, 2024.

[CGG+20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, Ashutosh Kumar, Xin Li, Raghu Meka, and David Zuckerman. Extractors and secret sharing against bounded collusion protocols. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1226–1242, 2020.

[CGH+96] Robert M. Corless, G. H. Gonnet, D. E. G. Hare, D. J. Jeffrey, and D. E. Knuth. On the Lambert W function. *Advances in Computational Mathematics*, 5(1):329–359, 1996.

[CGS22] Alessandro Chiesa, Tom Gur, and Igor Shinkar. Relaxed locally correctable codes with nearly-linear block length and constant query complexity. *SIAM J. Comput.*, 51(6):1839–1865, 2022.

[CGW13] Victor Chen, Elena Grigorescu, and Ronald de Wolf. Error-correcting data structures. *SIAM J. Comput.*, 42(1):84–111, 2013.

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

[CKK+14] Ruiwen Chen, Valentine Kabanets, Antonina Kolokolova, Ronen Shaltiel, and David Zuckerman. Mining circuit lower bound proofs for meta-algorithms. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 262–273, 2014.

[CL21] Lijie Chen and Xin Lyu. Inverse-exponential correlation bounds and extremely rigid matrices from a new derandomized XOR lemma. In *STOC*, 2021.

[CL23] Eshan Chattopadhyay and Jyun-Jie Liao. Hardness Against Linear Branching Programs and More. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference (CCC 2023)*, volume 264 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:27, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

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

[CMV13] Kai-Min Chung, Michael Mitzenmacher, and Salil Vadhan. Why simple hash functions work: Exploiting the entropy in a data stream. *Theory of Computing*, 9(30):897–945, 2013.

[CRSW13] L. Elisa Celis, Omer Reingold, Gil Segev, and Udi Wieder. Balls and bins: Smaller hash families and faster evaluation. *SIAM Journal on Computing*, 42(3):1030–1050, 2013.

[CS16] Gil Cohen and Igor Shinkar. The complexity of dnf of parities. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, ITCS '16, page 47–58, New York, NY, USA, 2016. Association for Computing Machinery.

[CSS16] Ruiwen Chen, Rahul Santhanam, and Srikanth Srinivasan. Average-Case Lower Bounds and Satisfiability Algorithms for Small Threshold Circuits. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1:1–1:35, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[CW79] J. Lawrence Carter and Mark N. Wegman. Universal classes of hash functions. *Journal of Computer and System Sciences*, 18(2):143–154, 1979.

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

[CY23] Gil Cohen and Tal Yankovitz. Asymptotically-good RLCCs with $(\log n)^{2+o(1)}$ queries. Technical Report TR23-110, Electronic Colloquium on Computational Complexity (ECCC), July 2023.

[DD22] Manik Dhar and Zeev Dvir. Linear hashing with ℓ_∞ guarantees and two-sided kakeya bounds. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science*, pages 419–428, 2022.

[DEL+22] Irit Dinur, Shai Evra, Ron Livne, Alexander Lubotzky, and Shahar Mozes. Locally testable codes with constant rate, distance, and locality. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 357–374, New York, NY, USA, 2022. Association for Computing Machinery.

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

[DGY11] Zeev Dvir, Parikshit Gopalan, and Sergey Yekhanin. Matching vector codes. *SIAM Journal on Computing*, 40(4):1154–1178, 2011.

[DMOZ22] Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Nearly optimal pseudorandomness from hardness. *Journal of the ACM*, 69(6), 2022.

[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.

[Dvi12] Zeev Dvir. Incidence theorems and their applications. *CoRR*, abs/1208.5073, 2012.

[Dvi16a] Zeev Dvir. Lecture notes on linear locally decodable codes. `https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC1.pdf`, Fall 2016.

[Dvi16b] Zeev Dvir. Lecture notes on linear locally decodable codes. `https://www.cs.princeton.edu/~zdvir/LDCnotes/LDC8.pdf`, Fall 2016.

[Efr09] Klim Efremenko. 3-query locally decodable codes of subexponential length. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009, Bethesda, MD, USA, May 31 - June 2, 2009*, pages 39–44. ACM, 2009.

[FG13] Jeff Ford and Anna Gál. Hadamard tensors and lower bounds on multiparty communication complexity. *Comput. Complex.*, 22(3):595–622, 2013.

[FKS84] Michael L. Fredman, János Komlós, and Endre Szemerédi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31(3):538–544, 1984.

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, Circuits, and the Polynomial-Time Hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[GHSZ02] V. Guruswami, J. Hastad, M. Sudan, and D. Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1034, 2002.

[GIL+90] Oded Goldreich, Russell Impagliazzo, Leonid A. Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[GKMM26] Elena Grigorescu, Vinayak M. Kumar, Peter Manohar, and Geoffrey Mon. Relaxed vs. full local decodability with few queries: Equivalence and separations for linear codes. In *Proceedings of the 58th Annual ACM*

Symposium on Theory of Computing, STOC '26, page 1409–1416, New York, NY, USA, 2026. Association for Computing Machinery.

[GKO+18] Sivakanth Gopi, Swastik Kopparty, Rafael Oliveira, Noga Ron-Zewi, and Shubhangi Saraf. Locally testable and locally correctable codes approaching the Gilbert–Varshamov bound. *IEEE Trans. Inform. Theory*, 64(8):5813–5831, 2018.

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

[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.

[GL89] Oded Goldreich and Leonid A. Levin. A hard-core predicate for all one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, pages 25–32, 1989.

[GL21] Tom Gur and Oded Lachish. On the power of relaxed local decoding algorithms. *SIAM J. Comput*, 50(2):788–813, 2021.

[GMRZ13] Parikshit Gopalan, Raghu Meka, Omer Reingold, and David Zuckerman. Pseudorandom generators for combinatorial shapes. *SIAM Journal on Computing*, 42(3):1051–1076, 2013.

[Gol23a] Guy Goldberg. Linear relaxed locally decodable and correctable codes do not need adaptivity and two-sided error. Technical Report TR23-067, Electronic Colloquium on Computational Complexity (ECCC), May 2023.

[Gol23b] Oded Goldreich. On the lower bound on the length of relaxed locally decodable codes. Technical Report TR23-064, Electronic Colloquium on Computational Complexity (ECCC), May 2023.

[Gol24] Oded Goldreich. On the relaxed LDC of BGHSV: a survey that corrects the record. *Electron. Colloquium Comput. Complex.*, TR24-078, 2024.

[GPT22] Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. Linear branching programs and directional affine extractors. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[GRR20] Tom Gur, Govind Ramnarayan, and Ron Rothblum. Relaxed locally correctable codes. *Theory Comput.*, 16(18):1–68, 2020.

[Hås86] Johan Håstad. *Computational limitations for small depth circuits*. PhD thesis, Massachusetts Institute of Technology, 1986.

[HG90] J. Hastad and M. Goldmann. On the power of small-depth threshold circuits. In *Proceedings [1990] 31st Annual Symposium on Foundations of Computer Science*, pages 610–618 vol.2, 1990.

[HH24] Pooya Hatami and William M. Hoza. Paradigms for unconditional pseudorandom generators. *Foundations and Trends in Theoretical Computer Science*, 16, 2024.

[HHTT22] Pooya Hatami, William M. Hoza, Avishay Tal, and Roei Tell. Fooling constant-depth threshold circuits (extended abstract). In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 104–115, 2022.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[HKM+24] Jun-Ting Hsieh, Pravesh K. Kothari, Sidhanth Mohanty, David Munhá Correia, and Benny Sudakov. Small even covers, locally decodable codes and restricted subgraphs of edge-colored Kikuchi graphs. *CoRR*, abs/2401.11590, 2024.

[HO08] Brett Hemenway and Rafail Ostrovsky. Public-key locally-decodable codes. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 126–143, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[Hol05] Thomas Holenstein. Key agreement from weak bit agreement. In *STOC*, 2005.

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

[HOW15] Brett Hemenway, Rafail Ostrovsky, and Mary Wootters. Local correctability of expander codes. *Inform. and Comput.*, 243:178–190, 2015. 40th International Colloquium on Automata, Languages and Programming (ICALP 2013).

[IK99] Yuval Ishai and Eyal Kushilevitz. Improved upper bounds on information-theoretic private information retrieval (extended abstract). In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 79–88. ACM, 1999.

[ILL89] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 12–24, New York, NY, USA, 1989. Association for Computing Machinery.

[Imp95] Russell Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, 1995.

[JKZ25] Michael Jaber, Vinayak M. Kumar, and David Zuckerman. Linear hashing is optimal. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 245–255, 2025.

[JM24] Oliver Janzer and Peter Manohar. A $k^{q/q-2}$ lower bound for odd query locally decodable codes from bipartite Kikuchi graphs. *CoRR*, abs/2411.14276, 2024.

[Jus72] J. Justesen. Class of constructive asymptotically good algebraic codes. *IEEE Transactions on Information Theory*, 18(5):652–656, 1972.

[KKL17] Valentine Kabanets, Daniel M. Kane, and Zhenjian Lu. A polynomial restriction lemma with applications. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2017, page 615–628, New York, NY, USA, 2017. Association for Computing Machinery.

[KM24a] Pravesh K. Kothari and Peter Manohar. An exponential lower bound for linear 3-query locally correctable codes. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*, pages 776–787. ACM, 2024.

[KM24b] Pravesh K. Kothari and Peter Manohar. Exponential lower bounds for smooth 3-LCCs and sharp bounds for designs. In *65th IEEE Annual*

Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024. IEEE, 2024.

[KM24c] Vinayak M. Kumar and Geoffrey Mon. Relaxed local correctability from local testing. *SIAM Journal on Computing*, 0(0):STOC24–100–STOC24–119, 2024.

[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.

[Knu19] Mathias Bæk Tejs Knudsen. Linear hashing is awesome. *SIAM Journal on Computing*, 48(2):736–741, 2019.

[Kra94] Hugo Krawczyk. LFSR-based hashing and authentication. In *Advances in Cryptology – CRYPTO '94*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139, 1994.

[KRT13] Ilan Komargodski, Ran Raz, and Avishay Tal. Improved average-case lower bounds for demorgan formula size. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 588–597, 2013.

[KS99] Adam R. Klivans and Rocco A. Servedio. Boosting and hard-core sets. In *FOCS*, 1999.

[KSY14] Swastik Kopparty, Shubhangi Saraf, and Sergey Yekhanin. High-rate codes with sublinear-time decoding. *J. ACM*, 61(5), September 2014.

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

[Kum25] Vinayak M. Kumar. New Pseudorandom Generators and Correlation Bounds Using Extractors. In Raghu Meka, editor, *16th Innovations in Theoretical Computer Science Conference (ITCS 2025)*, volume 325 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 68:1–68:23, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[Kum26] Vinayak M. Kumar. Most juntas saturate the hardcore lemma. In *2026 SIAM Symposium on Simplicity in Algorithms (SOSA)*, pages 142–150, 2026.

[KW03] Iordanis Kerenidis and Ronald de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 106–115, New York, NY, USA, 2003. Association for Computing Machinery.

[KZ07] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2007.

[LFKN92] Carsten Lund, Lance Fortnow, Howard J. Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *J. ACM*, 39(4):859–868, 1992.

[Lip90] Richard J. Lipton. Efficient checking of computations. In *Proceedings of the 7th Annual Symposium on Theoretical Aspects of Computer Science*, STACS '90, pages 207–215, Berlin, Heidelberg, 1990. Springer-Verlag.

[LN96] Rudolf Lidl and Harald Niederreiter. *Finite Fields*. Encyclopedia of Mathematics and Its Applications. Cambridge University Press, Cambridge, 2 edition, 1996.

[LS11] Shachar Lovett and Srikanth Srinivasan. Correlation bounds for poly-size ac0 circuits with n(1-o(1)) symmetric gates. In Leslie Ann Goldberg, Klaus Jansen, R. Ravi, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 640–651, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[LTW11] Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu. Complexity of hard-core set proofs. *Computational Complexity*, 2011.

[LVW93] M. Luby, B. Velickovic, and A. Wigderson. Deterministic approximate counting of depth-2 circuits. In *[1993] The 2nd Israel Symposium on Theory and Computing Systems*, pages 18–24, 1993.

[LW21] Ray Li and Mary Wootters. Improved list-decodability of random linear binary codes. *IEEE Trans. Inf. Theor.*, 67(3):1522–1536, March 2021.

[LY22] Jiatu Li and Tianqi Yang. 3.1n - o(n) circuit lower bounds for explicit functions. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 1180–1193, New York, NY, USA, 2022. Association for Computing Machinery.

[Lyu22] Xin Lyu. Improved pseudorandom generators for ac0 circuits. In *Proceedings of the 37th Computational Complexity Conference*, CCC '22, Dagstuhl, DEU, 2022. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[LZ24] Xin Li and Yan Zhong. Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs. In Rahul Santhanam, editor, *39th Computational Complexity Conference (CCC 2024)*, volume 300 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:14, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[MCW78] George Markowsky, J. Lawrence Carter, and Mark N. Wegman. Analysis of a universal class of hash functions. In *Mathematical Foundations of Computer Science 1978*, volume 64 of *Lecture Notes in Computer Science*, pages 345–354, 1978.

[Mei08] Or Meir. Combinatorial construction of locally testable codes. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, pages 285–294, New York, NY, USA, 2008. Association for Computing Machinery.

[MNT93] Yishay Mansour, Noam Nisan, and Prasoon Tiwari. The computational complexity of universal hashing. *Theoretical Computer Science*, 107(1):121–133, 1993.

[MRRR14] Raghu Meka, Omer Reingold, Guy N. Rothblum, and Ron D. Rothblum. Fast pseudorandomness for independence and load balancing. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming*, volume 8572 of *Lecture Notes in Computer Science*, pages 859–870, 2014.

[MU05] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomized Algorithms and Probabilistic Analysis*. Cambridge University Press, Cambridge, 2005.

[MV84] Kurt Mehlhorn and Uzi Vishkin. Randomized and deterministic simulations of PRAMs by parallel machines with restricted granularity of parallel memories. *Acta Informatica*, 21(4):339–374, 1984.

[Nis93] Noam Nisan. The communication complexity of threshold gates. Combinatorics, Paul Erdős is eighty, Vol. 1, 1993.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994.

[O'D02] Ryan O'Donnell. Hardness amplification within NP. In *STOC*, 2002.

[OP15] Rafail Ostrovsky and Anat Paskin-Cherniavsky. Locally decodable codes for edit distance. In Anja Lehmann and Stefan Wolf, editors, *Information Theoretic Security*, pages 236–249, Cham, 2015. Springer International Publishing.

[OPS07] Rafail Ostrovsky, Omkant Pandey, and Amit Sahai. Private locally decodable codes. In Lars Arge, Christian Cachin, Tomasz Jurdziński, and Andrzej Tarlecki, editors, *Automata, Languages and Programming*, pages 387–398, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

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

[PT12] Mihai Pătraşcu and Mikkel Thorup. The power of simple tabulation hashing. *Journal of the ACM*, 59(3):1–50, 2012.

[Raz87] Alexander A. Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical notes of the Academy of Sciences of the USSR*, 41:333–338, 1987.

[Rob55] Herbert Robbins. A remark on Stirling's formula. *The American Mathematical Monthly*, 62(1):26–29, 1955.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM J. Comput*, 25(2):252–271, 1996.

[RTTV08] Omer Reingold, Luca Trevisan, Madhur Tulsiani, and Salil Vadhan. Dense subsets of pseudorandom sets. In *FOCS*, 2008.

[RW93] Alexander Razborov and Avi Wigderson. $\Omega(\log n)$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993.

[Sch85] W. H. Schikhof. *Ultrametric Calculus: An Introduction to p-Adic Analysis*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, Cambridge, 1985.

[SFI+11] M. Sasaki, M. Fujiwara, H. Ishizuka, W. Klaus, K. Wakui, M. Takeoka, S. Miki, T. Yamashita, Z. Wang, A. Tanaka, K. Yoshino, Y. Nambu, S. Takahashi, A. Tajima, A. Tomita, T. Domeki, T. Hasegawa, Y. Sakai, H. Kobayashi, T. Asai, K. Shimizu, T. Tokura, T. Tsurumaru, M. Matsui, T. Honjo, K. Tamaki, H. Takesue, Y. Tokura, J. F. Dynes, A. R. Dixon, A. W. Sharpe, Z. L. Yuan, A. J. Shields, S. Uchikoga, M. Legré, S. Robyr, P. Trinkler, L. Monat, J.-B. Page, G. Ribordy, A. Poppe, A. Allacher, O. Maurhart, T. Länger, M. Peev, and A. Zeilinger. Field test of quantum key distribution in the tokyo QKD network. *Optics Express*, 19(11):10387–10409, 2011.

[Sha49] Claude E. Shannon. The synthesis of two-terminal switching circuits. *The Bell System Technical Journal*, 28(1):59–98, 1949.

[Sin64] R. Singleton. Maximum distance q-nary codes. *IEEE Transactions on Information Theory*, 10(2):116–118, 1964.

[Smo87] R. Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 77–82, New York, NY, USA, 1987. Association for Computing Machinery.

[ST18] Rocco A. Servedio and Li-Yang Tan. Luby-Velickovic-Wigderson Revisited: Improved Correlation Bounds and Pseudorandom Generators for

Depth-Two Circuits. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, volume 116 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 56:1–56:20, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[ST22] Rocco A. Servedio and Li-Yang Tan. Improved pseudorandom generators from pseudorandom multi-switching lemmas. *Theory Comput.*, 18:1–46, 2022.

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma (extended abstract). In *STOC*, 1999.

[Tre04] Luca Trevisan. Some applications of coding theory in computational complexity. *arXiv preprint cs/0409044*, 2004.

[Tre05] Luca Trevisan. On uniform amplification of hardness in NP. In *STOC*, 2005.

[Tre09] Luca Trevisan. Approximating a boolean function via small circuits. `https://lucatrevisan.wordpress.com/2009/11/06/approximating-a-boolean-function-via-small-circuits/`, 2009. Accessed: 2025-08-08.

[Vio07] Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. *SIAM Journal on Computing*, 36(5):1387–1403, 2007.

[Vio08] Emanuele Viola. The sum of d small-bias generators fools polynomials of degree d. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 124–127, 2008.

[Vio09] Emanuele Viola. Guest column: Correlation bounds for polynomials over $\{0,1\}$. *SIGACT News*, 40(1):27–44, 2009.

[Vio22] Emanuele Viola. Correlation bounds against polynomials. *Electron. Colloquium Comput. Complex.*, TR22-142, 2022.

[VW07] Emanuele Viola and Avi Wigderson. Norms, xor lemmas, and lower bounds for gf(2) polynomials and multiparty protocols. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 141–154, 2007.

[Wat11] Thomas Watson. Pseudorandom generators for combinatorial checkerboards. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 232–242, 2011.

[WC81] Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981.

[Woo07] David Woodruff. New lower bounds for general locally decodable codes. Technical Report TR07-006, Electronic Colloquium on Computational Complexity (ECCC), January 2007.

[Woo14] Mary Katherine Wootters. *Any Errors in This Dissertation Are Probably Fixable: Topics in Probability and Error Correcting Codes*. PhD thesis, University of Michigan, 2014.

[Yan24] Tal Yankovitz. A stronger bound for linear 3-LCC. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*. IEEE, 2024.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles. In *26th Annual Symposium on Foundations of Computer Science*, pages 1–10, 1985.

[Yek08] Sergey Yekhanin. Towards 3-query locally decodable codes of subexponential length. *Journal of the ACM (JACM)*, 55(1):1–16, 2008.

[Yek12] Sergey Yekhanin. Locally decodable codes. *Found. Trends Theor. Comput. Sci.*, 6(3):139–255, 2012.

Vita

Vinayak Mahesh Kumar was born in Fremont, California on August 10, 1999. In 2017, he graduated from Dougherty Valley High School in San Ramon. He then attended the California Institute of Technology, where he received a B.S. in mathematics and computer science in 2021. Shortly after, he joined the University of Texas at Austin to pursue a Ph.D. in computer science.

Address: 910 Duncan Ln
Austin, TX 78705

This dissertation was typeset with LaTeX[†] by the author, with typesetting assistance from ChatGPT 5.6.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.