Seedless Extractors

Jesse Patrick McGrenra Goodman
Cornell University

December 2023

Abstract

Randomness is a powerful tool, exploited almost everywhere in computer science – from cryptography, to distributed computing, to algorithm design and more. Unfortunately, most of these applications require access to *perfectly uniform bits*, while randomness harvested from nature (e.g., atmospheric noise, radioactive decay, other quantum phenomena) rarely looks so pure. This motivates the study of *randomness extractors*, which are deterministic algorithms that convert weak sources of randomness into uniformly random bits.

The study of extractors dates back over 70 years, and has blossomed into a beautiful theory with deep connections to cryptography, complexity theory, and combinatorics. As it is impossible to construct a single extractor that works for *all* weak sources of randomness, research on extractors has split into two complementary settings: (1) the *seeded* setting, where the extractor is equipped with a short uniform seed to help extract bits from the weak source; and (2) the *seedless* setting, where the extractor is given no seed, but the weak source is equipped with some additional structure. While a rich body of work has culminated in near-optimal *seeded* extractors, we are still far from constructing optimal *seedless* extractors.

In this thesis, we explicitly construct *new seedless extractors* that significantly improve the previous state-of-the-art. To build our extractors, we exploit new reductions within pseudorandomness, and unearth new connections to extremal combinatorics, communication complexity, and coding theory. Along the way, we unlock exciting new applications in cryptography, complexity theory, and beyond.

Acknowledgements

I would not still be in academia if it weren't for Eshan. Over the past 5 years, Eshan Chattopadhyay has served as the most excellent PhD advisor - generous with his time, his praise, and his support. I always looked forward to our weekly meetings, and will forever be thankful for the opportunity he has given me.

To my committee members Bobby Kleinberg and Éva Tardos, thank you for being such friendly and familiar faces around Gates. I will always view Bobby as a role model, in both the intellectual and moral sense. Éva, thank you for all the life advice.

To the Bayen lab at UC Berkeley, thank you for providing me with a welcoming place to stay between my undergrad and graduate years. Alex Keimer, I miss our daily walks to the corner shop to pick up chocolate.

To my collaborators, I am grateful for all of the wonderful things you have taught me. A special thanks to Vipul Goyal, Xin Li, and David Zuckerman, for working with me so early on in my PhD. Omar Alrabiah, Jonathan Mosheiff, and João Ribeiro: I like spending the first 15 minutes of every meeting just catching up. Jyun-Jie Liao, thanks for putting up with my nonstop questions, and keeping me company at the lab. Abhishek Shetty, thank you for being my academic shaman. Sergey, thank you for the cheesecake at 5am.

To the theory lab at Cornell, it's fun to share an office with you guys. Especially Shijin Rajakrishnan, Makis Arsenis, and Jyun-Jie (JJ) Liao.

Franklin Gong, thanks for making Ithaca feel like a home.

To my friends in SF: thank you for listening to my stories. To my friends in NY: thank you for asking me to visit more. And to my friends back home: thanks for helping me get away from it all.

To my family in Ireland, thanks for all the summers growing up. Angela, thanks for all the candy. JJ, thanks for carrying me around the house in a sleeping bag.

Mom, thank you for being my cheerleader.

Dad, thank you for teaching me discipline through piano.

And Emmett, thanks for being my brother.

Contents

Chapter 1

Introduction

Randomness is a powerful tool, with rich applications across all of computer science. Unfortunately, nearly all of these applications require access to a stream of *perfectly uniform bits*, while the randomness that can be found in nature is often plagued with imperfections. This motivates the construction of *randomness extractors*, which are devices that can extract clean, uniform random bits from dirty sources of randomness. Since their introduction in the 1950s, randomness extractors have inspired a beautiful line of work, and have become a central object of study in theoretical computer science. In this chapter, we take a deeper look at this story, and highlight the contributions of this thesis to the magnificent theory of *randomness extractors*.

1.1 Randomness in computation

A perfectly fair coin is a surprisingly powerful tool. In everyday life, we have all appreciated its ability to make a quick, impartial decision - whether that be *where to eat dinner* on a certain night, or *who has to take out the trash*. But perhaps the best way to exploit its power is to remove it from the hands of a human, and put it in the possession of a *computer*. Indeed, the use of *randomness in computation* is one of the great success stories in computer science, unearthing deep applications in every corner of this rich field [Vad12].

Algorithms One of the most celebrated applications of randomness in computer science is in the design of elegant, efficient algorithms [MR95]. While it is widely believed that every efficient algorithm that *flips coins* has a deterministic counterpart that does not,[1] the randomized algorithm is usually much simpler, and noticeably more efficient. For example, a simple and quick (polynomial-time) randomized algorithm for *testing whether a number is prime* has been known for nearly 50 years [Mil76, Rab80], while its (slower and more complex) deterministic counterpart was only found quite recently, in a landmark paper of Agrawal, Kayal, and Saxena [AKS04]. Meanwhile, many natural problems (such as *polynomial identity testing*) that admit polynomial-time randomized algorithms are still not known to be in P [Sch80, Zip79, Sax09].

Cryptography Randomness is vital to the construction of almost all cryptographic primitives, such as encryption, secret sharing, and multi-party computation [DOPS04]. As an example, consider the classical *one-time pad* [Sha49], a simple encryption scheme that uses randomness to *hide the contents of a message* sent over an insecure channel. In this scheme, Alice and Bob get together beforehand (in private) and agree

[1] In other words, there is strong evidence that $\mathsf{BPP} = \mathsf{P}$ [NW94, IW97, DMOZ22], or rather that every polynomial-time randomized algorithm can be converted into a deterministic one, at the expense of at most a *polynomial slowdown* in runtime. However, there is also strong evidence that *some* amount of slowdown will always be required [Wil16].

on a uniformly random *secret key* $\mathbf{Y} \sim \{0,1\}^n$. Later, when Alice needs to send a message $m \in \{0,1\}^n$ to Bob, she instead sends its bitwise XOR with $\mathbf{Y}$; namely, $\widehat{m} := m \oplus \mathbf{Y}$. Upon receiving $\widehat{m}$, Bob can easily recover Alice's message, by computing $\widehat{m} \oplus \mathbf{Y} = (m \oplus \mathbf{Y}) \oplus \mathbf{Y} = m$. However, since Alice and Bob are the only ones that know the key $\mathbf{Y}$, any adversary that sees the communication $\widehat{m}$ just sees a sequence of totally random bits. Thus, *no information* can be obtained about m, and Alice can rest assured her message remains completely hidden. For more great examples of randomness in cryptography, see the book [KL20].

Distributed computing Several protocols in distributed computing also benefit from randomness [GSV05]. Perhaps this is best demonstrated by the classical task of *Byzantine agreement* [PSL80], where a collection of processors wish to agree on a *common value*. The problem is that some of the processors are *adversarial*, and may try to prevent the honest processors from reaching a consensus. Depending on the model, the processors communicate either synchronously or asynchronously. In each case, randomness enables protocols that are otherwise impossible. Using a randomized protocol, consensus can be achieved as long as the fraction of bad processors is less than $1/3$, in both the synchronous [CC85, BOPV06] and asynchronous settings [BO83, Bra84]. Without randomness, the best possible *synchronous* protocol can handle the same number of faulty processors, but requires many more rounds of communication [PSL80]. In the *asynchronous* setting, no deterministic protocol can exist, even given just *one* bad processor [FLP85]!

Scientific computing Randomness is the core ingredient in *Monte Carlo* methods, which are used by scientists to simulate complex systems in the physical world [HH64, PTVF07]. Such methods are, for example, used to model the Earth's climate [BT13], predict how proteins will fold [LS87], and understand nuclear interactions [CGP$^+$15]. Outside of the hard sciences, Monte Carlo methods have been exploited to price various financial instruments [Boy77], and design delicious recipes for chicken [TSG23].

Combinatorics Finally, randomness can be used to construct a variety of delightful combinatorial objects, which themselves have important applications across many areas of computer science [Vad12]. More formally, using the *probabilistic method* [AS16], it is easy to construct key combinatorial primitives such as expander graphs [HLW06] and error-correcting codes [GRS22]. *Expander graphs* are sparse graphs that are nevertheless well-connected, and are naturally suited as cheap, robust communication networks. *Error-correcting codes*, on the other hand, provide a way to add a little redundancy to data so that it can be recovered even in the presence of many errors, and are key tools in reliable data storage and long-distance communication. Surprisingly, these two objects are intimately connected [Vad12], and enjoy a host of unexpected applications in algorithms [ST04], complexity theory [Din07], and cryptography [McE78].

Needless to say, randomness is a powerful force in computer science. Simply by giving a computer some random bits, one can unlock an impressive suite of applications in algorithm design, cryptography, distributed computing, and more. Of course, this raises the question,

How do we give a computer random bits?

Surely we cannot just fix a robotic arm onto a computer and ask it to flip a physical coin. Right?

1.2 Randomness in the wild

In the real world, computers use devices called *true random number generators* (TRNGs) to obtain their random bits [JK99]. These devices generate bits by measuring unpredictable events in *nature*, such as mouse movements, keystrokes, atmospheric noise, radioactive decay, and other quantum phenomena [HG17].[2] Unfortunately, even with this wide range of sources, TRNGs can only produce bits that are *weakly random*: these bits contain *some* level of unpredictability, but almost always have unwanted correlations and biases.[3] On the other hand, almost all applications of randomness in computation (like those in Section 1.1) assume access to bits that are *perfectly random*: that is, completely independent and unbiased. This is a big problem.

Almost all applications of randomness in computer science assume access to *perfectly random* bits,
but nature can only provide us with *weakly random* bits.

Before we panic, it is natural to ask if this is really such a big deal. Indeed, even though these applications *expect* to receive perfectly random bits, perhaps they will do just fine if we give them the weak bits that are available, instead. Unfortunately, this is not the case. For example, it has been known for nearly 20 years that basic cryptographic primitives (like encryption) are impossible to implement with weak randomness alone [DOPS04]. And even before that, the privacy of real-world data has been put at serious risk by TRNGs that fail to generate sufficiently random strings [GW96]. More generally, however, giving imperfect randomness to *any* algorithm expecting perfect randomness means *throwing away all theoretical guarantees* - a move that could result in *unexpected*, and potentially even *dangerous*, behavior [FLW92]. Surely, it would be quite concerning to board an airplane that runs on randomized algorithms with no theoretical guarantees.

As can be seen, there is an alarming gap between the *weak randomness* available in practice, and the *perfect randomness* used in theory. In order to truly unlock the treasure trove of applications in Section 1.1, we must bridge this gap, and find some device that can *purify* weak randomness into perfectly random bits.

1.3 Randomness extractors

Randomness extractors are the exact devices we need. True to their name, extractors are powerful tools that can *extract* perfectly random bits out of a weak source of randomness. First introduced by von Neumann in 1951 [vN51], these devices can easily be used to purify the output of a TRNG, immediately solving the problem that troubled us in Section 1.2. However, ever since they were first introduced, extractors have revealed themselves to be much more fundamental objects than ever anticipated, unearthing a host of rich applications that extend far beyond their original intended use. Most notably, extractors have emerged as central figures in the wonderful world of *theoretical computer science* [AB09], where they have had a profound impact on an incredible number of fields, including cryptography [BBR88], complexity theory [LY22], coding theory [Gur04], combinatorics [Li23], pseudorandomness [Vad12], and more. As a result,

The study of extractors has blossomed into its own beautiful theory,
and the goal of this thesis is to contribute to it.

To get things started, this section will provide a gentle introduction to *extractor theory*. We start by formally defining *extractors*, connecting their definition with the original motivation from Section 1.2. Then, in Sections 1.3.1 to 1.3.3, we overview the history of extractors, describing the origin story of each *flavor* of extractor, while discussing some exciting, unexpected applications. Finally, in Section 1.3.4, we highlight *our contributions* to this beautiful field of study, and provide a detailed roadmap for the rest of this thesis.

[2] For a crystal clear introduction to *how computers generate random bits*, we highly recommend the beautiful survey [HG17].

[3] This is due to various imperfections in the natural random sources, or the measurement process used by the TRNG.

A formal definition

As promised, let us begin by making everything a bit more formal. Returning to our original motivation, recall that we wish to extract *perfectly random bits* out of a *weak source of randomness*. To formalize this, we model the perfectly random bits as a random variable $\mathbf{U}_m$ that is uniform over the set $\{0,1\}^m$ of bitstrings of length m. Our weak source, on the other hand, is modeled by a random variable $\mathbf{X}$ over $\{0,1\}^n$, which holds the *entire sequence* of random bits that can be harvested from the source.[4] Naturally, the exact distribution of $\mathbf{X}$ is unknown, but we assume it comes from some known family $\mathcal{X}$. Finally, the extractor is modeled as a single deterministic function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$, such that given as input any random variable $\mathbf{X} \in \mathcal{X}$,[5] the output $\mathsf{Ext}(\mathbf{X})$ should look *close to* uniform $\mathbf{U}_m$.

Now, before we put everything together and formally define an extractor, we must elaborate on what it means for the output to look *close to* uniform, and why this is even okay - after all, we originally wanted the output to be *perfect*, right? As it turns out, demanding the output $\mathsf{Ext}(\mathbf{X})$ to be *exactly* uniform is unnecessarily restrictive, and prevents the existence of extractors in even the simplest of settings [CFG$^+$85]. Instead, following the suggestion of Santha and Vazirani [SV86], the extractor community has realized it makes much more sense to simply ask that $\mathsf{Ext}(\mathbf{X})$ is *close to* uniform $\mathbf{U}_m$ in statistical distance, defined as

$$\begin{aligned} |\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| &:= \max_{S \subseteq \{0,1\}^m} |\Pr[\mathsf{Ext}(\mathbf{X}) \in S] - \Pr[\mathbf{U}_m \in S]| \\ &= \frac{1}{2} \sum_{z \in \{0,1\}^m} |\Pr[\mathsf{Ext}(\mathbf{X}) = z] - \Pr[\mathbf{U}_m = z]|. \end{aligned}$$

Given a guarantee that this distance is at most ε, one can ensure that any randomized algorithm or protocol that expects uniformly random bits *can instead use the bits produced by* $\mathsf{Ext}(\mathbf{X})$, and err with probability at most ε. Assuming ε is sufficiently small, this guarantee is good enough for pretty much any application of randomness in computer science.[6] With all of this in mind, we can now formally define the star of the show.

Definition 1.1 (Randomness extractor). *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *is called an* extractor for $\mathcal{X}$ with error ε *if for every* $\mathbf{X} \in \mathcal{X}$,

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \le \varepsilon.$$

Given the formal definition of an extractor, it is now easy to state the fundamental goal of *extractor theory*:

Construct the best possible extractors,
and discover new connections to other areas of theoretical computer science.

However, staring at Definition 1.1, there are a lot of parameters flying around - indeed, it may not even be clear what it means for an extractor to be *good* (not to mention *best possible*). Towards this end, there are *four key elements* that constitute an excellent randomness extractor. We go into more detail, below.

[4] In other words, *no more randomness* can be harvested from the source beyond the bits in $\mathbf{X}$. If multiple samples can be taken from the source, then this is modeled by having $\mathbf{X}$ represent the concatenation of those samples.

[5] For convenience, we often slightly abuse notation and let $\mathbf{X}$ denote both a random variable and its underlying distribution, and refer to it as a *source*.

[6] We emphasize that this guarantee is *much stronger* than any guarantee offered by the *weak randomness* harvested directly from nature. Such weak randomness can only offer a guarantee on *entropy*, and we will soon see that it is possible to have an excellent entropy guarantee while unfortunately being *extremely far* from uniform in statistical distance.

The four essential elements of a great randomness extractor

First and foremost, we would like the extractor to work for a family of distributions $\mathcal{X}$ that is *as general as possible*. The more general $\mathcal{X}$ is, the more likely it is to capture distributions one might actually find in nature. At the same time, an extractor for a general $\mathcal{X}$ is much more likely to enjoy unexpected applications *outside* the world of randomness extraction.

Second, we would like the extractor to output *as many bits as possible*, m. The reason for this is pretty self-explanatory: the more uniform bits that we can harvest from a weak source, the better. Surprisingly, however, it is often the case that extracting just $m = 1$ bit of randomness is the hardest part. Given such an extractor, well-known techniques can often be applied to boost its output length [Vaz87b, GRS06, Sha08].

Third, we would like the error ε of the extractor to be *as small as possible*, so that its output is as close as possible to being perfectly uniform. Just like with the output length, even getting *nontrivial* error $\varepsilon < 1/2$ is extremely challenging.[7] Unlike the output length, however, there are no standard ways to bootstrap an extractor with nontrivial error into one that achieves notably low error. This is troublesome, given the fact that many cryptographic applications require random bits that are extremely close to uniform [DOPS04].

Finally, we would like the extractor to be *explicit*, in the sense that it should be *deterministic* and *efficiently computable*. More formally, given an input $x \in \{0,1\}^n$, the output of the extractor $\mathsf{Ext}(x)$ should be computable by a deterministic Turing machine in time *polynomial in* n.[8] This restriction on runtime comes from the standard notion of *efficiency* in complexity theory, while the desire for the Turing machine be *deterministic* (instead of randomized) comes from the original motivating application of an extractor.[9] Surprisingly, if we were actually allowed to use unlimited random bits, it is often trivial to construct an extractor. Indeed, assuming an extractor actually exists for a family $\mathcal{X}$, it is almost always the case that a *uniformly random function* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is an *optimal* extractor for $\mathcal{X}$, and this can be shown via a straightforward application of the probabilistic method (Section 1.1). A key goal in extractor theory is to come up with explicit constructions with parameters that come close to matching these probabilistic ones.

From which distributions should we extract?

Needless to say, there are many factors to keep in mind when trying to construct a randomness extractor. However, they can all be easily remembered as follows.

> We would like to construct an *explicit* extractor for the *most general possible family* $\mathcal{X}$,
> which outputs *as many bits as possible* m with the *lowest possible error* ε.

Thus, after picking a family of distributions $\mathcal{X}$, the goal is clear: construct an explicit extractor for $\mathcal{X}$ that has the longest possible output length and lowest possible error. It is less clear, however, *which* family of distributions $\mathcal{X}$ we should try to extract from. Of course, as discussed, we would like $\mathcal{X}$ to be as general as possible - but it is not exactly clear what *is* possible. In particular, if $\mathcal{X}$ contained *all* distributions over $\{0,1\}^n$, then there is clearly *no way* to extract from this family. This is simply because it would include random variables $\mathbf{X} \in \mathcal{X}$ that always output the same value, which would force any extractor called on $\mathbf{X}$ to always output the same value. Thus, the first question we must ask is:

[7] In fact, such nontrivial extractors even have their own name (*dispersers*), and are deeply connected to combinatorics [Li23].

[8] To make this condition completely formal, we must actually think of the family of distributions $\mathcal{X}$ as an *infinite sequence of families* $(\mathcal{X}_n)_{n\in\mathbb{N}}$, where each $\mathcal{X}_n$ contains distributions over $\{0,1\}^n$. Similarly, the extractor actually consists of a *family of extractors* $(\mathsf{Ext}_n)_{n\in\mathbb{N}}$ where each Ext_n is of the form $\mathsf{Ext}_n : \{0,1\}^n \to \{0,1\}^m$. Then, we say the *family* of extractors $(\mathsf{Ext}_n)_{n\in\mathbb{N}}$ is explicit if there exists a *single* deterministic Turing machine M that computes $M(x) := \mathsf{Ext}_{|x|}(x)$ in time polynomial in $|x|$.

[9] Extractors were introduced to cope with the fact that perfect coin flips are not available in nature. Thus, allowing the extractor to be computable by a *randomized* Turing machine (which assumes access to perfect coin flips) seems circular, to say the least.

What is the most general family of distributions $\mathcal{X}$
that we can (nontrivially) extract from?

As it turns out, this is the question that has fueled the development of much of extractor theory over the past 70 years. In the sections that follow (Sections 1.3.1 to 1.3.3), we tell this story.

1.3.1 On the existence of extractors

The origins of extractor theory can be traced all the way back to the 1950s, when von Neumann [vN51] asked whether it is possible to simulate a *perfect coin* by flipping one that is far from perfect.[10] In other words, von Neumann asked to *extract* from the family of distributions $\mathcal{X}$ of the form $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_n)$, where each $\mathbf{X}_i$ is an independent random bit with some fixed unknown bias δ. He provided an elegant solution,[11] and unknowingly launched a brand new field at the intersection of computer science and mathematics.

However, in the decades that followed, interest in this question remained surprisingly sparse. While a handful of papers emerged from the statistics community that sought to improve upon von Neumann's work [Sam68, HS70, Eli72, Dwa72, BL73, SW84], it wasn't until FOCS 1984 that extractors truly caught the eye of theoretical computer scientists. There, two of the earliest known papers on *extractor theory* were published [Blu86, SV86], and the quest to extract from the most general possible family $\mathcal{X}$ began.

In the first paper, Blum [Blu86] considered a model that generalized von Neumann's *independent coin flips* by allowing the flips to be *somewhat dependent*. More formally, he studied the family $\mathcal{X}$ of sources generated by *finite state Markov chains*. In the second paper, Santha and Vazirani [SV86] considered an even more general family $\mathcal{X}$, which they referred to as *semi-random sources*.[12] These sources could consist of *arbitrarily correlated* bits, as long as each bit had *bounded bias* conditioned on any fixing of the bits that came before it. The following year at STOC and FOCS, a flurry of new papers (and models) appeared [Vaz87b, VV85, CG88, CFG+85, BL85], and a foundation for the theory of extractors began to take hold.

On the introduction of entropy to extractor theory However, as more and more models emerged, it became less clear where things were headed. While the ultimate goal was to extract from the most general possible family $\mathcal{X}$, it was still not obvious what such a family looks like. One reason was the approach taken above, in which one would start with the simple model of von Neumann, remove some assumption, and hope to still extract. This led to a motley crew of distribution families, with no clear unifying theme. It soon became clear that in order to extract from the most general possible family $\mathcal{X}$, one should *approach the problem from the other end*. In particular, one should start with the family of *all distributions*, add some (minimal) assumption, and hope to *now* be able to extract. But what (minimal) assumption should we add?

Returning to our example from just before Section 1.3.1, let's recall the simple reason it was impossible to extract from the family $\mathcal{X}$ of all distributions: this family contains *constant random variables*, which force the output of any extractor to be constant. Thus, if we want to have any hope of extracting, the *absolute minimum* assumption we can make is that $\mathcal{X}$ *does not contain any constant random variables*. That is,

In order for randomness extraction to be possible,
each source must contain *some* amount of randomness.

Of course, this immediately raises the question of *how to define the notion of randomness*. Here, the most tempting option might be via *entropy*, which has long been the most classical way to measure randomness,

[10] For those seeking a job in finance, take note: this is often asked as an interview question at various hedge funds.

[11] Pair up the bits, look for the first mismatched pair, and output the first bit in that pair. If no such pair exists, simply output 0.

[12] Today, these are affectionately known as *Santha-Vazirani* sources.

ever since it was first introduced by Shannon in his seminal paper that founded information theory [Sha48]. Given a random variable $\mathbf{X} \sim \{0,1\}^n$, Shannon defined its entropy as

$$H(\mathbf{X}) := \mathbb{E}_{x \sim \mathbf{X}} \log\left(\frac{1}{\Pr[\mathbf{X}=x]}\right) = \sum_{x \in \text{support}(\mathbf{X})} \Pr[\mathbf{X}=x] \cdot \log\left(\frac{1}{\Pr[\mathbf{X}=x]}\right).$$

Now known as *Shannon entropy*, this quantity measures the average amount of "information" revealed by observing a random variable. It is easy to verify that Shannon entropy of a random variable $\mathbf{X} \sim \{0,1\}^n$ ranges between 0 (for a constant random variable) and n (for a uniform random variable) - meaning that the higher the Shannon entropy of a random variable, the more randomness it contains, and the easier it should be to extract. Thus, it is natural to ask whether *Shannon entropy* can help provide the minimal assumption we are looking for in order to enable randomness extraction. In other words, one might ask,

Can we extract from the family $\mathcal{X}$ of all distributions
with sufficiently high *Shannon entropy*?

Unfortunately, the answer to this question is a resounding *no*. In fact, there even exist families $\mathcal{X}$ consisting of just *one* random variable $\mathbf{X} \sim \{0,1\}^n$ with high Shannon entropy from which (high quality) extraction is impossible. For example, consider a source $\mathbf{X}$ that outputs 0 with probability 0.99, and otherwise samples uniformly from $\{0,1\}^n$. The Shannon entropy of this source is nearly $0.01n$, but any extractor receiving $\mathbf{X}$ as input must output some value with probability 0.99. Thus, it is impossible to construct a high quality extractor in this setting. If we hope to eventually extract, we need a different notion of *randomness*.

Looking back at the challenges encountered above, one might notice that we had difficulty extracting from a source $\mathbf{X} \sim \{0,1\}^n$ with high *Shannon entropy* because such a source can still assign a large probability to some $x \in \{0,1\}^n$. Thus, if we want to avoid these challenges, it is natural to propose a definition of randomness that forces $\Pr[\mathbf{X} = x]$ to be small for *all* $x \in \{0,1\}^n$. As it turns out, such a definition of randomness already exists, and is known as *min-entropy*.

Definition 1.2 (Min-entropy)**.** *The* min-entropy *of a random variable* $\mathbf{X} \sim \{0,1\}^n$ *is defined as*

$$H_\infty(\mathbf{X}) := \min_{x \in \textit{support}(\mathbf{X})} \log\left(\frac{1}{\Pr[\mathbf{X}=x]}\right).$$

In other words, we have that $H_\infty(\mathbf{X}) \geq k$ *if and only if* $\Pr[\mathbf{X}=x] \leq 2^{-k}$ *for all* $x \in$ *support*$(\mathbf{X})$.

Just like Shannon entropy, the *min-entropy* of a random variable $\mathbf{X} \sim \{0,1\}^n$ ranges between 0 and n, and the higher the min-entropy of $\mathbf{X}$, the more randomness it contains, and the easier it should be to extract. *Unlike* Shannon entropy, having high min-entropy means that no value is hit with too high of a probability, and the bad situation from before is no longer possible. In fact, it is not hard to see that min-entropy is a strictly *stronger* notion of randomness, and we always have $H_\infty(\mathbf{X}) \leq H(\mathbf{X})$. This means that a guarantee on min-entropy is always more powerful than a guarantee on Shannon entropy, which raises the question:

Can we extract from the family $\mathcal{X}$ of all distributions
with sufficiently high *min-entropy*?

Inspired by the seminal paper of Santha and Vazirani [SV86], this question was first asked by Chor and Goldreich [CG88], and later championed by Zuckerman [Zuc90]. And since it was first asked, this question has helped lay the foundation for most of modern-day extractor theory, with the family $\mathcal{X}$ of *sources with* *sufficiently high min-entropy* becoming the most fundamental model in the field. Indeed, this family of sources, known as (n, k) *sources*,[13] not only captures the *minimum possible assumption* one can make on a family $\mathcal{X}$ in order to have any hope of extracting, but it also generalizes the early models of Blum [Blu86] and Santha and Vazirani [SV86]. So, without further ado, let us answer: can we extract from such sources?

An impossibility result Unfortunately, as the title of this section might suggest, the answer to this question is surprisingly *no*! In fact, even if each source $\mathbf{X} \in \mathcal{X}$ has nearly the *maximum* possible amount of min-entropy, then extraction is *still* impossible. Moreover, as it turns out, this result is not too difficult to show.[14]

Fact 1.1 (Classic impossibility result [CG88])**.** *There does not exist an extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ *with error* $\varepsilon < 1/2$ *for the family* $\mathcal{X}$ *of all distributions with min-entropy* $k \geq n-1$.

Proof. Let $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ be an arbitrary candidate extractor for $\mathcal{X}$. Since Ext is a deterministic function, at least half of its inputs map to the same output. Without loss of generality, assume that half of its inputs map to 0. That is, $|\mathsf{Ext}^{-1}(0)| \geq 2^{n-1}$. Consider now a random variable $\mathbf{X} \sim \{0,1\}^n$ that is uniform over the set $\mathsf{Ext}^{-1}(0)$. Clearly $\mathsf{Ext}(\mathbf{X})$ is always 0, by definition of $\mathbf{X}$. On the other hand, $\mathbf{X}$ is uniform over a set of size at least 2^{n-1}, meaning it has min-entropy $k \geq n-1$. Thus we have found a source $\mathbf{X} \in \mathcal{X}$ for which $|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_1| = |0 - \mathbf{U}_1| = 1/2$, meaning Ext cannot be an extractor for $\mathcal{X}$ with error $\varepsilon < 1/2$. □

Thus, even if each source $\mathbf{X} \in \mathcal{X}$ has extremely high min-entropy, we still cannot extract. So, where should we go from here? Well, let us recap what we have discussed thus far. Originally, our goal was to construct an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for the *most general possible* family $\mathcal{X}$, which has the longest possible output length m and lowest possible error ε. After discussing some early work on extractors, we realized that in order to extract from the most general possible $\mathcal{X}$, we should start with the family $\mathcal{X}$ of *all distributions*, and add the fewest possible assumptions in order to enable extraction. We started by realizing that the *minimum possible assumption* is that each $\mathbf{X} \in \mathcal{X}$ contains *some randomness*. To formalize the notion of randomness, we initially tried to use a lower bound on *Shannon entropy*, but realized this is far too weak of a guarantee to enable extraction of any kind. Thus, we turned to a lower bound on *min-entropy*, a much stronger notion than Shannon entropy. However, we proved that even in this setting, extraction is still impossible![15] So, what else can we do? The answer is simple: *add another assumption*. Towards this end, researchers have considered adding one of *two possible assumptions* to circumvent the impossibility result in Fact 1.1.

- **Option 1** (*seed*): the extractor is equipped with a *short, uniform seed* to help extract from the source.
- **Option 2** (*structure*): the source is equipped with some *additional structure* to help enable extraction.

As it turns out, by making just one of these two assumptions, it suddenly becomes possible to extract![16] And indeed, ever since the impossibility results of the 80s [SV86, CG88], almost all of extractor theory has branched off into *two complementary directions*, dedicated to the study of extractors under each assumption.[17] Extractors built under the first assumption are called *seeded extractors*, and we provide an overview

[13] Formally, the family $\mathcal{X}$ of (n, k) *sources* consists of random variables $\mathbf{X} \sim \{0,1\}^n$ with min-entropy at least k.

[14] A (weaker) impossibility result was first shown by Santha and Vazirani [SV86] for the (smaller) family of semi-random sources.

[15] Recall that the impossibility result for Shannon entropy is much stronger (and more prohibitive) than the impossibility result for min-entropy. Indeed, the former shows that *there exists* a source $\mathbf{X}$ with high Shannon entropy such that *for all* functions $f : \{0,1\}^n \to \{0,1\}^m$, it holds that f does not extract from $\mathbf{X}$. The latter shows that *for all* functions f *there exists* a source $\mathbf{X}$ with high min-entropy such that f does not extract from $\mathbf{X}$.

[16] We emphasize that these assumptions are made *in addition to* the assumption that each source has sufficiently high min-entropy.

[17] As we will see, the origins of this bifurcation in extractor theory can be traced back to the pioneering work of Vazirani [Vaz87b].

of their history in Section 1.3.2. Extractors built under the second assumption are called *seedless extractors* (or *deterministic extractors*), and we discuss them at length in Section 1.3.3. After that, we will have finally completed our tour of extractor theory, and we will be ready to outline *our contributions* in Section 1.3.4.

1.3.2 Seeded extractors

While the origins of both seeded and seedless extractors can be traced back to the 80s [Vaz87b], the first two decades of extractor theory were focused almost exclusively on the construction of *seeded extractors*. As mentioned above, a seeded extractor is a device that extracts uniformly random bits from the family $\mathcal{X}$ of sources of min-entropy at least k, with the help of an additional *short, uniformly random seed*. More formally, a (k, ε)-seeded extractor is a deterministic function $\mathsf{sExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ such that given any source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy at least k, and an independent uniform seed $\mathbf{Y} \sim \{0,1\}^d$,

$$|\mathsf{sExt}(\mathbf{X}, \mathbf{Y}) - \mathbf{U}_m| \leq \varepsilon.$$

As always, the goal is to construct an *explicit* extractor, which maximizes the output length, minimizes the error, and works for the most general possible family $\mathcal{X}$ (which, here, means minimizing the min-entropy requirement k). Now, however, we have an additional parameter, d, which corresponds to the *length of the seed*. Of course, we would also like this to be as small as possible.

But before we continue, you may be wondering *why* seeded extractors are interesting in the first place. Indeed, extractors were originally introduced to cope with the fact that *uniform randomness is not available in nature*. So, why should we be allowed to assume access to a short, uniform seed? As it turns out, in some applications, it is actually possible to use a seeded extractor *without* making such an assumption. In particular, it is sometimes possible to *simulate* the uniform seed via a *brute-force enumeration*. For example, suppose you want to run a *randomized algorithm* to compute a Boolean function, but you only have access to a weak random source $\mathbf{X} \sim \{0,1\}^n$. While the randomized algorithm *expects* to receive uniformly random bits $\mathbf{U}_m$ to aid in its computation, we can instead try to run the algorithm using the bits $\mathsf{sExt}(\mathbf{X}, y)$, for each fixed seed $y \in \{0,1\}^d$. If we take a *majority vote* of the outputs across these 2^d trials, the result will actually look *almost identical* to the output of the randomized algorithm had it received *truly uniform bits* $\mathbf{U}_m$.[18] Moreover, we will have increased the runtime by a factor of at most 2^d, which is small if d is small.[19]

Thus, assuming we have a good enough explicit seeded extractor, it is actually possible to take *any* randomized algorithm that uses uniformly random bits, and make it work with *weakly random bits*, instead. In other words, such an extractor could be used to *simulate the complexity class* BPP *using weak random sources*. In fact, the desire to achieve such a simulation actually *precedes* the introduction of the seeded extractor, and is what directly led to its invention! We go into a little more detail, below.

History Motivated by the classical impossibility result of Santha and Vazirani [SV86] for extracting from *semi-random sources*, Vazirani was the first to ask about simulating randomized algorithms using such weak sources [Vaz87b], while Vazirani and Vazirani were the first to *show* that this can actually be done [VV85]. Following these early works, several subsequent papers studied this question under increasingly general models of *weak random sources* [CG88, Vaz87a, HILL99, CW89], until Zuckerman proposed simulating randomized algorithms using weak random sources that only have a guarantee on their *min-entropy* [Zuc90]. These results were later improved in a paper by Zuckerman [Zuc96b], which ultimately led to the seminal work of Nisan and Zuckerman [NZ96] that introduced the notion of *seeded extractors*, as defined above.

[18] For a rigorous treatment of this claim, we refer the reader to [Vad12, Proposition 6.15].

[19] This claim ignores the runtime of the seeded extractor, which is not a big deal as long as the seeded extractor is *explicit*.

Like many fascinating combinatorial objects, excellent seeded extractors are easy to construct via the *probabilistic method* (Section 1.1). In particular, by selecting a function $\mathsf{sExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ uniformly at random, it is not too difficult to show that it is a (k, ε)-seeded extractor (with high probability) as long as it has seed length $d \geq \log(n-k)+2\log(1/\varepsilon)+O(1)$ and output length $m \leq k+d-2\log(1/\varepsilon)-O(1)$ [Vad12, Theorem 6.14]. Moreover, it is known that these are the best possible parameters, up to additive constants [RT00]. However, as always, the goal is to construct *explicit* extractors with similarly great parameters, and ever since the seminal work of Nisan and Zuckerman [NZ96], researchers have worked tirelessly in this direction [WZ99, GW97, SZ99, SSZ98, Ta-96, Zuc97, ACRT99, Ta-02, NT99, Tre01, RRV02, ISW00, RVW02, RSW06, TUZ07, TZS06, SU05, LRVW03, GUV09, DW11, DKSS13, TU12]. After nearly two decades of hard work, we now have near-optimal explicit constructions [LRVW03, GUV09, DKSS13], and in particular can achieve seed length $d = O(\log(n/\varepsilon))$ and output length $m = \Omega(k)$ [GUV09].

Applications Today, the best known explicit seeded extractors can easily be used to solve the original motivating problem of *simulating* BPP *using weak random sources*.[20] However, ever since seeded extractors were first introduced [NZ96], they have unearthed a wide array of applications that extend far beyond this original motivation. For example, seeded extractors have become a crucial ingredient in many cryptographic protocols [BBR88, Lu04, Vad04, DORS08], and have played a key role in several *hardness of approximation* results [Zuc96a, Uma99, MU02, Zuc07]. Seedless extractors have also been used to construct error-correcting codes [TZ04, Gur04], pseudorandom generators [NZ96], expander graphs [CRVW02], and even extractors themselves [Rao09a]! Each of these objects is incredibly powerful in its own right, boasting its own rich collection of unexpected applications [Vad12].

Thus, seeded extractors have carved out an important legacy, which has not only impacted *extractor theory*, but much of theoretical computer science as a whole. However, as we will soon see, they are not the focus of this thesis. For a detailed survey on seeded extractors, we refer the reader to [NT99, Sha04a, Sha11b].

1.3.3 Seedless extractors

After two decades of great success in constructing *seeded extractors*, most of modern extractor theory has shifted its focus towards the construction of *seedless extractors*. As discussed above, a seedless extractor is a device that extracts uniformly random bits from a family $\mathcal{X}$ of sources that not only have a min-entropy guarantee of at least k, but also exhibit some sort of *extra structure*. More formally, a seedless extractor for such a family $\mathcal{X}$ with error ε is a deterministic function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ such that for any $\mathbf{X} \in \mathcal{X}$,

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \varepsilon.$$

Indeed, such extractors are exactly captured by Definition 1.1. However, since we are now interested in extracting from families $\mathcal{X}$ that exhibit some sort of *extra structure*, an immediate question comes to mind:

What sort of *extra structure* should we consider?

[20] A BPP algorithm $\mathcal{A}$ for computing a function $f : \{0,1\}^t \to \{0,1\}$ can only be simulated with a weak source $\mathbf{X} \sim \{0,1\}^n$ if that source has min-entropy $k \geq n^{\Omega(1)}$, because otherwise the length of the source would need to be of the form $n \geq k^{\omega(1)} \geq t^{\omega(1)}$. Thus, the best possible simulation of BPP just requires an explicit seeded extractor $\mathsf{sExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ that works for sources of min-entropy $k \geq n^{\Omega(1)}$, which has seed length $d \leq O(\log n)$, output length $m \geq k^{\Omega(1)}$, and error $\varepsilon = 0.01$. Such an extractor was first constructed by Trevisan [Tre01], though earlier work achieved an equally good simulation of BPP using different techniques [ACRT99].

One silly idea is to assume that each source $\mathbf{X} \in \mathcal{X}$ consists of a (long) prefix with some min-entropy, followed by a (short) independent suffix that is uniform. Taking a second to think, one might realize that seedless extractors for such sources are *exactly the same thing as seeded extractors*! In this way, seedless extractors *strictly generalize* seeded extractors. However, these two types of extractors have historically been treated as *two distinct objects*. This is because the study of seedless extractors has focused on extracting from *other definitions of structure*, motivated by the fact that a uniform seed may not always be available.[21] While a variety of different definitions of *structure* have been examined, they all fall into *three broad categories*.

Over the years, *three key flavors* of structure have been considered,
giving rise to three key flavors of seedless extractors.

In what follows, we give a quick overview of each key flavor of structure, and point towards later chapters of this thesis that provide more comprehensive surveys. Then, we will finally be ready to outline our contributions to the wonderful world of seedless extractors, in Section 1.3.4.

Extractors for independent sources

The first key flavor of *structure* studied in the world of seedless extraction assumes that each source $\mathbf{X} \in \mathcal{X}$ actually consists of *several independent chunks* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$, each with their own min-entropy guarantee. Known as the *independent source model*, this may very well be the oldest and most well-studied setting in *all of extractor theory*, dating back to the very paper of von Neumann [vN51] that started the field! However, it wasn't until a few decades later that the modern definition of extractors for independent sources truly began to take hold, in the seminal works of Vazirani [Vaz87b] and Chor and Goldreich [CG88]. But ever since, they have been the subject of relentless attention [Vaz87a, BIW06, Bou05, Raz05, BKS+10, BGK06, Rao09a, BRSW12, RZ08, BSRZ15, Li11a, Li12b, Li13b, Li13a, Li15, Coh16a, Coh19, CZ19, CL16b, Li16, CS16a, CL16a, Coh16b, BDT19, Coh17, Li17, Lew19, Li19, Li23].

It is easy to appreciate *why* the independent source model has been so popular. Going back to the origin story of extractor theory as a whole, recall that the original motivating goal was to purify the *weak randomness* found in nature (Section 1.2), so as to unlock the rich suite of applications that *perfect randomness* enjoys in computer science (Section 1.1). While Fact 1.1 tells us this is simply impossible using a *single* source of randomness, it says nothing about doing so using *multiple, independent* sources. And furthermore, it is not unreasonable to assume that it might actually be possible to find such sources in nature - for example, in *geographically isolated locations*, or emerging from *completely unrelated physical phenomena*. If this is truly the case, then an extractor for independent sources can solve all of our woes.

On a more mathematical note, extractors for independent sources naturally generalize *seeded extractors* (Section 1.3.2), and thereby inherit their luxurious collection of applications. But they also boast several applications of their own, spreading across a wide variety of fields such as communication complexity [Vaz87b], combinatorics [BRSW12], distributed computing [GSV05], and even learning theory [GRT18]. Motivated by all of this and more, the extractor community has been hard at work, eagerly pursuing new-and-improved extractors for this setting. Today, we are happy to report that near-optimal explicit extractors are now known [Li23].

For a detailed survey on extractors for independent sources, we refer the reader to Part I (Section 3.1).

[21] Moreover, in many applications (like in *cryptography* and *distributed computing*), it is not possible to *simulate a uniform seed* via *brute-force enumeration*, as is done in Section 1.3.2 for the application of *algorithm design*.

Extractors for algebraic sources

While the independent source model may very well be the most popular setting in which to study seedless extraction, the *algebraic source model* is certainly a close second. For this next key flavor of seedless extractor, one assumes that each source $\mathbf{X} \in \mathcal{X}$ exhibits some sort of *algebraic structure*. The origins of this setting also date back to the 80s, with the introduction of *oblivious bit-fixing sources* [Vaz87b, BBR88, CFG+85]. Motivated by applications in distributed computing [CFG+85] and exposure-resilient cryptography [CDH+00, Dod00, KZ07], oblivious bit-fixing sources $\mathbf{X} \sim \{0,1\}^n$ consist of completely independent bits, but only a few of which are guaranteed to be uniform (while the rest are fixed to constants). A long line of work has successfully constructed extractors for this setting [Vaz87b, BBR88, CFG+85, CW89, Fri92, KJS01, BS00, KZ07, GRS06, Sha08, Rao09b, GVW15, CL18], and researchers have since turned to study a variety of other, more exotic, algebraic models.

By and far, however, the most popular model to study in the algebraic source setting is the *affine source*. First introduced at the turn of the century [BSHR+01, BKS+10], affine sources offer a natural generalization of oblivious bit-fixing sources, and are formally defined as distributions $\mathbf{X}$ that are uniform over an affine subspace of $\mathbb{F}_2^n$. Despite this seemingly artificial definition, extractors for affine sources have emerged as a central figure in pseudorandomness and beyond, finding surprising applications in circuit lower bounds [DK11, FGHK16, LY22] and extractor theory itself [DW12, Vio14, CG21]. Because of this, they have received a lot of attention over the past three decades [BSHR+01, BKS+10, GR08, Bou07, BSK12, Rao09b, DG10, Yeh11, BSRZ15, Li11b, Sha11a, CT15, BDL16, Li16, CGL21, HIV22, Li23], and researchers have even turned to consider *even more general* models, known as *polynomial* [DGW09, BSG12, GVJZ23] and *variety sources* [Dvi12, CT15, Rem16, LZ19, GVJZ23]. These families may very well represent the pinnacle of the algebraic source model, unlocking new applications unable to be achieved by affine sources [HR15, GK16, GKW21]. However, with pleasure comes pain, and few explicit constructions are known for these extremely general models.

For a detailed survey on extractors for algebraic sources, we refer the reader to Part II (Section 6.1).

Extractors for low-complexity sources

The third and final flavor of seedless extraction steps out of the world of algebra, and back into the world of computer science. Here, we are talking about the model of *low-complexity sources*. As their name might suggest, sources in this model exhibit a sort of *computational* structure. A little more formally, low-complexity sources are typically defined as distributions $\mathbf{X}$ that can be generated by low-complexity computational models (such as *low-depth circuits* or *space-bounded algorithms*), by feeding them with uniformly random bits. Like the other two classical settings discussed above, the origins of low-complexity sources can be found in the 80s, with the pioneering work of Vazirani [Vaz87b]. However, it wasn't until the seminal papers of Trevisan and Vadhan [TV00] and Kamp, Rao, Vadhan and Zuckerman [KRVZ11] that this model truly took off, and they have since enjoyed their fair share of time in the spotlight [Vaz87b, Vaz87a, TV00, KM04, KM05, KRVZ11, DW12, Vio14, Vio12b, CL16b, Li16, CGGL20, CG21, CL22, ACG+22, Li23, BDSGM23].

Unlike the other two models in which to study seedless extraction, the motivation for low-complexity sources is a bit more philosophical in nature. As advocated for by Trevisan and Vadhan [TV00] and scholars before them [Lev86], there is an argument to be made that such sources might *actually* be a good model for distributions that one might find in the wild. Indeed, it is not unreasonable to think that nature may behave like a low-complexity process, acting on some *true randomness* out there somewhere in the universe (say, arising from quantum phenomena), and the random bits that we can harvest are simply an outcome of this low-complexity process.

In case you remain unconvinced of the motivation above, there are, of course, many reasons to study low-complexity sources from a mathematical perspective. Most notably, due to their definition, they are intimately connected to complexity theory. In particular, not only do extractors for such sources offer another concrete approach towards new circuit lower bounds [LZ19], but they also play a fundamental role in burgeoning new area of complexity called the *complexity of sampling* [Vio12a]. Moreover, low-complexity sources are exceedingly general, capturing a host of other well-studied models in extractor theory.

For a detailed survey on extractors for low-complexity sources, we refer the reader to Part III (Section 7.1).

1.3.4 Our contributions

We have finally finished our brief tour through the wonderful world of randomness extractors. As we have seen, extractors have become fundamental figures in complexity theory and beyond, and were originally motivated by the fact that nearly all applications of randomness in computer science expect access to *perfectly uniform bits* (Section 1.1), yet nature simply doesn't have the capacity to produce such perfect randomness (Section 1.2). And while we would like to cope by constructing a single extractor that can purify *all* sources of randomness, we have observed that this is just not possible (Fact 1.1). Thus, research on extractors has split into two complementary settings: the *seeded setting* (Section 1.3.2), and the *seedless setting* (Section 1.3.3), which has further been split into three different flavors of sources: independent sources, algebraic sources, and low-complexity sources. However, while a beautiful line of work has culminated in near-optimal *seeded extractors*, there is still much work to be done on *seedless extractors*.

This brings us to the main contribution of this thesis:

We construct new-and-improved seedless extractors
for each of the three most popular flavors of structure.

In the blurbs that follow, we give a sneak peak into some of these results, and lay out a roadmap for the remainder of this thesis. Should the reader ever get lost, we strongly encourage them to frequently check back in at the table of contents. Indeed, great care was taken to ensure that this thesis can be easily navigated - and understood - simply by looking there. With all of that being said, let's trek onward to give a brief view of some of our main results, before embarking on a journey into the body of this thesis.

Part I Extractors for independent sources

As we saw above, the independent source model is the oldest and most well-studied setting in seedless extraction. Recall that in this setting, one assumes access to a weak source $\mathbf{X}$ that actually consists of *several independent sources* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$, each guaranteed to contain some min-entropy. However, while a rich line of work has culminated in excellent extractors for this setting, it is not always clear how realistic this setting really is. In particular, in real life, nature may sometimes produce samples with no entropy at all, or - worse still - samples that are *correlated*. Motivated by this and applications in cryptography, Part I of this thesis focuses on constructing *robust* extractors for independent sources, which can easily handle these types of tricky situations.

Chapter 3 Extractors for adversarial sources Our first type of *robust extractor for independent sources* can extract uniform bits from so-called *adversarial sources*. Adversarial sources naturally generalize the independent source model, by consisting of N independent sources $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_N)$, but allowing some of the sources $\mathbf{X}_i$ to be *bad* in the sense that they contain no entropy at all. This setting captures unreliable sources (e.g., Zener diodes) which are known to occasionally produce bits with no entropy, and furthermore has applications in generating a cryptographic common random string (CRS) in the presence of adversaries. Here, our main contribution is as follows.

We explicitly construct the first
extractors for adversarial sources for polylogarithmic entropy.

Our adversarial source extractors enjoy an entropy requirement that is similar to the best known extractors for the (less general) independent source setting [Li15, Li23]. Furthermore, the previous best extractors for adversarial sources (implicit in [KRVZ11]) required min-entropy $k \geq n^{0.99}$, and thus our $k \geq \text{polylog}(n)$ entropy requirement is an exponential improvement. In order to construct our extractors, we establish new connections to coding theory, communication complexity, and extremal combinatorics, and build a variety of new pseudorandom objects along the way.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. In *52nd Annual Symposium on Theory of Computing (STOC 2020)*, pages 1184–1197. ACM, 2020

[CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021

Chapter 4 Leakage-resilient extractors Our second type of *robust extractor for independent sources* can extract uniform bits from a source $\mathbf{X}$ consisting of N independent sources $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$, and its output remains uniform *even conditioned on* the output of multiple leakage functions, each called on $0.99N$ of the input sources. Such an object, called a *leakage-resilient extractor*, was originally motivated by applications in leakage-resilient cryptography [KMS19]. There, the authors posed as an open problem the explicit construction of a leakage-resilient extractor for sublinear entropy. In this chapter,

We explicitly construct the first
leakage-resilient extractors for polylogarithmic entropy.

Thus, we answer the question of Kumar, Meka, and Sahai [KMS19], and in fact give an extractor that works for exponentially less entropy. Our extractor construction is extremely simple, yet powerful enough to have interesting implications in multiparty communication complexity and leakage-resilient cryptography. In cryptography, we obtain leakage-resilient secret-sharing schemes that are exponentially better than the previous state-of-the-art [KMS19]. In communication complexity, we obtain strong average-case lower bounds against a rich spectrum of multiparty communication protocols that interpolate between the well-studied *number-in-hand* (NIH) and *number-on-forehead* (NOF) protocols. One of our lower bounds (against a slightly weaker definition of NOF protocols) is strong enough that, if proven in the standard NOF setting, would yield breakthrough circuit lower bounds [Yao90, BT94, RW93].

[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 *61st Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1226–1242. IEEE, 2020

Chapter 5 Explicit extremal designs For the third and final chapter of Part I we construct a new object which, at first glance, looks totally unrelated to extractors. In particular, we construct an *extremal combinatorial design*. Designs are a special type of *well-balanced hypergraph*, which have become fundamental primitives in the theory of pseudorandomness. More formally, an (n, r, s)-design is just a set system over n elements, where each set has size r, and every pair of sets have intersection size $< s$. Most applications of designs in pseudorandomness [NW94, Tre01, RRV02] require them to be *extremal* in the sense that they have a lot of hyperedges, and it is well-known how to construct such designs via the method of conditional expectations [NW94]. In this chapter, we explicitly construct a different flavor of extremal designs altogether. In particular,

We explicitly construct the first
designs with small independence number.

Prior to our work, the only known constructions of such designs were probabilistic [RŠ94], and our explicit designs achieve parameters that are nearly as good. Our construction is elementary, and simply combines some old results from coding theory [BRC60, Hoc59] with some recent bounds from additive combinatorics [Sid18]. Moreover, these designs play a crucial role in the construction of adversarial source extractors in Chapter 3.

[CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021

Part II Extractors for algebraic sources

In the next part of the thesis, we turn our attention towards the model of *algebraic sources*. Here, recall from above that the most well-studied setting within this family are *affine sources*, which are simply distributions that are uniform over an affine subspace of $\mathbb{F}_2^n$. Our main contribution is a new state-of-the-art extractor.

Chapter 6 Extractors for affine sources In a little more detail, we are motivated to study affine extractors given that they have been gradually emerging as fundamental primitives within complexity theory and the theory of pseudorandomness. In particular, not only have they recently been used to get new state-of-the-art circuit lower bounds [LY22], but a number of surprising connections have also recently been established between affine sources and other well-studied models [DW12, Vio14, CG21]. Despite this, prior to our work, the state-of-the-art extractors for affine sources noticeably lagged behind the state-of-the-art in extractors for independent sources. In this chapter, we resolve this discrepancy.

We explicitly construct the first
extractors for affine sources for almost logarithmic entropy.

Prior to our work, the best-known affine extractors [Li16] required entropy $k \geq \log^C n$ for some large constant C, and our improved $k \geq \log^{1+o(1)}(n)$ entropy requirement is near-optimal [ABGSD21]. Our techniques closely follow a line of work that achieves a similar entropy improvement for the independent source setting [CL16a, Coh16b, Mek17, Coh17, Li17, Li19], except we are forced to develop a host of new pseudorandom objects to deal with the affine correlations that arise from using just one source.

> [CGL21] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 622–633. IEEE, 2021

Part III Extractors for low-complexity sources

For our final act, we study the third and final flavor of structured sources, namely, *low-complexity sources*. Within this model, the two most well-studied settings are sources generated by space-bounded algorithms, and sources generated by low-depth circuits. We give results in both directions, and launch a systematic study into a closely related new subfield of complexity theory.

Chapter 7 Extractors for small-space sources Within the family of low-complexity sources, the model of *small-space sources* may very well be the oldest. Indeed, while such sources were formally introduced in 2006 [KRVZ11], their roots trace all the way back to the 80s [Vaz87b]. Small-space sources, as their name suggests, model distributions that can be generated by space-bounded algorithms. Beyond their natural definition, these sources capture a variety of other well-studied models, meaning that an extractor for small-space sources can be quite the versatile tool. In this chapter, we construct new state-of-the-art extractors for this setting.

We explicitly construct the first
extractors for small-space sources for polylogarithmic entropy.

Previously, the best extractors [CL16b] required entropy $k \geq 2^{\log^{0.51}(n)}$, and thus the $k \geq \text{polylog}(n)$ entropy requirement of our extractors offers a near-exponential improvement. In the more challenging *low-error* setting, we improve the entropy requirement from $k \geq n^{0.99}$ [KRVZ11] to $k \geq n^{0.51}$. In order to construct our low-error extractors, we exploit a known reduction from small-space sources to adversarial source, by combining it with our new extractors for adversarial sources from Chapter 3. On the other hand, in order to obtain our extractors that work for polylogarithmic entropy, we provide a novel reduction from small-space sources to *affine sources*, and combine it with the excellent affine extractor of Li [Li16] (or, for an even better entropy requirement, our new affine extractors from Chapter 6). Beyond its application in getting better extractors for small-space sources, we view our new reduction as an independently interesting structural result, which helps highlight the role of affine sources as a unifying model in extractor theory.

> [CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021

Chapter 8 Extractors for circuit sources Next, we explore the other well-studied flavor of low-complexity sources: namely, *circuit sources*. Circuit sources model distributions that can be sampled by

Boolean circuits, and they were the original model studied in the seminal paper of Trevisan and Vadhan [TV00]. Depending on the exact type of circuit that generates the source, the difficulty of extracting from it can greatly vary. Towards this end, researchers have generally focused on the simplest possible type of circuits: namely, *low-depth circuits*. However, even here, constructing extractors for such sources is remarkably hard. In particular, even for circuits of *constant depth*, the state-of-the-art extractor still requires min-entropy $k \geq \sqrt{n}$, and has been stuck here for quite some time [DW12, Vio14]. This is in stark contrast to almost *every other basic model* that has been studied - most of which readily admit extractors for polylogarithmic entropy by now [Li23]. In this chapter, we provide some hope that this barrier will soon be broken.

We show that random degree 3 *polynomials break the* $\sqrt{n}$ *barrier,*
and fully characterize the power of low-degree polynomials as extractors for such sources.

The reason we study (random) low-degree polynomials as candidate extractors is simple: not only is this class of functions *small* (thereby yielding *semi-explicit* constructions), but they are also well-known to do a great job at extracting from stubborn classes of sources [CT15]. Most notably, however, understanding the power of low-degree polynomials as extractors amounts to proving interesting *structural results* about these fundamental algebraic objects. In this vein, we prove a new type of *Chevalley-Warning theorem*, which asserts the existence of *low-weight* solutions to systems of polynomial equations. We also obtain a new reduction from constant-depth circuit sources to a much more specialized model, which is guaranteed to have uniform bits at certain (unknown) locations. We hope that this new reduction will ultimately facilitate the construction of truly explicit extractors for this challenging model.

[ACG$^+$22] Omar Alrabiah, Eshan Chattopadhyay, Jesse Goodman, Xin Li, and João Ribeiro. Low-degree polynomials extract from local sources. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022

Chapter 9 The space complexity of sampling Up until this point, Part III has focused on constructing extractors for sources that can be sampled by low-complexity computational models. For the final chapter of this thesis, we ask an even more fundamental question: *What do these sources even look like?* In other words, we ask what distributions can - *and cannot* - be sampled by algorithms with bounded computational resources. This question lies at the heart of a rich new area of complexity theory, called the *complexity of sampling* [Vio12a]. In this new area of study, the goal is to understand the power of our favorite computational models for the task of *sampling* from distributions. Lower bounds in this setting are more difficult to obtain than classical lower bounds, and often lead to unexpected tools, tricks, and techniques. As a result, the complexity of sampling has recently seen a huge wave of interest. Despite this exciting progress, prior to our work, nothing was known about the complexity of sampling with *limited memory*. In this chapter, we initiate such a study.

We launch a systematic study into the space complexity of sampling.
We obtain strong explicit lower bounds, and prove a more general direct product theorem.

Our direct product theorem is the first result of its type, and our sampling lower bounds imply (via a known connection) new data structure lower bounds for storing codewords. Moreover, in order to lay the groundwork for this new direction, we prove several other results, including a collection of *equivalence* *theorems* that provide a black-box way to convert sampling lower bounds into computing lower bounds. We view the *complexity of sampling* as a fundamental new field at the intersection of extractors and complexity theory, and we are see excited to see where it goes next.

[CGZ22] Eshan Chattopadhyay, Jesse Goodman, and David Zuckerman. The space complexity of sampling. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *LIPIcs*, pages 40:1–40:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022

Chapter 2

Preliminaries

We start with some preliminaries that will be used throughout. We aim to keep this chapter light, and only include a few basic definitions and results that are exploited throughout the entirety of this thesis. All other tools and techniques that we need will be introduced naturally in the chapters in which they are used.

2.1 Notation

To start, let us introduce some basic notation. We let $\mathbb{N} := \{1, 2, \ldots, \}$ denote the natural numbers, and for any $n \in \mathbb{N}$, we define $[n] := \{1, 2, \ldots, n\}$. As usual, $\mathbb{R}$ is the set of real numbers, $\mathbb{C}$ is the set of complex numbers, and for a prime power q, $\mathbb{F}_q$ is the finite field of size q. This thesis will mostly focus on the finite field $\mathbb{F}_q = \mathbb{F}_2$, which is just the set $\{0, 1\}$ with addition and multiplication modulo 2. Occasionally, we will also consider the n-dimensional vector space over this field, which we denote as $\mathbb{F}_2^n$. Throughout the thesis, we treat N and n as different variables, with no relation to one another (in particular, we *do not* adopt the typical convention that $N = 2^n$). Finally, all logarithms are base 2.

Indexing

For a bitstring $x \in \{0, 1\}^n$, we let $x_i \in \{0, 1\}$ denote the value held at its i^{th} coordinate, and for a set $S \subseteq [n]$ we let $x_S \in \{0, 1\}^{|S|}$ denote the the concatenation of bits $x_i, i \in S$ in increasing order of i. We also exploit the shorthand x_{-i} to denote the string $x_{[n]\setminus\{i\}}$, and we let $x_{i\to j} := (x_i, x_{i+1}, \ldots, x_j)$. We build on the latter notation, and let $x_{\leq i} := x_{1\to i}$ and $x_{<i} := x_{i\to i-1}$. Similarly, $x_{\geq i} := x_{i\to n}$ and $x_{>i} := x_{i+1\to n}$.

We extend all of this to domains beyond $\{0, 1\}$, and heavily use the following notation, which is crucial to internalize. Given a domain $\mathcal{D}$ and some string $x \in \mathcal{D}^n$, we let $x_i \in \mathcal{D}$ denote the value held at its i^{th} coordinate, and for a set $S \subseteq [n]$ we let $x_S \in \mathcal{D}^{|S|}$ denote the concatenation of values $x_i, i \in S$ in increasing order of i. For example, if $x \in (\{0, 1\}^t)^n$, then $x_1 \in \{0, 1\}^t$ holds the first t consecutive bits in x, x_2 holds the next t consecutive bits, and so on, all the way to x_n. Finally, for $x \in (\{0, 1\}^t)^n$, we define $x_{i,j} := (x_i)_j$. One last note about bitstrings: we define the *support* of $x \in \{0, 1\}^n$ as the set of coordinates i where $x_i = 1$.

Operations

To complete this section, we explain our notation for several common operations. We use the standard notation $\binom{n}{k}$ to denote a binomial coefficient, and we write $\binom{n}{\leq k} := \sum_{i=0}^{k} \binom{n}{i}$. We overload this notation for sets as follows: given a set S, we let $\binom{S}{k}$ denote the collection of all subsets of S of size k, and $\binom{S}{\leq k}$ the collection of subsets of size at most k. Speaking of sets, given some set $S \subseteq [n]$ and $x \in \{0, 1\}^n$, we define

$x^S := \prod_{i \in S} x_i$, which should not be confused with the notation x_S from above. Naturally, we let $x^\emptyset := 1$. Given a set $S \subseteq [n]$, we write $\overline{S}$ to denote its complement, and given sets $A, B \subseteq \mathbb{F}_2^n$, we let

$$A + B := \{a + b : a \in A, b \in B\}$$

denote the *sumset* of A and B. Returning to bitstrings, given some bitstrings $x, y \in \{0,1\}^n$, we let $x \circ y \in \{0,1\}^{2n}$ denote their *concatenation*: notably, this notation *does not* denote function composition in this thesis. Finally, we write $x \oplus y$ to denote the bitwise XOR of these strings - or rather their *sum* when they are treated as vectors in $\mathbb{F}_2^n$. For this reason, we interchangeably write $x \oplus y$ and $x + y$.

2.2 Probability

We now overview some basic notions from probability theory that are heavily used throughout this thesis. Notably, we will only ever need *discrete* probability theory. Here, random variables are denoted by boldface letters such as $\mathbf{X}$, and we let $\text{support}(\mathbf{X})$ denote its support. For a set S, we write $\mathbf{X} \sim S$ to denote that $\text{support}(\mathbf{X}) \subseteq S$. Moreover, all notation from Section 2.1 carries over to any discussion of random variables. For example, given a random variable $\mathbf{X} \sim \{0,1\}^n$, the notation $\mathbf{X}_i$ denotes the value held at its i^{th} bit. Finally, the uniform distribution over $\{0,1\}^m$ is denoted by $\mathbf{U}_m$, and whenever it appears in an expression with other random variables, it is assumed to be independent of them all. However, whenever $\mathbf{U}_m$ appears multiple times in the same expression or formula, this represents multiple copies of the *same* random variable. For example, the random variable $(\mathbf{U}_m, \mathbf{U}_m)$ hits each string $(x, x) \in \{0,1\}^m \times \{0,1\}^m$ with probability 2^{-m}.

Statistical distance

The primary way we measure the *distance* between two random variables is via *statistical distance*.

Definition 2.1 (Statistical distance)**.** *For any random variables $\mathbf{X}, \mathbf{Y} \sim V$, we define*

$$|\mathbf{X} - \mathbf{Y}| := \max_{S \subseteq V} |\Pr[\mathbf{X} \in V] - \Pr[\mathbf{Y} \in V]| = \frac{1}{2} \sum_{v \in V} |\Pr[\mathbf{X} = v] - \Pr[\mathbf{Y} = v]|$$

as the statistical distance *(or* total variation distance*) between $\mathbf{X}$ and $\mathbf{Y}$.*

We say that $\mathbf{X}, \mathbf{Y}$ are *ε-close* if $|\mathbf{X} - \mathbf{Y}| \leq \varepsilon$, and denote this as $\mathbf{X} \approx_\varepsilon \mathbf{Y}$. On the other hand, we say that $\mathbf{X}, \mathbf{Y}$ are *ε-far* if $|\mathbf{X} - \mathbf{Y}| > \varepsilon$, and denote this by $\mathbf{X} \not\approx_\varepsilon \mathbf{Y}$. As it turns out, statistical distance is truly a *distance* (i.e., metric), in every sense of the word. We record the following result.

Fact 2.1 (Statistical distance is a metric)**.** *For any random variables $\mathbf{X}, \mathbf{Y} \sim V$, all of the following hold.*

- $|\mathbf{X} - \mathbf{Y}| \geq 0$, *with equality holding if and only if* $\mathbf{X} \equiv \mathbf{Y}$. (Positivity)
- $|\mathbf{X} - \mathbf{Y}| = |\mathbf{Y} - \mathbf{X}|$. (Symmetry)
- $|\mathbf{X} - \mathbf{Y}| \leq |\mathbf{X} - \mathbf{Z}| + |\mathbf{Z} - \mathbf{Y}|$, *for any random variable* $\mathbf{Z} \sim V$. (Triangle inequality)

The triangle inequality will be especially useful. Next, if two random variables have identically distributed marginals, we can condition on these without changing their statistical distance (in expectation).

Fact 2.2 (Averaging principle)**.** *For any random variables* $\mathbf{X}, \mathbf{Y} \sim V$ *and* $\mathbf{Z}, \mathbf{Z}' \sim Z$ *such that* $\mathbf{Z} \equiv \mathbf{Z}'$*,*

$$|\mathbf{X} \circ \mathbf{Z} - \mathbf{Y} \circ \mathbf{Z}'| = \mathbb{E}_{z \sim \mathbf{Z}}[|(\mathbf{X} \mid \mathbf{Z} = z) - (\mathbf{Y} \mid \mathbf{Z}' = z)|].$$

Proof. By the definition of statistical distance, we have

$$\begin{aligned}
2|\mathbf{X} \circ \mathbf{Z} - \mathbf{Y} \circ \mathbf{Z}'| &= \sum_{(v,z)} |\Pr[\mathbf{X} \circ \mathbf{Z} = (v,z)] - \Pr[\mathbf{Y} \circ \mathbf{Z}' = (v,z)]| \\
&= \sum_{(v,z)} \Pr[\mathbf{Z} = z] |\Pr[\mathbf{X} = v \mid \mathbf{Z} = z] - \Pr[\mathbf{Y} = v \mid \mathbf{Z}' = z]| \\
&= \sum_{z} \Pr[\mathbf{Z} = z] \sum_{v} |\Pr[\mathbf{X} = v \mid \mathbf{Z} = z] - \Pr[\mathbf{Y} = v \mid \mathbf{Z}' = z]| \\
&= 2\mathbb{E}_{z \sim \mathbf{Z}}[|(\mathbf{X} \mid \mathbf{Z} = z) - (\mathbf{Y} \mid \mathbf{Z}' = z)|].
\end{aligned}$$

□

Data-processing inequality

Perhaps the most fundamental fact about statistical distance is the data-processing inequality. Almost all facts about statistical distance can be obtained via this simple observation.

Fact 2.3 (Data-processing inequality)**.** *For any random variables* $\mathbf{X}, \mathbf{Y} \sim V$ *and any function* $f : V \to W$*,*

$$|\mathbf{X} - \mathbf{Y}| \geq |f(\mathbf{X}) - f(\mathbf{Y})|.$$

Proof. By definition of statistical distance, there is a set $T \subseteq W$ where $|\Pr[f(\mathbf{X}) \in T] - \Pr[f(\mathbf{Y}) \in T]| = |f(\mathbf{X}) - f(\mathbf{Y})|$. Now, consider the set $S^* \subseteq V$ defined as $S^* := \{v \in V : f(v) \in T\}$. We have

$$\begin{aligned}
|\mathbf{X} - \mathbf{Y}| &\geq |\Pr[\mathbf{X} \in S^*] - \Pr[\mathbf{Y} \in S^*]| \\
&= |\Pr[f(\mathbf{X}) \in T] - \Pr[f(\mathbf{Y}) \in T]| = |f(\mathbf{X}) - f(\mathbf{Y})|.
\end{aligned}$$

□

As it turns out, if f is injective over the supports of $\mathbf{X}, \mathbf{Y}$, then this inequality becomes tight. In other words, the fact below follows immediately from two applications of the data-processing inequality.

Corollary 2.1 (Data-processing equality)**.** *For any random variables* $\mathbf{X}, \mathbf{Y} \sim V$ *and function* $f : V \to W$ *that is injective over support*$(\mathbf{X}) \cup$ *support*$(\mathbf{Y})$*,*

$$|\mathbf{X} - \mathbf{Y}| = |f(\mathbf{X}) - f(\mathbf{Y})|.$$

Proof. Since f is injective over support$(\mathbf{X}) \cup$ support$(\mathbf{Y})$, there is some function $f^{-1} : W \to V$ such that $f^{-1}(f(\mathbf{X})) \equiv \mathbf{X}$ and $f^{-1}(f(\mathbf{Y})) \equiv \mathbf{Y}$. Thus, by the data-processing inequality (Fact 2.3),

$$|\mathbf{X} - \mathbf{Y}| \geq |f(\mathbf{X}) - f(\mathbf{Y})| \geq |f^{-1}(f(\mathbf{X})) - f^{-1}(f(\mathbf{Y}))| = |\mathbf{X} - \mathbf{Y}|,$$

which immediately implies the claimed result. □

Convex combinations

Finally, we say that $\mathbf{X}$ is a *convex combination* of distributions $\{\mathbf{Y}_i\}$ if there exist probabilities $\{p_i\}$ summing to 1 such that $\mathbf{X} = \sum_i p_i \mathbf{Y}_i$, meaning that $\mathbf{X}$ samples from $\mathbf{Y}_i$ with probability p_i. To start, one silly (but important) observation is that a convex combination of convex combinations is a convex combination. More formally, if $\mathbf{X}$ is a convex combination of distributions $\{\mathbf{Y}_i\}$, and each $\mathbf{Y}_i$ is a convex combination of some $\{\mathbf{Z}_i\}$, then $\mathbf{X}$ is a convex combination of distributions $\{\mathbf{Z}_i\}$. Getting a little more serious, the following fact is crucial in many extractor proofs.

Fact 2.4 (Convex combination of close distributions is close)**.** *Let $\mathbf{X}, \mathbf{Q} \sim V$ be any two random variables. Suppose that $\mathbf{X}$ is a convex combination of distributions $\mathbf{X} = \sum_{i\in[t]} p_i \mathbf{Y}_i$, where for each $\mathbf{Y}_i$ we have $|\mathbf{Y}_i - \mathbf{Q}| \leq \varepsilon$. Then*

$$|\mathbf{X} - \mathbf{Q}| \leq \varepsilon.$$

Proof. By definition of statistical distance, there is a set $S \subseteq V$ such that $|\mathbf{X} - \mathbf{Q}| = |\Pr[\mathbf{X} \in S] - \Pr[\mathbf{Q} \in S]|$. And by definition of convex combination,

$$\begin{aligned} |\Pr[\mathbf{X} \in S] - \Pr[\mathbf{Q} \in S]| &= \left| \sum_i p_i \Pr[\mathbf{Y}_i \in S] - \sum_i p_i \Pr[\mathbf{Q} \in S] \right| \\ &\leq \sum_i p_i |\Pr[\mathbf{Y}_i \in S] - \Pr[\mathbf{Q} \in S]| \leq \varepsilon, \end{aligned}$$

which completes the proof. □

It is natural to ask whether the *opposite* of such a result can be proven. Namely, is a convex combination of *far* distributions *far*? With a little thought, one might realize the answer must is a resounding *no*: given an arbitrary distribution $\mathbf{Q}$, we can write it as a convex combination of random variables $\mathbf{Q} = \sum_i p_i \mathbf{Q}_i$ such that each $\mathbf{Q}_i$ is *constant*. In this case, each $|\mathbf{Y}_i - \mathbf{Q}|$ is huge, yet clearly, $|\sum_i p_i \mathbf{Y}_i - \mathbf{Q}| = 0$. As it turns out, however, if there are *not too many participants* in the convex combination, then such a result can be proven! We show that for any distribution $\mathbf{Q}$, a convex combination of not-too-many distributions, each far from $\mathbf{Q}$, is itself far from $\mathbf{Q}$. This slightly generalizes a result from [Vio20], where it was shown to hold for convex combinations of the form $\sum_i p_i \mathbf{Y}_i$, where each p_i is the same.

Fact 2.5 (Convex combination of (not too many) far distributions is far)**.** *Let $\mathbf{X}, \mathbf{Q} \sim V$ be any two random variables. Suppose that $\mathbf{X}$ is a convex combination of t distributions $\mathbf{X} = \sum_{i\in[t]} p_i \mathbf{Y}_i$, where for each $\mathbf{Y}_i$ we have $|\mathbf{Y}_i - \mathbf{Q}| \geq 1 - \varepsilon$. Then*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - t\varepsilon.$$

Proof. Let Q denote the support of $\mathbf{Q}$. By definition of statistical distance, we know that for each $i \in [t]$ there is a set S_i such that $|\mathbf{Y}_i - \mathbf{Q}| = \Pr[\mathbf{Q} \in Q - S_i] - \Pr[\mathbf{Y}_i \in Q - S_i] = 1 - (\Pr[\mathbf{Q} \in S_i] + \Pr[\mathbf{Y}_i \in Q - S_i])$. And by the lemma hypothesis, we know that $\Pr[\mathbf{Q} \in S_i] + \Pr[\mathbf{Y}_i \in Q - S_i] \leq \varepsilon$.

Now, define $S = \bigcup_i S_i$. By definition of statistical distance, $|\mathbf{X} - \mathbf{Q}| \geq 1 - (\Pr[\mathbf{Q} \in S] + \Pr[\mathbf{X} \in Q - S])$. Thus it suffices to show $\Pr[\mathbf{Q} \in S] + \Pr[\mathbf{X} \in Q - S] \leq t\varepsilon$ to complete the proof. Towards this end, observe that we can write 1 as $\sum_j p_j$ to obtain

$$\Pr[\mathbf{Q} \in S] \leq \sum_{i\in[t]} \Pr[\mathbf{Q} \in S_i] = \sum_{i,j\in[t]} p_j \Pr[\mathbf{Q} \in S_i].$$

Next, observe that

$$\Pr[\mathbf{X} \in Q - S] = \sum_{j\in[t]} p_j \Pr[\mathbf{Y}_j \in Q - S] \leq \sum_{i,j\in[t]} p_j \Pr[\mathbf{Y}_j \in Q - S_i].$$

Thus we have

$$\begin{aligned}\Pr[\mathbf{Q} \in S] + \Pr[\mathbf{X} \in Q - S] &\leq \sum_{i,j\in[t]} p_j \cdot (\Pr[\mathbf{Q} \in S_i] + \Pr[\mathbf{Y}_j \in Q - S_i]) \\ &\leq \sum_{i,j\in[t]} p_j \cdot \varepsilon \leq t\varepsilon,\end{aligned}$$

as desired. □

Fact 2.6 (Closeness to convex combinations)**.** *Let $\mathcal{X}$ be a family of distributions over V, and let $\mathbf{X} \sim V$ be a random variable for which there exists some $\mathbf{Z} \sim Z$ such that*

$$\Pr_{z\sim\mathbf{Z}}[(\mathbf{X} \mid \mathbf{Z} = z) \notin \mathcal{X}] \leq \varepsilon.$$

Then $\mathbf{X}$ is ε-close to a convex combination of distributions from $\mathcal{X}$.

Proof. Let $Z^* \subseteq Z$ hold all z such that $(\mathbf{X} \mid \mathbf{Z} = z) \in \mathcal{X}$. Pick an arbitrary $z^* \in Z^*$ and consider the convex combination

$$\mathbf{X}' := \sum_{z\in Z^*} \Pr[\mathbf{Z} = z] \cdot (\mathbf{X} \mid \mathbf{Z} = z) + \sum_{z\notin Z^*} \Pr[\mathbf{Z} = z] \cdot (\mathbf{X} \mid \mathbf{Z} = z^*).$$

By construction, $\mathbf{X}'$ is a convex combination of distributions from $\mathcal{X}$. On the other hand, $\mathbf{X}$ is a convex combination of the form

$$\mathbf{X} = \sum_{z\in Z^*} \Pr[\mathbf{Z} = z] \cdot (\mathbf{X} \mid \mathbf{Z} = z) + \sum_{z\notin Z^*} \Pr[\mathbf{Z} = z] \cdot (\mathbf{X} \mid \mathbf{Z} = z).$$

By definition of convex combination, it is straightforward to verify that for any $S \subseteq \{0,1\}^n$,

$$\begin{aligned}&|\Pr[\mathbf{X} \in S] - \Pr[\mathbf{X}' \in S]| \\ &= |\sum_{z\notin Z^*} \Pr[\mathbf{Z} = z] \cdot (\Pr[\mathbf{X} \in S \mid \mathbf{Z} = z^*] - \Pr[\mathbf{X} \in S \mid \mathbf{Z} = z])| \\ &\leq \Pr[\mathbf{Z} \notin Z^*] \\ &\leq \varepsilon.\end{aligned}$$

Thus, we get that $\mathbf{X}$ is ε-close to $\mathbf{X}'$, a convex combination of distributions from $\mathcal{X}$, as desired. □

2.3 Extractors

We now turn towards some preliminaries on extractors. We start by reviewing some basic definitions that we highlighted in the introduction, before discussing the powerful notion of *reductions* between extractors.

The basics

We repeat some basic definitions from the introduction, so that all of the necessary technical preliminaries can be found in one place. First, let us recall the definition of an extractor.

Definition 2.2 (Extractor)**.** *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *is called an* extractor for $\mathcal{X}$ with error ε *if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \varepsilon.$$

For short, we will sometimes simply call such an object an *ε-extractor for $\mathcal{X}$*. Now, a nontrivial extractor has error $\varepsilon < 1 - 2^{-m}$, as a function that outputs the all zeroes string always achieves error $\varepsilon = 1 - 2^{-m}$. A weaker object, which also boasts an impressive list of applications, is a *disperser*.

Definition 2.3 (Disperser)**.** *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function* $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}^m$ *is called a* disperser *for $\mathcal{X}$ with error ε if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathit{support}(\mathsf{Disp}(\mathbf{X}))| \geq (1 - \varepsilon) \cdot 2^m.$$

A nontrivial disperser has error $\varepsilon < 1 - 2^{-m}$, as a function that outputs the all zeroes string always achieves error $\varepsilon = 1 - 2^{-m}$. Furthermore, notice that an extractor for $\mathcal{X}$ with error ε is automatically a disperser for $\mathcal{X}$ with error ε, because if it missed $> \varepsilon 2^m$ points in the support, then it must have statistical distance $> \varepsilon 2^m \cdot 2^{-m} = \varepsilon$, as witnessed by these points. Given a family $\mathcal{X}$ of distributions, the simplest possible object we can ask to construct is a nontrivial disperser with 1 bit of output for $\mathcal{X}$. This is equivalent to a function that is not constant on any $\mathbf{X} \in \mathcal{X}$.

When trying to figure out what we should extract from, we observed that a minimum possible requirement is that the source has some entropy. Following the lead of Chor and Goldreich [CG88] and Zuckerman [Zuc90] and the rest of the extractor community, we used *min-entropy*, defined again below.

Definition 2.4 (Min-entropy)**.** *The* min-entropy *of a random variable $\mathbf{X} \sim \{0,1\}^n$ is defined as*

$$H_\infty(\mathbf{X}) := \min_{x \in \mathit{support}(\mathbf{X})} \log\left(\frac{1}{\Pr[\mathbf{X} = x]}\right).$$

In other words, we have that $H_\infty(\mathbf{X}) \geq k$ if and only if $\Pr[\mathbf{X} = x] \leq 2^{-k}$ for all $x \in \mathit{support}(\mathbf{X})$.

However, we observed that this alone is not enough to even allow a nontrivial disperser. More formally, let us say that an (n,k)-source is a random variable $\mathbf{X} \sim \{0,1\}^n$ with min-entropy at least k. We have the following classic impossibility result.

Fact 2.7 (Classic impossibility result [SV86, CG88])**.** *There cannot exist an extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ *with error $\varepsilon < 1/2$ for the family $\mathcal{X}$ $(n, n-1)$-sources.*

Proof. Let $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ be an arbitrary candidate extractor for $\mathcal{X}$. Since Ext is a deterministic function, at least half of its inputs map to the same output. Without loss of generality, assume that half of its inputs map to 0. That is, $|\mathsf{Ext}^{-1}(0)| \geq 2^{n-1}$. Consider now a random variable $\mathbf{X} \sim \{0,1\}^n$ that is uniform over the set $\mathsf{Ext}^{-1}(0)$. Clearly $\mathsf{Ext}(\mathbf{X})$ is always 0, by definition of $\mathbf{X}$. On the other hand, $\mathbf{X}$ is uniform over a set of size at least 2^{n-1}, meaning it has min-entropy $k \geq n - 1$. Thus we have found a source $\mathbf{X} \in \mathcal{X}$ for which $|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_1| = |0 - \mathbf{U}_1| = 1/2$, meaning Ext cannot be an extractor for $\mathcal{X}$ with error $\varepsilon < 1/2$. □

Recall that given this impossibility result, there are two options: construct a *seeded extractor*, or *seedless extractor*. We define them, below.

Definition 2.5 (Seeded extractor). *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function* $\mathsf{sExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is called a* seeded extractor *for $\mathcal{X}$ with error ε if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}, \mathbf{U}_d) - \mathbf{U}_m| \le \varepsilon.$$

We say that the extractor is strong *if we have the stronger condition* $|\mathsf{Ext}(\mathbf{X}, \mathbf{U}_d) \circ \mathbf{U}_d - \mathbf{U}_m \circ \mathbf{U}_d| \le \varepsilon$.

Note that the (non-strong) definition of a seeded extractor is captured by the general definition of an extractor (Definition 2.2), by redefining the distribution family $\mathcal{X}$ to consist of random variables over d more bits, where the suffix is uniform and independent of the prefix. However, it is often more helpful to just treat them as separate primitives. Recall that near-optimal seeded extractors are now known, and this thesis focuses on constructing seed*less* extractors, defined below.

Definition 2.6 (Seedless extractor). *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *is called a* seedless extractor *for $\mathcal{X}$ with error ε if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \le \varepsilon.$$

Indeed, the keen reader will notice that this is the exact same definition as Definition 2.2. In the introduction, we mentioned that there are three different "flavors" of seedless extractors: extractors for independent sources, extractors for algebraic sources, and extractors for low-complexity sources. These different extractors simply refer to the specific "flavor" of distribution class $\mathcal{X}$ which we try to extract from. We go into much greater detail about these flavors in Part I, Part II, and Part III, respectively. For now, however, we focus on a common technique present across almost all of the chapters: reductions between extractors.

Reductions between extractors

It is often the case that we know how to extract from a *well-behaved* family $\mathcal{X}$, and are presented with a complicated source $\mathbf{Y}$ that we do not know how to extract from. The following fact is a crucial tool in extractor theory, which says that as long as $\mathbf{Y}$ is close to a convex combination of sources from $\mathcal{X}$, then the extractor for $\mathcal{X}$ automatically works on $\mathbf{Y}$!

Fact 2.8 (Convex combinations suffice). *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$, and let* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *be an extractor for $\mathcal{X}$ with error ε. Then for any $\mathbf{Y}$ that is δ-close to a convex combination of distributions from $\mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{Y}) - \mathbf{U}_m| \le \delta + \varepsilon.$$

In other words, Ext *is also an extractor for $\mathbf{Y}$ with error $\delta + \varepsilon$.*

Proof. Let $\mathbf{X} = \sum_i p_i \mathbf{X}_i$ be the convex combination of distributions from $\mathcal{X}$ that $\mathbf{Y}$ is δ-close to. Then

$$\begin{aligned}
|\mathsf{Ext}(\mathbf{Y}) - \mathbf{U}_m| &\le |\mathsf{Ext}(\mathbf{Y}) - \mathsf{Ext}(\mathbf{X})| + |\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| && \text{(triangle inequality)}\\
&= |\mathbf{Y} - \mathbf{X}| + |\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| && \text{(data-processing inequality)}\\
&\le \delta + \sum_i p_i \cdot |\mathsf{Ext}(\mathbf{X}_i) - \mathbf{U}_m| && \text{(definition of } \mathbf{X} \text{ and triangle inequality)}\\
&\le \delta + \varepsilon,
\end{aligned}$$

as desired. □

Let us see one example of the above fact in action. Suppose we wish to construct an extractor for independent sources. Each source is independent, and guaranteed to have min-entropy at least k. Apriori, we cannot assume that each source has any more structure than that. However, one might reason that it could be much easier to extract from these sources if we could assume that they are *uniform* over their support. Such sources, called *flat sources*, have a much more "combinatorial" flavor than standard (n, k) sources.

Definition 2.7 (Flat sources)**.** *We call a source* $\mathbf{X}$ flat *if it is uniformly distributed over its support.*

As it turns out, the following classical result shows that, in fact, we *can* assume that an arbitrary (n, k) source is flat when trying to extract![1]

Fact 2.9 (Flatness for free [CG88])**.** *Every* (n, k)*-source is a convex combination of flat* (n, k) *sources.*

The proof that we provide is adapted from [Vad12, Lemma 6.10], and is due to Kalai.

Proof. Suppose we could define a sequence ("coupling") of random variables $(\mathbf{X}_1, \dots, \mathbf{X}_{2^k})$ over the same space such that each marginal $\mathbf{X}_i \sim \{0,1\}^n$ is distributed according to $\mathbf{X}$, and such that there are *no collisions* between marginals: namely, for any $i \neq i'$, we have $\Pr[\mathbf{X}_i = \mathbf{X}_{i'}] = 0$. Then, $\mathbf{X}$ is a convex combination of flat (n, k) sources. To see why, consider the function $f : (\{0,1\}^n)^{2^k} \times [k] \to \{0,1\}^n$ defined as $f(x_1, \dots, x_{2^k}, i) := x_i$, and the random variable

$$\mathbf{X}' := f(\mathbf{X}_1, \dots, \mathbf{X}_{2^k}, \mathbf{U}),$$

where $\mathbf{U}$ is uniform and independent of $\mathbf{X}_1, \dots, \mathbf{X}_{2^k}$. Observe that $\mathbf{X}' \equiv \mathbf{X}$, since each $(\mathbf{X}' \mid \mathbf{U} = u) = \mathbf{X}_u \equiv \mathbf{X}$. Furthermore, observe that $\mathbf{X}'$ is a convex combination of flat (n, k)-sources, since each $(\mathbf{X}' \mid \mathbf{X}_1 = x_1, \dots, \mathbf{X}_{2^k} = x_{2^k})$ is a flat (n, k) source. Thus $\mathbf{X}'$ (and therefore $\mathbf{X}$) is a convex combination of flat (n, k)-sources. All that remains is to design the sequence $(\mathbf{X}_1, \dots, \mathbf{X}_{2^k})$ from the beginning.

Towards this end, as suggested by Kalai and described in [Vad12], define $\{z_1, \dots, z_\ell\}$ to be the support of $\mathbf{X}$, and consider the unit circle divided into ℓ half-open intervals $I_1, \dots, I_\ell$, where I_j has length $\Pr[\mathbf{X} = z_j]$. Place 2^k points around the circle $v_1, \dots, v_{2^k}$ such that they are evenly spaced. Perform a uniformly random rotation of the circle, and define the random variable $\mathbf{X}_i$ as z_j if and only if v_i lands on interval I_j. It is straightforward to verify that each $\mathbf{X}_i \equiv \mathbf{X}$. To see why each $\mathbf{X}_i$ will realize to a different value, recall that the points are evenly spaced, and thus are at least 2^{-k} units apart on the perimeter of the circle. Each interval, on the other hand, has length $\Pr[\mathbf{X} = z_j] \leq 2^{-k}$ and is half-open. Thus, no two distinct $\mathbf{X}_i, \mathbf{X}_{i'}$ can land in the same interval. □

While the above demonstrates one way to execute and exploit Fact 2.8, its main use is in showing that an extractor for flat *independent sources* automatically works for arbitrary *independent sources*. For other sources $\mathbf{X}$, the goal is to construct a random variable $\mathbf{Z}$ such that $(\mathbf{X} \mid \mathbf{Z} = z)$ is in the family $\mathcal{X}$, with high probability over fixing $\mathbf{Z} = z$ (and then apply Fact 2.6). But recall that the family $\mathcal{X}$ is typically defined via (1) some min-entropy requirement, and (2) some *structure*. It is always easy to construct a random variable $\mathbf{Z}$ such that $(\mathbf{X} \mid \mathbf{Z} = z)$ contains the desired structure. Indeed, if you fix all the randomness, you just get a constant random variable, which is as structured as possible. However, ensuring that (1) is true at the same time is what makes this task challenging. Towards this end, the key tool in every extractor theorist's toolbox is the following result. It says that as long as $\mathbf{Z}$ has support size at most 2^ℓ, then the min-entropy of $\mathbf{X}$ will drop by at most (roughly) ℓ bits.

[1] However, caution is warranted. While the result states that an (n, k) source $\mathbf{X}$ is a convex combination of flat (n, k) sources, if we originally knew that $\mathbf{X}$ had even more structure than being an (n, k) source, than this proof may remove that structure. Thus it cannot be applied to assume that certain *structured* (n, k) sources are flat.

Lemma 2.1 (Min-entropy chain rule [MW97])**.** *For any random variables $\mathbf{X} \sim X$, $\mathbf{Y} \sim Y$, and any $\varepsilon > 0$,*

$$\Pr_{y \sim \mathbf{Y}}[H_\infty(\mathbf{X} \mid \mathbf{Y} = y) \geq H_\infty(\mathbf{X}) - \log |Y| - \log(1/\varepsilon)] > 1 - \varepsilon.$$

Proof. Define $k := H_\infty(\mathbf{X})$. Fix some $y \in Y$. Let's analyze the value of $k_y := H_\infty(\mathbf{X} \mid \mathbf{Y} = y)$. Towards this end, note that for any x we have

$$\Pr[\mathbf{X} = x \mid \mathbf{Y} = y] = \frac{\Pr[\mathbf{X} = x \wedge \mathbf{Y} = y]}{\Pr[\mathbf{Y} = y]} \leq \Pr[\mathbf{X} = x] / \Pr[\mathbf{Y} = y].$$

Thus, by definition of min-entropy, we have $k_y \geq k - \log(1/\Pr[\mathbf{Y} = y])$. Now, suppose that $k_y < k - \log |Y| - \log(1/\varepsilon)$. By the above, we know $\Pr[\mathbf{Y} = y] < \varepsilon/|Y|$. Thus we have

$$\Pr_{y \sim \mathbf{Y}}[k_y < k - \log |Y| - \log(1/\varepsilon)] = \sum_{y : k_y < k - \log |Y| - \log(1/\varepsilon)} \Pr[\mathbf{Y} = y] < \sum_y \varepsilon/|Y| = \varepsilon,$$

as desired. □

In fact, the random variable $\mathbf{Y}$ that we are conditioning on often has much more structure. In many cases, $\mathbf{Y}$ is in fact *flat*. In this case, it is easy to obtain the following, cleaner version of the min-entropy chain rule.

Lemma 2.2 (Min-entropy chain rule, special case)**.** *For any random variables $\mathbf{X} \sim X, \mathbf{Y} \sim Y$ such that $\mathbf{Y}$ is uniform over Y, and any $y \in Y$,*

$$H_\infty(\mathbf{X} | \mathbf{Y} = y) \geq H_\infty(\mathbf{X}) - \log |Y|.$$

Proof. As in the proof to Lemma 2.1, we know that $H_\infty(\mathbf{X} \mid \mathbf{Y} = y) \geq H_\infty(\mathbf{X}) - \log(1/\Pr[\mathbf{Y} = y])$. Since $\mathbf{Y}$ is uniform over Y, we know that $\Pr[\mathbf{Y} = y] = 1/|Y|$, and the result immediately follows. □

Equipped with these tools, we are now ready to fight even the hairiest, worst-behaved sources.

2.4 Codes

Finally, we will complete our preliminaries by reviewing standard notions from coding theory. We start with some basic definitions, before formally giving the definition of a code (and one of its well-known generalizations). We conclude by recording some classical bounds on the (non)existence of such objects.

The basics

We now review some basics of coding theory, which will be used throughout the thesis. First, given a string $v \in \{0,1\}^n$, we let $\Delta(v) := \{i \in [n] : v_i\}$ denote its *Hamming weight*, and for any strings $u, v \in \{0,1\}^n$, we let $\Delta(u, v) := \{i \in [n] : u_i \neq v_i\}$ denote their *Hamming distance*. Then, given any string $v \in \{0,1\}^n$ and integer $r \in \{0, 1, \dots, n\}$, we let

$$\mathcal{B}(v, r) := \{v' \in \{0,1\}^n : \Delta(v, v') \leq r\}$$

denote the (closed) Hamming ball of radius r around (i.e., centered at) v. If we replace the inequality "$\leq$" above with a strict inequality "$<$", we call this object the *open* Hamming ball of radius r around v, and denote it by $\mathcal{B}^-(v, r)$. Notice that the closed Hamming ball has size $\binom{n}{\leq r}$, whereas the open Hamming ball has size $\binom{n}{\leq r-1}$. With these definitions in hand, we can now define error-correcting codes (*codes*, for short).

The codes

A *code* is simply a subset of $\{0,1\}^n$ such that any pair of points in the subset are far apart.

Definition 2.8 (Code)**.** *An* (n,k,d) code *is a subset* $Q \subseteq \{0,1\}^n$ *of size* 2^k *such that the Hamming distance between any distinct* $x, y \in Q$ *is at least* d*. We say* Q *is a* linear $[n,k,d]$ *code if it is also a subspace of* $\mathbb{F}_2^n$*.*

We say an (n,k,d)-code has dimension k, rate k/n, distance d, and relative distance d/n. Next, a standard relaxation of (n,k,d) codes are *list-decodable codes*.

Definition 2.9 (List-decodable code)**.** *A subset* $Q \subseteq \{0,1\}^n$ *is a* (ρ, L)*-list-decodable code if every Hamming ball in* $\{0,1\}^n$ *of radius at most* ρn *contains at most* L *codewords.*

A straightforward application of the triangle inequality shows that every (n,k,d) code has the following list decoding properties:

Fact 2.10. *If* $Q \subseteq \{0,1\}^n$ *is an* (n,k,d) *code, then* Q *is* (ρ, L) *list decodable for* $L = 1$ *and any* $\rho < \frac{d}{2n}$*.*

Now that we know what codes are, it's natural to ask whether such objects exist. We answer this, next.

The bounds

We will exploit two different classical bounds from coding theory. First, the following theorem shows that great (linear) codes exist, but does not give an explicit way to construct them.

Theorem 2.1 (Gilbert-Varshamov bound [Gil52, Var57])**.** *There exists an* $[n,k,d]$ *code for all* n, k, d *where*

$$2^k \leq \frac{2^n}{\binom{n}{\leq d-1}}.$$

On the other hand, this next bound shows that great codes *cannot* exist (for slightly better parameters). We state a version for list-decodable codes, and provide its simple proof, for completeness.

Theorem 2.2 (Hamming bound [Ham50])**.** *For any* (ρ, L)*-list decodable code* $Q \subseteq \{0,1\}^n$ *of size* 2^k*,*

$$2^k \leq \frac{2^n L}{\binom{n}{\leq \rho n}}$$

Proof. Consider the quantity $\sum_{q \in Q} |\mathcal{B}(q, \rho n)|$. Notice that for any fixed $v \in \{0,1\}^n$, there are at most L codewords $q \in Q$ such that $v \in \mathcal{B}(q, \rho n)$, because otherwise $|\mathcal{B}(v, \rho n) \cap Q| > L$, contradicting the list-decodability of Q. Thus

$$|Q| \cdot \binom{n}{\leq \rho n} = \sum_{q \in Q} |\mathcal{B}(q, \rho n)| \leq \sum_{v \in \{0,1\}^n} L \leq 2^n L,$$

and the inequality follows. □

This completes our extremely brief discussion on coding theory. For a detailed exposition of this beautiful field, we refer the reader to the excellent textbook of Guruswami, Rudra, and Sudan [GRS22].

Part I

Extractors for independent sources

Chapter 3

Extractors for adversarial sources

The *independent source* model is the oldest and most well-studied setting in which to study randomness extraction. Here, one assumes access to multiple independent weak random sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$ over n bits, each guaranteed to contain some min-entropy k. The origins of this model can be traced back to the 50s, and after a beautiful line of work spanning several decades, we now have near-optimal explicit extractors for independent sources.

In practice, however, it is unclear whether the independent source model is a realistic one. In real life, natural sources of randomness may sometimes produce samples with no entropy at all. It may therefore be more realistic to consider a model where only K of the independent sources $\mathbf{X}_1, \dots, \mathbf{X}_N$ are guaranteed to have any min-entropy k, with the location of these good sources being unknown. We call such sources (N, K, n, k)-adversarial sources.

In this chapter, we launch the first systematic study of randomness extraction from adversarial sources. Prior to our work, the best low-error extractors for this model required each good source to have min-entropy $k \geq n^{0.99}$, and furthermore required $K \geq \sqrt{N}$ of the sources to be good. In our work, we show how to construct low-error extractors for *exponentially less* min-entropy $k \geq \mathrm{polylog}(n)$, and furthermore our extractors require even fewer good sources $K \geq N^{0.01}$. In fact, in most settings, our extractors perform even better than this, and can handle $K \geq \mathrm{polylog}(N)$ good sources or even bad sources that *depend on* many of the good sources.

The class of adversarial sources generalizes several well-studied classes of sources, and our extractors for this new model find applications in cryptography, distributed computing, and pseudorandomness itself. In order to construct our extractors for adversarial sources, we combine advanced machinery from the theory of extractors in a novel way, via various sorts of explicit extremal hypergraphs. Our constructions establish new connections to coding theory, communication complexity, and extremal combinatorics (e.g., cap set bounds and explicit Ramsey graphs), and may therefore be of independent interest.

The results presented in this chapter are based on the following two joint works:

- [CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. In *52nd Annual Symposium on Theory of Computing (STOC 2020)*, pages 1184–1197. ACM, 2020
- [CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021

3.1 Introduction

The *independent source* model is the oldest model in all of extractor theory, with origins dating back to the very paper of von Neumann [vN51] that started the field. In this setting, one assumes access to multiple independent sources of randomness, each guaranteed to contain some min-entropy. Given such sources, the goal is to produce a uniformly random string:

Definition 3.1 (Extractor for independent sources [SV86, Vaz87b, CG88]). *A function* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *is called an* extractor for independent sources *of min-entropy* k *with error* ε *if the following holds. For any* N *independent sources* $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N \sim \{0,1\}^n$*, each with min-entropy at least* k*,*

$$|\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N) - \mathbf{U}_m| \leq \varepsilon.$$

The motivation behind this model comes from the belief that if we harvest random strings from *unrelated* (e.g., geographically isolated) sources, then these strings are likely to be *independent*. Given an extractor for independent sources, we can then convert these strings into a stream of uniformly random bits, and unlock the treasure trove of applications of randomness in computer science (Section 1.1).

Beyond the above classical motivation from extractor theory, extractors for independent sources are *powerful pseudorandom primitives* that naturally generalize seeded extractors, which already boast a stunning display of applications in hardness of approximation [Zuc96a, Uma99, MU02, Zuc07] and the construction of error-correcting codes [TZ04, Gur04], pseudorandom generators [NZ96], and expander graphs [CRVW02] (which have their own suite of applications [HLW06]). But independent source extractors have also made a name for themselves, uncovering a wide array of "bonus" applications in communication complexity [Vaz87b], combinatorics [BRSW12], distributed computing [GSV05], and even learning theory [GRT18]. As always, however, almost all of these applications require the extractor to be *explicit*.

Extractors for independent sources

After a beautiful line of work spanning four decades, we now have near-optimal explicit extractors for independent sources. *Non-explicitly*, it is not hard to show that excellent extractors exist, even for just $N = 2$ sources with extremely low min-entropy $k \geq O(\log n)$ and extremely low error $\varepsilon = 2^{-\Omega(k)}$. *Explicitly* (which is crucial for applications), the best extractors known today are as follows.

- *For* $N = 2$ *sources:* There exist explicit (constant-error) extractors for min-entropy $k \geq O(\log n)$.

  In the low-error setting, the best explicit extractors still require min-entropy just below $k \geq n/2$.

- *For* $N = 3$ *sources:* There exist explicit low-error extractors for min-entropy $k \geq \mathrm{polylog}(n)$.

These intricate constructions are the culmination of dozens of papers published over nearly 40 years. Below, we give a whirlwind tour of this story.

The early days The first explicit extractor for independent sources was given by Chor and Goldreich [CG88], who introduced this model[1] and showed that *inner product (mod* 2*)* extracts from two sources

[1] The earliest extractors for this setting can actually be traced back to von Neumann [vN51] (and the follow-ups [HS70, Eli72, Dwa72, BL73, SW84]), but these constructions only worked for *identically distributed sources* and very large N. Santha and Vazirani [SV86] showed how to construct low-error extractors for $N = \omega(\log n)$ arbitrary *semi-random sources*, a slightly less general model than min-entropy sources. Vazirani [Vaz87b] greatly improved this result, showing how to extract from just $N = 2$ semi-random sources with exponentially small error. Chor and Goldreich [CG88] generalized this result, introducing the notion of *min-entropy sources* and showing that the function from [Vaz87b] still works.

with min-entropy $k \geq 0.501n$ and error $\varepsilon = 2^{-\Omega(k)}$. No progress was made for nearly 20 years, until a breakthrough result of Bourgain [Bou05] exploited recent advances in additive combinatorics to *break the* $(n/2)$*-entropy barrier* and extract from two sources with min-entropy $k \geq 0.499n$ and error $\varepsilon = 2^{-\Omega(k)}$.[2] Raz [Raz05] constructed a different two-source extractor, which still required one source with min-entropy $k_1 \geq 0.501n$, but allowed the other source to have min-entropy $k_2 \geq O(\log n)$, while maintaining low error $\varepsilon = 2^{-\Omega(k_2)}$. That same year, the entropy requirement for two-source *dispersers* (extractors with nontrivial error $\varepsilon < 1/2$) was improved to $k \geq 0.01n$ [BKS+10]. The following year saw an even more dramatic improvement, when Barak, Rao, Shaltiel and Wigderson decreased the entropy requirement for two-source dispersers to $k \geq n^{o(1)}$ [BRSW12]. For the next 10 years, progress on the two-source setting came to a halt.

Near-optimal three-source extractors In an attempt to further decrease the min-entropy requirement, researchers turned towards using *more sources*. The first result in this direction was published by Barak, Impagliazzo, and Wigderson [BIW06],[3] who showed how to extract from a constant number of sources $N = O(1)$ with min-entropy $k \geq 0.01n$ and error $\varepsilon = 2^{-\Omega(k)}$. (A simpler construction, which obtains the same parameters, was given by Bourgain, Glibichuk, Konyagin [BGK06].) The following year, the same entropy requirement of $k \geq 0.01n$ was achieved for just $N = 3$ sources, albeit with worse error $\varepsilon = 0.01$ [BKS+10]. A major improvement was then given by Rao [Rao09a], who extracted from a constant number of sources $N = O(1)$ with entropy $k \geq n^{0.01}$. Rao's extractor has error $\varepsilon = 1/k^{\Omega(1)}$, which was improved to $\varepsilon = 2^{-k^{\Omega(1)}}$ in [BRSW12]. Finally, in an exciting line of work [Li11a, Li13b, Li13a, Li15], Li dramatically improved these parameters, constructing *near-optimal* extractors for just $N = 3$ sources: his extractors work for min-entropy just $k \geq \log^C n$ for some constant C, and have exponentially small error $\varepsilon = 2^{-k^{\Omega(1)}}$.[4]

Near-optimal two-source extractors In the year following Li's near-optimal three-source extractors, the two-source setting saw its first improvement in a decade - and it was groundbreaking. In independent works at STOC 2016, Cohen constructed a two-source *disperser* for min-entropy $k \geq \log^C n$ [Coh19], while Chattopadhyay and Zuckerman constructed a two-source *extractor* for the same min-entropy [CZ19].[5] While the error of the Chattopadhyay-Zuckerman extractor is only polynomially small $\varepsilon = n^{-\Omega(1)}$, their entropy requirement offered an exponential improvement over the previous best result of Bourgain [Bou05]. Following these breakthrough results, Li [Li16] improved the output length of the extractor, while a flurry of other work successfully brought the entropy requirement down from $k \geq \log^C n$ to $k \geq \log^{1+o(1)} n$ [CS16a, CL16a, Coh16b, Mek17, BDT19, Coh17, Li17, Li19]. Very recently, a preprint of Li [Li23] finally achieves the asymptotically optimal entropy requirement of $k \geq O(\log n)$. Notably, however, none of these constructions achieve *low-error*: here, the only improvement since Bourgain has been the low-error two-source extractor of Lewko [Lew19], which requires min-entropy about $k \geq 0.45n$.

Needless to say, extractors for independent sources have received an extraordinary amount of attention over the past 40 years, and the progress has been staggering. While a significant amount of work remains to be done for the low-error, two-source setting, we now have near-optimal explicit constructions of low-error three-source extractors and high-error two-source extractors. This begs the question,

What's next?

[2] We refer the reader to [Rao07] for an excellent exposition of this result.

[3] In fact, this was a year *before* the result of Bourgain [Bou05], and may therefore be considered the catalyst that restarted work on independent source extraction.

[4] A later work [BCDT19] gives an alternative construction that achieves these same parameters.

[5] See [Cha20] for a great overview.

Extractors for adversarial sources

While the independent source model has enjoyed a rich history in the theory of randomness extractors, it is not completely clear how *realistic* the model really is. In particular, recall that this model assumes access to several independent sources $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$, each with its own min-entropy guarantee. However, in reality, computers rely on sources (such as keystrokes, mouse movements, and processor temperatures) that can sometimes be completely predictable.

In such situations, some of the sources may contain no entropy at all.

Today's independent source extractors cannot handle this adversarial situation, and will fail to output uniformly random bits. Developing a theory of randomness extraction in the presence of *sources with missing entropy* would provide a robust generalization of the independent source model, and may eventually help us build better random number generators for computers. In this chapter, we initiate a systematic study of this question, via a model we call *adversarial sources*.

Definition 3.2 (Adversarial source)**.** *A source $\mathbf{X} \sim (\{0,1\}^n)^N$ is called an (N, K, n, k)*-adversarial source *if it is of the form $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$, where each $\mathbf{X}_i$ is an independent source over n bits, and at least K of them are* good*: i.e., there is some set $S \subseteq [N]$ of size $|S| \geq K$ such that $H_\infty(\mathbf{X}_i) \geq k$ for all $i \in S$.*

In this model, we crucially do not know which of the sources are good. We would like to construct a single function that extracts from *any* adversarial source:

Definition 3.3 (Extractor for adversarial sources)**.** *A function* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *is called an* extractor for (N, K, n, k)-adversarial sources with error ε *if the following holds. For any N independent sources $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N \sim \{0,1\}^n$, where at least K of them have min-entropy at least k,*

$$|\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N) - \mathbf{U}_m| \leq \varepsilon.$$

Comparing Definitions 3.1 and 3.3, it can be seen that extractors for adversarial sources are a simple, robust generalization of extractors for independent sources. As we will see, these new, more general extractors also unlock new applications in pseudorandomness (Section 3.6), distributed computing (Section 3.7), and cryptography (Section 3.8). Despite all of this, there has never been a comprehensive study of this setting. However, there do exist explicit constructions of extractors for adversarial sources in the wild - either implicitly, or under a different name. We outline these, below.

Prior constructions In several works [KM04, KM05, GSV05, LLTT05, CL16b], it has been observed that a (black-box) two-source extractor can be used to extract from a collection of sources where only two of them have entropy (with roughly the same parameters). This gives high-error extractors for adversarial sources with $K \geq 2$ good sources of min-entropy $k \geq O(\log n)$, and low-error extractors for adversarial sources with $K \geq 2$ good sources of min-entropy just below $k \geq n/2$. In [KRVZ11], the authors observed that the (white-box) constant-source extractor of [BGK06] can also handle sources with no entropy. This gives low-error extractors for adversarial sources with $K \geq O(1)$ good sources with $k \geq 0.01n$. However, to achieve *sublinear entropy* in the low-error setting, the only known construction [KRVZ11, Theorem 5.6] requires *many more* good sources: $K \geq O(\sqrt{N})$ good sources of min-entropy $k \geq n^{0.99}$. To the best of our knowledge, these are the only known constructions of extractors for adversarial sources.[6]

[6] Other known generalizations of two-source extractors (like interleaved-source extractors [RY11, CZ16, CL20] and sumset extractors [CL16b, CL22]) can, of course, also extract from adversarial sources. However, these settings are even more challenging than the two-source setting, and cannot be used to improve the parameters above.

3.2 Our results

In this chapter, we launch a systematic study of *extractors for adversarial sources*. As we have seen, these extractors are more robust versions of classical extractors for independent sources. We focus here on *low-error* extractors, motivated by applications in cryptography.[7] Our results are as follows.

The main extractor

In our main result, we construct significantly improved low-error extractors for adversarial sources. Previously, the only known low-error extractor for adversarial sources that could handle sublinear entropy required $K \geq O(\sqrt{N})$ good sources of min-entropy $k \geq n^{0.99}$ [KRVZ11]. We exponentially improve the min-entropy requirement to $k \geq \mathrm{polylog}(n)$, and do so using fewer good sources $K \geq N^{0.01}$.

Theorem 3.1 (The main extractor)**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \geq CN^\delta$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

Our construction relies on a novel way of combining new tools from extractor theory and extremal combinatorics, which are built using fresh connections to communication complexity and coding theory. As an application of our extractor, we also obtain improved extractors for more traditional settings (Section 3.6). Next, we show that in most regimes, we can actually improve the parameters of Theorem 3.1 even further.

Optimized extractors for a few long sources

In every adversarial source, there are N sources of length n. Depending on the relationship between these parameters, it may be easier or harder to extract. In our main result (Theorem 3.1), we show how to extract no matter how these parameters are related. Next, we show that in the setting of $N \ll n$ (i.e., when there are a few long sources), we can obtain a significantly better result. That is, we obtain an exponential improvement on the number of good sources required, from $K \geq N^{0.01}$ to $K \geq \mathrm{polylog}(N)$.

Theorem 3.2 (A better extractor, in the few-long-sources setting)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \geq \log^C N$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$, provided that we also have $k \geq N^C$.*

As it turns out, we will show later on that we can get a similar exponential improvement in the complementary *many-short-sources* setting.[8] However, we note that the *few-long-sources* setting tackled by Theorem 3.2 is significantly more challenging (and more relevant to applications). To construct these extractors, we combine tools from extractor theory and extremal combinatorics in a similar way as in our proof of Theorem 3.1, except the tools are older and the analysis is more involved. In fact, by extending this analysis, we actually show that we can handle adversarial sources where the bad sources are not only devoid of min-entropy, but actually *correlated* with the good sources! We call such sources *adversarial sources of locality d*, where d indicates the number of good sources that each bad source may depend on.[9] We prove the following.

[7] It is known that basic cryptographic tasks (such as bit commitment, encryption, and secret sharing) *require* access to high-quality randomness [DOPS04]. Low-error extractors are crucial for such applications.

[8] There is a slight gap between these two settings, which is why these two results don't supersede Theorem 3.1.

[9] Adversarial sources studied thus far correspond to *adversarial sources of locality* 0. See Definition 3.9.

Theorem 3.3 (An extractor for high locality, in the few-long-sources setting)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for (N, K, n, k)-adversarial sources of locality $d \leq K^\gamma$ with $K \geq N^{1-\gamma}$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$, provided that we also have $k \geq N^C$.*

In particular, at the cost of increasing the number of good sources back up to $K \geq N^{0.99}$, we can handle bad sources that depend on *polynomially many* good sources $d \leq K^{0.01}$, while still allowing the good sources to have min-entropy just $k \geq \operatorname{polylog}(n)$. To the best of our knowledge, this is the first explicit construction of an extractor for adversarial sources that can handle *any* positive locality.[10]

Optimized extractors for many short sources

As promised, we can also get similar significant improvements to Theorem 3.1 in the *many-short-sources* setting $N \gg n$ as in the *few-long-sources* setting (discussed above). In particular, we again obtain an exponential improvement on the number of good sources required, from $K \geq N^{0.01}$ to $K \geq \operatorname{polylog}(N)$.

Theorem 3.4 (A better extractor, in the many-short-sources setting)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for (N, K, n, k)-adversarial sources with $K \geq \log^C N$ good sources of min-entropy $k \geq 1$, which has output length $m \geq K^\gamma$ and error $\varepsilon = 2^{-K^\gamma}$, provided that we also have $K \geq n^C$.*

Comparing the improvements obtained above with the improvements gained in the few-long-sources setting (Theorem 3.2), notice that we now only require each source to have *one bit of min-entropy*! The proof of Theorem 3.4 is also much easier than that of Theorem 3.2, and involves one simple new trick: we show that *any source with one bit of min-entropy is a convex combination of affine sources*. This allows us to prove that existing affine extractors already extract from the many-short-sources setting with the parameters claimed above. In fact, we slightly extend the analysis, and show that these extractors even work for adversarial sources of high locality. We prove the following.

Theorem 3.5 (An extractor for high locality, in the many-short-sources setting)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for (N, K, n, k)-adversarial sources of locality $d \leq K^\gamma$ with $K \geq N^{1-\gamma}$ good sources of min-entropy $k \geq 1$, which has output length $m \geq K^\gamma$ and error $\varepsilon = 2^{-K^\gamma}$, provided that we also have $K \geq n^C$.*

Comparing the improvements obtained above with the few-long-sources setting (Theorem 3.3), note that we again can handle polynomial locality in the many-short-sources, with the additional bonus of just requiring one bit of min-entropy in each source. This completes the overview of our results.

Organization

We start with our main extractor (Theorem 3.1) in Section 3.3, gradually building up intuition in a step-by-step fashion before ultimately providing its construction. Next, we construct our optimized extractors for the few-long-sources setting (Theorems 3.2 and 3.3) in Section 3.4, and our optimized extractors for the

[10] In an independent work [AOR+20], Aggarwal, Obremski, Ribeiro, Siniscalchi, and Visconti study a model similar to ours. However, in their model, called SHELA sources, each bad source can depend on *every* good source preceding it. They show that extraction is impossible here, and focus instead on constructing weaker objects known as *somewhere extractors*. In a concurrent work [BGM22], Ball, Goldreich, and Malkin study a somewhat related model, called *sources of bounded coordination*. However, they prove that their model behaves similarly to the independent source model (with entropy guarantees), which is not true for ours.

many-short-sources setting (Theorems 3.4 and 3.5) in Section 3.5. After, we discuss some bonus applications of our new extractors in pseudorandomness, distributed computing, and cryptography (in Section 3.6, Section 3.7, and Section 3.8, respectively). Finally, we conclude with some open problems in Section 3.9.

3.3 The main extractor

We are now ready to proceed with the technical portion of the chapter. In this section, we focus on constructing our main extractor, Theorem 3.1. Instead of diving straight in, let's begin with what is known.

3.3.1 A known construction: two-source extractors everywhere

Given standard equipment from extractor theory, it is not too difficult to extract from adversarial sources with just 2 good sources, provided that they have enough min-entropy. Indeed, the following result has been independently observed several times a couple times over the past few years.

Theorem 3.6 (The known extractor for adversarial sources)**.** *For any fixed $\delta > 0$, there exists a constant $\gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \geq 2$ good sources of min-entropy $k \geq (1/2 + \delta) \cdot n$, which has output length $m \geq \gamma k$ and error $\varepsilon = 2^{-\gamma k}$.*

In order to prove this result, one natural approach is to take an off-the-shelf two-source extractor, and simply call it on every pair of sources in the adversarial source $\mathbf{X}$. Indeed, upon doing so, as long as there are just 2 good sources in $\mathbf{X}$, we are guaranteed that one of these calls will output uniform bits. Furthermore, by the min-entropy chain rule (Lemma 2.1), we can roughly assume that these outputs are independent, allowing us to XOR them together to propagate the output of the good call to the overall output of the extractor. More formally, one can prove the following.

Lemma 3.1 (Extractor for adversarial sources via two-source extractors over all pairs)**.** *Let $2\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ be a two-source extractor for min-entropy k with error ε. Then the function $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ defined as*

$$\mathsf{Ext}(x_1, \dots, x_N) := \bigoplus_{1 \leq i < j \leq N} 2\mathsf{Ext}(x_i, x_j)$$

is an extractor for $(N, 2, n, k + m + \log(1/\varepsilon))$-adversarial sources with error 3ε.

Proof. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an $(N, 2, n, k')$-adversarial source, for some k' we will set later. Since the extractor Ext treats every pair of sources the same, we may assume (without loss of generality) that $\mathbf{X}_1, \mathbf{X}_2$ are the good sources in $\mathbf{X}$. In other words, we may assume that $\mathbf{X}_1, \mathbf{X}_2$ each have min-entropy at least k'. We now use the data-processing inequality (Fact 2.3) to fix all other sources. More formally, we have

$$\begin{aligned} |\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| &\leq |\mathsf{Ext}(\mathbf{X}) \circ \mathbf{X}_3 \circ \cdots \circ \mathbf{X}_N - \mathbf{U}_m \circ \mathbf{X}_3 \circ \cdots \circ \mathbf{X}_N| \\ &= \mathbb{E}_{x_3 \sim \mathbf{X}_3, \dots, x_N \sim \mathbf{X}_N}[|\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, x_3, \dots, x_N) - \mathbf{U}_m|] \\ &\leq |\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, x_3^*, \dots, x_N^*) - \mathbf{U}_m| \end{aligned}$$

for some worst-case $x_3^*, \dots, x_N^* \in \{0,1\}^n$. Consider now the random variables

$$\begin{aligned}\mathbf{Y}_1 &:= \bigoplus_{3 \le j \le N} \mathsf{2Ext}(\mathbf{X}_1, x_j^*)\\ \mathbf{Y}_2 &:= \bigoplus_{3 \le j \le N} \mathsf{2Ext}(\mathbf{X}_2, x_j^*),\\ z &:= \bigoplus_{3 \le i < j \le N} \mathsf{2Ext}(x_i^*, x_j^*),\end{aligned}$$

Now, notice that we may write

$$\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, x_3^*, \dots, x_N^*) = \mathsf{2Ext}(\mathbf{X}_1, \mathbf{X}_2) \oplus \mathbf{Y}_1 \oplus \mathbf{Y}_2 \oplus z,$$

where $\mathbf{Y}_1 \sim \{0,1\}^m$ is a deterministic function of $\mathbf{X}_1$, and $\mathbf{Y}_2 \sim \{0,1\}^m$ is a deterministic function of $\mathbf{X}_2$, and z is fixed. Thus if we condition on $\mathbf{Y}_1, \mathbf{Y}_2$ being equal to any values y_1, y_2, it holds that $\mathbf{X}_1, \mathbf{X}_2$ remain independent. By the min-entropy chain rule (Lemma 2.1), we know that with probability $\ge 1 - \varepsilon$ over fixing $\mathbf{Y}_1$ to some y_1, it holds that $(\mathbf{X}_1 \mid \mathbf{Y}_1 = y)$ has min-entropy at least $k' - m - \log(1/\varepsilon)$. The same is true for $\mathbf{X}_2$ and $\mathbf{Y}_2$. Thus, there exist some y_1^*, y_2^* such that $\mathbf{X}_1^* := (\mathbf{X}_1 \mid \mathbf{Y}_1 = y_1^*)$ and $\mathbf{X}_2^* := (\mathbf{X}_2 \mid \mathbf{Y}_2 = y_2^*)$ are independent and each have min-entropy at least $k' - m - \log(1/\varepsilon)$, and such that we can continue our inequality (from above) as follows:

$$\begin{aligned}|\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, x_3^*, \dots, x_N^*) - \mathbf{U}_m| &= |\mathsf{2Ext}(\mathbf{X}_1, \mathbf{X}_2) \oplus \mathbf{Y}_1 \oplus \mathbf{Y}_2 \oplus z - \mathbf{U}_m|\\ &\le |\mathsf{2Ext}(\mathbf{X}_1^*, \mathbf{X}_2^*) \oplus y_1^* \oplus y_2^* \oplus z - \mathbf{U}_m| + 2\varepsilon\\ &= |\mathsf{2Ext}(\mathbf{X}_1^*, \mathbf{X}_2^*) - \mathbf{U}_m| + 2\varepsilon,\end{aligned}$$

where the last equality follows from the fact that shifting a random variable does not change its statistical distance from uniform. Finally, as long as $k' - m - \log(1/\varepsilon) \ge k$, can bound the above by

$$\le \varepsilon + 2\varepsilon = 3\varepsilon.$$

Thus, the adversarial source extractor Ext has error 3ε, as long as $k' \ge k + m + \log(1/\varepsilon)$, as desired. □

Now, to instantiate the above framework, we can just take a standard low-error two-source extractor, and plug it in. Our two-source extractor of choice will be the following.

Lemma 3.2 (Two-source extractor [Vaz87b, CG88])**.** *For every fixed $\delta > 0$, there exists a constant $\gamma > 0$ such that the following holds. There exists an explicit two-source extractor $\mathsf{2Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ for min-entropy $k \ge (1/2 + \delta) \cdot n$, which has output length $m \ge \gamma k$ and error $\varepsilon = 2^{-\gamma k}$.*

Now, let's plug it into the above framework.

Proof of Theorem 3.6. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an $(N, K \ge 2, n, k)$-adversarial source. By plugging in the explicit two-source extractor from Lemma 3.2 into our compiler (Lemma 3.1), we know that for every fixed $\delta' > 0$ there is a constant $\delta' > \gamma' > 0$ such that the following holds.[11] We can explicitly extract $m \ge \gamma' \cdot (1/2 + \delta') \cdot n$ bits from $\mathbf{X}$ with error $3\varepsilon = 3 \cdot 2^{-\gamma' \cdot (1/2 + \delta') \cdot n}$, as long as $k \ge (1/2 + \delta') \cdot n + m + \log(1/\varepsilon)$, or rather $k \ge (1/2 + \delta') \cdot (1 + 2\delta') \cdot n$. The result follows by picking δ' based on δ, and γ based on γ'. □

[11] Note that we can enforce an arbitrary upper bound on γ' because this can only weaken the claim made in Lemma 3.2.

The above construction of adversarial source extractors was simple, quick, and worked for just $K = 2$ good sources. However, it left something to be desired. In particular, the min-entropy requirement of the good sources is huge, and we would like to try to decrease it further. In order to do so, we will need at least one more source of randomness, unless we want to immediately solve one of the biggest open questions in extractor theory (improved low-error extractors for two independent sources). Now, the question is, is this enough to get better extractors?

One might hope that the answer is yes. Indeed, while there are no known low-error two-source extractors that significantly outperform the one above, there *are*, in fact, three-source extractors of this type. In particular, in a seminal paper of Li [Li15], low-error three-source extractors were constructed for just *polylogarithmic* min-entropy. In order to exploit these objects to obtain better adversarial source extractors, one might try to just run the above argument once again. In particular, perhaps we can just call Li's three-source extractor on every *triple*, XOR the results, and hope for the best. Now, if there are at least 3 good sources with polylogarithmic min-entropy, we are certainly guaranteed that *some* three-source extractor call will be *activated*, i.e., output uniform bits. However, upon some reflection, there is no way to guarantee that the XOR call won't completely destroy these uniform bits. Indeed, since some extractor calls may now share up to *two* sources with the activated call, we have no way to fix these other calls without compromising the independence of the sources feeding the activated call (and thereby compromising its uniform output).

3.3.2 A warm-up construction: three-source extractors and cap set bounds

In this section, we will provide a solution to the above conundrum, and prove the following result.

Theorem 3.7 (The warm-up extractor for adversarial sources)**.** *There exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \geq CN^{0.9228}$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

In order to prove this result, notice that we ran into a problem above due to the fact that there exist extractor calls that share two of their input sources. Thus, there is a clear way to prevent this from happening: simply make sure that your adversarial source extractor never makes such overlapping calls! In particular, instead of calling Li's three-source extractor over *every* triple of sources, it makes more sense to call it over a collection of triples such that each pair of triples share at most one source. As it turns out, this is a well-studied structure in combinatorics.

Definition 3.4 (Partial Steiner triple system (STS))**.** *A* partial Steiner triple system *is a 3-uniform hypergraph $G = (V, \mathcal{E})$ where any two edges share at most one vertex.*

Now, given this structure, we have all-but-defined our new candidate extractor for adversarial sources: namely, just overlay an STS over the sources making up the adversarial source, call a three-source extractor call on each hyperedge of the STS, and XOR the results. Thinking about this a little, it is not too hard to show that the output of an activated three-source extractor call will look roughly independent of the rest. However, another problem arises: we are no longer guaranteed to have an activated call at all! Indeed, the *empty* hypergraph is an STS, yet clearly the corresponding extractor cannot even extract when there are $K \geq N$ good sources. Indeed, what we want is to make sure that for the smallest possible number of good sources K, we are always guaranteed to *cover* a hyperedge (no matter where the good sources are placed). In other words, we want to make sure that the STS has a small *independence number*, defined to be the size of the largest set of vertices that *fail* to cover any hyperedge. With all of this in mind, we are just about ready to define our new framework for extracting. But one quick last note: since hyperedges are *not* inherently ordered, while the inputs to a function *are*, we need some way to order the hyperedges. Towards this end, we will simply define partial Steiner triple systems over a vertex sets identified with $[N]$, and always assume that the edges in $\mathcal{E}$ are ordered in increasing order of vertex label. At last, let us construct and prove the correctness of our new extractor.

Lemma 3.3 (Extractor for adversarial sources via three-source extractors over a Steiner triple system)**.** *Let* $\mathsf{3Ext} : (\{0,1\}^n)^3 \to \{0,1\}^m$ *be a three-source extractor for min-entropy* k *with error* ε*, and let* $G = ([N], \mathcal{E})$ *be a partial Steiner triple system with independence number* α*. Then the function* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *defined as*

$$\mathsf{Ext}(x_1, \dots, x_N) := \bigoplus_{(i_1,i_2,i_3)\in\mathcal{E}} \mathsf{3Ext}(x_{i_1}, x_{i_2}, x_{i_3})$$

is an extractor for $(N, \alpha+1, n, k+m+\log(1/\varepsilon))$*-adversarial sources with error* 4ε*.*

Proof. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an $(N, K = \alpha+1, n, k')$-adversarial source, for some k' we will set later, and observe that since $K > \alpha$, there must be some edge $F := (v_1, v_2, v_3) \in \mathcal{E}$ that is completely covered by good sources. Now, for each $(h, i, j) \in \mathcal{E}$, define the random variable $\mathbf{Y}_{(h,i,j)} := \mathsf{3Ext}(\mathbf{X}_h, \mathbf{X}_i, \mathbf{X}_j)$, and note that $\mathsf{Ext}(\mathbf{X}) = \bigoplus_{(h,i,j)\in\mathcal{E}} \mathbf{Y}_{(h,i,j)}$. Observe that since G is an STS, we can partition $\mathcal{E} \setminus (v_1, v_2, v_3)$ into $\mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4$ such that the triples in $\mathcal{E}_4$ have an empty intersection with F, while the triples in $\mathcal{E}_\ell$, for $\ell \in [3]$, intersect with F at exactly v_ℓ. Thus, if we define $\mathbf{Z}_\ell := \bigoplus_{(h,i,j)\in\mathcal{E}_\ell} \mathbf{Y}_{(h,i,j)}$, for each $\ell \in [4]$, then:

$$\mathsf{Ext}(\mathbf{X}) = \mathbf{Y}_{(v_1,v_2,v_3)} \oplus \mathbf{Z}_1 \oplus \mathbf{Z}_2 \oplus \mathbf{Z}_3 \oplus \mathbf{Z}_4.$$

Now, fix all random variables $\mathbf{X}_i$ such that $i \notin \{v_1, v_2, v_3\}$. Notice that this fixes $\mathbf{Z}_4$, whereas each $\mathbf{Z}_\ell$, for $\ell \in [3]$, becomes a deterministic function of $\mathbf{X}_\ell$. Next, fix $\mathbf{Z}_1$. This only affects $\mathbf{X}_{v_1}$, and by the min-entropy chain rule (Lemma 2.1), it will still have at least $k' - m - \log(1/\varepsilon)$ bits of min-entropy, except with probability at most ε. Next, do the same for $\mathbf{Z}_2$ and $\mathbf{Z}_3$. By a union bound, each of $\mathbf{X}_{v_1}, \mathbf{X}_{v_2}, \mathbf{X}_{v_3}$ will still have min-entropy $k' - m - \log(1/\varepsilon)$, except with probability at most 3ε. Furthermore, these three random variables are still independent, because we were just fixing deterministic functions that act on just one of them each. After all these fixings, we now have

$$\mathsf{Ext}(\mathbf{X}) = \mathbf{Y}_{(v_1,v_2,v_3)} \oplus b = \mathsf{3Ext}(\mathbf{X}_{v_1}, \mathbf{X}_{v_2}, \mathbf{X}_{v_3}) \oplus b$$

for some constant $b \in \{0,1\}^n$. Thus, as long as $k' - m - \log(1/\varepsilon) \geq k$, this output will be ε-close to uniform. Adding in the error from our fixings, we see that the original output of $\mathsf{Ext}(\mathbf{X})$ is $(\varepsilon + 3\varepsilon)$-close to uniform, provided that $k' \geq k + m + \log(1/\varepsilon)$. This completes the proof. □

Now, let us prepare to instantiate the above framework. First, we'll need a formal statement of Li's three-source extractor.

Lemma 3.4 (Three-source extractor [Li15])**.** *There exist constants* $C, \gamma > 0$ *such that the following holds. There exists an explicit three-source extractor* $\mathsf{3Ext} : (\{0,1\}^n)^3 \to \{0,1\}^m$ *for min-entropy* $k \geq \log^C n$*, which has output length* $m \geq \gamma k$ *and* $\varepsilon = 2^{-k^\gamma}$*.*

Next, we'll need an explicit construction of an STS with no big independence set. But wait, where can we even find an object? In the following lemma, we show that they can be easily constructed by exploiting the recent breakthrough on *cap set bounds*.

Lemma 3.5 (Partial Steiner triple system)**.** *There exists a constant $C > 0$ such that there exists an explicit partial Steiner triple system $G = ([N], \mathcal{E})$ with independence number $\alpha(G) \le CN^{0.9228}$.*

Proof. For now, assume that N is a power of 3. Define a hypergraph $G = ([N], \mathcal{E})$ that identifies $[N]$ with the points in $\mathbb{F}_3^{\log_3 N}$ and $\mathcal{E}$ with the lines in $\mathbb{F}_3^{\log_3 N}$. G is clearly a partial Steiner triple system, because no two lines intersect at more than one point. Furthermore, notice that an independent set S in G is exactly a set of points in $\mathbb{F}_3^{\log N}$ such that no three are in a line. Such a set is called a *cap set*, and it is known that a cap set in $\mathbb{F}_3^{\log N}$ has size at most $2.756^{\log_3 N} \le N^{0.9228}$ [CLP17, EG17]. It is straightforward to extend this construction to when N is not a power of three, by writing N in base-3, doing the above construction for every component of its base-3 representation, and adding the max cap set sizes from each component. □

Now, given these two ingredients and our new framework for extracting, Theorem 3.7 is immediate.

Proof of Theorem 3.7. Simply instantiate Lemma 3.3 with Lemmas 3.4 and 3.5. □

Now that we have successfully constructed low-error adversarial source extractors for polylogarithmic min-entropy, where should we look next? Well, one issue with the above construction is that it requires *a lot* of the sources to be good, namely, $K \ge N^{0.9228}$. This seems like quite a steep price to pay in order to reduce the min-entropy of each good source. Is there any way to reduce the required number of good sources, while still extracting from $k = \mathrm{polylog}(n)$ min-entropy? One natural approach is to simply try to construct better STS's (i.e., with smaller independence number). However, it is well-known that *every* STS over N vertices has independence number at least $\Omega(\sqrt{N \log N})$ [RŠ94]. Thus, if we want to extract from $K \ll \sqrt{N}$ good sources, we need a new idea. One possibility is to think about set systems with *bigger* hyperedges, that still have pairwise edge intersections of at most 1. However, this can only make matters worse, as the independent set size can only *increase* [RŠ94]. Indeed, we somehow need to allow for *bigger* intersections if we hope to extract from fewer good sources. But wasn't this the idea that got us into trouble in the first place?

3.3.3 The general construction: leakage-resilient extractors and extremal designs

We now show how to resolve the issues above, and extract from just $K \ge N^{0.01}$ good sources.

Theorem 3.8 (The main extractor for adversarial sources - Theorem 3.1, restated)**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \ge CN^\delta$ good sources of min-entropy $k \ge \log^C n$, which has output length $m \ge k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

In order to construct good extractors for adversarial sources, the key idea is to strike a perfect balance between the *structure* offered by the underlying hypergraph, and the *robustness* offered by the extractor being called as a subroutine. Indeed, as we have seen above, if the hypergraph has big intersections, then we need our three-source extractor to be extremely robust (in the sense that its output should look uniform even conditioned on other three-source extractors acting on up to two of the same inputs - and indeed, we don't know of such objects). On the other hand, if the hypergraph has small intersections, then we need many good sources in order to active a hyperedge. So, what should we do?

The key idea is to use an object constructed in a different part of this thesis (Chapter 4), which was originally motivated by applications in cryptography. I'm talking about a *leakage-resilient extractor* (LRE), which is a robust version of a classical independent source extractor, in the sense that its output is guaranteed to look uniform and *stay* uniform, even conditioned on the output of several other *leakage* functions acting on the same set of sources. Indeed, their definition, presented below, is eerily similar to the exact property we requested above.[12]

Definition 3.5 (Leakage-resilient extractor)**.** *A function* $\mathsf{Ext} : (\{0,1\}^n)^r \to \{0,1\}^m$ *is called an* (r,s)-leakage-resilient extractor for independent sources *of min-entropy* k *with error* ε *if the following holds. For any* r *independent sources* $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_r \sim \{0,1\}^n$*, each with min-entropy at least* k*, and for any collection of functions* $\{\mathsf{Leak}_A : (\{0,1\}^n)^{s-1} \to \{0,1\}^m\}_{A\in\mathcal{A}}$ *where* $\mathcal{A} := \binom{[r]}{s-1}$*, it holds that*

$$|\mathsf{LRE}(\mathbf{X}) \circ (\mathsf{Leak}_A(\mathbf{X}_A))_{A\in\mathcal{A}} - \mathbf{U}_m \circ (\mathsf{Leak}_A(\mathbf{X}_A))_{A\in\mathcal{A}}| \leq \varepsilon.$$

Informally, an (r,s)-LRE is just an extractor for r independent sources, whose output remains uniform even conditioned on leaks that act on all possible $(s-1)$-tuples of the inputs. Whether we realize it or not, we have actually already referred to LREs implicitly many times throughout this section. In particular, in our constructions that *worked* we were using the fact that every three-source extractor is automatically an $(3,2)$-LRE, via the min-entropy chain rule. And in our constructions that failed, they failed because it is not guaranteed that every three-source extractor is a $(3,3)$-LRE. Furthermore, it is not clear at all how hard such objects are to construct.

In Chapter 4, we do not give full-blown $(3,3)$-LREs. However we *do* construct *low-error* LREs that vary greatly in terms of how many sources they use, and the intersections they can handle. And furthermore, they can even extract from polylogarithmic min-entropy. Thus, the key idea behind our main extractor is to figure out how to exploit these, by constructing combinatorial objects that are more general than STS's. As it turns out, the exact objects that we want are well-known, and referred to as *combinatorial designs*.

Definition 3.6 (Design)**.** *An* (N,r,s)-design *is an* r*-uniform hypergraph* $G=(V,\mathcal{E})$ *over* N *vertices where any two edges intersect at less than* s *vertices.*

Since our LREs will work for polylogarithmic min-entropy, the hope is that these more general designs will offer a way to achieve a smaller independence number, and thus extract from a smaller number K of good sources that still have just polylogarithmic min-entropy. As suggested by the matching parameters, the idea will be to call an (r,s)-LRE over the hyperedges of an (N,r,s)-design. And almost by definition, we immediately get the following framework.

Lemma 3.6 (Extractor for adversarial sources via leakage-resilient extractors over a combinatorial design)**.** *Let* $\mathsf{LRE} : (\{0,1\}^n)^r \to \{0,1\}^m$ *be an* (r,s)*-leakage-resilient extractor for min-entropy* k *with error* ε*, and let* $G=([N],\mathcal{E})$ *be an* (N,r,s)*-design with independence number* α*. Then the function* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *defined as*

$$\mathsf{Ext}(x_1,\dots,x_N) := \bigoplus_{(i_1,\dots,i_r)\in\mathcal{E}} \mathsf{LRE}(x_{i_1},\dots,x_{i_r})$$

is an extractor for $(N,\alpha+1,n,k)$*-adversarial sources with error* ε*.*

Proof. Let $\mathbf{X}$ be an $(N,K=\alpha+1,n,k)$-adversarial source, and observe that since $K>\alpha$, there must be some edge $F:=(v_1,\dots,v_r)\in\mathcal{E}$ that is completely covered by good sources. Now, since G is an (N,r,s)-design, it follows that F (which has size r) shares at most $s-1$ vertices with any other edge. Thus, we may partition $\mathcal{E}\setminus\{F\}$ into $\binom{r}{s-1}$ sets, depending on the intersection behavior of each edge with F. In particular, for each $S\in\binom{F}{s-1}$, we define

$$\mathcal{W}_S := \{e\in\mathcal{E} : e\cap F\subseteq S\}.$$

[12] The definition presented here is actually an LRE against so-called *non-adaptive bounded collusion protocols*. We construct even stronger LREs in Chapter 4.

If any $e \in \mathcal{E}$ ends up in more than one $\mathcal{W}_S$, we simply remove it from all but one of these sets, to form a true partition. Now, for each $S \in \binom{F}{s-1}$, we define the function $\mathsf{Leak}_S : (\{0,1\}^n)^N \to \{0,1\}^m$ as

$$\mathsf{Leak}_S(x_1, \dots, x_N) := \bigoplus_{(i_1,\dots,i_r)\in\mathcal{W}_S} \mathsf{LRE}(x_{i_1}, \dots, x_{i_r}),$$

so that we may write

$$\mathsf{Ext}(\mathbf{X}) = \mathsf{LRE}(\mathbf{X}_{v_1}, \dots, \mathbf{X}_{v_r}) \oplus \bigoplus_{S\in\binom{F}{s-1}} \mathsf{Leak}_S(\mathbf{X}_1, \dots, \mathbf{X}_N).$$

By combining this expression with (an appropriate function plugged into) the data-processing inequality (Fact 2.3), we know that

$$\begin{aligned}|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq |&\mathsf{LRE}(\mathbf{X}_{v_1}, \dots, \mathbf{X}_{v_r}) \circ (\mathsf{Leak}_S(\mathbf{X}_1, \dots, \mathbf{X}_N))_{S\in\binom{F}{s-1}} \\ &-\mathbf{U}_m \circ (\mathsf{Leak}_S(\mathbf{X}_1, \dots, \mathbf{X}_N))_{S\in\binom{F}{s-1}}|.\end{aligned}$$

But notice now that if fix all $\mathbf{X}_i$ such that $i \notin \{v_1, \dots, v_r\}$, then we have not affected $\mathbf{X}_{v_1}, \dots, \mathbf{X}_{v_r}$ at all (due to all the sources being independent), while the leaks restricted under these fixings obtain the same structure as the leaks in Definition 3.5. Thus, since the good sources started out with min-entropy k (and still have min-entropy k after the fixings), the leakage-resilience of LRE immediately implies that the above is upper bounded by ε. This completes the proof. □

Now, before we instantiate this framework, we must figure out how to achieve the smallest possible independence number. Well, returning to the paper that originally yielded a $\sqrt{N}$ barrier for STSs [RŠ94], the authors show that for constant r and s, there exist (N, r, s)-designs G with independence number $\alpha(G) \leq O(N^{\frac{r-s}{r-1}}(\log n)^{\frac{1}{r-1}})$. In other words, this decreases as s gets closer to r, and thus these are the parameters we should shoot for. Towards this end, we show in Chapter 4 how to construct explicit (r, s)-LREs for almost the most extreme setting of these parameters, in particular, when $s = r - 1$. We prove the following.

Lemma 3.7 (Leakage-resilient extractor - Theorem 4.2, specialization)**.** *For every constant $r \geq 3$ there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit $(r, r-1)$-leakage-resilient extractor $\mathsf{LRE} : (\{0,1\}^n)^r \to \{0,1\}^m$ for min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

We are almost ready to instantiate our main framework, but one clear problem remains - where can we find our explicit designs? Indeed, even though it was shown in [RŠ94] that $(N, r, r-1)$-designs exist with very small independence number exist, they did not provide a way to efficiently (and deterministically) construct them. However, in Chapter 5, we provide an efficient algorithm to do exactly that, explicitly constructing designs that match the probabilistic constructions up to a square. In particular, setting $s = r-1$ in our construction yields the following result.

Lemma 3.8 (Design - Theorem 5.1, specialization)**.** *For every even constant $r \geq 3$ there exists a constant $C > 0$ such that the following holds. There exists an explicit $(N, r, r-1)$-design $G = ([N], \mathcal{E})$ with independence number $\alpha(G) \leq Cn^{2/r}$.*

Now, with these two results in hand, the proof of our main extractor follows immediately.

Proof of Theorem 3.8. Instantiate Lemma 3.6 with Lemmas 3.7 and 3.8. □

3.4 Optimized extractors for a few long sources

Now that we have completed the construction of our main extractor for adversarial sources, we turn to show that for the setting of *a few long sources*, we can achieve significantly improved parameters.

A construction of three-source correlation breakers

The key idea is to switch back to the setting of *three-source extractors*, and hope that we can somehow make them robust enough to deal with pairwise intersections of size 2. In an ideal world, we would like a three-source extractor that is, in some sense, *self-resilient*: namely, we want its output to look uniform on input sources $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$, even if it is called several more times on any (strict) subsets of the same input (combined with other random variables or fixed strings). While such an object remains out of reach, we try to construct the next best thing. In particular, while we won't be able to ensure resilience against *all* possible subsets of size 2, we will be able to do so against most. To make things more formal, we will use the notion of *correlation breakers*.

In more detail, we must start by defining *valid orderings* and *valid tampering functions*. Towards this end, we say that an ordering $\sigma : [3] \to [3]$ is *valid* if the sequence $(\sigma(1), \sigma(2), \sigma(3))$ does not contain *both* 1 or 2; or if it does, they must both appear in the first two positions. In symbols, we require that

$$\{1,2\} \not\subseteq \{\sigma(1), \sigma(2), \sigma(3)\} \text{ or } (\sigma(1), \sigma(2), \sigma(3)) \in \{(1,2,3), (2,1,3)\}.[13]$$

We note that the word *ordering* here is a bit of a misnomer, as the image of σ is allowed to have repeated elements. Next, we say that $f : (\{0,1\}^n)^3 \to (\{0,1\}^n)^3$ is a *valid tampering function* if there exist some $f_1, f_2, f_3 : \{0,1\}^n \times \{0,1\}^n$ and valid ordering $\sigma : [3] \to [3]$ such that

$$f(x_1, x_2, x_3) = (f_1(x_{\sigma(1)}), f_2(x_{\sigma(2)}), f_3(x_{\sigma(3)}))$$

Given these definitions, we are now ready to define our correlation breaker.

Definition 3.7 (Three-source correlation breaker). *A function* $\mathsf{3CB} : (\{0,1\}^n)^3 \times \{0,1\}^a \to \{0,1\}^m$ *is called a* three-source correlation breaker *for min-entropy k, advice length a, tampering degree t, and error ε if the following holds. For any three independent sources $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \sim \{0,1\}^n$, each with min-entropy at least k, any t valid tampering functions $f_1, \dots, f_t : (\{0,1\}^n)^3 \to (\{0,1\}^n)^3$, and any pieces of "advice" $\alpha_0, \alpha_1, \dots, \alpha_t \in \{0,1\}^a$ with α_0 distinct from the rest,*

$$\begin{aligned}|\mathsf{3CB}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \alpha_0) \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_t) \\ - \mathbf{U}_m \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_t)| \leq \varepsilon.\end{aligned}$$

Next, we spend just a moment constructing these objects, as they appear in the remaining constructions.

Lemma 3.9 (Three-source correlation breaker). *There exist constants $C, \gamma > 0$ such that the following holds. There exist an explicit three-source correlation breaker* $\mathsf{3CB} : (\{0,1\}^n)^3 \times \{0,1\}^a \to \{0,1\}^m$ *for min-entropy $k \geq \log^C n$, tampering degree $t \geq k^\gamma$, and advice length $a = \log t$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

In order to construct our correlation breaker, we combine two classic pseudorandom objects: a two-source non-malleable extractor, and a two-source condenser. We start with the latter, which will allow our correlation breaker to work for very low min-entropy.

[13] We encourage the reader to not get caught up in this definition: it is just reverse engineered so our fixings ultimately work out.

Lemma 3.10 (Two-source condenser [BCDT19]). *For any constant $\beta_1 > 0$ there exist constants $C_1, \gamma_1 > 0$ such that the following holds. There exists an explicit function $\mathsf{2Cond} : \{0,1\}^{n_1} \times \{0,1\}^{n_1} \to \{0,1\}^{m_1}$ such that for any two independent sources $\mathbf{X}_1, \mathbf{X}_2 \sim \{0,1\}^{n_1}$, each with min-entropy at least $k_1 \geq \log^{C_1} n_1$,*

$$\Pr_{x_2 \sim \mathbf{X}_2}[H_\infty(\mathsf{2Cond}(\mathbf{X}_1, x_2)) \geq m_1 - m_1^{\beta_1}] \geq 1 - \varepsilon_1,$$

where $m_1 \geq k_1^{\gamma}$ and $\varepsilon_1 = 2^{-k_1^{\gamma/2}}$.

Next, we turn towards the two-source non-malleable extractor that will be baked into our correlation breaker. It will be responsible for equipping our correlation breaker with its robustness. This non-malleable extractor acts as a two-source extractor that is leakage-resilient against itself.

Lemma 3.11 (Two-source non-malleable extractor [CGL20]). *There exists a constant $\gamma_2 > 0$ such that the following holds. There exists an explicit function $\mathsf{2nmExt} : \{0,1\}^{n_2} \times \{0,1\}^{n_2} \to \{0,1\}^{m_2}$ such that for any two sources $\mathbf{X}_1, \mathbf{X}_2 \sim \{0,1\}^{n_2}$, each with min-entropy at least $k_2 \geq n_2 - n_2^{\gamma_2}$, the following holds. For any "tampering degree" $t_2 \leq n_2^{\gamma_2}$, any functions $f_1, \dots, f_t, g_1, \dots, g_t : \{0,1\}^{n_2} \to \{0,1\}^{n_2}$ with no fixed points, and any $\sigma_1, \dots, \sigma_t : [2] \to [2]$,*

$$\begin{aligned}|&\mathsf{2nmExt}(\mathbf{X}_1, \mathbf{X}_2) \circ \mathsf{2nmExt}(f_1(\mathbf{X}_{\sigma_1(1)}), g_1(\mathbf{X}_{\sigma_1(2)})) \circ \cdots \circ \mathsf{2nmExt}(f_t(\mathbf{X}_{\sigma_t(1)}), g_t(\mathbf{X}_{\sigma_t(2)})) \\ &- \mathbf{U}_m \circ \mathsf{2nmExt}(f_1(\mathbf{X}_{\sigma_1(1)}), g_1(\mathbf{X}_{\sigma_1(2)})) \circ \cdots \circ \mathsf{2nmExt}(f_t(\mathbf{X}_{\sigma_t(1)}), g_t(\mathbf{X}_{\sigma_t(2)}))| \leq \varepsilon_2,\end{aligned}$$

where $m_2 \geq k_2^{\gamma_2}$ and $\varepsilon_2 = 2^{-k_2^{\gamma_2}}$.

Given the above two ingredients, we can now turn to construct our three-source correlation breaker, which simply dedicates one of its input sources to *condensing* the other two, which are then passed into a two-source nonmalleable extractor call.

Proof of Lemma 3.9. Let $\mathsf{2Cond} : \{0,1\}^{n_1} \times \{0,1\}^{n_1} \to \{0,1\}^{m_1}$ be the function from Lemma 3.10 for min-entropy k_1 with associated parameters $\beta_1, C_1, \gamma_1, \varepsilon_1$ as provided in the lemma statement. Then, let $\mathsf{2nmExt} : \{0,1\}^{n_2} \times \{0,1\}^{n_2} \to \{0,1\}^{m_2}$ be the function from Lemma 3.11 for min-entropy k_2 with its associated parameters $\gamma_2, t_2, \varepsilon_2$. Let $n_1 := n, n_2 := m_1 + a, m_2 := m$, and $k_1 := k$. Now, define $\mathsf{3CB} : (\{0,1\}^n)^3 \times \{0,1\}^a \to \{0,1\}^m$ as

$$\mathsf{CB}(x_1, x_2, x_3, \alpha) := \mathsf{2nmExt}((\alpha, \mathsf{2Cond}(x_1, x_3)), (\alpha, \mathsf{2Cond}(x_2, x_3))).$$

Consider now any three independent sources $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3 \sim \{0,1\}^n$ with min-entropy at least k, any t valid tampering functions $f_1, \dots, f_t$, and any pieces of advice $\alpha_0, \alpha_1, \dots, \alpha_t \in \{0,1\}^a$ with α_0 distinct from the rest. We want to show

$$\begin{aligned}|&\mathsf{3CB}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \alpha_0) \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_t) \\ &- \mathbf{U}_m \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_t)| \leq \varepsilon.\end{aligned}$$

If $k \geq \log^{C_1} n$, then by the properties of $\mathsf{2Cond}$ (and a union bound) we get that

$$\begin{aligned}&|\mathsf{3CB}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, \alpha_0) \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_t) \\ &\qquad - \mathbf{U}_m \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3), \alpha_t)| \\ &\leq |\mathsf{3CB}(\mathbf{X}_1, \mathbf{X}_2, x_3, \alpha_0) \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_t) \\ &\qquad - \mathbf{U}_m \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_t)| + 2\varepsilon_1,\end{aligned}$$

where $\mathsf{2Cond}(\mathbf{X}_1, x_3)$ and $\mathsf{2Cond}(\mathbf{X}_2, x_3)$ both have min-entropy at least $m_1 - m_1^{\beta_1}$, and are of course independent. Towards this end, define $\mathbf{Y}_1 := \mathsf{2Cond}(\mathbf{X}_1, x_3)$ and $\mathbf{Y}_2 := \mathsf{2Cond}(\mathbf{X}_2, x_3)$ so that we may write the above expression as

$$\leq |\mathsf{2nmExt}((\alpha, \mathbf{Y}_1), (\alpha, \mathbf{Y}_2)) \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_t)$$
$$-\mathbf{U}_m \circ \mathsf{3CB}(f_1(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_1) \circ \cdots \circ \mathsf{3CB}(f_t(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_t)| + 2\varepsilon_1.$$

It is now straightforward to verify that since each f_i is a valid tampering function, it cannot produce outputs $\mathbf{X}_1', \mathbf{X}_2'$ that depend, respectively, on $\mathbf{X}_1, \mathbf{X}_2$ *and* end up in the $\mathsf{2Cond}$ call. Thus, once we unwrap the calls to $\mathsf{3CB}$, even on the tampered inputs $f_i(\mathbf{X}_1, \mathbf{X}_2, x_3)$, the output of each $\mathsf{2Cond}$ call will depend on exactly one of $\mathbf{X}_1, \mathbf{X}_2$. In other words, there exist some $\sigma_1, \dots, \sigma_t : [2] \to [2]$ and deterministic functions $g_1, \dots, g_t, h_1, \dots, h_t : \{0,1\}^n \to \{0,1\}^{m_1}$ such that each occurrence of $\mathsf{3CB}(f_i(\mathbf{X}_1, \mathbf{X}_2, x_3), \alpha_i)$ in the expression above can be replaced with $\mathsf{2nmExt}((\alpha_i, g_i(\mathbf{X}_{\sigma_i(1)})), (\alpha_i, h_i(\mathbf{X}_{\sigma_i(2)})))$. Moreover, if $\mathbf{X}_1$ contains any randomness that $\mathbf{Y}_1$ does not, we can fix it, and same for $\mathbf{X}_2$ and $\mathbf{Y}_2$. This means that we can actually replace every expression with $\mathsf{2nmExt}((\alpha_i, g_i(\mathbf{Y}_{\sigma_i(1)})), (\alpha_i, h_i(\mathbf{Y}_{\sigma_i(2)})))$. Thus we arrive at exactly a situation that the two-source non-malleable extractor can protect against (note that the advice strings prevent fixed points).

Provided everything works, the overall error is $2\varepsilon_1 + \varepsilon_2 = 2^{-k_1^{\gamma_1}} + 2^{-k_2^{\gamma_2}}$, where $k_1 = k$ and $k_2 = m_1 - m_1^{\beta_1}$, and $m_1 = k_1^{\gamma_1} = k^{\gamma_1}$. To make sure the two-source condenser worked, we required $k = k_1 \geq \log^{C_1} n$. And to make sure that the non-malleable extractor works, we just need $t \leq n_2^{\gamma_2}$ and $k_2 \geq n_2 - n_2^{\gamma_2}$, where $n_2 = m_1 + a$. Setting $a = \log t$ and β_1 small enough yields the result. □

3.4.1 Polylogarithmic good sources via correlation breakers and Ramsey graphs

Now that we have constructed our correlation breaker, we will put it good use, and prove the following.

Theorem 3.9 (A better extractor, in the few-long-sources setting - Theorem 3.2, restated)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for (N, K, n, k)-adversarial sources with $K \geq \log^C N$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$, provided that we also have $k \geq N^C$.*

In order to get this improvement in the few-long-sources regime, the key idea is to call our three-source correlation breaker over triples that exhibit a pairwise structure that three-source correlation breakers can defend against. As it turns out, looking back at the definition of *valid ordering* from the beginning of this section, one way to ensure that this happens is to call the three-source correlation breaker over the *wedges* of a graph. As a reminder, a wedge in a graph $G = ([N], E)$ is just a subset $S \subseteq [N]$ of three vertices such that the induced subgraph $G[S]$ contains exactly two edges. The vertices of degree 1 in $G[S]$ are called the *terminals* of the wedge, whereas the other vertex (of degree 2) is called the *anchor*. Moreover, since we will be calling a function on each wedge, we must give each wedge an orientation. Here, we orient any given wedge as a triple (w_1, w_2, w_3) so that w_1, w_2 are its terminals and $w_1 < w_2$. Furthermore, we will refer to the collection of all oriented wedges in the graph as the *oriented wedge set* $\mathcal{W} \subseteq [N]^3$.

Now, since our three-source correlation breaker is acting as a robust three-source *extractor*, we will still want to make sure that some call gets *activated*. Thus, we will want an explicit graph such that every big enough subset of vertices contains a wedge. As it turns out, *Ramsey graphs*, defined below, will do the trick.

Definition 3.8. *A graph $G = (V, E)$ is called α-*Ramsey *if it has no clique nor independent set of size $> \alpha$.*

Lemma 3.12 (Finding wedges in Ramsey graphs)**.** *Let $G = ([N], E)$ be an α-Ramsey graph. Let $S \subseteq [N]$ be a subset of size t that contains no wedge. Then $t \leq \alpha^2$.*

Proof. If the induced subgraph $G[S]$ truly contains no wedge, then each connected component in $G[S]$ must be a clique.[14] Suppose there are ℓ such components. We know that $\ell \leq \alpha$, or else we could take one vertex from each component and form an independent set of size $> \alpha$. On the other hand, some connected component must have size $\geq t/\ell$, simply by an averaging argument. Thus we need $t/\ell \leq \alpha$ as well, or else we've found a clique of size $> \alpha$. Combining these two bounds yields $t \leq \alpha\ell \leq \alpha^2$, as desired. □

Given these objects, we are ready to define our new framework for extracting from adversarial sources.

Lemma 3.13 (Extractors for adversarial sources via correlation breakers over a Ramsey graph)**.** *Let* $\mathsf{3CB} : (\{0,1\}^n)^3 \times [t] \to \{0,1\}^m$ *be a three-source correlation breaker for min-entropy k, tampering degree $t = N^3$, and error ε. Let $G = ([N], E)$ be an α-Ramsey graph, and let $\mathcal{W} \subseteq [N]^3$ be its oriented wedge set. Then the function* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *defined as*

$$\mathsf{Ext}(x_1, \ldots, x_N) := \bigoplus_{(i_1,i_2,i_3)\in\mathcal{W}} \mathsf{3CB}(x_{i_1}, x_{i_2}, x_{i_3}, (i_1, i_2, i_3))$$

is an extractor for $(N, \alpha^2 + 1, n, k)$-adversarial sources with error ε.

Proof. Let $\mathbf{X}$ be an $(N, K = \alpha^2 + 1, n, k)$-adversarial source, and observe that since $K > \alpha^2$, and G is an α-Ramsey graph, our wedge-finding lemma (Lemma 3.12) tells us that some oriented wedge $(v_1, v_2, v_3) \in \mathcal{W}$ is completely covered by good sources. To make notation convenient, let us assume (without loss of generality) that $(v_1, v_2, v_3) = (1, 2, 3)$. Now, to upper bound $|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m|$, let us start by fixing all other sources $\mathbf{X}_i$ such that $i \notin \{1, 2, 3\}$. Consider now any oriented wedge $(i_1, i_2, i_3) \in \mathcal{W} \setminus (1, 2, 3)$. We want to examine the sources $(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3})$ and see how they relate to $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$. There are two possibilities here:

1. The terminals $1, 2$ both appear among i_1, i_2, i_3.
2. The terminals $1, 2$ don't both appear among i_1, i_2, i_3.

Let us deal with each possibility separately. In the first case, we *must* have $i_1 = 1$ and $i_2 = 2$. Indeed, notice that we must at least have $\{i_1, i_2\} = \{1, 2\}$, or else one wedge's terminal will be another wedge's anchor (which is impossible, because these objects have different degrees). Further, our orientation tells us that both $1 < 2$ and $i_1 < i_2$, so it's impossible that $i_2 = 1$ and $i_1 = 2$. Thus, we have $(i_1, i_2) = (1, 2)$. Now, notice that i_3 is either equal to 3, or some vertex outside $1, 2, 3$ (as a wedge must consist of three distinct vertices). In the former case, $\mathbf{X}_{i_3}$ can obviously be written as a deterministic function of $\mathbf{X}_3$, as they are the same source. In the latter case, however, this is still true, as we have already fixed all $\mathbf{X}_i, i \notin \{1, 2, 3\}$, and a fixed constant is a deterministic function of anything (as it just ignores its input). Thus, in this case we must have $(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}) = (f_1(\mathbf{X}_1), f_2(\mathbf{X}_2), f_3(\mathbf{X}_3))$ for some deterministic functions f_1, f_2, f_3.

For the second case, assume (without loss of generality) that 2 doesn't appear among i_1, i_2, i_3. Then $\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}$ are all either functions of $\mathbf{X}_1, \mathbf{X}_3$, or some fixed constants. This means there exists some $\sigma : [3] \to [3]$ and deterministic functions f_1, f_2, f_3 such that $(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}) = (f_1(\mathbf{X}_{\sigma(1)}), f_2(\mathbf{X}_{\sigma(2)}), f_3(\mathbf{X}_{\sigma(3)})$ where $1, 2$ do not both appear among $\sigma(1), \sigma(2), \sigma(3)$. Since these cases are exhaustive, this means that we can write

$$\begin{aligned}\mathsf{Ext}(\mathbf{X}) = {} & \mathsf{3CB}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, 1, 2, 3) \\ & \oplus \bigoplus_{w:=(i_1,i_2,i_3)\in\mathcal{W}\setminus(1,2,3)} \mathsf{3CB}(f_{w,1}(\mathbf{X}_{\sigma_w(1)}), f_{w,2}(\mathbf{X}_{\sigma_w(2)}), f_{w,3}(\mathbf{X}_{\sigma_w(3)}), (i_1, i_2, i_3))\end{aligned}$$

[14] This can be verified by induction on the number of vertices in the connected component.

for some deterministic functions $f_{w,c}$ and valid ordering σ. By combining this expression with (an appropriate function plugged into) the data-processing inequality (Fact 2.3), we have $|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m|$ is at most

$$\begin{aligned}|\mathsf{3CB}(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3, 1, 2, 3) \circ (\mathsf{3CB}(f_{w,1}(\mathbf{X}_{\sigma_w(1)}), f_{w,2}(\mathbf{X}_{\sigma_w(2)}), f_{w,3}(\mathbf{X}_{\sigma_w(3)})))_w \\ -\mathbf{U}_m \circ (\mathsf{3CB}(f_{w,1}(\mathbf{X}_{\sigma_w(1)}), f_{w,2}(\mathbf{X}_{\sigma_w(2)}), f_{w,3}(\mathbf{X}_{\sigma_w(3)})))_w|.\end{aligned}$$

But since w ranges over at most N^3 wedges, and $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ each have min-entropy k, and all σ_w are valid orderings, the properties guaranteed of our three-source correlation breaker with tampering degree $t = N^3$ guarantee that the above is upper bounded by ε. This completes the proof. □

Now that we have our framework in place, all that is left to do is instantiate it. We already saw how to construct an explicit three-source correlation breaker, and thus the only thing we need is an explicit Ramsey graph. Luckily, Li recently constructed excellent Ramsey graphs in his breakthrough work.

Lemma 3.14 (Ramsey graphs [Li23])**.** *There exists a constant $C > 0$ such that there exists an explicit α-Ramsey graph $G = ([N], E)$ with $\alpha \leq \log^C N$.*

Finally, we can obtain our significantly improved extractors for the few-long-sources setting.

Proof of Theorem 3.9. Instantiate Lemma 3.13 with Lemmas 3.9 and 3.14. □

It is worth pointing out that the extra bound of the form $k \geq N^C$ in Theorem 3.9 arises from the fact that our framework requires our correlation breaker to handle tampering degree $t = N^3$, while the highest possible degree that it can handle is of the form $t = k^\gamma$ for some tiny constant γ.

3.4.2 Polynomial locality via correlation breakers and extremal C_4-free graphs

We now turn towards an even more surprising application of the wedge extractor and correlation breaker above: this framework can even be used to extract from adversarial sources where the bad sources *depend* on the good sources! To show this, let's first introduce a more general definition of adversarial sources.

Definition 3.9 (Adversarial source)**.** *A source $\mathbf{X} \sim (\{0,1\}^n)^N$ is called an (N, K, n, k)*-adversarial source of locality *d if it consists of N sources of n bits $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$, and both of the following hold:*

- ***Somewhere good sources**: There is a set $S \subseteq [N], |S| \geq K$ such that for any $i \in S, H_\infty(\mathbf{X}_i) \geq k$. These sources are independent, and are called good sources.*
- ***Bounded bad sources**: Every other source $\mathbf{X}_i, i \notin S$ is called a bad source, and is an arbitrary deterministic function of at most d good sources.*[15]

Notice that our original definition of adversarial sources (Definition 3.2) corresponds to $d = 0$. Now, given this more general definition, we will prove the following result.

Theorem 3.10 (An extractor for high locality, in the few-long-sources setting - Theorem 3.3, restated)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-sources of locality $d \leq K^\gamma$ with $K \geq N^{1-\gamma}$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$, provided that we also have $k \geq N^C$.*

[15] Without loss of generality, we can also allow the bad sources to depend on an arbitrary amount of additional randomness completely independent of the good sources, as we can start all our proofs by fixing this additional randomness.

In order to prove this result, we will once again use our three source correlation breaker over the wedges of a certain graph, just as in the previous section. Furthermore, we will again need to make sure that the good sources *land* on some wedge, so that one of our correlation breaker calls will output uniform bits. However, the question now is, how can we break all the correlation that will arise from having bad sources that are not independent? Following a work of Viola [Vio14], the first idea is to get rid of as many as these complex dependencies as possible. In this direction, the following lemma will be invaluable.

Lemma 3.15 (A reduction from d-local adversarial sources to low-weight 1-local adversarial sources)**.** *Let* $\mathbf{X} \sim (\{0,1\}^n)^N$ *be an* (N, K, n, k)*-adversarial source of locality* $d > 1$*. Then* $\mathbf{X}$ *is a convex combination of* $(N, \frac{K^2}{4Nd^2}, n, k)$*-adversarial sources of locality* 1 *and weight* $w \leq 2dN/K$*.*

Proof. Identify $\mathbf{X}$ with a bipartite graph $G = (V, E)$ over $K + N$ vertices as follows: partition V into L, R such that $|L| = K$ and $|R| = N$. Identify all N sources of $\mathbf{X}$ with R, identify the good sources of $\mathbf{X}$ with L, and draw an edge between $u \in L$ and $v \in R$ if $\mathbf{X}_u, \mathbf{X}_v$ are correlated. We will find a subset $L' \subseteq L$ of size $K' \geq K^2/(4Nd^2)$ such that every vertex in R has at ≤ 1 neighbor in L', and every vertex in L' has $\leq 2dN/K$ neighbors in R. Then, if we fix all sources in $\{\mathbf{X}_i\}_{i \in L \setminus L'}$, we are done: the locality is reduced to 1, K' good sources remain unfixed, and each unfixed good source has at most $2dN/K$ bad sources depending on it.

First, note that because $\mathbf{X}$ has locality d, every vertex in R has degree at most d, and thus $|E| \leq Nd$. Next, observe that there must be some subset $L^* \subseteq L$ of size at least $K/2$ where each $u \in L^*$ has degree at most $2dN/K$; otherwise L contains at least $K/2$ vertices of degree $> 2dN/K$, contradicting $|E| \leq Nd$. We will now show how to select L' from L^*.

For a vertex $v \in V$, we let $N(v)$ denote its neighbors, and for a set of vertices $S \subseteq V$, we let $N(S) = \bigcup_{v \in V} N(v)$. Now, we will greedily grow L' as follows: first, initialize $U \leftarrow L^*$. Then arbitrarily pick a vertex u from U and add it to L', remove $N(N(u))$ from U, and repeat while U is not empty. Because each $v \in R$ has degree at most d, and each $v \in L^*$ has degree at most $2dN/K$, the above process will produce a set L' of size at least $K' := |L^*|/(2d^2N/K) \geq K^2/(4Nd^2)$. Furthermore, our selection process ensures that we never select two vertices from L that share a neighbor, so no vertex in R has more than one neighbor in L'. Lastly, because $L' \subseteq L^*$, each vertex in L' has at most $2dN/K$ neighbors in R, so we are done. □

Given this reduction, we can pretty much assume that we are now just dealing with a 1-local adversarial source $\mathbf{X}$. But how can we go about extracting from these? The key idea is to visualize collecting all the sources of $\mathbf{X}$ into K different clouds, where each cloud includes one of the good sources, and all of the bad sources that depend on it. Since we are now dealing with a 1-local adversarial source, these clouds are nonempty and disjoint. Now, recall that we will be extracting over the wedges of a graph, so we must consider how these clouds interact with that graph. The idea is that if they form a joint structure that is special enough, then the wedge extractor will work! As it turns out, the joint structure we are looking for is something called a *cloud wedge*. Intuitively, a cloud wedge guarantees that the good sources not only land on a wedge, but the clouds emanating from each vertex in this wedge also have a special structure: namely, the clouds on the terminals of the wedge must not have any edges between them.

Definition 3.10 (Cloud-wedge)**.** *Let* $G = ([N], E)$ *be a graph. Let* $S_1, S_2, S_3 \subseteq [N]$ *be nonempty disjoint subsets, and let* $s_1, s_2, s_3 \in [N]$ *be vertices such that each* $s_i \in S_i$*. We say that* $(\{S_1, S_2, S_3\}, \{s_1, s_2, s_3\})$ *is a* cloud-wedge *if* $\{s_1, s_2, s_3\}$ *is a wedge and there exist no edges crossing between the two clouds that contain the terminals of this wedge.*

Given a 1-local adversarial source with K good sources, is it possible to construct a graph such that no matter *where* the good sources and their clouds are placed, we will always be able to find a cloud wedge?

Surprisingly, we actually show that the answer is *yes*. In fact, we show that if a graph has no cycle of length 4 as a subgraph (induced or otherwise), and its independence number is not too big, then one is always guaranteed to find a cloud wedge. We prove the following, which is the main structural result of this section.

Lemma 3.16 (Finding cloud-wedges in extremal C_4-free graphs)**.** *Let $G = ([N], E)$ be a C_4-free graph with independence number α. Let $S_1, \dots, S_t \subseteq [N]$ be nonempty disjoint subsets of size at most w, and let $s_1, \dots, s_t \in [N]$ be vertices such that each $s_i \in S_i$. Suppose there do not exist indices $i_1, i_2, i_3 \in [t]$ such that $(\{S_1, S_2, S_3\}, \{s_1, s_2, s_3\})$ is a cloud-wedge. Then $t < 36w^4\alpha$.*

Proof. Define $A := \{s_1, \dots, s_t\}$ and consider the induced subgraph $G[A]$. We start by looking for a big *star* in this induced subgraph: namely, a complete bipartite graph with one vertex on one side, and many vertices on the other side. Towards this end, let Δ denote the maximum degree of any vertex in $G[A]$, and suppose (without loss of generality) that s_t is a maximum degree vertex. To get a quick lower bound on Δ, recall that we can always find an independent set of size $t/(\Delta + 1)$ by iteratively grabbing vertices and throwing away their neighbors: thus, we have $\alpha \geq t/(\Delta + 1)$, or rather $\Delta \geq t/\alpha - 1$. We can henceforth assume this is at least 3, or else we would get an upper bound on t that trivially satisfies the lemma statement.

Now, if we let $\mathcal{N}$ denote the neighborhood of s_t in A, then the induced subgraph $G[\mathcal{N}]$ must have maximum degree 1: otherwise, s_t would form a four-cycle with three of its neighbors. This, in turn, means that there must be an independent set of size $\ell \geq \Delta/2 \geq t/(2\alpha) - 2$ among the neighbors of s_t. Without loss of generality, suppose they are the vertices $s_1, \dots, s_\ell$, and notice that the induced subgraph $G[\{s_1, \dots, s_\ell, s_t\}]$ indeed forms a star. This means that s_i, s_j, s_t form a wedge, for any $1 \leq i < j \leq \ell$.

If there is truly no cloud-wedge to be found, this means that there is an edge between every pair of clouds S_i, S_j for $1 \leq i < j \leq \ell$. Consider now the induced subgraph $G' := G[S_1 \cup \dots \cup S_\ell]$. Since there is an edge between every pair of clouds in G', we know that each cloud S_i has $\geq \ell - 1$ edges leaving it, which means it must contain some vertex v_i whose neighborhood $\mathcal{N}_i$ has size $\geq (\ell - 1)/w$. Since G' is a C_4-free graph, the shared neighborhood $\mathcal{N}_i \cap \mathcal{N}_j$ of any two distinct vertices v_i, v_j must have size 1.

Because of the discussion above, we know that any pair of vertices in G' can show up together in at most one neighborhood $\mathcal{N}_i$. Thus, since the number of total vertex pairs in G' is at most $\binom{w\ell}{2}$, and the number of vertex pairs in each $\mathcal{N}_i$ is some $m_i \geq \binom{(\ell-1)/w}{2}$, we have

$$\ell \cdot \binom{(\ell - 1)/w}{2} \leq \sum_{i \in [\ell]} m_i \leq \binom{w\ell}{2},$$

which implies $\ell < 16w^4$. Plugging in our lower bound from above $\ell \geq t/(2\alpha) - 2$ yields the result. □

Thus, we see that if we place a 1-local adversarial source onto the vertices of an extremal C_4-free graph, then we are guaranteed to be able to find a cloud-wedge, which will allow us to easily extract and break correlation. Thus, by starting off with our reduction from d-local adversarial sources to 1-local adversarial sources, we get the following framework for extracting from adversarial sources of high locality.

Lemma 3.17 (Extractors for adversarial sources via correlation breakers over a C_4-free graph)**.** *Let $\mathsf{3CB} : (\{0,1\}^n)^3 \times [t] \to \{0,1\}^m$ be a three-source correlation breaker for min-entropy k, tampering degree $t = N^3$, and error ε. Let $G = ([N], E)$ be a C_4-free graph with independence number α, and let $\mathcal{W} \subseteq [N]^3$ be its oriented wedge set. Then the function $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ defined as*

$$\mathsf{Ext}(x_1, \dots, x_N) := \bigoplus_{(i_1, i_2, i_3) \in \mathcal{W}} \mathsf{3CB}(x_{i_1}, x_{i_2}, x_{i_3}, (i_1, i_2, i_3))$$

is an extractor for $(N, 4dN^{5/6}\alpha^{1/6}, n, k)$-adversarial sources of locality d with error ε.

Proof. Let $\mathbf{X}$ be an (N, K, n, k)-adversarial source of locality d, where $K \geq 4dN^{5/6}\alpha^{1/6}$. By our reduction from d-local adversarial sources to low-weight 1-local adversarial sources (Lemma 3.15), we can assume (without loss of generality)[16] that $\mathbf{X}$ is an (N, K', n, k)-adversarial source of locality 1 and weight $w \leq 2dN/K$, with $K' \geq \frac{K^2}{4Nd^2}$ good sources. Without loss of generality (just for notational convenience), assume that the good sources in $\mathbf{X}$ are $\mathbf{X}_1, \ldots, \mathbf{X}_{K'}$.

Now, for every $i \in [K']$, let $D_i \subseteq [N]$ denote the collection of sources that depend on the good source i (including i itself). By definition of adversarial source of locality 1 and weight w, we know that these D_i's are nonempty, disjoint, and have size at most w. Let us now try to find a cloud-wedge among the subsets $D_1, \ldots, D_{K'} \subseteq [N]$ and vertices $1, 2, \ldots, K'$ where each $i \in D_i$. By our cloud-wedge-finding (Lemma 3.16), we know this is possible as long as $K' \geq 36w^4\alpha$. Plugging in our known values of K' and w^4, we just need

$$\frac{K^2}{4Nd^2} \geq 36\left(\frac{2dN}{K}\right)^4 \alpha,$$

or rather

$$K \geq (36)^{1/6} \cdot 2dN^{5/6}\alpha^{1/6}.$$

And, indeed, we were originally guaranteed $K \geq 4dN^{5/6}\alpha^{1/6}$. Thus, we know that among the subsets $D_1, \ldots, D_{K'}$ and vertices $1, \ldots, K'$ with each $i \in D_i$, there is some cloud-wedge present. Without loss of generality (again for notational convenience), let us assume $(\{D_1, D_2, D_3\}, 1, 2, 3)$ is a cloud-wedge, and that $(1, 2, 3)$ is the oriented version of the wedge $\{1, 2, 3\}$.

Now, let us fix all *good* sources $\mathbf{X}_i, i \notin [3]$. Notice the sources $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ are still independent and have not lost any entropy, and furthermore every other source (good or bad) is either fixed, or a deterministic function of *one of* $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$. Let $\delta : [N] \setminus [3] \to [3] \cup \{0\}$ capture this dependency; namely, if i is in some cloud $D_b, b \in [3]$, then $\delta(i) := b$ (recall that it can be in at most one cloud). On the other hand, if j does not belong to any of the clouds D_1, D_2, D_3, then $\delta(i) := 0$.

Now, consider any oriented wedge $(i_1, i_2, i_3) \in \mathcal{W} \setminus (1, 2, 3)$. We want to examine the sources $\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}$ and see how they relate to $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$. There are two possibilities here.

1. The terminals $1, 2$ both appear among $\delta(i_1), \delta(i_2), \delta(i_3)$.

2. The terminals $1, 2$ don't both appear among $\delta(i_1), \delta(i_2), \delta(i_3)$.

Let's deal with each possibility separately. In the first case, notice that we cannot have $\delta(i_3) \in \{1, 2\}$, or in other words: $i_3 \notin D_1 \cup D_2$. To see why, simply note that if i_3 were in either of these sets, say D_1, then i_1 or i_2 must be in D_2 by the case condition. But this is impossible, since (i_1, i_2, i_3) is an oriented wedge with terminals i_1, i_2, and thus the above situation would force an edge between D_1, D_2 (contradicting the definition of cloud-wedge). Therefore, in this case we must have $\delta(i_3) \in \{0, 3\}$ and $\{\delta(i_1), \delta(i_2)\} = \{1, 2\}$. This means that $\mathbf{X}_{i_3}$ can be written as a deterministic function (perhaps a constant one) of $\mathbf{X}_3$, whereas $\mathbf{X}_{i_1}, \mathbf{X}_{i_2}$ can be written as deterministic functions of $\mathbf{X}_1, \mathbf{X}_2$ (respectively), or as $\mathbf{X}_2, \mathbf{X}_1$ (respectively). Thus, there is a valid ordering $\sigma : [3] \to [3]$ and deterministic functions f_1, f_2, f_3 such that $(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}) = (f_1(\mathbf{X}_{\sigma(1)}), f_2(\mathbf{X}_{\sigma(2)}), f_3(\mathbf{X}_{\sigma(3)}))$.

In the second case, assume (without loss of generality) that 2 doesn't appear among $\delta(i_1), \delta(i_2), \delta(i_3)$. Then $\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}$ are all either functions of $\mathbf{X}_1, \mathbf{X}_3$, or some fixed constants. Thus there is some $\sigma : [3] \to [3]$ and deterministic functions f_1, f_2, f_3 such that $(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}) = (f_1(\mathbf{X}_{\sigma(1)}), f_2(\mathbf{X}_{\sigma(2)}), f_3(\mathbf{X}_{\sigma(3)}))$, where $1, 2$ do not both appear among $\sigma(1), \sigma(2), \sigma(3)$.

[16] As a reminder, this is because an extractor for a family of sources is also an extractor for convex combinations of members from that family.

Thus for any $(i_1, i_2, i_3) \in \mathcal{W} \setminus (1, 2, 3)$, there exist deterministic functions f_1, f_2, f_3 and a valid ordering $\sigma : [3] \to [3]$ such that $(\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \mathbf{X}_{i_3}) = (f_1(\mathbf{X}_{\sigma(1)}), f_2(\mathbf{X}_{\sigma(2)}), f_3(\mathbf{X}_{\sigma(3)}))$. The remainder of the proof is now identical to that of Lemma 3.13. □

With our framework in hand and our correlation breaker ready to go, the very last thing we need are explicit C_4-free graphs with small independence number. Luckily for us, such constructions exist.

Lemma 3.18 (Extremal C_4-free graphs [Alo86])**.** *There exists an explicit C_4-free graph $G = ([N], E)$ with independence number $\alpha(G) \leq 2N^{3/4}$.*

By plugging these C_4-free graphs and our correlation breakers into the framework from above, we finally get our adversarial source extractors for high locality.

Proof of Theorem 3.10. Instantiate Lemma 3.17 with Lemmas 3.9 and 3.18. □

3.5 Optimized extractors for many short sources

In the previous section, we showed that one can get much improved extractors for adversarial sources, should the adversarial source consist of just a few, long sources. It is natural to ask what happens in the other regime, when it consists of many, short sources. As it turns out, we can get the exact same improvements in this setting, and the proofs are much easier! The reason is because in such a setting, adversarial sources can be reduced to a much simpler model, known as a low-weight affine source.

Low-weight affine sources

As we will see in Chapter 6, affine sources are a unifying model in extractor theory, capturing a ton of seemingly unrelated families of sources. In this section, we'll add another one to the list. In a little more detail, an affine source $\mathbf{X}$ is simply a distribution that is uniform over some affine subspace of $\mathbb{F}_2^n$. More formally, these sources take the form $\mathbf{X} = A\mathbf{U}_k + b$, where $\mathbf{A} \in \mathbb{F}_2^{n \times k}$ is some matrix with full column rank, $b \in \mathbb{F}_2^n$ is an arbitrary vector, and $\mathbf{U}_k$ is a uniform random variable. While affine sources are generally quite challenging to extract from, certain specializations of them are much easier. One such specialization is the family of *weight* w affine sources (for some small w), defined to be those for which the columns of A each have Hamming weight at most w. While we still don't have many great low-error extractors for affine sources in general, we do have *excellent* low-error extractors for low-weight affine sources:

Lemma 3.19 (An explicit extractor for low-weight affine sources [Rao09b, DW12, Vio14])**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for affine sources of min-entropy $k \geq \log^C n$ and weight $w \leq k^{1-\delta}$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

With this in mind, the goal will be to show that in the many-short-source setting, adversarial sources can be effectively reduced to low-weight affine sources, from which extraction is easy via the lemma above. In fact, we will actually reduce our adversarial sources to an even more specialized model, which we might call 1-*local* weight w affine sources. Such sources not only guarantee that the column weights of A are at most w, but that the *row weights* of A are at most 1. The following lemma will be remarkably helpful.

Lemma 3.20 (A reduction from general sources to 1-local sources)**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a source of min-entropy $k \geq 1$. Then $\mathbf{X}$ is a convex combination of 1-local sources of min-entropy 1.*

Proof. Since $\mathbf{X}$ has min-entropy at least ≥ 1, it is a convex combination of flat sources with min-entropy exactly 1 (Fact 2.9). But any flat source $\mathbf{X}'$ with min-entropy 1 is (by definition) a uniform distribution over two distinct strings $x, y \in \{0,1\}^n$. And any two distinct strings must differ at *some* coordinate $\alpha \in [n]$. Thus $\mathbf{X}'_\alpha$ is a uniform bit, and it is easy to verify that every other bit $\mathbf{X}'_i, i \neq \alpha$ is either constantly 0, constantly 1, or exactly equal to $\mathbf{X}'_\alpha$ or $\mathbf{X}'_\alpha \oplus 1$. In other words, every other bit is a deterministic function of $\mathbf{X}'_\alpha$, meaning that $\mathbf{X}'$ is a 1-local source. □

A simple corollary of the above result, which will serve as a core tool in our extractors, is the following.

Corollary 3.1 (A reduction from 1-local adversarial sources to 1-local sources)**.** *Let* $\mathbf{X} \sim (\{0,1\}^n)^N$ *be an* $(N, K, n, k \geq 1)$*-adversarial source of locality* $d \leq 1$ *and weight* w*. Then,* $\mathbf{X}$ *is a convex combination of* 1*-local sources of min-entropy* K *and weight* wn*.*

Proof. By the definition of adversarial sources (Definition 3.9) and Lemma 3.20, we can assume (without loss of generality) that each good source in $\mathbf{X}$ is a 1-local source of min-entropy 1, while every bad source is a deterministic function of at most 1 good source. But since each good source $\mathbf{X}_i$ is a 1-local source of min-entropy 1, there is a single bit in $\mathbf{X}_i$ that determines all of its other bits. This means that any bad source depending on $\mathbf{X}_i$ is a deterministic function of that single bit alone. Thus, $\mathbf{X}$ consists of K uniform independent bits (one for each good source), with the remaining $Nn - K$ bits being a deterministic function of exactly 1 of these good bits each. In other words, $\mathbf{X}$ is a 1-local source of min-entropy K. Finally, since $\mathbf{X}$ was originally an adversarial source of weight w, each good source can influence up to w other sources (including itself), and every source (good or bad) has length n. Thus, the 1-local source discussed above must have weight wn, as desired. □

Given the tools from above, we are ready to construct our two new extractors for adversarial sources.

3.5.1 Polylogarithmic good sources via extractors for low-weight affine sources

Our first optimized extractor for the many-short-sources setting matches the improvements seen by our first optimized extractor for the few-long-sources setting (Theorem 3.9).

Theorem 3.11 (A better extractor, in the many-short-sources setting (Theorem 3.4, restated))**.** *There exist universal constants* $C, \gamma > 0$ *such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for* (N, K, n, k)*-adversarial sources with* $K \geq \log^C N$ *good sources of min-entropy* $k \geq 1$*, which has output length* $m \geq K^\gamma$ *and error* $\varepsilon = 2^{-K^\gamma}$*, provided that we also have* $K \geq n^C$*.*

Proof. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an $(N, K, n, k \geq 1)$-adversarial source (of locality 0). Note that since $\mathbf{X}$ is an adversarial source of locality 0, it is also an adversarial source of weight 1 (since the only source correlated with a good source is itself). Thus, applying our reduction from 0-local adversarial sources to 1-local sources (Corollary 3.1), we get that $\mathbf{X} \sim \{0,1\}^{Nn}$ is a convex combination of 1-local sources of min-entropy K and weight n. Given Rao's explicit low-weight affine extractor (Lemma 3.19), we know that we can extract $m \geq K^\gamma$ bits with error $\varepsilon = 2^{-K^\gamma}$, as long as $K \geq \log^C(Nn)$ and $n \leq K^{0.99}$, where C is a large enough constant and γ is a small enough one. Notice that the latter two conditions are satisfied as long as $K \geq n^{C'}$ and $K \geq \log^{C'} N$ for a large enough constant C'. The result follows. □

3.5.2 Polynomial locality via extractors for low-weight affine sources

Complementing the result above, our second optimized extractor for the many-short-sources setting matches the improvements seen by our second optimized extractor for the few-long-sources setting (Theorem 3.10).

Theorem 3.12 (An extractor for high locality, in the many-short-sources setting (Theorem 3.5, restated))**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for (N, K, n, k)-sources of locality $d \leq K^\gamma$ with $K \geq N^{1-\gamma}$ good sources of min-entropy $k \geq 1$, which has output length $m \geq K^\gamma$ and error $\varepsilon = 2^{-K^\gamma}$, provided that we also have $K \geq n^C$.*

Proof. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an $(N, K, n, k \geq 1)$-adversarial source of locality d. By our reduction from d-local adversarial sources to low-weight 1-local adversarial sources (Lemma 3.15), we know that $\mathbf{X}$ is a convex combination of $(N, \frac{K^2}{4Nd^2}, n, k)$-adversarial sources of locality 1 and weight at most $2dN/K$. Then, by further applying our reduction from 1-local adversarial sources to 1-local sources (Corollary 3.1), we get that $\mathbf{X} \sim \{0,1\}^{Nn}$ is a convex combination of 1-local sources of min-entropy $\Gamma \geq \frac{K^2}{4Nd^2}$ and weight $w \leq (2dN/K)n = 2dnN/K$. Given Rao's explicit low-weight affine extractor (Lemma 3.19), we can extract $m \geq \Gamma^\gamma$ bits with error $\varepsilon = 2^{-\Gamma^\gamma}$, as long as $\Gamma \geq \log^C(Nn)$ and $w \leq \Gamma^{0.99}$, where C is a large enough constant and γ is a small enough one. Plugging in our values for Γ and w, it is straightforward to verify that the last two conditions hold as long as $K \geq 2d\sqrt{N \log^C(Nn)}$ and $K \geq 2dN^{0.67}n^{0.34}$. Moreover, we can ensure $\Gamma \geq K^{0.01}$ (to ensure the correct output length and error) as long as $K \geq 4d^{1.01}N^{0.51}$. Notice that all three conditions are satisfied as long as $K \geq n^{C'}$ and $K \geq N^{1-\gamma'}$ and $d \leq K^{\gamma'}$ for a large enough constant C' and a small enough constant γ'. The result follows. □

This concludes the presentation, discussion, and proofs of our five main results (Theorems 3.1 to 3.5). Next, we move on to discuss some exciting applications.

3.6 Applications in pseudorandomness

As we have seen, adversarial sources are a natural generalization of the classical independent source model, and are therefore interesting to study in their own right. As it turns out, however, adversarial sources are also general enough to model several other important settings in pseudorandomness, distributed computing, and cryptography, yielding applications in each. Within pseudorandomness (in particular, the theory of extractors), adversarial sources capture two other well-studied models of weak random sources. As a result, our new extractors for adversarial sources immediately give new and improved extractors in these two settings.

3.6.1 Extractors for total-entropy sources

The first well-studied model for which we get new-and-improved extractors (via our extractors for adversarial sources) is the model of *total-entropy sources*. Total-entropy sources are a natural generalization of the classical independent source model, and are formally defined as follows.

Definition 3.11 (Total-entropy source)**.** *A random variable $\mathbf{X} \sim (\{0,1\}^n)^N$ is called an (N, n, Γ)-total-entropy source if each $\mathbf{X}_i \sim \{0,1\}^n$ is independent and $H_\infty(\mathbf{X}) \geq \Gamma$.*

Total-entropy sources were first introduced by Kamp, Rao, Vadhan and Zuckerman [KRVZ11], though similar models appear in the earlier works of König and Maurer [KM04, KM05]. Total-entropy sources are not only a more robust model of independent sources, but they also play a fundamental role in extracting from other, more well-studied, models of randomness [KRVZ11]. In this work, we construct significantly improved (low-error) extractors for total-entropy sources, and prove the following theorem.

Theorem 3.13 (Improved extractors for total-entropy sources)**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, n, Γ)-total entropy sources of min-entropy $\Gamma \geq CnN^\delta + C(nN)^{1/2+\delta}$, which has output length $m \geq \Gamma^\gamma$ and error $\varepsilon = 2^{-\Gamma^\gamma}$.*

Previously, the best-known low-error explicit extractors for total-entropy sources required min-entropy $\Gamma = O(nN^{1/2} + (nN)^{1-\gamma})$ [KRVZ11, Theorem 1.6], whereas non-explicitly it is known that there exist such extractors for min-entropy $\Gamma = O(n + \log N)$ [KRVZ11, Theorem 1.10].[17] Thus, while there is still a lot of room for improvement, our total-entropy extractor is almost optimal when the source does not consist of too many subsources.

Remark 3.1. *The entropy requirement in Theorem 3.13 becomes $\Gamma = O(n^{1+\delta})$ when $N \leq n$, which is close to the optimal entropy requirement of $\Gamma = O(n)$.*

In order to prove Theorem 3.13, we simply use our new extractor for adversarial sources (Theorem 3.1), without any further modifications. In order to do so, we establish a connection between total-entropy sources and adversarial sources. This connection has been observed in prior work (under different terminology) - we emphasize that the new contribution here is the new extractor for adversarial sources that we plug into this known connection. We formally describe this connection below, and show how it can be used (in conjunction with Theorem 3.1) to obtain our new-and-improved extractors for total-entropy sources.

A known reduction from total-entropy sources to adversarial sources As hinted at before, total-entropy sources are very similar to adversarial sources, with one key difference. While an (N, K, n, k)-adversarial source $\mathbf{X} \sim (\{0,1\}^n)^N$ offers the guarantee that at least K subsources each have min-entropy at least k, an (N, n, Γ)-total-entropy source $\mathbf{X} \sim (\{0,1\}^n)^N$ only guarantees that the total min-entropy across *all* the subsources is at least Γ. Thus, while an (N, K, n, k)-adversarial source is always an (N, n, Kk)-total-entropy source, the converse is not necessarily true. So, how can we use our improved adversarial source extractors to obtain Theorem 3.13?

While an (N, n, Kk)-total-entropy source is not necessarily an (N, K, n, k)-adversarial source, it does turn out to be an adversarial source with weaker parameters. The intuition is that if the total-entropy source $\mathbf{X} \sim (\{0,1\}^n)^N$ has enough min-entropy, then there must be at least a few subsources each with at least a little bit of min-entropy: otherwise, every single subsource would have extremely little entropy, and the total-entropy of $\mathbf{X}$ would not be very high at all. This can be formalized via the following lemma, which is proved using a Markov-type argument.

Lemma 3.21 (A reduction from total-entropy sources to adversarial sources [KRVZ11, Lemma 4.1, adapted])**.** *If $\mathbf{X}$ is an (N, n, Γ)-total-entropy source, then it is also an (N, K, n, k)-adversarial source, as long as*

$$\Gamma \geq Kn + Nk.$$

Proof. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an (N, n, Γ)-total-entropy source. Let $S \subseteq [N]$ be the indices of the subsources in $\mathbf{X}$ with entropy at least k. Observe that each subsource $\mathbf{X}_i$ with $i \in S$ has min-entropy at most n, while each subsource $\mathbf{X}_i$ with $i \notin S$ has min-entropy less than k. If $|S| < K$, then the total entropy in $\mathbf{X}$ is $\Gamma < |S| \cdot n + |[N] \setminus S| \cdot k < Kn + Nk$, which contradicts the guaranteed lower bound on Γ. □

[17] Furthermore, it is known that this non-explicit entropy requirement is near-optimal, as the classic extractor impossibility result (Fact 1.1) rules out the existence of extractors for total-entropy sources with $\Gamma = n - 1$.

Given this reduction from total-entropy sources to adversarial sources, it is now a straightforward (but tedious) calculation to obtain Theorem 3.13 using our new extractors for adversarial sources.

Proof of Theorem 3.13. Let $\mathbf{X} \sim (\{0,1\}^n)^N$ be an (N, n, Γ)-total-entropy source, and let $t \in \mathbb{N}$ be an arbitrary integer that divides N, which we will set later. If we consider every t consecutive subsources of $\mathbf{X}$ as a single subsource, we may instead interpret $\mathbf{X}$ as a random variable over $(\{0,1\}^{nt})^{N/t}$, meaning that $\mathbf{X}$ is also an $(N/t, nt, \Gamma)$-total-entropy source. And by the reduction from total-entropy sources to adversarial sources (Lemma 3.21), we know that $\mathbf{X}$ is also an $(N/t, K, nt, k)$-adversarial source, as long as

$$\Gamma \geq Knt + Nk/t. \tag{3.1}$$

Now, applying our new extractor for adversarial sources (Theorem 3.1), we know that we can extract $m \geq k^\gamma$ bits with error $\varepsilon = 2^{-k^\gamma}$, as long as $K \geq C(N/t)^\delta$ and $k \geq C(nt)^\delta$.[18] Plugging this into Equation (3.1), we get that for any $t \in \mathbb{N}$ that divides N, there exists an explicit extractor for (N, n, Γ)-total-entropy sources with output length $m \geq C(nt)^\delta$ and error $\varepsilon = 2^{-C(nt)^{\delta\gamma}}$, as long as

$$\Gamma \geq C(N/t)^\delta nt + CN(nt)^\delta/t = CnN^\delta t^{1-\delta} + CNn^\delta/t^{1-\delta}.$$

All that remains is to set t to minimize the entropy requirement. We would like to set $t = \sqrt{N/n}$ in order to two terms in the above expression and obtain $\Gamma \geq 2C(Nn)^{1/2+\delta/2}$. However, this is clearly not possible when $N < n$. Thus we consider two cases.

1. If $N < n$, we set $t = 1$. In this case, we get an explicit extractor for (N, n, Γ)-total-entropy sources with output length $m \geq C(nt)^\delta = Cn^\delta \geq C(Nn)^{\delta/2} \geq C\Gamma^{\delta/2}$ and error $\varepsilon = 2^{-C(nt)^{\delta\gamma}} \leq 2^{-C\Gamma^{\delta\gamma/2}}$, as long as $\Gamma \geq CnN^\delta + CNn^\delta$, where the latter expression is at most $CnN^\delta + C(N^\delta n^{1-\delta})n^\delta \leq 2CN^\delta n$.

2. If $N \geq n$, we set $t = \sqrt{N/n}$, which is now guaranteed to be at least 1.[19] In this case, we get an explicit extractor for (N, n, Γ)-total-entropy sources with output length $m \geq C(nt)^\delta = C(Nn)^{\delta/2} \geq C\Gamma^{\delta/2}$ and error $\varepsilon = 2^{-C(nt)^{\delta\gamma}} = 2^{-C(Nn)^{\delta\gamma/2}} \leq 2^{-C\Gamma^{\delta\gamma/2}}$, as long as $\Gamma \geq CnN^\delta(\sqrt{N/n})^{1-\delta} + CNn^\delta/(\sqrt{N/n})^{1-\delta} = 2C(Nn)^{1/2+\delta/2}$.

Thus in both cases, we extract $m \geq C\Gamma^{\delta/2}$ bits with error $\varepsilon \leq 2^{-C\Gamma^{\delta\gamma/2}}$, as long as $\Gamma \geq 2C\max\{nN^\delta, (Nn)^{1/2+\delta/2}\}$. Resetting constants δ, C, γ appropriately completes the proof. □

Above, we have seen that our new extractors for adversarial sources immediately yield new-and-improved extractors for total-entropy sources. One reason why total-entropy sources are interesting to study is because they play a fundamental role in extracting from another, more well-studied model of weak randomness. In the next subsection, we discuss this connection further and demonstrate how our improved extractors for total-entropy sources also yield improved extractors in this more well-studied setting.

[18] Note that we can actually handle $k \geq \log^C(nt)$, but we will instead enforce $k \geq C(nt)^\delta$ to keep our error small and output length large.

[19] While t is now guaranteed to be at least 1, it is not guaranteed to divide N. However, it is straightforward to slightly modify the argument in such a case: i.e., set $t = \lfloor\sqrt{N/n}\rfloor$ and restart the proof with the prefix of $\mathbf{X}$ over $(\{0,1\}^{nt})^{\lfloor N/t\rfloor}$. Since this removes $< t$ subsources from $\mathbf{X}$, it removes $< tn \leq \sqrt{Nn}$ bits of entropy from $\mathbf{X}$. This loss in entropy can be captured by the hidden constant C in the entropy requirement.

3.6.2 Extractors for small-space sources

The second well-studied model for which we get new-and-improved extractors (via our extractors for adversarial sources) is the model of *small-space sources*. At a high-level, small-space sources are distributions that can be sampled by algorithms with limited memory. To model an algorithm with limited memory, we use *branching programs*. A branching program of width w and length n is a directed acyclic graph with $n + 1$ layers, where the first layer has one node, the remaining layers have w nodes each, and every edge starting in layer i terminates in layer $i + 1$. Small-space sources are then defined as follows.

Definition 3.12 (Small-space source)**.** *A random variable* $\mathbf{X} \sim \{0, 1\}^n$ *is called a* space s source *if it is generated by a random walk starting on the first layer of a branching program of width* 2^s *and length* n*, where each edge is labeled with an output bit and some transition probability.*

Small-space sources were first introduced by Kamp, Rao, Vadhan and Zuckerman [KRVZ11], though similar models were studied in the earlier works of Blum [Blu86], Vazirani [Vaz87a], and König and Maurer [KM04, KM05]. More generally, small-space sources fit into the line of work initiated by Trevisan and Vadhan [TV00] on extracting from distributions that can be sampled by algorithms with limited resources. The motivation to study such sources comes from the belief that they model distributions that one might actually find in nature [Lev86, TV00]. Beyond this motivation, small-space sources are also general enough to capture several other well-studied models of randomness [vN51, Blu86, Vaz87a, KM04, KM05, CFG$^+$85, KZ07, CG88]. In this work, we construct significantly improved (low-error) extractors for small-space sources, and prove the following theorem.[20]

Theorem 3.14 (Improved extractors for small-space sources)**.** *For any fixed* $0 < \delta \leq 1/2$*, there exist constants* $C, \gamma > 0$ *such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0, 1\}^n \to \{0, 1\}^m$ *for space* s *sources of min-entropy* $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$*, which has output length* $m \geq k^\gamma$ *and error* $\varepsilon = 2^{-k^\gamma}$.

Previously, the best-known low-error explicit extractors for small-space sources required min-entropy $k \geq O(n^{1-\gamma} + n^{2/3}s^{1/3})$, where $\gamma > 0$ is some tiny constant [KRVZ11, Theorem 1.2]. As a result, we improve the entropy requirement from $k \geq O(n^{1-\gamma})$ to $k \geq O(n^{1/2+\delta})$ for all small-space sources with $s \leq n^\delta$. For even larger settings of s, a straightforward calculation shows that our new entropy requirement is just as good as (and usually significantly better than) the previous state-of-the-art.[21] Non-explicitly, on the other hand, it is known that there exist low-error extractors for space s sources with min-entropy $k \geq O(s + \log n)$ [KRVZ11, Theorem 1.5].[22] Thus, while there is still a lot of room for improvement, we note that any further progress would require substantially new techniques.

Remark 3.2. *Like the previous-best extractors for small-space sources [KRVZ11], our new extractors follow from a reduction to total-entropy sources. As we will see, it is impossible to achieve min-entropy* $k < n^{1/2}$ *using such a reduction. Thus, our requirement of* $k \geq O(n^{1/2+\delta}s^{1/2-\delta})$ *is near-optimal for this technique.*

[20] In this thesis, we construct several new-and-improved extractors for small-space sources. Here, we only briefly highlight the result that comes for free from our new extractors for adversarial sources. In Chapter 7, we provide an in-depth treatment of all of our new extractors for small-space sources.

[21] Indeed, note that $n^{2/3}s^{1/3} \geq n^{1/2+\delta}s^{1/2-\delta}$ as long as $s \leq n$. For $s \geq n$, both bounds enforce a trivial entropy requirement of $k \geq n$.

[22] It is known that this non-explicit entropy requirement is near-optimal, as the classic extractor impossibility result rules out the existence of extractors for space s sources of min-entropy $k = s - 1$ (since a space s source can sample an arbitrary distribution over 2^s strings).

In order to prove Theorem 3.14, we simply use our new extractor for total-entropy sources (Theorem 3.13), without any further modifications.[23] In order to do so we establish a connection between small-space sources and total-entropy sources. This connection has been observed in prior work (under different terminology) - we emphasize that the new contribution here is the new extractor for total-entropy sources that we plug into this known connection. We formally describe this connection below, and show how it can be used (in conjunction with Theorem 3.13) to obtain our new-and-improved extractors for small-space sources.

A known reduction from small-space sources to total-entropy sources At first glance, small-space sources and total-entropy sources seem quite different. After all, total-entropy sources consist of independent subsources, while small-space sources are certainly not guaranteed to have this property. However, it turns out that small-space sources are a *convex combination* of sources with this property. Indeed, the key insight (which can be traced back to the work of König and Maurer [KM04, KM05]) is the following. Given a small-space source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy k, if you condition on hitting a certain node v in the middle layer of the branching program that generates $\mathbf{X}$, then $\mathbf{X}$ is broken into two independent parts: the bits produced before hitting the middle layer, and the bits produced after hitting the middle layer. Thus, by considering all possible nodes v that could be hit in the middle layer (by the random walk generating $\mathbf{X}$), we get that $\mathbf{X}$ is a convex combination of sources $\{\mathbf{X}'\}$, each consisting of two independent subsources $\mathbf{X}' = (\mathbf{X}'_1, \mathbf{X}'_2) \sim (\{0,1\}^{n/2})^2$. Furthermore, by the entropy chain rule (Lemma 2.1), each $\mathbf{X}'$ has min-entropy roughly $k - s$. Thus, using the inner product extractor, one can extract as long as we have $k - s > n/2$ [KM04, KM05].

Unfortunately, the above technique fails if one hopes to extract from min-entropy $k < n/2$, since all of the min-entropy could be stuck in $\mathbf{X}'_1$ or $\mathbf{X}'_2$, and the classic extractor impossibility result (Fact 1.1) says that it is impossible to extract from such sources. In order to get around this $k = n/2$ barrier, the key idea of Kamp, Rao, Vadhan and Zuckerman [KRVZ11] is to generalize the above argument to create *several* independent subsources, each of length $\ll n/2$ (so that the entropy cannot get stuck in a single subsource). To do this, instead of conditioning on hitting a certain node v in the middle layer of the branching program that generates $\mathbf{X}$, they condition on hitting certain nodes $v_1, \dots, v_r$ across *several* (equally-spaced) layers. Indeed, under such a conditioning, $\mathbf{X}$ becomes a sequence of several independent subsources (of the same length). Furthermore, depending on how many layers are fixed, we lose just a little more entropy than before. This is formalized via the following lemma.

Lemma 3.22 (A reduction from small-space sources to total-entropy sources [KRVZ11, Lemma 3.1, adapted])**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source with min-entropy k. If $r = k/(4s)$ and $\ell = 4ns/k$ are positive integers, then $\mathbf{X}$ is $2^{-k/4}$-close to a convex combination of $(r, \ell, k/2)$-total entropy sources.*

Proof. Let $\mathbf{W} = (\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_n) \sim (\{0,1\}^s)^{n+1}$ be a random variable holding the states reached in layers $0, 1, \dots, n$ of the branching program in the random walk that generates $\mathbf{X}$. Observe that fixing any $\mathbf{W}_i$ breaks $\mathbf{X}$ into two independent sources. More generally, observe that if we define $\mathbf{W}^* := (\mathbf{W}_\ell, \mathbf{W}_{2\ell}, \dots, \mathbf{W}_{r\ell})$, then if we condition $\mathbf{X}$ on any fixing of $\mathbf{W}^*$, it must hold that $\mathbf{X}$ becomes an (r, ℓ, Γ)-total entropy source, for *some* Γ. And by the entropy chain rule (Lemma 2.1), we know that Γ is close to k with high probability. More formally,

$$\Pr_{w \sim \mathbf{W}^*}[H_\infty(\mathbf{X} \mid \mathbf{W}^* = w) \geq k - rs - k/4 = k/2] \geq 1 - 2^{-k/4}.$$

[23] Furthermore, since our extractor for total-entropy sources is simply our new extractor for adversarial sources (Theorem 3.1), we are ultimately using an extractor for adversarial sources (without any further modifications) in order to extract from small-space sources.

In other words, the random variable $(\mathbf{X} \mid \mathbf{W}^* = w)$ is an $(r, \ell, k/2)$-total entropy source with probability at least $1 - 2^{-k/4}$ over $w \sim \mathbf{W}^*$. This completes the proof. □

Given this reduction from small-space sources to total-entropy sources, it is now a straightforward (but tedious) calculation to obtain Theorem 3.14 using our new extractors for total-entropy sources.

Proof of Theorem 3.14. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy at least k. By the reduction from small-space sources to total-entropy sources (Lemma 3.22), we know that $\mathbf{X}$ is $2^{-k/4}$-close to a convex combination of $(k/(4s), 4ns/k, k/2)$-total entropy sources.[24] Now, applying our new extractor for total-entropy sources (Theorem 3.13), we know that we can extract $m \geq (k/2)^\gamma$ bits with error $\varepsilon = 2^{-(k/2)^\gamma} + 2^{-k/4}$, as long as

$$(k/2) \geq C(4ns/k)(k/(4s))^\delta + Cn^{1/2+\delta}.$$

In order for this to hold, it suffices to have both $k \geq 4C(4ns/k)(k/(4s))^\delta$ and $k \geq 4Cn^{1/2+\delta}$. In order for the first requirement to hold, it suffices to have $k \geq 4C^{\frac{1}{2-\delta}} n^{\frac{1}{2-\delta}} s^{\frac{1-\delta}{2-\delta}} = 4s \cdot (Cn/s)^{\frac{1}{2-\delta}}$. And since $\frac{1}{2-\delta} \leq 1/2 + \delta$ for the current setting of $\delta \in (0, 1/2]$, it suffices to have $k \geq 4s \cdot (Cn/s)^{1/2+\delta}$, or rather $k \geq 4Cs \cdot (n/s)^{1/2+\delta} = 4Cn^{1/2+\delta}s^{1/2-\delta}$. Since we are assuming that $s \geq 1$, this also implies $k \geq 4Cn^{1/2+\delta}$. Thus, summarizing the above, we can extract $m \geq (k/2)^\gamma$ bits with error $\varepsilon = 2^{-(k/2)^\gamma} + 2^{-k/4}$, as long as

$$k \geq 4Cn^{1/2+\delta}s^{1/2-\delta}.$$

Resetting constants C, γ appropriately completes the proof. □

Above, we have seen that our new extractors for total-entropy sources immediately yield new-and-improved extractors for small-space sources. And since our extractors for total-entropy sources are simply our new extractors for adversarial sources, we see that our new extractors for adversarial sources immediately yield new-and-improved extractors for small-space sources.

Our new low-error extractors for space s sources improve the entropy requirement from $k = O(n^{1-\gamma})$ to $k = O(n^{1/2+\delta})$ when s is not too big. Previously, we noted (in Remark 3.2) that this is almost the best possible entropy requirement that can be achieved using the above technique. In particular, it is impossible to obtain an entropy requirement of $k < \sqrt{n}$ using a reduction from small-space sources to total-entropy sources - no matter how we set the number r of subsources and the length ℓ of each subsource. To see why, note that since $r\ell = n$, it must be the case that either $\ell > \sqrt{n}$ or $\ell \geq \sqrt{n}$. In the first case, all of the entropy could be trapped in a single source of length $> k$, from which extraction is impossible. In the second case, the application of the entropy chain rule in the proof of Lemma 3.22 would leave the source with 0 bits of entropy, from which extraction is also impossible. In a later chapter (Chapter 7), we will see how to overcome this barrier via a new type of reduction, and construct (higher error) extractors for small-space sources with min-entropy $k \ll \sqrt{n}$.

3.7 Applications in distributed computing

Randomness is vital in distributed computing, enabling a whole host of applications that would not otherwise be possible (Section 1.1). However faulty processors are everywhere in this field, and simply cannot be trusted. This is the perfect playground for adversarial source extractors, which are capable of of producing uniform bits even in such unreliable environments. We highlight one such application, below.

[24] We assume that $k/(4s)$ and $4ns/k$ are positive integers, as it is easy to extend the argument when this is not the case (i.e., using similar tricks to those described in the proof of Theorem 3.13).

3.7.1 Collective coin flipping

In the *collective coin flipping* problem, N parties get together in an effort to produce a common random bit. However, the catch is that not all of them are honest, and some of them may be trying to influence the outcome. Ever since it was first introduced by Ben-Or and Linial in 1985 [BL85] this has become a fundamental problem in not only distributed computing [Dod06], but also pseudorandomness [CZ19]. Indeed, this problem is equivalent to constructing extractors for *non-oblivious bit-fixing (NOBF) sources*, which have played a big role in modern day extractor theory, along with their close cousins, the *oblivious bit-fixing source* [Vaz87b, BBR88, CFG$^+$85] and *symbol-fixing source* [KM04, KZ07]. Our model of adversarial sources generalize all of these - most notably, to a setting where each party may hold some imperfect randomness. Thus, it would be interesting to see if our extractors (or the techniques that go into building them) could be applied to improve existing results on distributed computing with weak randomness [GSV05, KLRZ08, KLR09].

3.8 Applications in cryptography

Just like with distributed computing, cryptography is plagued with the problem of needing randomness, but trusting no one. We believe that our adversarial source extractors may help.

3.8.1 The common reference string (CRS) model

The common reference string (CRS) model is popular setting in which to study cryptography, in which it is assumed that all parties have access to a common random string that is *trusted*. Given such an assumption, one can construct a variety of important cryptographic primitives that would otherwise be impossible in the standard model, including *non-interactive zero-knowledge (NIZK) proofs* [BFM88] and more [CF01]. However, this immediately raises the question of *where this trusted string should come from*. Adversarial source extractors provide one solution: for example, our Theorem 3.3 could be used to generate a reliable random string among a reasonable number of parties, even if the honest players have extremely low min-entropy, there are much more bad players, and the bad players can see the randomness of (a large fraction) of the honest players. It appears that a few works have recently began to study this exact question [GK08, GGJS11, GO14], and it remains an interesting future direction to explore how adversarial source extractors fit into the picture.

3.9 Future directions

Extractors for independent sources are one of the most well-studied primitives in the world of extractors. In the past decade alone, there have been several exciting breakthroughs in constructing these objects [Li15, CZ19, Li19], and just a few months ago, a huge announcement was made [Li23] that we now have explicit two-source extractors (with constant error) that can handle *truly logarithmic entropy* $k \geq O(\log n)$. However, despite these amazing advancements, there has been little attention given to extracting from independent sources in more robust settings. While some existing extractors can handle such settings [RY11, CL16b, CZ16, CL20, CL22], they all require high entropy, or have high error. This is especially problematic, due to the natural applicability of these objects in cryptography.

In this chapter, we launched a systematic study of *adversarial sources*, which make the classical independent source model *more robust* by allowing some of the sources to have no entropy at all. We studied

(N, K, n, k)-adversarial sources, which consist of N sources of length n, where at least K of them have min-entropy at least k. We presented a variety of constructions across a large range of parameters, and exploited new tools from extractor theory and combinatorics (which are built via new connections to communication complexity and coding theory). Since our work was published [CGGL20, CG21], several exciting follow-ups have emerged, which study adversarial source extractors under computational assumptions [KS21], and build on our techniques to construct significantly improved extractors for other classical settings [GSZ21]. Nevertheless, several exciting questions remain, three of which we outline, below.

Adversarial source extractors for fewer good sources Our explicit extractors for adversarial sources can handle $K \geq N^{0.01}$ good sources of min-entropy $k \geq \mathrm{polylog}(n)$, which is a significant improvement to the previous best requirement [KRVZ11] of $K \geq O(\sqrt{N})$ good sources of min-entropy $k \geq n^{0.99}$. *Non-explicitly*, however, we know there exist low-error extractors that can handle just $K = 2$ good sources of min-entropy $k \geq O(\log n)$.[25] Can we come up with explicit constructions that come closer to matching these existential results? Here, one potential approach could be to construct improved *leakage-resilient extractors*.

Unconditional extractors for nontrivial locality In Sections 3.4 and 3.5, we showed that our extractors can be optimized to handle bad sources that *depend on a large number of good sources* - as long as there are *not too many* sources in total, or *very many* sources in total. That is, unlike our main extractor for adversarial sources (Theorem 3.1), our extractors that can handle dependent bad sources only work in certain restricted settings. It is natural to ask whether it is possible to construct an extractor for adversarial sources of locality $d > 0$ that works in *any* setting (i.e., for any number of sources N of any length n). This would be interesting even for sources of locality $d = 1$, especially since we know that higher locality sources can be reduced to sources of locality 1 (Lemma 3.15).

Improved extractors for total-entropy sources Finally, our extractors for adversarial sources combine a host of new objects from extractor theory and combinatorics in order to extract from very few good sources. It would be exciting to see whether these new objects and techniques can be exploited to obtain improved extractors for other settings. One natural setting is the class of *total entropy sources*. While our adversarial source extractors immediately yield improved extractors for these sources (Section 3.6.1), they do so in *black-box* manner. Perhaps our techniques can be carefully tailored to total-entropy sources (in a *white-box* manner) to achieve even better extractors for this classical model.

[25] This can be achieved by exploiting an optimal (non-explicit) two-source extractor, as described in Section 3.3.1.

Chapter 4

Leakage-resilient extractors

In the previous chapter, we initiated a study of *extractors for adversarial sources*, a new class of objects which can be viewed as *robust* extractors for independent sources. In this chapter, we study a different flavor of robust extractors for independent sources, which we call *leakage-resilient extractors* (LREs). Unlike extractors for adversarial sources, LREs once again require that each of the n-bit input sources $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$ contain some min-entropy. Instead, their robustness comes from the following, equally powerful guarantee: while classical independent source extractors only need to extract uniform bits from $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N$, an LRE must extract uniform bits that *remain uniform, even conditioned on the output of multiple leakage functions called over the input sources*.

In this chapter, we construct explicit LREs that dramatically improve the previous state-of-the-art. Prior to our work, the best LREs required each input source to have min-entropy $k \geq 0.99n$, even if each leakage function acts on just $p = 2$ input sources. Recently, Kumar, Meka, and Sahai [KMS19] posed the open problem of reducing the entropy requirement to $k \geq 0.01n$. In our work, we construct LREs that resolve this open problem, and in fact handle *exponentially less* min-entropy $k \geq \mathrm{polylog}(n)$. Furthermore, the bits produced by our LREs remain uniform even against leakage functions that depend on up to $p = N - 2$ of the input sources.

LREs were originally inspired by applications in communication complexity, cryptography, and pseudorandomness, and our constructions have important consequences in each of these areas. In communication complexity, we obtain strong average-case lower bounds against a rich spectrum of multiparty communication protocols that interpolate between the well-studied *number-in-hand* (NIH) and *number-on-forehead* protocols. One of our lower bounds (against a slightly weaker definition of NOF protocols) is strong enough that, if proven in the standard NOF setting, would yield breakthrough circuit lower bounds. In cryptography, our LREs yield leakage-resilient secret sharing schemes that are exponentially better than the previous state-of-the-art. In pseudorandomness, our LREs act as a critical ingredient in our new extractors for adversarial sources.

The results presented in this chapter are based on the following joint work:

- [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 *61st Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1226–1242. IEEE, 2020

4.1 Introduction

As we have seen, the *independent source* model is an extremely well-studied model in randomness extraction, but it is not entirely robust. In the previous chapter, we explored one way to address this, via *extractors for adversarial sources*. In this chapter, we study a different flavor of *robust* extractors for independent sources, which we call *leakage-resilient extractors*.

Definition 4.1 (Leakage-resilient extractor for independent sources)**.** *Let $\mathcal{F}$ be a family of functions of the form $f : (\{0,1\}^n)^N \to \{0,1\}^\mu$. A function* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *is called an $\mathcal{F}$-*leakage-resilient extractor for independent sources *of min-entropy k with error ε if the following holds. For any N independent sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N \sim \{0,1\}^n$, each with min-entropy at least k, and for any $f \in \mathcal{F}$,*

$$|\mathsf{Ext}(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N) \circ f(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N) - \mathbf{U}_m \circ f(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)| \leq \varepsilon.$$

In order to digest this definition, we can compare it against classical extractors for independent sources (Definition 3.1) and extractors for adversarial sources (Definition 3.3). While adversarial source extractors make the independent source setting more robust by allowing some of the sources to be *missing entropy*, leakage-resilient extractors once again require each source to have some min-entropy guarantee. On the other hand, leakage-resilient extractors offer a new robustness guarantee (that adversarial source extractors do not): the output of a leakage-resilient extractor must look uniform and *remain uniform* even conditioned on some *leaked information* $f \in \mathcal{F}$. As we will see, this simple extension of the classical extractor definition will ultimately yield powerful applications in cryptography, complexity theory, and more.

To begin our study of leakage-resilient extractors, the first question that clearly demands attention is, *What should we select as our family $\mathcal{F}$ of leakage functions?* One idea is to consider functions that leak a few bits of information about each source: that is, where each leakage function $f \in \mathcal{F}$ is of the form $f(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N) = (f_1(\mathbf{X}_1), f_2(\mathbf{X}_2), \dots, f_N(\mathbf{X}_N))$. As it turns out, this setting can easily be handled by classical extractors for independent sources.[1] A much more interesting question, then, is whether we can construct leakage-resilient extractors for the setting where each leaked bit f_i may *act on multiple sources* $\mathbf{X}_{i_1}, \mathbf{X}_{i_2}, \dots, \mathbf{X}_{i_p}$. Such leaks induce correlations between the original sources, making classical independent source extractors fail.

In this chapter, we construct leakage-resilient extractors that can successfully handle these challenging leaks and more. Before we dive into these constructions, we visit the land of *communication complexity*, which is home to the original motivation for these leakage-resilient extractors.

Communication complexity

In communication complexity, the goal is to understand the amount of *communication* required to compute some Boolean function $f : (\{0,1\}^n)^N \to \{0,1\}$. This field was initiated by a seminal paper of Yao [Yao79], who considered the *two-party setting*. In this setting, there are $N = 2$ parties, each holding n bits of input. Each party can only see the input they are holding, and they must work together to determine the value of f (by communicating as few bits as possible). The question is, *how many bits must be communicated?*[2]

[1] This follows via a standard application of the *min-entropy chain rule* (see Chapter 2).

[2] Note that a trivial protocol would have one party send over their entire input, which the other party can then use (combined with their own input) to determine the value of f. This trivial protocol requires n bits of communication, and the goal is to design protocols that require much fewer bits of communication $\ll n$ (or prove they do not exist). For example, if f is the *parity* function, then the first party only needs to send 1 bit of information (the parity of their input) to the second party before the second party knows the value of f.

Since its introduction, communication complexity has blossomed into a central area of complexity theory, with deep connections to many other fields.[3] While the two-party setting remains the most classical, the *multiparty* setting has received a growing amount of attention over the past few decades. In this more general setting, there can be an *arbitrary* number of parties, N, each holding n bits of input. They once again wish to determine the value of $f : (\{0,1\}^n)^N \to \{0,1\}$, communicating as few bits as possible. In the multiparty setting, however, it is less clear *how* the parties will work together to do so.[4] Historically, two different models have been studied: (1) *number-in-hand protocols*, and (2) *number-on-forehead protocols*.

Number-in-hand protocols In a number-in-hand (NIH) protocol [DF92], each party can only see the n bits of input in their hand. In order to compute the value of $f : (\{0,1\}^n)^N \to \{0,1\}$, the parties have access to a *public blackboard*, which they may use to communicate. They take turns writing on the blackboard (some information about the input they are holding), until every party knows the value of $f : (\{0,1\}^n)^N \to \{0,1\}$. The *NIH communication complexity* of f, denoted $\mathsf{CC}^{\mathsf{NIH}}(f)$, is then defined as the number of bits that must be communicated by any such protocol (in the worst case over all inputs).

Number-on-forehead protocols In a number-on-forehead (NOF) protocol [CFL83], each party can see all of the input to f *except* the n bits in their hand (metaphorically, these n bits are actually written *on their forehead*). Once again, the parties have access to a public blackboard, which they use to communicate. Every turn, one party writes on the blackboard (some information about the input they are *not* holding), until everyone knows the value of $f : (\{0,1\}^n)^N \to \{0,1\}$.[5] The *NOF communication complexity* of f, denoted $\mathsf{CC}^{\mathsf{NOF}}(f)$, is then defined analogously to $\mathsf{CC}^{\mathsf{NIH}}(f)$. Because each party can see much more of the input in NOF protocols than in NIH protocols, NOF protocols are much more powerful and we always have

$$\mathsf{CC}^{\mathsf{NOF}}(f) \leq \mathsf{CC}^{\mathsf{NIH}}(f).$$

Lower bounds and applications Multiparty communication protocols offer an attractive model in which to pursue lower bounds, for two reasons. First, these protocols are simple enough to reason about combinatorially: if we write down a Boolean function $f : (\{0,1\}^n)^N \to \{-1,1\}$ in the cells of a multi-dimensional matrix M_f, we can get lower bounds on $\mathsf{CC}^{\mathsf{NIH}}(f)$ and $\mathsf{CC}^{\mathsf{NOF}}(f)$ by upper bounding the *discrepancy* of certain well-structured subsets of M_f. Second, these protocols appear to be rich enough to capture seemingly unrelated models of computation. As a result, NIH lower bounds imply lower bounds against streaming algorithms [AMS99], while NOF lower bounds imply lower bounds in proof complexity [BPS07] and circuit complexity [All89, Yao90, HG91, RW93, BT94].[6] Lower bounds against these protocols also find great applicability in settings where hardness is considered "good" (like cryptography) [KMS19].

Bounded collusion protocols Given the beautiful connections above, it is natural to wonder whether NIH and NOF protocols can be placed into a broader family of protocols, whose exploration might offer even more insight into these applications and lower bounds. In a recent work, Kumar, Meka, and Sahai introduced exactly such a family, which they called *bounded collusion protocols* (BCPs) [KMS19].

Like NIH and NOF protocols, BCPs are multiparty communication protocols that seek to compute a function $f : (\{0,1\}^n)^N \to \{0,1\}$ whose input is split among N parties. Each party once again holds n bits

[3] For an overview of the area, we recommend the excellent textbooks of Kushilevitz, Nisan [KN96], Rao and Yehudayoff [RY20].

[4] In the two-party setting, the two parties simply take turns sending 1 bit of information about their input to the other party.

[5] An equivalent way to define NOF protocols is as following: each party can only see the n bits of input in their hand, and every turn, some $N-1$ parties get together to write something on the blackboard, which *depends on all* of their $N-1$ pieces of input.

[6] For more connections, we refer the reader to the excellent survey of Lee and Shraibman [LS09].

of input, and they communicate via a public blackboard. Unlike NIH and NOF protocols, however, BCPs come equipped with an additional parameter p, known as the *collusion bound*. In every round of the BCP, p parties get together to write a bit on the blackboard, using all of the input in their possession. We identify the BCP with all of the bits written on the blackboard over the course of the protocol, and formally define them as follows.

Definition 4.2 (Bounded collusion protocol (BCP))**.** *A function $g : (\{0,1\}^n)^N \to \{0,1\}^\mu$ is called a* bounded collusion protocol (BCP) with collusion bound p *if each output bit $y_i := g(X_1, X_2, \dots, X_N)_i$ is a deterministic function of the previous output bits $y_1, \dots, y_{i-1}$ and some p inputs $X_{i_1}, \dots, X_{i_p}$.*[7]

BCPs define a rich spectrum of communication protocols, which is induced by the collusion bound p (and gets more powerful as p increases). In fact, NIH and NOF protocols are precisely the endpoints of this spectrum: at $p = 1$ and $p = N - 1$, respectively. It is therefore natural to ask about the *communication complexity* of a function $f : (\{0,1\}^n)^N \to \{0,1\}$ with respect to BCPs (of collusion bound p), which we can simply define as the length μ of the shortest BCP (of collusion bound p) that computes f.[8] If we denote this quantity as $\mathsf{CC}^p(f)$, we get a vivid illustration of the spectrum cut out by BCPs:

$$\mathsf{CC}^{\mathsf{NOF}}(f) = \mathsf{CC}^{N-1}(f) \leq \mathsf{CC}^{N-2}(f) \leq \cdots \leq \mathsf{CC}^2(f) \leq \mathsf{CC}^1(f) = \mathsf{CC}^{\mathsf{NIH}}(f).$$

Thus, BCPs offer a natural generalization of the classical NIH and NOF models. If we can obtain a better understanding of BCPs, then we can also obtain a better understanding of these well-studied settings. This may not only help us make progress on longstanding barriers in these areas, but could also help us build new connections from communication complexity to other areas of theoretical computer science. Despite all of this,

BCPs have never before been studied as a first-class object.

Indeed, in the work that originally introduced these protocols [KMS19], BCPs are motivated and studied from the context of *cryptography*, and many fundamental questions are left unanswered. In this chapter, we seek to launch the first systematic study of these objects, and answer several of these key questions. We provide an overview of these key questions, next.

Key questions

Lower bounds against BCPs Just like any subfield of complexity theory, a central goal in communication complexity is to obtain *explicit lower bounds*. In the NIH model, it is relatively straightforward to come up with an explicit (poly-time computable) function $f : (\{0,1\}^n)^N \to \{0,1\}$ such that $\mathsf{CC}^{\mathsf{NIH}}(f) \geq n$.[9] On the other hand, getting strong explicit lower bounds against NOF protocols is much more difficult: here, the best known results are of the form $\mathsf{CC}^{\mathsf{NOF}}(f) \geq \Omega(n/2^N)$ [BNS92]. Given this, it is natural to wonder how explicit lower bounds against BCPs (and the difficulty of proving them) evolve from the NIH side of the spectrum to the NOF side of the spectrum. This leads to our first key question:

Question 4.1. *Can we get explicit lower bounds against BCPs that exhibit a collusion-complexity tradeoff?*

[7] Note that the indices $i_1, \dots, i_p$ may depend on $y_1, \dots, y_{i-1}$.

[8] Formally, we say that a BCP $g : (\{0,1\}^n)^N \to \{0,1\}^\mu$ *computes* $f : (\{0,1\}^n)^N \to \{0,1\}$ if $g(x)_\mu = f(x)$ for all x.

[9] Note that any lower bound in the *two-party* setting also applies to NIH protocols with $N > 2$ parties, as increasing the number of parties in the NIH setting will decrease the fraction of total input held by each party. Thus the lower bound $\mathsf{CC}^{\mathsf{NIH}}(f) \geq n$ (and, in fact, an even lower bound) follows from classical results in the two-party setting [Yao79].

It is especially interesting to understand if we can obtain (nontrivial) explicit lower bounds in the setting $N \gg \log n$, as no such lower bounds are known for NOF protocols. In fact, overcoming this "logarithmic barrier" is a longstanding challenge in complexity theory.[10] In our first main result (Theorem 4.1), we show that in the setting $N \gg \log n$, if we slightly reduce the collusion bound from $p = N - 1$ (NOF protocols) to $p = 0.99N$, we can immediately move from the trivial lower bound $\mathsf{CC}^{\mathsf{NOF}}(f) \geq \Omega(1)$ to extremely strong lower bounds of the form $\mathsf{CC}^p(f) \geq n^{\Omega(1)}$. In fact, we show that these lower bounds *always apply* against such BCPs with $p = 0.99N$, *regardless of the relationship between N and n.*

Leakage-resilient extractors against BCPs If we can successfully answer Question 4.1 and construct explicit *worst-case* lower bounds against BCPs, it is natural to ask whether we can strengthen these results and obtain explicit *average-case* lower bounds against BCPs. In particular, we would like to explore whether it is possible to explicitly construct a function f such that every BCP requires a large amount of communication to compute its value, even on just *slightly more than half* of all possible inputs.[11] As it turns out, this is precisely the same task as constructing an explicit *leakage-resilient extractor* (Definition 4.1) for min-entropy $k = n$ and output length $m = 1$ against the family $\mathcal{F}$ of BCPs.[12]

Given this extractor-based characterization of average-case lower bounds against BCPs, it is natural to wonder whether we can also construct leakage-resilient extractors (LREs) against BCPs that work for min-entropy $k \ll n$. This question was asked in the original paper of Kumar, Meka, and Sahai [KMS19], who introduced these objects (LREs against BCPs) under the name *extractors for cylinder intersections*. They posed the following as an open question.

Question 4.2. *Can we construct leakage-resilient extractors against BCPs with collusion $p = 2$, which work for min-entropy $k = 0.01n$?*

In our second main result (Theorem 4.2), we answer this question in the affirmative, and in fact construct LREs that work for *exponentially less* min-entropy $k \geq \mathrm{polylog}(n)$. Furthermore, our LREs can even handle BCPs with up to $p = N-2$ collusion, and more generally achieve a "leakage-collusion" tradeoff that mirrors the complexity-collusion tradeoff in our answer to Question 4.1. These LREs against BCPs (which work for low entropy) not only act as even stronger average-case lower bounds against BCPs, but also have important applications in the world of extractors, as well. Indeed, as we have seen, they serve as *robust* versions of classical extractors for independent sources (Section 4.1), and have applications in constructing extractors for adversarial sources (Section 3.3).

Leakage-resilient secret sharing against BCPs For our final main question, we visit the original motivating application considered by Kumar, Meka, and Sahai [KMS19] in their introduction of BCPs: *secret sharing schemes*. Secret sharing schemes are a fundamental cryptographic primitive, originally introduced in the seminal works of Blakley [Bla79] and Shamir [Sha79]. These schemes capture the natural setting of a central authority who wishes to share some secret (e.g., missile launch codes) among a group of N somewhat trusted individuals. In the simplest model, the goal is to allocate a portion (or *share*) of the secret to each individual so that if they *all* work together, they can reconstruct the secret, but if *any fewer than N*

[10] Indeed, it has been shown that any significant improvements to the best NOF lower bounds would yield a breakthrough in circuit complexity, by providing new lower bounds against the circuit class ACC^0 [Yao90, RW93, BT94].

[11] Note that there is always a trivial BCP that computes f on at least half of all possible inputs: the BCP that always outputs 0, or the BCP that always outputs 1.

[12] This is a straightforward exercise that follows from the definition of statistical distance.

of them work together, they cannot recover any information about the secret. These schemes are known as *N-out-of-N secret sharing schemes*, and are the most basic building block in this subarea of cryptography.[13]

Kumar, Meka, and Sahai [KMS19] study a powerful twist on classical secret sharing, known as *leakage-resilient secret sharing* (LRSS). LRSS schemes can be viewed as more *robust* versions of classical secret sharing schemes, and have recently received a lot of attention [DP07, DDV10, GK18a, GK18b, BDIR21, ADN$^+$19, SV19, KMS19, NS20, LCG$^+$20].[14] In additional to the classical guarantees offered by a traditional N-out-of-N scheme (i.e., secrecy against some $N-1$ parties working together), a *leakage-resilient* N-out-of-N scheme further guarantees that the secret will remain hidden even if some amount of information is leaked by *all* of the N parties. Kumar, Meka, and Sahai consider *bounded collusion protocols* as their model of leakage, in an effort to generalize and strengthen several previous models in leakage-resilient secret sharing.

In their work [KMS19], Kumar, Meka, and Sahai successfully construct LRSS schemes against BCPs, by leveraging tools from communication complexity. Indeed, by applying explicit functions with high NOF communication complexity, they are able to construct efficient LRSS schemes against BCPs with collusion $p = O(\log N)$.[15] However, they are unable to tolerate collusion $p = \omega(\log N)$, largely due to their application of NOF lower bounds in a *black-box* manner. This begs the question of whether stronger explicit lower bounds, which apply directly to BCPs (i.e., do not come from a black-box application of NOF lower bounds), can be exploited to overcome this "logarithmic barrier." As luck has it, we have explicit bounds of exactly this nature (i.e., in our answer to Question 4.1). This motivates our third and final question.

Question 4.3. *Can we use our new explicit lower bounds against BCPs to construct efficient LRSS schemes against BCPs with collusion $p = \omega(\log N)$?*

In our third main result (Theorem 4.3), we answer this question in the affirmative, and in fact construct efficient LRSS schemes that can even handle BCPs with *exponentially higher* collusion $p = 0.99N$. With our *three key questions* about BCPs in mind, we now proceed to formally state our main theorems.

4.2 Our results

In this chapter, we construct a new flavor of robust extractors for independent sources, which we call *leakage-resilient extractors*. As we have seen, LREs are extremely powerful objects, capable of extracting uniform bits that *remain uniform*, even conditioned on the output of multiple leakage functions. Here, we consider a leakage model where each leaked bit depends on a bounded number p of the input sources.

In fact, we actually consider an even more general model, known as *bounded collusion protocols* (BCPs). As discussed, BCPs are a rich suite of communication protocols that interpolate between the classical NIH and NOF settings. But despite their natural definition, they have never before been studied as a first-class object. Thus, we seek to initiate here a systematic study of these well-motivated protocols, by constructing *three key objects* related to BCPs - and indeed, LREs against BCPs constitute just one of these constructions.

[13] A large variety of more general secret sharing schemes have been studied, such as *t-out-of-N schemes* and *schemes for general access structures* (see, e.g., the survey [Bei11]). However, in many settings, classical N-out-of-N schemes can be "compiled" into these more general schemes by exploiting now-standard machinery from the secret sharing literature.

[14] More generally, leakage-resilient secret sharing fits into the long line of work on *leakage-resilience* in cryptography [KR19].

[15] A secret sharing scheme over N parties is said to be *efficient* if each party receives a share of length $\mathrm{poly}(N)$.

Lower bounds against BCPs

In our first main theorem, we establish explicit lower bounds against the entire spectrum of BCPs. These lower bounds exhibit a *collusion-complexity tradeoff*, answering Question 4.1.

Theorem 4.1 (Lower bounds against BCPs)**.** *There is a universal constant $c > 0$ such that for all $N, n \in \mathbb{N}$ and $p \leq N - 1$, there exists an explicit function $f : (\{0,1\}^n)^N \to \{0,1\}$ with*

$$\mathsf{CC}^p(f) \geq c \cdot n^{\frac{\log(N/p)}{\log(N/p)+1}}.$$

As can be seen, as the collusion bound p decreases, the lower bound in Theorem 4.1 increases. Most notably, however, even if the setting of BCPs with very high collusion $p = 0.99N$, we obtain a strong lower bound of the form $\mathsf{CC}^p(f) \geq n^{\Omega(1)}$. Previously, the best known result [KMS19] followed immediately from (black-box) lower bounds against NOF protocols [BNS92], and was of the form $\mathsf{CC}^p(f) \geq \Omega(n/2^p)$.[16] Thus, all previous bounds against $p = 0.99N$ collusion were of the form $\mathsf{CC}^p(f) \geq \Omega(n/2^{0.99N})$ and thus become trivial when $N > 1.1 \log n$, whereas our bounds remain polynomially large *for all settings of* N, n.

Leakage-resilient extractors against BCPs

In our second main theorem, we significantly strengthen our explicit lower bounds to *average-case* lower bounds, and in fact strengthen these even further to produce leakage-resilient extractors against BCPs for *polylogarithmic entropy*. Furthermore, our extractors exhibit a collusion-leakage tradeoff that mirrors the collusion-complexity tradeoff in our worst-case lower bounds (Theorem 4.1). We record our result below, which answers Question 4.2 and resolves an open problem of Kumar, Meka, and Sahai [KMS19].

Theorem 4.2 (Leakage-resilient extractors against BCPs)**.** *There exist universal constants $C, \gamma > 0$ such that for all $N, n, p \in \mathbb{N}$ with $N \geq 3$ and $p \leq N - 2$, the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for independent sources with min-entropy $k \geq \log^C n$, which is leakage-resilient against BCPs of collusion p and length μ, and has output length $m \geq \mu$ and error $\varepsilon = 2^{-\mu}$, where*

$$\mu := k^{\frac{\gamma \log(N/p)}{\log(N/p)+1}}.$$

Notably, Theorem 4.2 provides an explicit extractor for $k \geq \mathrm{polylog}(n)$ entropy that can handle $\mu = k^{\Omega(1)}$ bits of leakage from BCPs with collusion $p = 0.99N$. This is an *exponential improvement* over the previous best result [BNS92, KMS19], which required min-entropy $k \geq 0.99n$, even for BCPs of collusion $p = O(1)$. Our leakage-resilient extractor against BCPs is a powerful new primitive in extractor theory, and is the strongest object that we construct in this chapter. Indeed, all other results can be viewed as an *application* of this construction.

Leakage-resilient secret sharing against BCPs

In our third and final main theorem, we exploit the new tools we have developed thus far to construct significantly improved leakage-resilient secret sharing schemes, answering Question 4.3.

Theorem 4.3 (Leakage-resilient secret sharing against BCPs)**.** *There exists an efficient N-out-of-N secret sharing scheme that is leakage-resilient against BCPs of collusion $p = 0.99N$ and length $\mu = \mathrm{poly}(N)$.*

[16] We note that our explicit function in Theorem 4.1 can also achieve this bound, which is slightly better when, e.g., $p = O(1)$.

This is an *exponential improvement* over the previous best result [KMS19], which could only handle BCPs of collusion $p = O(\log N)$. Our new secret sharing schemes crucially rely on our new LREs against BCPs (Theorem 4.2), which have exponentially low error and can be viewed as strong average-case lower bounds. Interestingly, though, this application only requires LREs for *full min-entropy* $k = n$, while our LREs can actually tolerate exponentially less min-entropy $k \geq \operatorname{polylog}(n)$. It is therefore natural to ask whether these low-entropy LREs can help us construct even better secret sharing schemes. In Section 4.5, we exhibit exactly such a setting, via a new (more general) flavor of secret sharing called *seeded secret sharing*.

Organization

We organize this chapter around our new leakage-resilient extractors, as they are the most powerful object that we construct. In Section 4.3, we construct our new LREs against BCPs, by starting with a simple (but specific) function that is hard for NOF protocols, and gradually adding features until we arrive at our full-blown leakage-resilient extractor (Theorem 4.2). Next, in Section 4.4, we show how these new LREs immediately give explicit lower bounds against the entire spectrum of BCPs (Theorem 4.1). Then, in Section 4.5, we apply our new LREs to get exponentially stronger leakage-resilient secret sharing schemes (Theorem 4.3). As a third application of our LREs, we remind the reader in Section 4.6 of the crucial role they play in the construction of our extractors for adversarial sources (from Chapter 3). Finally, we conclude with some open problems in Section 4.7.

4.3 The extractor

We are now ready to begin the technical portion of the chapter. In this section, we obtain our leakage-resilient extractors against BCPs (Theorem 4.2), from which all our other main results will precipitate.

Theorem 4.4 (Leakage-resilient extractors against BCPs - Theorem 4.2, restated)**.** *There exist universal constants $C, \gamma > 0$ such that for all $N, n, p \in \mathbb{N}$ with $N \geq 3$ and $p \leq N - 2$, the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for independent sources with min-entropy $k \geq \log^C n$, which is leakage-resilient against BCPs of collusion p and length μ, and has output length $m \geq \mu$ and error $\varepsilon = 2^{-\mu}$, where*

$$\mu := k^{\frac{\gamma \log(N/p)}{\log(N/p)+1}}.$$

As discussed, the high-level idea is start with a function that is hard for NOF protocols, and gradually equip it with more features until we arrive at Theorem 4.4. But before we do so, it will be helpful to introduce a little bit of notation. Towards this end, we let $\mathsf{BCP}(p, \mu)$ denote the collection of all BCPs of the form $g : (\{0,1\}^n)^N \to \{0,1\}^\mu$ with collusion bound p, as defined by Definition 4.2. Additionally, we let $\mathsf{nBCP}(p, \mu)$ contain all functions of the form $g : (\{0,1\}^n)^N \to \{0,1\}^\mu$ where each output bit $y_i := g(X_1, X_2, \dots, X_N)_i$ is a deterministic function of some p inputs $X_{i_1}, \dots, X_{i_p}$ and notably does *not* depend on the previous output bits $y_1, \dots, y_{i-1}$. In other words, $\mathsf{nBCP}(p, \mu)$ contains the class of *non-adaptive* bounded collusion protocols. Finally, we let $\mathsf{NOF}(p, \mu)$ and $\mathsf{nNOF}(p, \mu)$ refer to the corresponding classes of *number-on-forehead* protocols. With these definitions in place, we're ready to get started.

4.3.1 The NOF-hard function

The NOF-hard function that will form the foundation of our leakage-resilient extractors is the *finite field multiplication function* $\mathsf{FFM}_N : (\{0,1\}^n)^N \to \{0,1\}$. This function takes as input N bitstrings of length n, treats them as elements of $\mathbb{F}_{2^n}$, takes their product over this field, interprets the result again as a bitstring over $\{0,1\}^n$, and outputs the first bit. In [FG13], Ford and Gál demonstrate strong average-case lower bounds against this function, of the form $\Omega(n/2^N)$. However, as we would ultimately like to output many bits, we must slightly modify their construction so it can do so (while maintaining some notion of average-case hardness). Towards this end, we will define the *product extractor* to be equivalent to the finite field multiplication function, except that it will output the first m bits instead of just the first bit. More formally, for $m \leq n$, we let $\sigma_{n,m} : \mathbb{F}_{2^n} \to \mathbb{F}_{2^m}$ denote the function that interprets its input as an element of $\mathbb{F}_2^n$ and projects it onto its first m coordinates, and we define a multi-bit output version of FFM_N as follows.

Definition 4.3. *For any $N, n, m \in \mathbb{N}$ with $n \geq m$, the* product extractor $\mathsf{prodExt} : (\{0,1\}^n)^N \to \{0,1\}^m$ *is defined as:*

$$\mathsf{prodExt}(x_1, x_2, \ldots, x_N) := \sigma_{n,m}(x_1 \cdot x_2 \cdot \cdots \cdot x_N),$$

where the input/output are interpreted as elements of $\mathbb{F}_{2^n}$, and the product is taken over this field.

We remark that, using the standard encoding of $\mathbb{F}_{2^n}$, it is straightforward to perform all the above operations in $\mathrm{poly}(n, N)$ time (see, e.g., [Vad12]), and thus this is an efficiently computable function. Now, the main goal of this section is to prove that the product extractor define above is a leakage-resilient extractor for full min-entropy against *non-adaptive* number-on-forehead protocols. In other words, in some sense, we want to show that it is a "multi-output" average-case hard function. In particular we will prove the following.

Lemma 4.1 (The NOF-hard function)**.** *There is a universal constant $c > 0$ such that for all $N, n \in \mathbb{N}$ such that $N \geq 2$, the product extractor* $\mathsf{prodExt} : (\{0,1\}^n)^N \to \{0,1\}^m$ *from Definition 4.3 is an explicit leakage-resilient extractor against* $\mathsf{nNOF}(\mu)$ *for entropy $k = n$ and leakage $\mu = \xi$, output length $m \geq \xi$, and error $\varepsilon = 2^{-\xi}$, where*

$$\xi = cn/2^N.$$

A natural to prove this result might simply apply the correlation bounds of Ford and Gál in a black-box manner. However, this is not possible, since their correlation bounds correspond to just the first bit produced by the finite field multiplication function. Instead, we will need to dig into their proof a little bit. Towards this end, we must first give a brief reminder that a *homomorphism* between groups $(G, +)$ and $(H, \cdot)$ is just a function $\phi : G \to H$ such that $\phi(u + v) = \phi(u) \cdot \phi(v)$ for all $u, v \in G$. Then, a *character* of G is just defined as a homomorphism into the (multiplicative group of the) complex numbers, namely $\psi : G \to \mathbb{C}^\times$. Finally, we say that ψ is nontrivial if it is not the constant 1. Now, the reason why characters are useful is because if you have a random variable $\mathbf{X} \sim G$, and the expectation of $\psi(\mathbf{X})$ is small (in absolute value) for every nontrivial character, then it follows that $\mathbf{X}$ is close to uniform. Such a claim is known as an XOR Lemma, and we record such a result from Rao, below.

Lemma 4.2 (XOR Lemma [Rao07])**.** *Let $\mathbf{X}$ be a random variable over a finite abelian group, G. Then if $\mathbf{U}$ is the uniform random variable over G and Ψ is the collection of all nontrivial characters $\psi : G \to \mathbb{C}^\times$,*

$$|\mathbf{X} - \mathbf{U}| \leq \sqrt{|G|} \cdot \max_{\psi \in \Psi} |\mathbb{E}[\psi(\mathbf{X})]|.$$

Given this XOR lemma, it is actually easy to extend the result of Ford and Gál to an extractor that outputs multiple bits. The reason for this is because they actually proved a bound on the appropriate expectations (also referred to as *character sums*) in their paper! In particular, they showed the following.

Lemma 4.3 ([FG13, Theorems 4.7 and 4.11])**.** *For any $N, n, \mu \in \mathbb{N}$ such that $N \geq 2$, any $\mathsf{Leak} : (\{0,1\}^n)^N \to \{0,1\}^\mu \in \mathsf{nNOF}(\mu)$, any $y \in \{0,1\}^\mu$, and any nontrivial character $\chi : \mathbb{F}_{2^n} \to \mathbb{C}^\times$,*

$$\left| \sum_{x \in (\{0,1\}^n)^N : \mathsf{Leak}(x)=y} \chi(x_1 \cdot x_2 \cdot \dots \cdot x_N) \right| \leq 2^{Nn+1-n/2^{N-1}}.$$

By combining this character sum with Rao's XOR lemma, it is straightforward to obtain Lemma 4.1.

Proof of Lemma 4.1. We must show that there is some $\xi = \Omega(n/2^N)$ such that for all $N, n \in \mathbb{N}$ such that $N \geq 2$, and any $\mathsf{Leak} \in \mathsf{nNOF}(\mu)$ with $\mu \leq \xi$,

$$\Delta := |\mathsf{prodExt}(\mathbf{U}_{Nn}) \circ \mathsf{Leak}(\mathbf{U}_{Nn}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{U}_{Nn})| \leq 2^{-\xi}.$$

We start by observing that

$$\begin{aligned} \Delta &= \mathbb{E}_{\mathsf{Leak}(\mathbf{U}_{Nn})}[|\mathsf{prodExt}(\mathbf{U}_{Nn}) - \mathbf{U}_m|] \\ &= \sum_y \Pr[\mathsf{Leak}(\mathbf{U}_{Nn}) = y] \cdot |\mathsf{prodExt}(\mathbf{U}_{Nn} \mid \mathsf{Leak}(\mathbf{U}_{Nn}) = y) - \mathbf{U}_m| \\ &\leq 2^\mu \max_{y \in \{0,1\}^\mu} \Pr[\mathsf{Leak}(\mathbf{U}_{Nn}) = y] \cdot |\mathsf{prodExt}(\mathbf{U}_{Nn} \mid \mathsf{Leak}(\mathbf{U}_{Nn}) = y) - \mathbf{U}_m|, \end{aligned}$$

where y runs over $\{0,1\}^\mu$. Next, we can applying Lemma 4.2 to upper bound the above by

$$2^\mu \max_{y,\psi} \Pr[\mathsf{Leak}(\mathbf{U}_{Nn} = y)] \cdot |\mathbb{E}[\psi(\mathsf{prodExt}(\mathbf{U}_{Nn} \mid \mathsf{Leak}(\mathbf{U}_{Nn}) = y))]| \cdot 2^{m/2},$$

where ψ runs over all nontrivial characters of the additive group $\mathbb{F}_{2^m}$. Plugging in the definition of this expectation, we get:

$$2^{\mu+m/2} \max_{y,\psi} \Pr[\mathsf{Leak}(\mathbf{U}_{Nn}) = y] \cdot \left| \sum_{x \in (\{0,1\}^n)^N : \mathsf{Leak}(x)=y} \frac{\Pr[\mathbf{U}_{Nn} = x]}{\Pr[\mathsf{Leak}(\mathbf{U}_{Nn}) = y]} \cdot \psi(\mathsf{prodExt}(x)) \right|,$$

which clearly equals

$$2^{\mu+m/2-Nn} \cdot \max_{y,\psi} \left| \sum_{x \in (\{0,1\}^n)^N : \mathsf{Leak}(x)=y} \psi(\mathsf{prodExt}(x)) \right|.$$

Plugging in the definition of prodExt, we get:

$$2^{\mu+m/2-Nn} \cdot \max_{y,\psi} \left| \sum_{x \in (\{0,1\}^n)^N : \mathsf{Leak}(x)=y} \psi(\sigma_{n,m}(x_1 \cdot x_2 \cdots x_N)) \right|.$$

It is straightforward to show that $\sigma_{n,m}$, as defined earlier, is a surjective homomorphism between the additive groups $\mathbb{F}_{2^n}, \mathbb{F}_{2^m}$. Thus, for any nontrivial character $\psi : \mathbb{F}_{2^m} \to \mathbb{C}^\times$, the composition $\psi(\sigma_{n,m}) : \mathbb{F}_{2^n} \to \mathbb{C}^\times$ is a nontrivial character of the additive group $\mathbb{F}_{2^n}$. Thus, we can upper bound the above quantity by

$$2^{\mu+m/2-Nn} \cdot \max_{y,\chi} \left| \sum_{x \in (\{0,1\}^n)^N : \mathsf{Leak}(x)=y} \chi(x_1 \cdot x_2 \cdots x_N) \right|,$$

where χ runs over all nontrivial characters of $\mathbb{F}_{2^n}$. We note that the general technique seen above (defining a surjective homomorphism $\sigma_{n,m}$ in order to use an XOR lemma over a smaller domain in order to incur a smaller blow-up in error) is borrowed from [Rao07]. Finally, using the character sum from Lemma 4.3, we can upper bound the above by

$$2^{\mu+m/2+1-n/2^{N-1}}.$$

Thus, all that remains is to show $2^{\mu+m/2+1-n/2^{N-1}} \leq 2^{-\xi}$, for some $\xi = \Omega(n/2^N)$. We recall that $\mu \leq \xi$ and $m \leq \xi$, and note that we may assume $\xi \geq 1$, because otherwise $\mathsf{prodExt}$ has no output and is trivially a leakage-resilient extractor. Thus, the above bound indeed holds for, say, $\xi = 0.5n/2^N$, as desired. □

4.3.2 Handling more leakage under less collusion via basic combinatorics

Next, we show that without any further modifications, $\mathsf{prodExt}$ can handle leakage from BCPs across a very general range of parameters, and furthermore achieve a nontrivial tradeoff between leakage (complexity) and collusion. In particular, we prove the following lemma, and note that we optimize to pick a good setting for t in Section 4.3.4.

Lemma 4.4. *There exists a universal constant $c > 0$ such that the following holds. For all sufficiently large $N, n \in \mathbb{N}$ and any $t, p \in \mathbb{N}$ such that $t \leq N$ and $p \leq N - 1$, the product extractor $\mathsf{prodExt} : (\{0,1\}^n)^N \to \{0,1\}^m$ from Definition 4.3 is an explicit leakage-resilient extractor against $\mathsf{nBCP}(p, \mu)$ for entropy $k = n$ and leakage $\mu < \min\{\xi, \binom{N}{t}/\binom{p}{t}\}$, with output length $m \geq \xi$ and error $\varepsilon = 2^{t-\xi}$, where $\xi = cn/2^t$.*

Proof. We must show that for N independent uniform sources $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$ and any $\mathsf{Leak} \in \mathsf{nBCP}(p, \mu)$ with $\mu < \xi$ and $\mu\binom{p}{t} < \binom{N}{t}$,

$$|\mathsf{prodExt}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})| \leq \varepsilon,$$

where $\varepsilon = 2^{-\xi}$. For brevity, we let $\mathbf{Z}_1 := \mathsf{prodExt}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X})$ and $\mathbf{Z}_2 := \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})$ and show $|\mathbf{Z}_1 - \mathbf{Z}_2| \leq \varepsilon$. By the definition of BCPs, there must exist some $S_1, \ldots, S_\mu \subseteq [N]$, each of size p, and some functions $g_1, \ldots, g_\mu : (\{0,1\}^n)^p \to \{0,1\}$ such that $\mathsf{Leak}(\mathbf{X}) = (g_1(\mathbf{X}_{S_1}), \ldots, g_\mu(\mathbf{X}_{S_\mu}))$. Thus:

$$\begin{aligned}\mathbf{Z}_1 &:= \mathsf{prodExt}(\mathbf{X}) \circ g_1(\mathbf{X}_{S_1}) \circ \cdots \circ g_\mu(\mathbf{X}_{S_\mu}),\\ \mathbf{Z}_2 &:= \mathbf{U}_m \circ g_1(\mathbf{X}_{S_1}) \circ \cdots \circ g_\mu(\mathbf{X}_{S_\mu}).\end{aligned}$$

The goal now is to perform fixings so as to reduce the analysis to let us apply Lemma 4.1 over t sources. We proceed as follows. First, notice that each of the μ subsets S_i has size p, and can therefore hold at most $\binom{p}{t}$ distinct subsets of size t. Thus, since we are told $\mu\binom{p}{t} < \binom{N}{t}$, there must be some *good* $G \in \binom{[N]}{t}$ where $G \not\subseteq S_i, \forall i \in [\mu]$. Without loss of generality, we may assume $G = [t]$ (any other case uses almost exactly the same ideas that follow, but the notation gets a little cumbersome). We let $\overline{G} = [N] \setminus G$, and by definition of statistical distance, we know that $|\mathbf{Z}_1 - \mathbf{Z}_2| \leq |\mathbf{Z}_1 \circ \mathbf{X}_{\overline{G}} - \mathbf{Z}_2 \circ \mathbf{X}_{\overline{G}}|$. This means there is some fixed $X^* \in (\{0,1\}^n)^{N-t}$ such that

$$|\mathbf{Z}_1 - \mathbf{Z}_2| \leq |(\mathbf{Z}_1 \mid \mathbf{X}_{\overline{G}} = X^*) - (\mathbf{Z}_2 \mid \mathbf{X}_{\overline{G}} = X^*)|.$$

Now, to see that this quantity is bounded above by ε, we just have to carefully rewrite the random variables. First, we note that it is safe to assume that $X_i^* \neq \vec{0}$, for all $i \in [N-t]$, at the expense of incurring a factor of 2^t in the error - this is because we can in fact preprocess the input to redirect zeroes to ones before calling $\mathsf{prodExt}$; this decreases the entropy of each source by 1 bit, and we will see (in Lemma 4.5) that we can handle such a situation at the expense of blowing up the error by an exponential factor in the number of sources placed into the final $\mathsf{prodExt}$ call (which, here, is t sources). Let us now examine the conditioned versions of $\mathbf{Z}_1$ and $\mathbf{Z}_2$.

We start by observing that $(\mathsf{prodExt}(\mathbf{X}) \mid \mathbf{X}_{\overline{G}} = X^*) = \sigma_{n,m}(\mathbf{X}_1 \cdot \dots \cdot \mathbf{X}_t \cdot X_1^* \cdot \dots \cdot X_{N-t}^*) = \mathsf{prodExt}(\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{t-1}, \mathbf{X}_t \cdot \pi)$, where $\pi = X_1^* \cdot \dots \cdot X_{N-t}^*$ is some fixed nonzero value in $\mathbb{F}_{2^n}$. Furthermore, for any $i \in [\mu]$, observe that $(g_i(\mathbf{X}_{S_i}) \mid \mathbf{X}_{\overline{G}} = X^*)$ becomes the function $g_i'(\mathbf{X}_{S_i \cap G})$, which is the restriction of g_i obtained by fixing the variables $\mathbf{X}_i, i \in S_i \cap \overline{G}$ according to X^*. By definition of G, we know $G \not\subseteq S_i$, and thus if we define $S_i' := S_i \cap G$, we know $S_i' \subsetneq G$, and furthermore $(g_i(\mathbf{X}_{S_i}) \mid \mathbf{X}_{\overline{G}} = X^*) = g_i'(\mathbf{X}_{S_i'})$. Thus we have:

$$\begin{aligned}(\mathbf{Z}_1 \mid \mathbf{X}_{\overline{G}} = X^*) &= \mathsf{prodExt}(\mathbf{X}_1, \dots, \mathbf{X}_t \cdot \pi) \circ g_1'(\mathbf{X}_{S_1'}) \circ \dots \circ g_\mu'(\mathbf{X}_{S_\mu'}), \text{ and}\\ (\mathbf{Z}_2 \mid \mathbf{X}_{\overline{G}} = X^*) &= \mathbf{U}_m \circ g_1'(\mathbf{X}_{S_1'}) \circ \dots \circ g_\mu'(\mathbf{X}_{S_\mu'}).\end{aligned}$$

Suppose now that we define $\mathbf{Y}_i, i \in [t]$ as $\mathbf{Y}_i := \mathbf{X}_i$ when $i \in [t-1]$, and $\mathbf{Y}_i := \mathbf{X}_t \cdot \pi$ when $i = t$. Then, for each $i \in [\mu]$, we define g_i'' to be identical to g_i', except for the fact that if g_i' receives $\mathbf{X}_t$ as an input - say, as its j^{th} argument - then g_i'' multiplies its j^{th} input by *the inverse of* π (in $\mathbb{F}_{2^n}^\times$) before passing all its input into g_i', and returning the result. By construction, such a function guarantees $g_i''(\mathbf{Y}_{S_i'}) = g_i'(\mathbf{X}_{S_i'})$. And thus, we see that we can write:

$$\begin{aligned}(\mathbf{Z}_1 \mid \mathbf{X}_{\overline{G}} = X^*) &= \mathsf{prodExt}(\mathbf{Y}_1, \dots, \mathbf{Y}_t) \circ g_1''(\mathbf{Y}_{S_1'}) \circ \dots \circ g_\mu''(\mathbf{Y}_{S_\mu'}), \text{ and}\\ (\mathbf{Z}_2 \mid \mathbf{X}_{\overline{G}} = X^*) &= \mathbf{U}_m \circ g_1''(\mathbf{Y}_{S_1'}) \circ \dots \circ g_\mu''(\mathbf{Y}_{S_\mu'}).\end{aligned}$$

Since $\mathbf{Y}_t$ is just a permutation of $\mathbf{X}_t$, it must have the same entropy, and furthermore note that each g_i'' acts as non-adaptive NOF leakage on $\mathbf{Y}_1, \dots, \mathbf{Y}_t$, since $S_i' \subsetneq [t]$. Thus we can use Lemma 4.1 to bound the difference $|(\mathbf{Z}_1 \mid \mathbf{X}_{\overline{G}} = X^*) - (\mathbf{Z}_2 \mid \mathbf{X}_{\overline{G}} = X^*)|$ as desired, completing the proof. □

4.3.3 Dropping the entropy requirement via two-source condensers and leakage lifting

In this section, we will show how to improve the entropy requirement of $\mathsf{prodExt}$ from $k = n$ all the way to $k = \operatorname{polylog} n$. The first step we take in this direction is a modest one: we show that without any further modificiations, $\mathsf{prodExt}$ will still work if its inputs are missing just a little entropy. More generally, we prove the following result:

Lemma 4.5. *Let* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *be a leakage-resilient extractor against* $\mathsf{nNOF}(\mu)$ *for entropy* $k = n$ *and error* ε. *Then for any* $0 < k \le n$, Ext *is also a leakage-resilient extractor against* $\mathsf{nNOF}(\mu - 2)$ *for entropy* k *and error* $\varepsilon \cdot 2^{N(n-k)}$.

Proof. Given N independent (n, k) sources $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$ and any $\mathsf{Leak} \in \mathsf{nNOF}(\mu - 2)$, we wish to upper bound the quantity

$$|\mathsf{Ext}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})|. \tag{4.1}$$

We may assume that each source $\mathbf{X}_i$ is *flat*; i.e., uniform over some $T_i \subseteq \{0,1\}^n$ of size 2^k. The main idea of this proof is to treat the missing entropy as leakage on uniform sources, by defining a function belonging to $\mathsf{nNOF}(2)$ that identifies the support of $(\mathbf{X}_1, \dots, \mathbf{X}_N)$. In particular, we define the indicator function $\mathsf{id} : (\{0,1\}^n)^N \to \{0,1\}^2$ as the map $(x_1, \dots, x_N) \mapsto (y_1, y_2)$, where $y_1 = 1$ if and only if $x_i \in T_i$, for all $i \in [N-1]$, and $y_2 = 1$ if and only if $x_N \in T_N$. Furthermore, we define $\vec{1} := (1, 1)$. Given these definitions, it is straightforward to verify that $\mathsf{id}(x) = \vec{1}$ if and only if $x \in T_1 \times \cdots \times T_N$, and that $\mathsf{id} \in \mathsf{nNOF}(2)$. Thus, if we define a function $\mathsf{Leak}' : (\{0,1\}^n)^N \to \{0,1\}^\mu$ that maps $x \mapsto (\mathsf{Leak}(x), \mathsf{id}(x))$, then $\mathsf{Leak}' \in \mathsf{nNOF}(\mu)$, and we may use it to upper bound Equation (4.1):

$$\begin{aligned}
&|\mathsf{Ext}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})| \\
&= |\mathsf{Ext}(\mathbf{U}_{Nn} \mid \mathsf{id}(\mathbf{U}_{Nn}) = \vec{1}) \circ \mathsf{Leak}(\mathbf{U}_{Nn} \mid \mathsf{id}(\mathbf{U}_{Nn}) = \vec{1}) \\
&\quad - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{U}_{Nn} \mid \mathsf{id}(\mathbf{U}_{Nn}) = \vec{1})| \\
&\leq |\mathsf{Ext}(\mathbf{U}_{Nn}) \circ \mathsf{Leak}(\mathbf{U}_{Nn}) \circ \mathsf{id}(\mathbf{U}_{Nn}) \\
&\quad - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{U}_{Nn}) \circ \mathsf{id}(\mathbf{U}_{Nn})| / \Pr[\mathsf{id}(\mathbf{U}_{Nn}) = \vec{1}] \\
&\leq \varepsilon \cdot 2^{N(n-k)},
\end{aligned}$$

where the first inequality is a Markov-type inequality, and the second follows from the hypothesis. □

Given the above, the key object we need in order to drop the entropy requirement of our extractor all the way down to $k \geq \mathrm{polylog}(n)$ is a *low-error strong two-source condenser*. Ben-Aroya, Cohen, Doron and Ta-Shma [BCDT19] recently constructed such objects with excellent parameters; most relevant to us here will be the following specialization of one of their more general constructions:

Theorem 4.5 ([BCDT19])**.** *There exist universal constants $C > 0$ and $\gamma := 1/C$ such that for every $n, k, m \in \mathbb{N}$ and $\varepsilon > 0$ satisfying $k \geq \log^C n$ and $k^\gamma \geq m$ and $\varepsilon \geq 2^{-k^{\gamma/2}}$, there exists an explicit function $\mathsf{2Cond} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ such that for any two independent (n,k) sources $\mathbf{X}_1, \mathbf{X}_2$, with probability $1-\varepsilon$ over $x_2 \sim \mathbf{X}_2$, the output has min-entropy $H_\infty(\mathsf{2Cond}(\mathbf{X}_1, x_2)) \geq m - \sqrt{m}$.*

With this condenser in hand, we're ready to define our final, low-entropy, version of the product-extractor:

Definition 4.4. *For a sufficiently large constant $C \geq 1$ and any $N, n, k, m_0, m \in \mathbb{N}$ satisfying $k \geq \log^C n$ and $k^{1/C} \geq m_0 \geq m$, let $\mathsf{2Cond} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^{m_0}$ be the condenser for (n,k) sources from Theorem 4.5, and let $\mathsf{prodExt}(\{0,1\}^{m_0})^{N-1} \to \{0,1\}^m$ be the product extractor from Definition 4.3. We define the* low entropy product extractor, *$\mathsf{leProdExt} : (\{0,1\}^n)^N \to \{0,1\}^m$, as*

$$\mathsf{leProdExt}(x_1, x_2, \ldots, x_N) := \mathsf{prodExt}((\mathsf{2Cond}(x_i, x_N))_{i\in[N-1]}).$$

In particular, the low-entropy product extractor simply takes its input and uses one of the sources as a common seed to a two-source condenser call on every other source. We now prove the main lemma of the section, which proves that $\mathsf{leProdExt}$ can in fact handle low-entropy, while simultaneously achieving a leakage-collusion tradeoff similar to Lemma 4.4. We remark again here that we will optimize to pick a good setting for t in Section 4.3.4.

Lemma 4.6. *There exist universal constants $C, \gamma > 0$ such that the following holds. For all sufficiently large $N, n \in \mathbb{N}$ and any $t, p \in \mathbb{N}$ such that $t \leq N$ and $p \leq N - 2$, the low-entropy product extractor $\mathsf{leProdExt} : (\{0,1\}^n)^N \to \{0,1\}^m$ from Definition 4.4 is an explicit leakage-resilient extractor against $\mathsf{nBCP}(p, \mu)$ for entropy $k \geq \log^C n$ and leakage $\mu < \min\{\xi, \binom{N-1}{t}/\binom{p}{t}\}$, with output length $m \geq \xi$ and error $\varepsilon = 2^{-\xi}$, where*

$$\xi = k^\gamma / 2^t.$$

Proof. We must show that for N independent (n,k) sources $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$, and any $\mathsf{Leak} \in \mathsf{nBCP}(p, \mu)$ with $\mathsf{Leak} \in \mathsf{nBCP}(p, \mu)$ with $\mu < \xi$ and $\mu\binom{p}{t} < \binom{N-1}{t}$,

$$|\mathsf{leProdExt}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})| \leq \varepsilon,$$

where $\varepsilon = 2^{-\xi}$. For brevity, we let $\mathbf{Z}_1 := \mathsf{leProdExt}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X})$ and $\mathbf{Z}_2 := \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})$ and show $|\mathbf{Z}_1 - \mathbf{Z}_2| \leq \varepsilon$. By definition of BCPs, there must exist some $S_1, \dots, S_\mu \subseteq [N]$, each of size p, and some functions $g_1, \dots, g_\mu : (\{0,1\}^n)^p \to \{0,1\}$ such that $\mathsf{Leak}(\mathbf{X}) = (g_1(\mathbf{X}_{S_1}), \dots, g_\mu(\mathbf{X}_{S_\mu}))$. Thus, substituting in the definition of leProdExt, we have

$$\mathbf{Z}_1 := \mathsf{prodExt}((\mathsf{2Cond}(\mathbf{X}_i, \mathbf{X}_N))_{i\in[N-1]}) \circ g_1(\mathbf{X}_{S_1}) \circ \cdots \circ g_\mu(\mathbf{X}_{S_\mu}), \text{ and}$$
$$\mathbf{Z}_2 := \mathbf{U}_m \circ g_1(\mathbf{X}_{S_1}) \circ \cdots \circ g_\mu(\mathbf{X}_{S_\mu}).$$

We remark that any parameters in the construction itself (like condenser output length, condenser error, product extractor output length, etc.) will be set at the end so that everything works out.

The goal now is to perform fixings to reduce the analysis to the analysis in Lemma 4.4. We proceed as follows. First, notice that each of the μ subsets S_i has size p, and can therefore hold at most $\binom{p}{t}$ distinct subsets of size t. Thus, since we are told $\mu\binom{p}{t} < \binom{N-1}{t}$, there must be some *good* $G \in \binom{[N-1]}{t}$ where $G \not\subseteq S_i, \forall i \in [\mu]$. Like in the proof to Lemma 4.4, we will assume, without loss of generality, that $G = [t]$.

Now, we let ε_1 be the error of 2Cond, meaning that with probability $1 - \varepsilon_1$ over $x_N \sim \mathbf{X}_N$, $\mathsf{2Cond}(\mathbf{X}_i, x_N)$ has entropy at least $m_0 - \sqrt{m_0}$, for any single $i \in [N-1]$. Thus, by a union bound, with probability $1 - t\varepsilon_1$ over $x_N \sim \mathbf{X}_N$, *every* random variable in $\{\mathsf{2Cond}(\mathbf{X}_i, x_N) : i \in G\}$ has entropy at least $m_0 - \sqrt{m_0}$. In other words, there is always some X_N^* such that the following is true:

$$\begin{aligned}|\mathbf{Z}_1 - \mathbf{Z}_2| &\leq |\mathbf{Z}_1 \circ \mathbf{X}_N - \mathbf{Z}_2 \circ \mathbf{X}_N| \\ &\leq t\varepsilon_1 + |(\mathbf{Z}_1 \mid \mathbf{X}_N = X_N^*) - (\mathbf{Z}_2 \mid \mathbf{X}_N = X_N^*)|,\end{aligned}$$

where $\mathsf{2Cond}(\mathbf{X}_i, X_N^*)$ has entropy at least $m_0 - \sqrt{m_0}$, for each $i \in G$. We now define $\mathbf{Y}_i := \mathsf{2Cond}(\mathbf{X}_i, X_N^*)$ for each $i \in [N-1]$, and we notice that this collection of random variables are mutually independent (because they are single-argument deterministic functions of the independent random variables $\{\mathbf{X}_i\}_{i\in[N-1]}$). We write $\mathbf{Y} = \mathbf{Y}_1 \circ \cdots \circ \mathbf{Y}_{N-1}$. We now fix each $\mathbf{X}_i, i \notin G \cup \{N\}$ to some $X_i^* \in \{0,1\}^n$ such that the following holds:

$$\begin{aligned}&|(\mathbf{Z}_1 \mid \mathbf{X}_N = X_N^*) - (\mathbf{Z}_2 \mid \mathbf{X}_N = X_N^*)| \\ &\quad\leq |(\mathbf{Z}_1 \mid \mathbf{X}_i = X_i^*, \forall i \notin G) - (\mathbf{Z}_2 \mid \mathbf{X}_i = X_i^*, \forall i \notin G)|.\end{aligned}$$

Notice that as a result, each $\mathbf{Y}_i, i \notin G \cup \{N\}$, gets fixed to $Y_i^* = \mathsf{2Cond}(X_i^*, X_N^*) \in \{0,1\}^{m_0}$, while each $\mathbf{Y}_i, i \in G$ still has entropy at least $m_0 - \sqrt{m_0}$, and the collection $\{\mathbf{Y}_i : i \in G\}$ remains mutually independent. By following the exact same reasoning as in the proof to Lemma 4.4 about restrictions and the structure of prodExt, we know that at this point we can write

$$\begin{aligned}&(\mathbf{Z}_1 \mid \mathbf{X}_i = X_i^*, \forall i \notin G) \\ &= \mathsf{prodExt}(\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_{t-1}, \mathbf{Y}_t \cdot \pi) \circ g_1'(\mathbf{X}_{S_1 \cap G}) \circ \cdots \circ g_\mu'(\mathbf{X}_{S_\mu \cap G}) =: \mathbf{Z}_1', \text{ and} \\ &(\mathbf{Z}_2 \mid \mathbf{X}_i = X_i^*, \forall i \notin G) \\ &= \mathbf{U}_m \circ g_1'(\mathbf{X}_{S_1 \cap G}) \circ \cdots \circ g_\mu'(\mathbf{X}_{S_\mu \cap G}) =: \mathbf{Z}_2',\end{aligned}$$

where $\pi = Y_{t+1} \cdot \dots \cdot Y_{N-1}$, and each g_i' is the appropriate restriction of g_i induced by the fixings of its inputs outside G. We are nearly ready to apply the leakage-resilience of prodExt against NOF protocols and complete the proof. In order to arrive at this situation, we must somehow write each leakage function g_i' as a function of random variables from $\{\mathbf{Y}_i : i \in G\}$ instead of $\{\mathbf{X}_i : i \in G\}$. It turns out this is not so difficult to do: we know that for each $i \in [N-1]$, $\mathbf{Y}_i$ is a deterministic function of $\mathbf{X}_i$. As such, for each

i, we can find some randomness $\mathbf{Q}_i$ and a deterministic function sample such that $\mathbf{Q}_i$ is independent of $\mathbf{Y}_i$, and $\mathsf{sample}(\mathbf{Y}_i, \mathbf{Q}_i)$ is identically distributed to $\mathbf{X}_i$: for any fixed $y \sim \mathbf{Y}_i$, the function call $\mathsf{sample}(y, \mathbf{Q}_i)$ simply uses $\mathbf{Q}_i$ to sample from $(\mathbf{X} \mid \mathbf{Y} = y)$.

As always, we can fix the random variables $\{\mathbf{Q}'_i : i \in G\}$ to some values $\{Q^*_i\}$ without reducing the distance between $\mathbf{Z}'_1, \mathbf{Z}'_2$:

$$|\mathbf{Z}'_1 - \mathbf{Z}'_1| \leq |(\mathbf{Z}'_1 \mid \mathbf{Q}_i = Q^*_i, \forall i \in G) - (\mathbf{Z}'_2 \mid \mathbf{Q}_i = Q^*_i, \forall i \in G)|.$$

Furthermore, under this conditioning, we know that each $g'_i(\mathbf{X}_{S_1 \cap G})$ obtains the form $g''_i(\mathbf{Y}_{S_1 \cap G})$ for some other deterministic function g''_i, since we saw above that $\mathbf{X}_i \approx \mathsf{sample}(\mathbf{Y}_i, \mathbf{Q}_i)$, and we have fixed all the variables $\{\mathbf{Q}_i : i \in G\}$. Thus, we may write:

$$\begin{aligned}
&(\mathbf{Z}'_1 \mid \mathbf{Q}_i = Q^*_i, \forall i \in G) \\
&\quad = \mathsf{prodExt}(\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_{t-1}, \mathbf{Y}_t \cdot \pi) \\
&\qquad\quad \circ g''_1(\mathbf{Y}_{S_1 \cap G}) \circ \dots \circ g''_\mu(\mathbf{Y}_{S_\mu \cap G}), \text{ and} \\
&(\mathbf{Z}'_2 \mid \mathbf{Q}_i = Q^*_i, \forall i \in G) \\
&\quad = \mathbf{U}_m \circ g''_1(\mathbf{Y}_{S_1 \cap G}) \circ \dots \circ g''_\mu(\mathbf{Y}_{S_\mu \cap G})
\end{aligned}$$

Using the same trick as in the proof to Lemma 4.4, we may remove π from the equation above, and thus we just assume now that it is no longer present. Thus, at last, we see that it suffices to upper bound $|\mathbf{Z}''_1 - \mathbf{Z}''_2|$, where

$$\begin{aligned}
\mathbf{Z}''_1 &:= \mathsf{prodExt}(\mathbf{Y}_1, \dots, \mathbf{Y}_t) \circ g''_1(\mathbf{Y}_{S_1 \cap G}) \circ \dots \circ g''_\mu(\mathbf{Y}_{S_\mu \cap G}), \text{ and} \\
\mathbf{Z}''_2 &:= \mathbf{U}_m \circ g''_1(\mathbf{Y}_{S_1 \cap G}) \circ \dots \circ g''_\mu(\mathbf{Y}_{S_\mu \cap G}).
\end{aligned}$$

These random variables have a very special structure: the collection $\{\mathbf{Y}_i : i \in [t]\}$ are independent, and each is an $(m_0, m_0 - \sqrt{m_0})$ source, due to the parameters of 2Cond. Furthermore, we know $G \cap S_i \subsetneq G$, for all $i \in [\mu]$, and thus each g_i is an NOF-leak on $\mathbf{Y} = \mathbf{Y}_1 \circ \dots \circ \mathbf{Y}_t$.

Thus, to conclude, we combine Lemma 4.1 with Lemma 4.5 to see that $\mathsf{prodExt} : (\{0,1\}^{m_0})^t \to \{0,1\}^m$ is a leakage-resilient extractor against $\mathsf{nNOF}(\mu_2 - 2)$ for entropy $m_0 - \sqrt{m_0}$, error $\varepsilon_3 = \varepsilon_2 \cdot 2^{t\sqrt{m_0}}$, and output $m = \gamma m_0 / 2^t$, where $\varepsilon_2 = 2^{-m}$ and $\mu_2 = m$, for some small universal constant $\gamma > 0$. Furthermore, we know the sources $\mathbf{Y}_i, i \in [t]$ end up with the promised entropy $m_0 - \sqrt{m_0}$ as long as $k \geq \log^C n$ and $k^{1/C} \geq m_0$, with error $\varepsilon_1 = 2^{-k^{1/(2C)}}$, for some universal constant $C \geq 1$. Thus, the condenser and extractor will both work as long as this entropy is guaranteed, and as long as $\mu < \mu_2 - 2$ (because this means the concatenation of leaks $g''_1, \dots, g''_\mu$ is in $\mathsf{nNOF}(\mu_2 - 2)$). Furthermore, its error will be

$$\varepsilon = |\mathbf{Z}_1 - \mathbf{Z}_2| \leq t\varepsilon_1 + |\mathbf{Z}''_1 - \mathbf{Z}''_2| = t\varepsilon_1 + \varepsilon_3.$$

Finally, recall that we required $\mu\binom{p}{t} < \binom{N-1}{t}$ at the beginning to ensure we could find a good set G. Thus, there exists some small constant $c > 0$ and function $\xi(k, t) := k^c / 2^t$ such that as long as $k \geq \log^C n$, and $\mu < \min\{\xi, \binom{N-1}{t} / \binom{p}{t}\}$, and $m \leq \xi$, it holds that $|\mathbf{Z}_1 - \mathbf{Z}_2| = \varepsilon \leq 2^{-\xi}$, which completes the proof. □

4.3.4 Adding adaptivity at (almost) no cost

For the final step, all that remains is to add adaptivity back into the equation. We prove the following.

Lemma 4.7. *Let* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *be a leakage-resilient extractor against* $\mathsf{nBCP}(p,\mu)$ *for entropy* k*, with error* ε*. Then* Ext *is also a leakage-resilient extractor against* $\mathsf{BCP}(p,\mu)$ *for entropy* k *with error at most* $(2^\mu + 1)\sqrt{\varepsilon}$*.*

Proof. Consider any function $\mathsf{Leak} : (\{0,1\}^n)^N \to \{0,1\}^\mu$ in $\mathsf{BCP}(p,\mu)$. We must show that for any N independent sources $\mathbf{X} := (\mathbf{X}_1, \dots, \mathbf{X}_N)$,

$$|\mathsf{Ext}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})| \leq (2^\mu + 1)\sqrt{\varepsilon}. \tag{4.2}$$

By the averaging principle (Fact 2.2), we know that

$$|\mathsf{Ext}(\mathbf{X}) \circ \mathsf{Leak}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}(\mathbf{X})| = \mathbb{E}_{\tau \sim \mathsf{Leak}(\mathbf{X})}[|(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{Leak}(\mathbf{X}) = \tau) - \mathbf{U}_m|].$$

We define a set of *bad leaks*, $\mathsf{Bad} := \{\tau \in \{0,1\}^\mu : |(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{Leak}(\mathbf{X}) = \tau) - \mathbf{U}_m| \geq \sqrt{\varepsilon}\}$, and can use it to upper bound the above as follows.

$$\begin{aligned}
\mathbb{E}_{\tau \sim \mathsf{Leak}(\mathbf{X})}[|(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{Leak}(\mathbf{X}) = \tau) - \mathbf{U}_m|] &< \Pr[\mathsf{Leak}(\mathbf{X}) \in \mathsf{Bad}] + \sqrt{\varepsilon} \\
&\leq |\mathsf{Bad}| \cdot \max_{\tau \in \mathsf{Bad}} \Pr[\mathsf{Leak}(\mathbf{X}) = \tau] + \sqrt{\varepsilon} \\
&\leq 2^\mu \cdot \max_{\tau \in \mathsf{Bad}} \Pr[\mathsf{Leak}(\mathbf{X}) = \tau] + \sqrt{\varepsilon}.
\end{aligned}$$

Thus, it suffices to show that for any $\tau \in \mathsf{Bad}$, we can upper bound $\Pr[\mathsf{Leak}(\mathbf{X}) = \tau]$ by $\sqrt{\varepsilon}$. Consider any $\tau \in \mathsf{Bad}$. We will define a function that agrees with Leak on exactly the inputs that map to τ, but is much simpler. In particular, note that by definition of bounded collusion protocols, we know that Leak is defined in a way such that for every $i \in [\mu]$, $\exists S_i : (\{0,1\}^n)^{i-1} \to \binom{[N]}{p}$ and $\mathsf{Leak}_i : (\{0,1\}^n)^{i-1} \times (\{0,1\}^n)^p \to \{0,1\}$ such that for every $x_1, \dots, x_N \in \{0,1\}^n$,

$$\mathsf{Leak}(x_1, \dots, x_N) = (y_1, \dots, y_\mu),$$

where

$$y_i := \mathsf{Leak}_i(y_1, \dots, y_{i-1}, x_{S_i(y_1, \dots, y_{i-1})}), \forall i \in [\mu].$$

Using this definition, we will define a restricted version of this function as follows. Define $\mathsf{nLeak}^\tau : (\{0,1\}^n)^N \to \{0,1\}^\mu$ such that for every $x_1, \dots, x_N \in \{0,1\}^n$,

$$\mathsf{nLeak}^\tau(x_1, \dots, x_N) := (z_1, \dots, z_\mu),$$

where

$$z_i := \mathsf{Leak}_i(\tau_1, \dots, \tau_{i-1}, x_{S_i(\tau_1, \dots, \tau_{i-1})}), \forall i \in [\mu].$$

Observe that given this definition, nLeak^τ is in the class $\mathsf{nBCP}(\mu, p)$, and most importantly, the preimage of τ under nLeak^τ and Leak are the same: i.e., for every $x \in (\{0,1\}^n)^N$, $\mathsf{nLeak}^\tau(x) = \tau \iff \mathsf{Leak}(x) = \tau$. Thus, we know that $\Pr[\mathsf{Leak}(\mathbf{X}) = \tau] = \Pr[\mathsf{nLeak}^\tau(\mathbf{X}) = \tau]$, and we know that $|(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{nLeak}^\tau(\mathbf{X}) = \tau) - \mathbf{U}_m| = |(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{Leak}(\mathbf{X}) = \tau) - \mathbf{U}_m| \geq \sqrt{\varepsilon}$, because we selected $\tau \in \mathsf{Bad}$. Since $\mathsf{nLeak}^\tau \in \mathsf{nBCP}(\mu, p)$ and Ext is resilient against this class, we know by definition that

$$\begin{aligned}
&|\mathsf{Ext}(\mathbf{X}) \circ \mathsf{nLeak}^\tau(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{nLeak}^\tau(\mathbf{X})| \\
&= \mathbb{E}_{\tau' \sim \mathsf{nLeak}^\tau(\mathbf{X})}[|(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{nLeak}^\tau(\mathbf{X}) = \tau') - \mathbf{U}_m|] \\
&\leq \varepsilon,
\end{aligned}$$

and thus with probability at most $\sqrt{\varepsilon}$ over sampling $\tau' \sim \mathsf{nLeak}^\tau(\mathbf{X})$, $|(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{nLeak}^\tau(\mathbf{X}) = \tau') - \mathbf{U}_m| \geq \sqrt{\varepsilon}$. Thus, because we saw above that $|(\mathsf{Ext}(\mathbf{X}) \mid \mathsf{nLeak}^\tau(\mathbf{X}) = \tau) - \mathbf{U}_m| \geq \sqrt{\varepsilon}$, we know that $\Pr[\mathsf{nLeak}^\tau(\mathbf{X}) = \tau] \leq \sqrt{\varepsilon}$. Thus, it also holds that $\Pr[\mathsf{Leak}(\mathbf{X}) = \tau] \leq \sqrt{\varepsilon}$, as desired. □

At last, we have all the ingredients needed to finish the proof of our full-blown leakage-resilient extractor (Theorem 4.4). It will follow immediately by combining our low-entropy leakage resilient extractor for non-adaptive BCPs (Lemma 4.6) with the above result showing that adaptivity comes for free (Lemma 4.7), while fixing t (which we have thus far left undefined) to the appropriate value.

Proof of Theorem 4.4. Simply combine Lemma 4.6 with Lemma 4.7 and set $t = \frac{\log(k^\gamma)}{\log((N-1)/p)+1}$. □

4.4 Applications in communication complexity

As we advertised in Section 4.1, a key application of our new leakage-resilient extractors are new explicit lower bounds against a wide range of multiparty communication protocols. We go into more detail, below.

4.4.1 Explicit lower bounds against multiparty communication protocols

Recall that leakage-resilient extractors against bounded collusion protocols are, in fact, *strictly stronger* than lower bounds against them. Indeed, they may be viewed as *average-case*, *multi-output* lower bounds against bounded collusion protocols. However, we find it still useful to record the *worst-case*, *single-output* lower bounds that come about, as these may be easier to compare against classical results in complexity theory.

Theorem 4.6 (Theorem 4.1, repeated)**.** *There is a universal constant $c > 0$ such that for all $N, n \in \mathbb{N}$ and $p \leq N - 1$, there exists an explicit function $f : (\{0,1\}^n)^N \to \{0,1\}$ with*

$$\mathsf{CC}^p(f) \geq c \cdot n^{\frac{\log(N/p)}{\log(N/p)+1}}.$$

In particular, the above lower bounds witness a collusion-complexity tradeoff. Most notably, these lower bounds are of the form $n^{\Omega(1)}$ even for $p = 0.99N$ (regardless of the dependence between N, n), which may be surprising given for certain values of N, no nontrivial lower bounds are known when p is pushed up to $p = N - 1$ (i.e., number-on-forehead protocols). Moreover, staring at the above lower bounds, one might notice that they are actually *stronger* than what one might obtain by instantiating our leakage-resilient extractor (Theorem 4.4) with full-entropy sources, due to the fact that they do not have an arbitrarily small constant attached to the power. Thus, in order to prove these lower bounds, we actually dig back into the proof presented in the previous section. But have no fear: this result is immediate, simply by combining our leakage-resilient extractor against non-adaptive BCPs for full min-entropy (Lemma 4.4) with the fact that adaptivity comes for free (Lemma 4.7), while setting the parameter t appropriately.

Proof of Theorem 4.6. Simply combine Lemma 4.4 with Lemma 4.7 and set $t = \frac{\log(n)}{\log(N/p)+1}$. □

4.5 Applications in cryptography

Now, let us turn to an application that takes a bit more work to prove.

4.5.1 Leakage-resilient secret sharing: an exponential improvement

One of the biggest motivating applications of leakage-resilient extractors is that they can provide better leakage-resilient secret sharing schemes. In this section, we will recall the definition of leakage-resilient secret sharing schemes, show that our explicit leakage-resilient extractors greatly improve existing results, and introduce a new, natural variant of secret sharing in which our low-entropy leakage-resilient extractor shines. We start with the standard definition of t-out-of-N secret sharing schemes, as given in [KMS19].

Definition 4.5. *Let* $\mathsf{Share} : \{0,1\}^m \to (\{0,1\}^n)^N$ *be a randomized* sharing *algorithm that maps a secret of length m into N shares of length n, and let* $\mathsf{Rec} : (\{0,1\}^n)^t \to \{0,1\}^m$ *be a deterministic* reconstruction *algorithm that recovers the secret from t shares. The tuple* $(\mathsf{Share}, \mathsf{Rec})$ *is defined as a* t-out-of-N secret sharing scheme *if both of the following hold.*

1. Perfect correctness*: Any t shares can recover the secret. Formally, for any secret $\Psi \in \{0,1\}^m$ and any $S \subseteq [N]$ of size t,*
$$\Pr[\mathsf{Rec}(\mathsf{Share}(\Psi)_S) = \Psi] = 1,$$
where the randomness is over Share.

2. Perfect secrecy*: Fewer than t shares reveal no information about the secret. Formally, for any secrets $\Psi_1, \Psi_2 \in \{0,1\}^m$, and any $S \subseteq [N]$ of size less than t,*
$$|\mathsf{Share}(\Psi_1) - \mathsf{Share}(\Psi_2)| = 0.$$

A *leakage-resilient* secret sharing scheme is defined to be a t-out-of-N scheme that can also hide the secret against more powerful adversaries (typically ones that simulate communication protocols). Formally:

Definition 4.6. *Let* $(\mathsf{Share}, \mathsf{Rec})$ *be a t-out-of-N secret sharing scheme, with* $\mathsf{Share} : \{0,1\}^m \to (\{0,1\}^n)^N$ *and* $\mathsf{Rec} : (\{0,1\}^n)^t \to \{0,1\}^m$. *Given a family of* leakage *functions $\mathcal{F}$ of the form $f : (\{0,1\}^n)^N \to \{0,1\}^\mu$, we say that* $(\mathsf{Share}, \mathsf{Rec})$ *is* leakage-resilient *against $\mathcal{F}$ with error ε if for any secrets $\Psi_1, \Psi_2 \in \{0,1\}^m$, and any $f \in \mathcal{F}$,*
$$|f(\mathsf{Share}(\Psi_1)) - f(\mathsf{Share}(\Psi_2))| \leq \varepsilon,$$
where the randomness comes from Share.

We now elucidate the connection between leakage-resilient secret sharing schemes and leakage-resilient extractors, by showing how the latter can be used to construct the former. This connection and the construction in our proof are borrowed from [KMS19, Lemma 1]. However, we include it for completeness, because we slightly generalize their result and proof to take advantage of multi-bit output leakage-resilient extractors, and because our proof can easily be generalized to any leakage class closed under restrictions.

Lemma 4.8. *Suppose there exists a leakage-resilient extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *against* $\mathsf{BCP}(p,\mu)$ *for entropy $k = n$ and error ε. Then there exists an N-out-of-N secret sharing scheme* $(\mathsf{Share}, \mathsf{Rec})$ *that is leakage-resilient against* $\mathsf{BCP}(p,\mu)$*, with error $2^{m+1} \cdot \varepsilon$, where* $\mathsf{Share} : \{0,1\}^m \to (\{0,1\}^{n+m})^N$, $\mathsf{Rec} : (\{0,1\}^{n+m})^N \to \{0,1\}^m$. *Furthermore, if* Ext *runs in* $\mathrm{poly}(N,n)$ *time, then so do* Share *and* Rec.

Proof. We first construct $(\mathsf{Share}, \mathsf{Rec})$ from Ext, and then prove its claimed properties. Let $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_N)$ be N uniform random variables over $\{0,1\}^n$, and let $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_{N-1})$ be $N-1$ uniform random variables over $\{0,1\}^m$, where all $2N-1$ random variables are independent. We define $\mathsf{Share} : \{0,1\}^m \to (\{0,1\}^{n+m})^N$, for each $\Psi \in \{0,1\}^m$, as
$$\mathsf{Share}(\Psi) := (\mathbf{X}_1, \mathbf{Y}_1) \circ (\mathbf{X}_2, \mathbf{Y}_2) \circ \cdots \circ (\mathbf{X}_{N-1}, \mathbf{Y}_{N-1}) \circ (\mathbf{X}_N, \Psi \oplus \mathsf{Ext}(\mathbf{X}) \oplus \mathbf{Y}_1 \oplus \cdots \oplus \mathbf{Y}_{N-1})$$

and we define $\mathsf{Rec} : (\{0,1\}^{n+m})^N \to \{0,1\}^m$, for each $((X_1, Y_1), \dots, (X_N, Y_N)) \in (\{0,1\}^{n+m})^N$, where each $X_i \in \{0,1\}^n, Y_i \in \{0,1\}^m$, as

$$\mathsf{Rec}((X_1, Y_1), \dots, (X_N, Y_N)) := \mathsf{Ext}(X_1, \dots, X_N) \oplus Y_1 \oplus \dots \oplus Y_{N-1} \oplus Y_N.$$

We claim that $(\mathsf{Share}, \mathsf{Rec})$ is an N-out-of-N secret sharing scheme that is leakage-resilient against $\mathcal{F}_{N,n+m}$ with error $2^{m+1} \cdot \varepsilon$. We must prove the *perfect correctness*, *perfect secrecy*, and *leakage-resilience* of this scheme. (The efficiency claim is clear.)

Perfect correctness follows easily by definition: for each $\Psi \in \{0,1\}^m$,

$$\begin{aligned}\mathsf{Rec}(\mathsf{Share}(\Psi)_{[N]}) &= \mathsf{Ext}(\mathbf{X}) \oplus \mathbf{Y}_1 \oplus \dots \oplus \mathbf{Y}_{N-1} \oplus (\Psi \oplus \mathsf{Ext}(\mathbf{X}) \oplus \mathbf{Y}_1 \oplus \dots \oplus \mathbf{Y}_{N-1}) \\ &= \Psi,\end{aligned}$$

and thus of course $\Pr[\mathsf{Rec}(\mathsf{Share}(\Psi))_{[N]} = \Psi] = 1$.

Perfect secrecy follows easily as well. For any $\Psi_1, \Psi_2 \in \{0,1\}^m$, and any $S \subseteq [N]$ of size $|S| < N$, either S contains N, or not. If $N \notin S$, $\mathsf{Share}(\Psi_1)_S = (\mathbf{X}_i, \mathbf{Y}_i)_{i \in S} = \mathsf{Share}(\Psi_2)_S$, and thus clearly $|\mathsf{Share}(\Psi_1) - \mathsf{Share}(\Psi_2)| = 0$. If $N \in S$, there must be some $\alpha \in [N-1] \setminus S$. Thus, there must exist some fixings $\{x_i \in \{0,1\}^n : i \in [N]\}$ and $\{y_i \in \{0,1\}^m : i \in [N-1] \setminus \{\alpha\}\}$ such that

$$|\mathsf{Share}(\Psi_1)_S - \mathsf{Share}(\Psi_2)_S| \leq |\mathbf{Z}_1 - \mathbf{Z}_2|,$$

where

$$\begin{aligned}\mathbf{Z}_1 &:= (\mathsf{Share}(\Psi_1)_S \mid \mathbf{X}_{[N]} = (x_i)_{i \in [N]}, \mathbf{Y}_{[N-1] \setminus \{\alpha\}} = (y_i)_{i \in [N-1] \setminus \{\alpha\}}), \\ \mathbf{Z}_2 &:= (\mathsf{Share}(\Psi_2)_S \mid \mathbf{X}_{[N]} = (x_i)_{i \in [N]}, \mathbf{Y}_{[N-1] \setminus \{\alpha\}} = (y_i)_{i \in [N-1] \setminus \{\alpha\}}).\end{aligned}$$

Notice, however, that

$$\begin{aligned}\mathbf{Z}_1 &= (x_i, y_i)_{i \in S \setminus \{N\}} \circ (x_N, \Psi_1 \oplus \mathsf{Ext}(x_1, \dots, x_N) \oplus \mathbf{Y}_\alpha \oplus_{i \in [N-1] \setminus \{\alpha\}} y_i) \\ &= (x_i, y_i)_{i \in S \setminus \{N\}} \circ (x_N, \mathbf{U}_m) \\ &= (x_i, y_i)_{i \in S \setminus \{N\}} \circ (x_N, \Psi_2 \oplus \mathsf{Ext}(x_1, \dots, x_N) \oplus \mathbf{Y}_\alpha \oplus_{i \in [N-1] \setminus \{\alpha\}} y_i) \\ &= \mathbf{Z}_2,\end{aligned}$$

and thus $|\mathsf{Share}(\Psi_1)_S - \mathsf{Share}(\Psi_2)_S| = 0$.

Finally, to show leakage-resilience, we must prove that for any $\Psi_1, \Psi_2 \in \{0,1\}^m$, and any $\mathsf{Leak} : (\{0,1\}^{n+m})^N \to \{0,1\}^\mu \in \mathcal{F}_{N,n+m}$, $|\mathsf{Leak}(\mathsf{Share}(\Psi_1)) - \mathsf{Leak}(\mathsf{Share}(\Psi_2))| \leq 2^{m+1} \cdot \varepsilon$. Now, we know that for some $Y^* \in (\{0,1\}^m)^{N-1}$, it holds that $|\mathsf{Leak}(\mathsf{Share}(\Psi_1)) - \mathsf{Leak}(\mathsf{Share}(\Psi_2))|$ is at most

$$|(\mathsf{Leak}(\mathsf{Share}(\Psi_1)) \mid \mathbf{Y} = Y^*) - (\mathsf{Leak}(\mathsf{Share}(\Psi_2)) \mid \mathbf{Y} = Y^*)|.$$

Thus, if we define the fixed quantities $\psi_1 := \Psi_1 \oplus Y_1^* \oplus \dots \oplus Y_{N-1}^*$ and $\psi_2 := \Psi_2 \oplus Y_1^* \oplus \dots \oplus Y_{N-1}^*$, the above statistical distance is precisely $|\mathbf{Z}_1 - \mathbf{Z}_2|$, where $\mathbf{Z}_\alpha, \forall \alpha \in [2]$ is defined as follows:

$$\mathbf{Z}_\alpha := \mathsf{Leak}((\mathbf{X}_1, Y_1^*) \circ (\mathbf{X}_2, Y_2^*) \circ \dots \circ (\mathbf{X}_{N-1}, Y_{N-1}^*) \circ (\mathbf{X}_N, \mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha)).$$

In order to bound $|\mathbf{Z}_1 - \mathbf{Z}_2|$, we will introduce some new random variables. First, we let $\mathbf{Q}$ denote a fresh new random variable that is uniform over $\{0,1\}^m$ and independent of everything we've seen thus far. Then, we define

$$\mathbf{Z}_0 := \mathsf{Leak}((\mathbf{X}_1, Y_1^*) \circ (\mathbf{X}_2, Y_2^*) \circ \dots \circ (\mathbf{X}_{N-1}, Y_{N-1}^*) \circ (\mathbf{X}_N, \mathbf{Q})).$$

Notice that by the triangle inequality, $|\mathbf{Z}_1 - \mathbf{Z}_2| \le |\mathbf{Z}_1 - \mathbf{Z}_0| + |\mathbf{Z}_0 - \mathbf{Z}_2|$. Thus, to complete the proof, it suffices to show $|\mathbf{Z}_\alpha - \mathbf{Z}_0| \le 2^m \cdot \varepsilon, \forall \alpha \in [2]$. To do so, we first use the data-processing inequality to get

$$|\mathbf{Z}_\alpha - \mathbf{Z}_0| \le |\mathbf{Z}_\alpha \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) - \mathbf{Z}_0 \circ \mathbf{Q}|.$$

Suppose now that we define, for each fixed $b \in \{0,1\}^m$, the random variable

$$\mathbf{Z}^{(b)} := \mathsf{Leak}((\mathbf{X}_1, Y_1^*) \circ (\mathbf{X}_2, Y_2^*) \circ \cdots \circ (\mathbf{X}_{N-1}, Y_{N-1}^*) \circ (\mathbf{X}_N, b)).$$

Then by definition of statistical distance, we have $|\mathbf{Z}_\alpha \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) - \mathbf{Z}_0 \circ \mathbf{Q}|$ is

$$\begin{aligned}
&= \frac{1}{2} \sum_{b \in \{0,1\}^m} \sum_{a \in \{0,1\}^\mu} |\Pr[\mathbf{Z}_\alpha \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) = a \circ b] - \Pr[\mathbf{Z}_0 \circ \mathbf{Q} = a \circ b]| \\
&= \frac{1}{2} \sum_{b \in \{0,1\}^m} \sum_{a \in \{0,1\}^\mu} |\Pr[\mathbf{Z}^{(b)} \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) = a \circ b] - \Pr[\mathbf{Z}^{(b)} \circ \mathbf{Q} = a \circ b]| \\
&\le 2^m \max_{b \in \{0,1\}^m} \frac{1}{2} \sum_{a \in \{0,1\}^\mu} |\Pr[\mathbf{Z}^{(b)} \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) = a \circ b] - \Pr[\mathbf{Z}^{(b)} \circ \mathbf{Q} = a \circ b]| \\
&\le 2^m \max_{b \in \{0,1\}^m} \frac{1}{2} \sum_{\substack{a \in \{0,1\}^\mu \\ b' \in \{0,1\}^m}} |\Pr[\mathbf{Z}^{(b)} \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) = a \circ b'] - \Pr[\mathbf{Z}^{(b)} \circ \mathbf{Q} = a \circ b']| \\
&= 2^m \max_{b \in \{0,1\}^m} |\mathbf{Z}^{(b)} \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) - \mathbf{Z}^{(b)} \circ \mathbf{Q}| \\
&= 2^m \max_{b \in \{0,1\}^m} |\mathsf{Ext}(\mathbf{X}) \circ \mathbf{Z}^{(b)} - \mathbf{U}_m \circ \mathbf{Z}^{(b)}|
\end{aligned}$$

where the last equality holds by $\mathbf{Q} = \mathbf{U}_m$. Finally, notice that for each $b \in \{0,1\}^m$, we may define a restriction string $\phi^{(b)} \in ((\{0,1\}^n \cup \{*\}) \times (\{0,1\}^m \cup \{*\}))^N$ such that $\phi_{i,1}^{(b)} = *, \forall i \in [N]$, and $\phi_{i,2}^{(b)} = Y_i^*, \forall i \in [N-1]$, and $\phi_{N,2}^{(b)} = b$, and note that if we define $\mathsf{Leak}^{(b)} : (\{0,1\}^n)^N \to \{0,1\}^\mu$ as the restriction of Leak under $\phi^{(b)}$, then $\mathbf{Z}^{(b)} = \mathsf{Leak}^{(b)}(\mathbf{X})$. Furthermore, it is easy to show this restriction $\mathsf{Leak}^{(b)} \in \mathsf{BCP}(p, \mu)$. Thus,

$$\begin{aligned}
|\mathbf{Z}_\alpha \circ (\mathsf{Ext}(\mathbf{X}) \oplus \psi_\alpha) - \mathbf{Z}_0 \circ \mathbf{Q}| &\le 2^m \max_{b \in \{0,1\}^m} |\mathsf{Ext}(\mathbf{X}) \circ \mathsf{Leak}^{(b)}(\mathbf{X}) - \mathbf{U}_m \circ \mathsf{Leak}^{(b)}(\mathbf{X})| \\
&\le 2^m \cdot \varepsilon,
\end{aligned}$$

where the last inequality follows by the leakage-resilience of Ext. □

Using this lemma, we are able to immediately use our leakage-resilient extractors to get much better leakage-resilient secret sharing schemes in the N-out-of-N setting. In leakage-resilient secret sharing, one typically wants the share size to be as small as possible. Thus, in the following theorem, one should notice how the requirement on share size, n, grows as we share larger secrets of size m, require less error ε, and give the adversary more power, in terms of its collusion bound p and number of rounds μ.

Theorem 4.7 (Theorem 4.3, formal)**.** *There exists a universal constant $C > 0$ and an N-out-of-N secret sharing scheme* (Share, Rec)*, of the form* Share : $\{0,1\}^m \to (\{0,1\}^n)^N$, Rec : $(\{0,1\}^n)^N \to \{0,1\}^m$, *that shares secrets of length m into N shares of length n that is leakage-resilient against* $\mathsf{BCP}(p, \mu)$ *with error ε. The shares have length at most*

$$n = C \cdot \left(m + 1 + \log(1/\varepsilon))^{\frac{\log(N/p)+1}{\log(N/p)}} + \mu^{\frac{\log(N/p)+1}{\log(N/p)}} \right),$$

and the scheme runs in time $\mathrm{poly}(N, n)$.

Proof. By Lemma 4.4, there exists an explicit leakage-resilient extractor $\mathsf{Ext} : (\{0,1\}^{n_0})^N \to \{0,1\}^m$ against $\mathsf{BCP}(p, \mu)$ for full entropy with error $\varepsilon \cdot 2^{-(m+1)}$ as long as $\mu < \xi, m \le \xi$, and $\varepsilon 2^{-(m+1)} \ge 2^{-\xi}$, where $\xi = \Omega(n_0^{\frac{\log(N/p)}{\log(N/p)+1}})$. Rewriting these restrictions, it is easy to show that this all holds if

$$\begin{aligned} n_0 &\ge O(\max\{\mu^{\frac{\log(N/p)+1}{\log(N/p)}}, m^{\frac{\log(N/p)+1}{\log(N/p)}}, (m + 1 + \log(1/\varepsilon))^{\frac{\log(N/p)+1}{\log(N/p)}}\}) \\ &= O((m + 1 + \log(1/\varepsilon))^{\frac{\log(N/p)+1}{\log(N/p)}} + \mu^{\frac{\log(N/p)+1}{\log(N/p)}}). \end{aligned}$$

Thus by setting $n := n_0 + m = O(n_0)$ and applying Lemma 4.8, the result is immediate. □

In order to put the parameters we can achieve in context, we note that the previous best requirement on share size n, given by [KMS19], was of the form

$$n = m(\log N)(\mu + \log(1/\varepsilon)) \cdot 2^{O(p)}.$$

The biggest difference here is that the share size in [KMS19] has an exponential dependency on the collusion bound, p, while our share size does not. Thus, because a secret sharing scheme is only considered efficient if share size grows polynomially in N, the maximum collusion that can be handled from the leakage-resilient schemes in [KMS19] is $p = O(\log N)$. Our schemes, however, are able to efficiently handle up to $p = 0.99N$ parties colluding, and furthermore remove any dependency of n on N in this setting.

Seeded leakage-resilient secret sharing Interestingly, it also seems like there is a natural setting in which such leakage-resilient secret sharing schemes can take advantage of our *low entropy* extractors. As discussed above, the key parameter to optimize in secret sharing is share size, n. Given a secret that is to be distributed amongst N parties and statistically hidden from any powerful adversary, we would like a scheme that produces the smallest shares possible. The motivation here is that if a secret is to be shared across multiple devices, we would like to require these devices to store as little information as possible. However, in such a setting, there is the assumption that the shares we are distributing are *new* pieces of information that each device must commit to its memory. If there was a way to somehow incorporate existing information on these devices into the secret sharing scheme, and only require each device to commit a very small amount of *new* information to store the share, then this seems just as good (modulo some user privacy concerns) as a secret sharing scheme with very small shares.

We develop this intuition into a new type of leakage-resilient secret sharing (LRSS) called *seeded leakage-resilient secret sharing*. In such a scheme, the Share function receives not only the secret that must be shared, but also some information from each device that will participate. The goal of Share is to output a small number of bits for each participating device such that when a device appends this new information onto their existing information, a secret sharing scheme is formed. In this scheme, a device's share becomes the concatenation of their old information with their new information.

If each participating device is storing some old information that is purely random, it is not difficult to do this job quite efficiently. While this is not a likely scenario, it *is* likely that a device is already storing bits that have *some* entropy, perhaps in the form of system log files. We will show that even if each device has a lot of old information with very low entropy, we can complete this with a very small (in fact, optimal) amount of new information in order to complete a leakage-resilient secret sharing scheme.

The old information on each device is referred to as a *seed*, and is represented by an (n, k) source. The amount of bits of *new* information that must be appended to each device is referred to as the *growth* of the scheme. We fully capture our new setting in the following formalism.

Definition 4.7. *Let* $\mathsf{Share} : \{0,1\}^m \times (\{0,1\}^n)^N \to (\{0,1\}^{m'})^N$ *be a randomized function, and let* $\mathsf{Rec} : (\{0,1\}^{n+m'})^N \to \{0,1\}^m$ *be a deterministic function. Furthermore, let* $\mathcal{F}$ *be a class of (deterministic) functions of the form* $f : (\{0,1\}^{n+m'})^N \to \{0,1\}^\mu$. *We say that* $(\mathsf{Share}, \mathsf{Rec})$ *is a* seeded leakage-resilient secret sharing scheme (S-LRSS) against $\mathcal{F}$ for seeds of type (n, k), with growth m' *and error* ε, *if given any* N (n, k) *sources* $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_N)$, *the maps* $(\mathsf{Share}^+, \mathsf{Rec}^+)$, *with* $\mathsf{Share}^+ : \{0,1\}^m \to (\{0,1\}^{n+m'})^N$ *and* $\mathsf{Rec}^+ : (\{0,1\}^{n+m'})^N \to \{0,1\}^m$ *and defined as*

$$\begin{aligned}\mathsf{Share}^+(\Psi) &:= ((\mathbf{X}_1, \mathsf{Share}(\Psi, \mathbf{X})_1), (\mathbf{X}_2, \mathsf{Share}(\Psi, \mathbf{X})_2), \dots, (\mathbf{X}_N, \mathsf{Share}(\Psi, \mathbf{X})_N)),\\ \mathsf{Rec}^+(\Psi) &:= \mathsf{Rec}(\Psi),\end{aligned}$$

are an N*-out-of-*N *leakage-resilient secret sharing scheme against* $\mathcal{F}$ *with error* ε.

By employing our leakage-resilient extractors against bounded collusion protocols for *low entropy*, we show that we can effectively move the requirement on share size from standard leakage-resilient secret sharing schemes to the requirement on *old information to be reused* in our new schemes. Furthermore, the old information that is harvested from each device barely has to look random at all. In particular, we can prove the following result.

Theorem 4.8. *There exists some constant universal constant* $C > 0$ *and an explicit S-LRSS* $(\mathsf{Share}, \mathsf{Rec})$, *of the form* $\mathsf{Share} : \{0,1\}^m \times (\{0,1\}^n)^N \to (\{0,1\}^m)^N$, $\mathsf{Rec} : (\{0,1\}^{n+m})^N \to \{0,1\}^m$, *against* $\mathsf{BCP}(p, \mu)$ *for seeds of type* (n, k) *with growth* m, *as long as* $k \geq \mathrm{polylog}(n)$ *and*

$$k \geq (m + 1 + \log(1/\varepsilon))^{C\cdot\frac{\log((N-1)/p)+1}{\log((N-1)/p)}} + \mu^{C\cdot\frac{\log((N-1)/p)+1}{\log((N-1)/p)}},$$

Proof. By Theorem 4.4, there exists an explicit leakage-resilient extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ against $\mathsf{BCP}(p, \mu)$ for entropy $k = \mathrm{polylog}(n)$ with error $\varepsilon \cdot 2^{-(m+1)}$ as long as $\mu < \xi, m \leq \xi$, and $\varepsilon 2^{-(m+1)} \geq 2^{-\xi}$, where $\xi = k^{\Omega(\frac{\log((N-1)/p)}{\log((N-1)/p)+1})}$. Rewriting these restrictions, it is easy to show that this all happens if

$$\begin{aligned}k &\geq \max\{\mu^{O(\frac{\log((N-1)/p)+1}{\log((N-1)/p)})}, m^{O(\frac{\log((N-1)/p)+1}{\log((N-1)/p)})}, (m + 1 + \log(1/\varepsilon))^{O(\frac{\log((N-1)/p)+1}{\log((N-1)/p)})}\}\\ &= (m + 1 + \log(1/\varepsilon))^{O\left(\frac{\log((N-1)/p)+1}{\log((N-1)/p)}\right)} + \mu^{O\left(\frac{\log((N-1)/p)+1}{\log((N-1)/p)}\right)}.\end{aligned}$$

Then, given $N - 1$ independent uniform random variables $\mathbf{Y}_1, \dots, \mathbf{Y}_N$ over $\{0,1\}^m$, we define Share as

$$\mathsf{Share}(\Psi, X_1, \dots, X_N) := \mathbf{Y}_1 \circ \mathbf{Y}_2 \circ \cdots \circ \mathbf{Y}_{N-1} \circ (\Psi \oplus \mathsf{Ext}(X_1, \dots, X_N) \oplus_{i\in[N-1]} \mathbf{Y}_i),$$

and we define Rec as

$$\mathsf{Rec}((X_1, Y_1), \dots, (X_N, Y_N)) := \mathsf{Ext}(X_1, \dots, X_N) \oplus_{i\in[N]} Y_i.$$

The result then follows immediately from the Definition 4.7 and the proof of Lemma 4.8. □

4.6 Applications in pseudorandomness

While the majority of this chapter has focused on leakage-resilient extractors, communication complexity, and secret sharing, we take a moment to recall a key application of leakage-resilient extractors in this thesis.

4.6.1 Extractors for adversarial sources

In Chapter 3, we constructed new state-of-the-art extractors for adversarial sources. As a reminder, an (N, K, n, k)-adversarial source is a sequence of N independent sources $\mathbf{X}_1, \dots, \mathbf{X}_N$ such that least K of them have min-entropy at least k. Such sources can capture unreliable sources from nature, our adversarial players participating in a protocol to generate a collective coin flip. By exploiting the *low-entropy* leakage-resilient extractors we constructed in this chapter (Theorem 4.2), we immediately obtained the following.

Theorem 4.9 (Theorem 3.1, repeated)**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \geq CN^\delta$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

Interestingly, beyond the new model of *seeded leakage-resilient secret sharing* presented in the previous section, this is the only known application of our leakage-resilient extractors that work for *low min-entropy*. Indeed, it is an interesting open question to find other applications of this powerful new object.

4.7 Future directions

In this chapter, we constructed a new, robust flavor of extractors for independent sources, known as a *leakage-resilient extractors*. Given a sequence of independent sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, such an extractor can not only produce a stream of uniformly random bits, but it can ensure that these bits *remain* uniform, even conditioned on leaks applied to some subset of the sources. Here, we studied leaks corresponding to *bounded collusion protocols* (BCPs), a rich spectrum of communication protocols that interpolate between the classical number-in-hand and number-on-forehead settings. As a result, our leakage-resilient extractors immediately yielded explicit lower bounds against this spectrum of protocols, in addition to new leakage-resilient secret sharing schemes. We believe that leakage-resilient extractors may be a fundamentally new type of pseudorandom primitive, and we suggest the following two directions.

Leakage-resilient extractors against NOF leakage While our leakage-resilient extractors for N independent sources can handle as little as *polylogarithmic min-entropy*, this is only for BCPs with collusion bound $p < N - 1$. In particular, for the setting of *number-on-forehead* leakage $p = N - 1$, we do not give any new constructions, and we still don't know how to handle min-entropy, say, $k = o(n)$. Given that such LREs would immediately imply better extractors for adversarial sources (Chapter 3), this seems like a very interesting direction to pursue. Furthermore, it is not clear how much harder this setting really is - indeed, perhaps our constructions rely too heavily on black-box tools, and a white-box approach may actually yield better LREs. More concretely, it would be interesting to see if *alternating extraction* [DP07] (a technique that is pre-baked into the two-source condensers we use) can be used to directly construct better LREs against NOF leakage.

Leakage-resilient extractors against other functions While this chapter studies LREs for independent sources against BCPs, it is natural to define and consider other types of LREs. For example, one might consider leakage classes $\mathcal{F}$ that can depend on *all* of the input sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, albeit in a computationally bounded way. In fact, if one considers leaks of the form $\mathsf{Ext}(f_1(\mathbf{X}_1), \dots, f_N(\mathbf{X}_N))$, where each f_i is a function with no fixed point and Ext is the LRE itself, then we immediately arrive at the definition of a *non-malleable extractor* [CG17]. This is a different type of robust independent source extractor, and its study has blossomed into a beautiful theory with connections to coding theory and beyond. Can a similarly useful theory be developed around LREs?

Chapter 5

Explicit extremal designs

Combinatorial designs are fundamental primitives in the theory of pseudorandomness. Given positive integers $n \geq r \geq s$, an (n, r, s)-design is simply a set system over n elements, where each set has size r, and any two sets intersect at $< s$ elements. In the language of hypergraphs, it is an r-uniform hypergraph over n vertices, with pairwise hyperedge intersections of size $< s$. While combinatorial designs have enjoyed many applications in pseudorandomness and beyond, these applications typically require designs that exhibit some sort of *extremal property* - that is, some property that is in direct tension with the *bounded intersection property* inherent to designs. Unfortunately, such designs can be difficult to construct.

Within pseudorandomness, the most famous applications of designs are in certain (celebrated) constructions of pseudorandom generators and seeded extractors. These applications require designs that are extremal in the sense that they have a large number of edges. Luckily, it is well-known how to efficiently construct such objects. In Chapter 3, we saw a new application of designs in the construction of extractors for adversarial sources. Unlike the prior two applications, these constructions require explicit designs that are extremal in the sense that they have a *small independence number*. Rödl and Šinajová proved the existence of such designs in 1994 [RŠ94], but prior to our work, no explicit constructions were known.

In this chapter, we give the first explicit construction of designs with small independence number. For all constants $r \geq s \in \mathbb{N}$ with r even, we explicitly construct (n, r, s)-designs $(G_n)_{n \in \mathbb{N}}$ with independence number $\alpha(G_n) \leq O(n^{\frac{2(r-s)}{r}})$. This matches, up to a square, the non-explicit designs of Rödl and Šinajová, which are known to be optimal up to constant factors. Our construction is elementary: we take a low-weight Hamming slice of a high-dimensional linear code, interpret it as a design, and analyze its independence number using recent bounds from additive combinatorics. This construction provides the last missing piece required to make our extractors for adversarial sources efficient, and it is an exciting open question to discover other applications of these new explicit extremal designs in the theory of pseudorandomness.

The results presented in this chapter are based on the following joint work:

- [CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021

5.1 Introduction

Part I of this thesis has largely focused on the explicit construction of (robust) extractors for independent sources. For the final chapter of this part, we turn our attention to the explicit construction of a different pseudorandom object: *combinatorial designs*. A combinatorial design is a special type of well-balanced set system, where each set has the same size, and no two sets intersect at too many points.

Definition 5.1 (Design)**.** *An* (n, r, s)-design*, or* (n, r, s)*-system, is an* r*-uniform hypergraph over* n *vertices, where any two distinct edges intersect at less than* s *vertices.*

Combinatorial designs have - time and again - popped up as key ingredients in a variety of pseudorandom constructions, and have gradually secured their spot as a fundamental primitive in this field. Perhaps the most notable application of designs is in the seminal work of Nisan and Wigderson [NW94], where they are used to construct *pseudorandom generators* (PRGs). In another landmark paper [Tre01], Trevisan demonstrated how designs can be used to construct excellent *seeded extractors*. Raz, Reingold, and Vadhan [RRV02] strengthened this connection even further, by showing that a slightly weaker notion of designs can be exploited to construct even better extractors. More recently, Li introduced a new hybrid object that combines seeded extractors with designs [Li12a], and used these objects to construct robust ultra-robust and ultra-efficient variants of classical objects in extractor theory [Li12a, CL18].

In all of the applications above (except the last one), the authors require designs that are *extremal*, in the sense that they have a very large number of hyperedges. While it is easy to show the existence of such designs via the probabilistic method, these applications (as always) require designs that are *explicit*. Luckily, in this case, it is easy to derandomize the probabilistic construction via *the method of conditional expectations* - see, for example [Vad12, Problem 3.2].

In Chapter 3, we gave a new application of designs that are extremal in a different sense: we showed that designs with *small independence number* can be used to construct state-of-the-art extractors for adversarial sources.[1] Unlike the first type of extremal designs, however, this second type of extremal designs are highly nontrivial to construct - *even using the probabilistic method*.[2] Luckily, in 1994, via a clever application of the Lovász Local Lemma, Rödl and Šinajová successfully showed that a random uniform hypergraph does indeed have small independence number - proving the existence of these designs for the very first time.

Theorem 5.0 ([RŠ94])**.** *Given any* $n \geq r \geq s \in \mathbb{N}$ *with* $r \geq 2$*, there exists an* (n, r, s)*-design* G *with independence number*

$$\alpha(G) \leq C_{r,s} \cdot n^{\frac{r-s}{r-1}} (\log n)^{\frac{1}{r-1}},$$

where $C_{r,s} = C(r, s)$ *depends only on* r, s*.*

Surprisingly, they also showed that this result is tight up to the term $C_{r,s}$. However, while their result proves the *existence* of extremal designs, it unfortunately does not provide an explicit way to construct them - and an explicit construction is needed if one hopes to apply these designs to construct other explicit objects. Most urgently, our application of these designs towards constructing adversarial source extractors absolutely requires these designs to be explicit. While a collection of work has emerged that improves on the result of Rödl and Šinajová [GPR95, EV13, Eus13, KMV14, TL18], all of these still rely on probabilistic constructions, and instead appear to focus on improving the term $C_{r,s}$ [EV13, Eus13] or extending the result to more general types of designs [GPR95, KMV14, TL18]. Thus, to date, there are no known explicit constructions of designs with small independence number.

[1] As a reminder, an independent set in a hypergraph $G = (V, E)$ is a subset of vertices $S \subseteq V$ that contains no edge, and the *independence number* α of a hypergraph is the size of its largest independent set.

[2] Moreover, it's a nice exercise to show extremal designs of the first type are not necessarily extremal designs of the second type.

5.2 Our results

In this chapter, we give the first explicit construction of designs with small independence number. This supplies the last missing ingredient needed to complete the construction of our adversarial source extractors, and provides the first efficient construction of a fundamental object from extremal combinatorics.

Theorem 5.1 (Explicit designs with small independence number)**.** *There exists an algorithm $\mathcal{A}$ such that given any $n \geq r \geq s \in \mathbb{N}$ as input with r an even number, $\mathcal{A}$ runs in time polynomial in $\binom{n}{r}$ and outputs an (n, r, s)-design G with independence number*

$$\alpha(G) \leq C_{r,s} \cdot n^{\frac{2(r-s)}{r}},$$

where $C_{r,s} = C(r, s)$ depends only on r, s.

For all constants $r \geq s \in \mathbb{N}$ where r is even, Theorem 5.1 gives an explicit family of (n, r, s)-designs $(G_n)_{n\in\mathbb{N}}$ with small independence number. The independence number of our explicit designs differs from the independence number of Rödl and Šinajová's non-explicit designs (Theorem 5.0, which is optimal up to constants) by just a square. Like their non-explicit designs, our explicit designs focus on the case where r, s are constant. However, we also give meaningful results when r, s are not constant, and show that we may take $C_{r,s} = Cr^4$ for some universal constant $C > 0$. It is also easy to extend Theorem 5.1 to work for odd r, at the expense of a small loss in parameters: indeed, to construct such an (n, r, s)-design G, we can simply construct an $(n, r + 1, s)$-design G' using Theorem 5.1 and remove an arbitrary vertex from each edge. The resulting (n, r, s)-design G will have independence number $\alpha(G) \leq C_{r+1,s} \cdot n^{\frac{2(r+1-s)}{r+1}}$.

As discussed, these explicit designs provide the last missing piece for our adversarial source extractors. More generally, however, we find these designs to be interesting in their own right, and we hope that our elementary construction (which exploits a new connection between extremal combinatorics and coding theory) provides further insight into them. Looking towards the future, it is exciting to think about where else these designs might find applications - in pseudorandomness, and beyond.

Organization

We present the explicit construction of our designs in Section 5.3, proving Theorem 5.1. Then, in Section 5.4, we provide a refresher on the application of these designs in our extractors for adversarial sources. Finally, we wrap up with some open problems in Section 5.5.

5.3 The design

In this section, we give our explicit construction of designs with small independence number, and thereby prove Theorem 5.1. Before we dive into the formal proof, we provide an overview of the construction.

An overview of the construction

In order to construct our (n, r, s)-designs $G = (V, E)$, we start with a linear code $Q \subseteq \mathbb{F}_2^n$ of distance $d > 2(r - s)$, and consider just the vectors $Q_r \subseteq Q$ of Hamming weight r. Our design $G = (V, E)$ is constructed by identifying V with $[n]$, and creating a hyperedge for each $x \in Q_r$ in the natural way. The distance of the code and the definition of Q_r immediately guarantee that G is an (n, r, s)-design.

Now, in order to upper bound the independence number $\alpha(G)$ of our design, we observe that any independent set in G corresponds to a subcube $A \subseteq \mathbb{F}_2^n$ that has no vector in Q_r. Since Q is a *linear* code, this means that the *subspace* $A^* := A \cap Q$ also has no vector of Hamming weight r. If our linear code Q had very high dimension, then even if the subcube A was relatively small, we would have found a relatively large subspace A^* containing no vector of Hamming weight r. But intuitively, it seems like as the dimension of a subspace grows large enough, at some point it must be guaranteed to have such a vector. It turns out this is true, and follows immediately from Sidorenko's recent bounds [Sid18] on the length of sequences in $\mathbb{F}_2^n$ with no subsequence of length r that sums to zero. Thus if Q has large enough dimension, then A cannot be too large, and thus neither can $\alpha(G)$. All that remains is to explicitly construct (the weight r vectors of) a high-dimensional linear code $Q \subseteq \mathbb{F}_2^n$ with distance $d > 2(r-s)$, which can easily be done using BCH codes [BRC60, Hoc59].

5.3.1 A simple construction via linear codes and Erdős-Ginzburg-Ziv constants

We now proceed with the full construction of our designs, and our formal proof of Theorem 5.1.

From designs to codes

We start with the simple observation that hypergraphs over n vertices can be identified with subsets of $\mathbb{F}_2^n$. In particular, notice that any subset $T \subseteq \mathbb{F}_2^n$ induces a hypergraph $G_T = (V, E)$ by identifying $[n]$ with V, and identifying each $x \in T$ with an edge $e \in E$ containing all i such that $x_i = 1$. Using this correspondence, we can instead focus on constructing special subsets of $\mathbb{F}_2^n$, and thereby leverage the tools of linear algebra and coding theory. Now, to obtain our designs, we just need to explicit construct a subset $T \subseteq \mathbb{F}_2^n$ such that:

1. G_T is an (n, r, s)-design, and
2. G_T has small independence number.

In order to make sure both of these things happen, we can use the following two facts, which describe how the above hypergraph properties translate to properties of the corresponding subsets in $\mathbb{F}_2^n$.

Fact 5.1 (The design property). *For any subset $T \subseteq \mathbb{F}_2^n$, it holds that G_T is an (n, r, s)-design if and only if: (i) every $x \in T$ has $\Delta(x) = r$, and (ii) any two distinct $x, y \in T$ have $\Delta(x, y) > 2(r-s)$.*

Proof. The two conditions are sufficient because (i) guarantees that G_T will be r-uniform, and (ii) guarantees that any two edges in G_T intersect at $< s$ points. They are both necessary because if (i) does not hold then G_T will not be r-uniform, and if (i) holds but (ii) does not, then some two edges share $\geq s$ points. □

Fact 5.2 (The small independence number property). *For any subset $T \subseteq \mathbb{F}_2^n$, the hypergraph G_T has $\alpha(G_T) < \ell$ if and only if every subcube $A \subseteq \mathbb{F}_2^n$ of dimension at least ℓ has at least one element in T.*

Proof. If $\alpha(G_T) \geq \ell$, there is an independent set $S \subseteq V = [n]$ of size at least ℓ, and thus the subcube $A := \mathsf{span}(\{e_i\}_{i \in S})$ of dimension ℓ has no points in T. If there is a subcube $A \subseteq \mathbb{F}_2^n$ of dimension ℓ with no points in T, the set $S \subseteq [n]$ indexing the standard basis vectors that span A must have size ℓ and constitute an independent set in G_T. □

Given the above facts, we can fully translate our task of constructing an (n, r, s)-design G with small independence number into the equivalent task of constructing a subset $T \subseteq \mathbb{F}_2^n$ with the following properties.

1. Every element in T has Hamming weight r, and any two distinct elements in T have Hamming distance $> 2(r - s)$.

2. Every subcube A of dimension $> \alpha$ has an element in T.

In order to construct a set $T \subseteq \mathbb{F}_2^n$ with these properties, we use connections to *coding theory* and *zero-sum problems*. Recall that an (n, k, d)-code is a subset $Q \subseteq \mathbb{F}_2^n$ of size 2^k such that any two distinct $x, y \in Q$ have Hamming distance $\Delta(x, y) \geq d$. Thus, if we take any (n, k, d)-code $Q \subseteq \mathbb{F}_2^n$ with $d > 2(r-s)$ and let Q_r denote the weight r elements of this code, then Q_r satisfies the first property above. In order to endow Q_r with the second property, we need to start with some code Q such that any relatively large subcube A has some element in Q_r. In other words, if we let Δ_r denote the set of all weight r vectors in $\mathbb{F}_2^n$, we want

$$A \cap Q_r = A \cap (Q \cap \Delta_r) = (A \cap Q) \cap \Delta_r \neq \emptyset.$$

Towards this end, the trick here is to start with a *linear* code Q. Recall that a *linear* $[n, k, d]$ code $Q \subseteq \mathbb{F}_2^n$ is simply an (n, k, d) code that is also a subspace. Given this, the condition $(A \cap Q) \cap \Delta_r \neq \emptyset$ requested above becomes much more concrete: since Q is a subspace, $A \cap Q$ is also a subspace, and thus we can make sure $(A \cap Q) \cap \Delta_r \neq \emptyset$ as long as we can show that

- $A \cap Q$ is big enough, and

- Every big enough subspace contains a vector of Hamming weight r.

Thus, to complete the proof, we roughly need to find a big enough linear code Q such that $d > 2(r-s)$, and we need to show that every big enough subspace contains a vector of Hamming weight r. To make things a little bit more formal, let $\Lambda_r(n)$ denote the dimension of the largest subspace in $\mathbb{F}_2^n$ containing no vector of Hamming weight r. We summarize our discussion above with the following formal lemma.

Lemma 5.1. *If $Q \subseteq \mathbb{F}_2^n$ is a linear $[n, k, d]$-code with $d > 2(r - s)$, then the hypergraph $G_{Q\cap\Delta_r}$ is an (n, r, s)-design with independence number α that obeys the following inequality:*

$$\alpha \leq (n - k) + \Lambda_r(\alpha)$$

Proof. It follows immediately from Fact 5.1 that $G_{Q\cap\Delta_r}$ is an (n, r, s)-design. By Fact 5.2, there is a subcube $A = span(e_{i_1}, \ldots, e_{i_\alpha}) \subseteq \mathbb{F}_2^n$ of dimension α that does not intersect $Q \cap \Delta_r$. Thus, if we define $A' := A \cap Q$, then A' contains no vector of Hamming weight r, and furthermore it has dimension $dim(A') = dim(A \cap Q) \geq dim(A) + dim(Q) - n = \alpha + k - n$. Notice now that if we define the projection $\pi : \mathbb{F}_2^n \to \mathbb{F}_2^\alpha$ as the map $(x_1, \ldots, x_n) \mapsto (x_{i_1}, \ldots, x_{i_\alpha})$, then the subset $\pi(A')$ is still a subspace (albeit now of $\mathbb{F}_2^\alpha$) of dimension $dim(\pi(A')) \geq \alpha + k - n$ containing no vector of Hamming weight r. Thus, by definition of Λ_r, it must hold that $\alpha + k - n \leq dim(\pi(A')) \leq \Lambda_r(\alpha)$. □

In what remains, we get good upper bounds on $n - k$ and $\Lambda_r(\alpha)$. Namely, we demonstrate an explicit linear code with very high dimension k (i.e., very low redundancy $n-k$), and we show that every big enough subspace contains a vector of Hamming weight r. We start with the latter.

Every big enough subspace contains a vector of Hamming weight r

Getting a good upper bound on $\Lambda_r(n)$ is closely related to the theory of *zero-sum problems*, from the fields of additive combinatorics and additive number theory. Here, the primary objects of study are the (generalized)

Erdős-Ginzburg-Ziv constants of a finite abelian group. If we consider the additive group of $\mathbb{F}_2^n$ and any even number $r \leq n$, this constant is defined as the smallest integer $s_r(n)$ such that any sequence of $s_r(n)$ (not necessarily distinct) elements of $\mathbb{F}_2^n$ contains a subsequence of length r that sums to zero.

As it turns out, the Erdős-Ginzburg-Ziv constant $s_r(n)$ is intimately related to $\Lambda_r(n)$, the dimension of largest subspace of $\mathbb{F}_2^n$ with no vector of Hamming weight r. This relationship is captured by the following fact, which has been observed in prior work (e.g., [Sid20]). We include its proof here, for completeness.

Fact 5.3. *For every $n \geq r \in \mathbb{N}$ where r is even,*

$$s_r(n - \Lambda_r(n)) > n.$$

Proof. Let $R \subseteq \mathbb{F}_2^n$ be a subspace of dimension $k := \Lambda_r(n)$ containing no vector of Hamming weight r. Let $M \in \mathbb{F}_2^{(n-k)\times n}$ be a matrix such that $R = \{x \in \mathbb{F}_2^n : Mx = 0\}$. Since R has no vector of weight r, it follows that no r columns of M sum to zero. Thus the columns of M are a sequence of n values in $\mathbb{F}_2^{n-k}$ containing no subsequence of length r that sums to zero. This means that $s_r(n-k) > n$, as desired. □

In other words, the above fact demonstrates that an upper bound on $s_r(n)$ immediately gives an upper bound on $\Lambda_r(n)$. With this connection in hand, all we need is a good upper bound on $s_r(n)$. In 2018, Sidorenko provided a very strong bound of this type:

Theorem 5.2 ([Sid18, Theorems 4.1 and 4.4])**.** *There is a universal constant $C > 0$ such that for every $n, r \in \mathbb{N}$ where r is even,*

$$s_r(n) \leq C \cdot r^3 \cdot 2^{2n/r}.$$

By plugging this bound into Fact 5.3, we immediately obtain the following.

Corollary 5.1. *There is a universal constant $C > 0$ such that for any $n \geq r \in \mathbb{N}$ where r is even, the largest subspace $S \subseteq \mathbb{F}_2^n$ with no vector of Hamming weight r has dimension*

$$\Lambda_r(n) \leq n - (r \log n - 3r \log r - r \log C)/2.$$

Thus, we have arrived at a nice upper bound on $\Lambda_r(n)$, the dimension of the largest subspace of $\mathbb{F}_2^n$ with no vector of Hamming weight r, as desired. To summarize the discussion above, this bound is immediate from Sidorenko's strong bounds on the *Erdős-Ginzburg-Ziv constants* of the additive group of $\mathbb{F}_2^n$ (Theorem 5.2).

A big enough (explicit) linear code

We are now almost ready to complete the construction of our designs. First, by exploiting the bounds on $\Lambda_r(n)$ from above, we are able to prove our main design lemma, which reduces the explicit construction of designs with small independence number to the explicit construction of a high dimensional linear code.

Lemma 5.2 (A reduction from extremal designs to high-dimensional linear codes)**.** *There is a universal constant $C > 0$ such that for every $n \geq r \geq s$ with r even, if $Q \subseteq \mathbb{F}_2^n$ is a linear $[n, k, d]$-code with $d > 2(r - s)$, then $G_{Q\cap\Delta_r}$ is an (n, r, s)-design with independence number*

$$\alpha(G_{Q\cap\Delta_r}) \leq C \cdot r^3 \cdot 2^{2(n-k)/r}.$$

Proof. Simply plug the bound on $\Lambda_r(\alpha)$ from Corollary 5.1 into Lemma 5.1. □

Given the above reduction, all we need to complete the proof of Theorem 5.1 is an explicit construction of linear codes with very high dimension. 60 years ago, Bose, Ray-Chaudhuri [BRC60], and Hocquenghem [Hoc59] explicitly constructed codes of exactly this type.[3] In particular, they proved the following theorem.

Theorem 5.3 ([BRC60, Hoc59])**.** *For every $m, t \in \mathbb{N}$, there exists an $[n, k, d]$-linear code $\mathbf{BCH}_{m,t} \subseteq \mathbb{F}_2^n$ with block length $n = 2^m - 1$, dimension $k \geq n - mt$, and distance $d > 2t$. Furthermore, there exists an Algorithm $\mathcal{B}$ that given any $m, t \in \mathbb{N}$ and $x \in \mathbb{F}_2^n$ as input, checks if $x \in \mathbf{BCH}_{m,t}$ in $\operatorname{poly}(n)$ time.*

By instantiating our reduction (Lemma 5.2) with these codes (Theorem 5.3), we can finally complete the construction of our designs, and prove Theorem 5.1.

Proof of Theorem 5.1. We start by assuming that $n = 2^m - 1$ for some $m \in \mathbb{N}$. Then, we let $t = r - s$, and use Theorem 5.3 to define the $[n, k, d]$-linear code $Q := \mathbf{BCH}_{m,t} \subseteq \mathbb{F}_2^n$, where $k \geq n - mt = n - m(r-s)$ and $d > 2t = 2(r - s)$. Algorithm $\mathcal{A}$ will simply output the hypergraph $G_{Q \cap \Delta_r}$. By Lemma 5.2, we know that $G_{Q\cap\Delta_r}$ is an (n, r, s)-design with independence number

$$\alpha(G_{Q\cap\Delta_r}) \leq C \cdot r^3 \cdot 2^{2(n-k)/r} \leq C \cdot r^3 \cdot 2^{2mt/r} = C \cdot r^3 \cdot (2^m)^{2(r-s)/r} \leq 2C \cdot r^3 \cdot n^{2(r-s)/r}.$$

Furthermore, note that $G_{Q\cap\Delta_r}$ can be constructed in $\operatorname{poly}(\binom{n}{r})$ time if $Q \cap \Delta_r$ can be constructed in $\operatorname{poly}(\binom{n}{r})$ time, and this can be done by simply checking (and appropriately including) whether each of the $\binom{n}{r}$ elements in Δ_r belong to Q, using Algorithm $\mathcal{B}$ from Theorem 5.3.

If n is of the form 2^m, we can follow the previous procedure to draw hyperedges around the first $n - 1$ vertices, and then add one more isolated vertex (contained in no edges) at the end to finish the hypergraph. Clearly we will still have $\alpha(G_{Q\cap\Delta_r}) \leq 3C \cdot r^3 \cdot n^{2(r-s)/r}$.

If n is not of the form $2^m - 1$ nor 2^m, then it can be written as a sum $x_0 2^0 + \cdots + x_d 2^d$ over $\mathbb{N}$, where $d = \lceil \log n \rceil$ and each $x_i \in \{0, 1\}$. We can then follow the most recent procedure to construct a graph G_i over 2^i vertices separately for each nonzero x_i. The final graph $G = \bigcup_i G_i$ is clearly still an (n, r, s)-design, and it has independence number

$$\begin{aligned}\alpha(G) = \sum_{i:x_i=1} \alpha(G_i) \leq \sum_{0\leq i\leq\lceil\log n\rceil} 3C \cdot r^3 \cdot (2^i)^{2(r-s)/r} &= 3C \cdot r^3 \sum_{0\leq i\leq\lceil\log n\rceil} (2^i)^{2(r-s)/r} \\ &= 3C \cdot r^3 \cdot \frac{(2^{\lceil\log n\rceil+1})^{2(r-s)/r} - 1}{2^{2(r-s)/r} - 1}.\end{aligned}$$

It is straightforward to verify that for a large enough universal constant C', the above fraction is bounded above by $C' \cdot r \cdot n^{2(r-s)/r}$, which completes the proof. □

On the efficiency of our construction

To conclude this section, it is worth making a brief remark on the efficiency of our construction. As claimed in Theorem 5.1 and proven above, our (n, r, s)-design G is constructed in time $\operatorname{poly}(\binom{n}{r})$. Should this be regarded as *efficient*? Well, if r is a constant, then we construct an n-vertex graph in time $\operatorname{poly}(n)$. By almost all definitions of efficiency, this would indeed qualify. But what if r is superconstant?

In the case of superconstant r, our algorithm is clearly not guaranteed to run in $\operatorname{poly}(n)$ time. However, perhaps this is to be expected: for example, if the algorithm is outputting a superpolynomial number m of hyperedges, then we certainly should not force it to run in $\operatorname{poly}(n)$ time in order to classify it as efficient. Indeed, a more reasonable way to check whether the algorithm is efficient is to confirm that it runs in $\operatorname{poly}(n, m)$ time. So, when does this happen? We make the following claim.

[3] See [GB10] for a great exposition of these codes, which are known as *BCH codes*.

Claim 5.1 (Explicitness of Theorem 5.1 for superconstant r)**.** *Suppose there is a constant $\varepsilon > 0$ such that $\alpha(G)$ in Theorem 5.1 is bounded above by $O(n^{1-\varepsilon})$. Then the algorithm $\mathcal{A}$ runs in time* $\mathrm{poly}(n, m)$.

Indeed, the above claim says that as long as Theorem 5.1 gives a nontrivial bound on the independence number in the first place, then the algorithm will run efficiently. This effectively covers all interesting regimes of r, s: indeed, the main application of selecting non-constant r, s would be to achieve independence bounds that are stronger than those achieved by constant r, s - which, in particular, are always of the form $\alpha(G) \le O(n^{1-\varepsilon})$, provided that they are less than n at all. Now, let us turn to prove Claim 5.1.

Proof of Claim 5.1. We use standard bounds on Turán numbers. The Turán number $T(n, \beta, r)$ is defined as the fewest number of edges in an r-uniform hypergraph with no independent set of size β, and it is known [Sid95] that $T(n, \beta, r) \ge \binom{n}{r}/\binom{\beta}{r}$. Thus, the fact that $\alpha(G) \le O(n^{1-\varepsilon})$ implies the number of edges m in the design is at least

$$T(n, Cn^{1-\epsilon} + 1, r) \ge T(n, n^{1-\epsilon/2}, r) \ge \binom{n}{r}/\binom{n^{1-\epsilon/2}}{r} \ge \binom{n}{r}/\binom{n}{r}^{1-\epsilon/4} \ge \binom{n}{r}^{\epsilon/4},$$

where we use the observation that the Turán number is non-increasing in its second argument, the fact that we can assume n, r are sufficiently large (since otherwise the efficiency claim is trivial), and an application of Stirling's formula. Thus, Algorithm $\mathcal{A}$ runs in time $\binom{n}{r} = \mathrm{poly}(n, m)$. In fact, since we gave a lower bound on m based on the independence number, it trivially holds that *any* algorithm that achieves independence numbers as small as $\mathcal{A}$ must output m edges, meaning the runtime of $\mathcal{A}$ is optimal up to constant powers. □

5.4 Applications in pseudorandomness

Our designs are built on error-correcting codes, which are inherently pseudorandom objects. In this section, we briefly review how our explicit designs contribute something back to the field of pseudorandomness.

5.4.1 Extractors for adversarial sources

The main application of our explicit extremal designs is to the construction of *extractors for adversarial sources*. As a reminder, an (N, K, n, k)-adversarial source is a sequence of N independent sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, each over n bits, with the guarantee that at least K of them have min-entropy at least k. In Chapter 3, we saw how explicit designs with small independence number (in particular, those constructed in Theorem 5.1) can be used to obtain the following improved extractors for adversarial sources.

Theorem 5.4 (Theorem 3.1, repeated)**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \ge CN^\delta$ good sources of min-entropy $k \ge \log^C n$, which has output length $m \ge k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

Moreover, we will see in Chapter 7 that these new extractors for adversarial sources play a big role in constructing new state-of-the-art extractor for small-space sources. Prior to the construction of our designs, the previous best extractors for adversarial sources required at least $K \ge N^{1/2+o(1)}$ of the sources to be good [CGGL20], and our new designs were a crucial ingredient in bringing this down to $K \ge N^\delta$. For more details about how this construction works, we refer the reader to Chapter 3.

5.5 Future directions

Combinatorial designs are fundamental building blocks in the construction of various pseudorandom objects. Historically, these applications have exploited designs that are extremal in the sense that they have *many edges*. Recently, a new application of designs was discovered in the construction of *adversarial source extractors* (Chapter 3). However, this application requires designs that are extremal in a different sense, in that they have *small independence number*. In this chapter, we gave explicit constructions of such designs, which are optimal up to a square (Theorem 5.1). Given that these are brand new pseudorandom objects, a few natural questions immediately spring to mind.

Improved explicit constructions Even though our explicit construction is optimal up to a square, there are still certain parameter regimes where our main theorem becomes trivial. In particular, our algorithm is unable to construct an explicit (n, r, s)-design $G = (V, E)$ with independence number $\alpha(G) < n$ whenever $s < r/2$: namely, whenever the hyperedges in the design are forced to have a pairwise intersection of *rate* less than half. A recent elegant work of Liu and Mubayi [LM22] successfully construct a handful of designs in this regime (by exploiting certain types of hypergraph products), but much work still remains to be done.

Other applications of designs with small independence number We find it exciting to have an explicit construction of a new pseudorandom object, which appears to be of a totally different flavor from those constructed in the past. It is an interesting open direction to figure out where else these designs might be useful. Some candidate application domains might be in the fields of *property testing* or *coding theory*. In particular, reviewing our code-based construction of designs, it is easy to verify that such an approach will always run into a barrier upon trying to construct (n, r, s)-designs with $s < r/2$. Thus, while the construction of our designs borrows some results from the field of coding theory, perhaps an improved construction (that achieves parameters that go beyond what is possible with a coding-theoretic approach) could instead *contribute something back* to the field of coding theory. For example, perhaps such designs could be used to construct new, more robust types of codes.

Part II

Extractors for algebraic sources

Chapter 6

Extractors for affine sources

Beyond the model of *independent sources*, one of the most classical settings in which to study seedless extraction is the model of *algebraic sources*. Instead of assuming access to several independent sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, we now assume access to just a single source $\mathbf{X}$ that exhibits some sort of *algebraic structure*. Here, the most well-studied setting is the model of *affine sources*, whose origins can be traced back over three decades.

An affine source $\mathbf{X}$ is simply a random variable that is uniform over some *affine subspace* of $\mathbb{F}_2^n$, and its min-entropy k corresponds exactly to the dimension of this subspace. Over the years, *extractors for affine sources* have emerged as a fundamental primitive in the theory of pseudorandomness, as they have found a growing number of applications in complexity theory and beyond (in fact, the current state-of-the-art circuit lower bounds are for affine extractors!). Despite this growing influence, the best-known affine extractors lag behind the best-known independent-source extractors in terms of their min-entropy requirement.

In this chapter, we construct extractors for affine sources that have a *near-optimal* min-entropy requirement. Prior to our work, the best-known extractors for affine sources required min-entropy $k \geq \log^C n$ for some large constant C. In our work, we show how to extract from affine sources with min-entropy $k \geq \log^{1+o(1)} n$, which is close to the optimal requirement of $k \geq O(\log n)$. In order to achieve this improvement, we take inspiration from an exciting line of work that achieves a similar improvement in the independent source setting. However, as we have only a single (affine) source to work with, we must develop a significant amount of new machinery to make everything go through. We exploit two key new ingredients (both of which can be viewed as powerful twists on the classical seeded extractor): (i) a linear seeded *somewhere extractor* for affine sources; and (ii) a *correlation breaker* for linearly correlated sources.

The results presented in this chapter are based on the following joint work:

- [CGL21] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 622–633. IEEE, 2021

6.1 Introduction

When it comes to popular settings in which to study seedless extraction, the *independent source* model may very well hold the crown. However, over the past few decades, the *algebraic source* model has quickly gained ground, becoming one of the most prominent models in all of extractor theory. In this part of the thesis, we shift our focus to this important setting. To get started, let's recall the definition of an extractor.

Definition 6.1 (Extractor). *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is called an* extractor for $\mathcal{X}$ with error ε *if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \varepsilon.$$

While the independent source model assumes that each source $\mathbf{X} \in \mathcal{X}$ consists of several independent sources $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$, the algebraic source model assumes that each source $\mathbf{X} \in \mathcal{X}$ exhibits some sort of *algebraic structure*. At first glance, this may appear as an artificial setting in which to study randomness extraction. However, over the years, *extractors for algebraic sources* have gradually proven themselves to be fundamental primitives in the theory of pseudorandomness, demonstrating surprising applications in cryptography, complexity theory, pseudorandomness, and more.

The origins of algebraic sources can be traced back to the founding papers of extractor theory, via the model of *oblivious bit-fixing sources* [Vaz87b, BBR88, CFG+85].[1] An oblivious bit-fixing source $\mathbf{X} \sim \{0,1\}^n$ consists of n independent bits, but only k of them are guaranteed to be uniform (while the rest are fixed to constants). Despite this simplicity, oblivious bit-fixing sources have become a classical setting in seedless extraction, finding applications in distributed computing [CFG+85], exposure-resilient cryptography [CDH+00, Dod00, KZ07], and complexity theory [GVW15].[2] In light of this, a long line of work has focused on constructing extractors for this setting [Vaz87b, BBR88, CFG+85, CW89, Fri92, KJS01, BS00, KZ07, GRS06, Sha08, Rao09b, GVW15, CL18], and we now have near-optimal constructions [Rao09b].

Given the great success of constructing extractors for oblivious bit-fixing sources, it is only natural to ask if we can achieve similar results for a more general model. Starting in 2001, researchers began to investigate exactly such a model, known as *affine sources* [BSHR+01, BKS+10, GR08].[3] Intuitively, an affine source $\mathbf{X}$ is a random variable that is uniform over some affine subspace of $\mathbb{F}_2^n$. Formally, they are defined as follows.

Definition 6.2 (Affine source). *A source $\mathbf{X} \sim \mathbb{F}_2^n$ is called an* affine source of dimension k *if there exists some matrix $A \in \mathbb{F}_2^{n\times k}$ with full column rank and a vector $b \in \mathbb{F}_2^n$ such that*

$$\mathbf{X} = A\mathbf{U}_k + b,$$

where $\mathbf{U}_k$ is a uniform random variable over $\mathbb{F}_2^k$.

Since their introduction, affine sources have drawn a tremendous amount of interest [BSHR+01, BKS+10, Bou07, BSK12, Rao09b, Yeh11, Li11b, Sha11a, CT15, Li16, CGL21, CL22, HIV22, Li23], and have enabled key new developments in both extractor theory [DW12, Vio14, BSRZ15, CG21] and complexity theory [DK11, FGHK16, LY22]. More generally, the introduction of affine sources ushered in a new era of randomness extraction from *algebraic sources*, where several exciting generalizations have been explored.

[1] These sources were first suggested by Vazirani [Vaz87b], but were not studied in detail until the works of Bennett, Brassard, and Robert [BBR88] and Chor, Friedman, Goldreich, Håstad, Rudich, and Smolensky [CFG+85].

[2] In fact, oblivious bit-fixing sources also have applications in extractor theory itself, as they also capture the oldest model studied in randomness extraction [vN51]. This was shown in [ACG+22], and we refer the reader to Section 8.4.2 for a proof.

[3] Other models that generalize oblivious bit-fixing sources have been studied [BL85, LLS89, KKL88, CW89], but affine sources are the most well-studied one.

Generalizations The most natural generalization extends Definition 6.2 to allow the affine source $\mathbf{X}$ to be uniform over an affine subspace of $\mathbb{F}^n$, where $\mathbb{F}$ can be *any* finite field (instead of just $\mathbb{F} = \mathbb{F}_2$). Several works have examined this direction [GR08, DG10, BGLZ15, BDL16, GVJZ23], yielding very surprising results.[4] A number of works have also considered pushing this generalization even further, to sources that are defined via *polynomials* (instead of just *degree* 1 maps). Here, two families have been considered.

- *Polynomial sources* [DGW09, BSG12], where the source $\mathbf{X}$ is the the output of some polynomial $p : \mathbb{F}^\ell \to \mathbb{F}^n$ applied to a uniform random variable $\mathbf{U} \sim \mathbb{F}^\ell$.
- *Variety sources* [Dvi12, CT15, Rem16, LZ19], where the source $\mathbf{X}$ is uniformly distributed over the solutions to some polynomial $p : \mathbb{F}^n \to \mathbb{F}^\ell$.

Finally, a very recent paper introduced the family of *image of variety* sources [GVJZ23], a model which captures both polynomial sources and variety sources, pushing these generalizations to the extreme.

Our focus In this chapter, we return to the most classical setting in algebraic extraction, and study affine sources over the field $\mathbb{F} = \mathbb{F}_2$ (Definition 6.2). Within the world of affine sources, this is considered the most challenging setting,[5] and the only one where the min-entropy of the source $\mathbf{X}$ directly corresponds to the dimension of the underlying subspace. Furthermore, given the natural correspondence between the vector space $\mathbb{F}_2^n$ with the collection of bitstrings $\{0, 1\}^n$, affine sources over $\mathbb{F}_2$ seem to strike a good balance between the worlds of algebra and computer science - and, indeed, from this balance is born a rich array of applications. In what follows, we highlight some of these applications, review what is known about affine extractors, and state the goal of this chapter.

Applications Over the years, affine sources (over $\mathbb{F}_2$) have emerged as a unifying model in seedless extraction. Returning to our original motivation, affine sources not only generalize *oblivious bit-fixing sources* [Vaz87b, BBR88, CFG$^+$85],[6] but can even be applied to extract from some basic *variety sources* [CT15]. Recently, a deep connection has also been established [DW12, Vio14, CG21] between affine sources and *samplable sources* [TV00]. Here, it was shown that extractors for affine sources also work for *circuit sources* [DW12, Vio14] and *small-space sources* [CG21].[7] Finally, for the *independent source* model, a beautiful paper of Ben-Sasson and Ron-Zewi [BSRZ15] proved (under plausible conjectures in additive combinatorics) that affine extractors can be used to construct *low-error two-source extractors* - a result that could help break longstanding barriers in the field [CG88, Bou05, Lew19].

Needless to say, affine extractors are a valuable currency in the land of seedless extraction. Surprisingly, however, they also have profound applications in *complexity theory*. As it turns out, affine extractors not only yield the best-known lower bounds against various specialized complexity classes [CS16b], but they also yield the best-known lower bounds against *general circuits* [DK11, FGHK16, LY22]. This connection was first discovered by Demenkov and Kulikov [DK11], and exploited a few years later in the breakthrough work of Find, Golovnev, Hirsch, and Kulikov [FGHK16], which smashed a 30-year-old record in circuit lower bounds. Today, the current state-of-the-art circuit lower bounds are still for affine extractors [LY22], and many of the most promising paths forward are based on the explicit construction of their algebraic cousins (namely, extractors for variety sources) [HR15, GK16, GKW21].

[4] For large enough fields, it turns out that extraction is possible even from *lines* (affine sources with dimension $k = 1$) [GR08]!

[5] Extractors for affine sources over $\mathbb{F}^n$ generally become more difficult to construct as the size of the field $\mathbb{F}$ becomes smaller, with $\mathbb{F} = \mathbb{F}_2$ representing the most challenging setting.

[6] This generalization actually played a key role in constructing the best extractors for oblivious bit-fixing sources [Rao09b].

[7] We discuss this connection in detail in Section 6.4.

Prior work For the reasons highlighted above, we direct our focus towards constructing affine extractors over $\mathbb{F}_2^n$. However, before we review what is known about these objects, we would be remiss to ignore a beautiful body of work on extracting from affine sources over larger fields.

The large field setting The first affine extractors over large fields $\mathbb{F}_q$ were obtained by Gabizon and Raz [GR08], who showed how to successfully extract from *lines*, or affine sources of dimension $k = 1$. Their extractors required a field of size roughly $q = O(n^2)$, and a later work [BGLZ15] reduced this requirement to $q = O(n)$. Meanwhile, an alternative construction of DeVos and Gabizon [DG10] showed how to reduce the field size q even further, at the expense of increasing the dimension k of the affine source. Here, they achieved a tradeoff of the form $q = O((n/k)^2)$, while also requiring the field to have large characteristic. All of the constructions above have relatively high error, which is (at best) of the form $\varepsilon = q^{-\Omega(1)}$.[8] In 2016, Bourgain, Dvir, and Leeman presented a construction that achieves exponentially small error $\varepsilon = q^{-\Omega(k)}$, at the expense of a larger field size of $q = n^{O(\log \log n)}$. Furthermore, their construction only worked for "typical" primes q, or rather primes q for which $q - 1$ doesn't have too many prime factors. In a subsequent work [GVJZ23], this restriction was successfully removed.

The small field setting We now turn to affine extractors over $\mathbb{F}_2$. As discussed, this is the most challenging setting in which to construct these objects. Here, it is no longer possible to extract from affine sources of dimension $k = 1$, and in fact this task remains impossible even for dimension $k = \log(n) - O(1)$ [CS16b, ABGSD21]. *Non-explicitly*, on other hand, a standard application of the probabilistic method shows the existence of affine extractors (over $\mathbb{F}_2$) for min-entropy $k \geq O(\log n)$ with extremely low error $\varepsilon = 2^{-\Omega(k)}$.[9] *Explicitly*, the best known extractors for affine sources (prior to our work) are as follows.

- *The low-error setting*: There exist explicit extractors for min-entropy $k \geq O(n/\sqrt{\log \log n})$.

  For the special case of *low-weight* affine sources,[10] the best extractors require $k \geq \operatorname{polylog}(n)$.

- *The high-error setting*: There exist explicit extractors for min-entropy $k \geq \operatorname{polylog}(n)$.

Just like with the independent source model, these constructions are the culmination of many years of work. We briefly review this history, below.

The first explicit affine extractors appeared in 2001 [BSHR$^+$01], when it was shown that the *inner product (mod 2)* function extracts from affine sources of min-entropy $k \geq 0.501n$ with error $\varepsilon = 2^{-\Omega(k)}$.[11] Several years later, this entropy requirement was significantly improved, when an explicit *disperser* appeared [BKS$^+$10] that could handle min-entropy $k \geq 0.01n$. Following this result, an exciting line of work by Bourgain [Bou07], Yehudayoff [Yeh11] and Li [Li11b] matched this entropy requirement for *low-error extractors*, and in fact constructed explicit affine extractors that could handle even less min-entropy $k \geq O(n/\sqrt{\log \log n})$ with exponentially small error. This remains the best known low-error extractor for general affine sources, while a fascinating construction of Rao [Rao09b] shows how to reduce the entropy requirement to $k \geq \operatorname{polylog}(n)$ for the special class of affine sources with a low-weight basis.[12]

In the high-error regime, significantly more progress has been made. At STOC 2009, Ben-Sasson and Kopparty [BSK12] constructed an affine disperser for min-entropy $k \geq O(n^{4/5})$. This was later improved

[8] In fact, the field sizes q claimed above must be raised to the power of (roughly) $1 + \Omega(1)$ to actually achieve this error.

[9] Recall that in the setting $\mathbb{F} = \mathbb{F}_2$, the dimension of an affine source is equal to its min-entropy.

[10] A low-weight affine source is uniform over an affine subspace whose basis consists of low (Hamming) weight vectors.

[11] Suspiciously, recall that this function achieves the same parameters for the independent source model!

[12] Rao's extractor handled basis vectors of weight $w \leq k^{0.01}$ [Rao09b], which was later optimized to $w \leq k^{0.99}$ [DW12, Vio14].

by Shaltiel [Sha11a], who constructed a disperser for min-entropy $k \geq 2^{(\log n)^{0.9}} = n^{o(1)}$. This remained the best known result for several years, until Li constructed an *extractor* which could handle dramatically less min-entropy $k \geq \text{polylog}(n)$ [Li16]. Li's affine extractor has polynomially small error, and is constructed by carefully extending a breakthrough result on independent source extractors [CZ19] to the affine setting.

Our goal While Li's construction [Li16] represents a dramatic improvement in the state-of-the-art, there is still progress to be made. In particular, with this result in hand, the best known explicit extractor for affine sources requires min-entropy $k \geq \log^C n$ for some large constant $C > 0$, whereas non-explicitly we know that it is possible to achieve $k \geq O(\log n)$. Naturally, we would like to find explicit constructions that come closer to matching this existential bound. The matter becomes more serious if we examine the progress that has been made on *independent source* extraction since the breakthrough work of Chattopadhyay and Zuckerman [CZ19], on which Li's affine extractor [Li16] is based. Here, a series of exciting work [CS16a, CL16a, Coh16b, Mek17, BDT19, Coh17, Li17, Li19] has successfully brought the entropy requirement down from $k \geq \log^C n$ to $k \geq \log^{1+o(1)} n$, whereas no such progress has been made for the affine setting. All of this begs the question,

Can we construct an affine extractor for min-entropy $k \geq \log^{1+o(1)} n$?

This is the main question we seek to answer.

6.2 Our results

In this chapter, we answer the main question presented in Section 6.1, and explicitly construct extractors for affine sources with a near-optimal min-entropy requirement. We prove the following theorem.

Theorem 6.1 (Affine extractors for almost logarithmic entropy)**.** *For any constant $\varepsilon > 0$, there exists a constant $C > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ for affine sources of min-entropy $k \geq C \log n \cdot \log\log n \cdot (\log\log\log n)^6$, which has error ε.*

As discussed, the previous best extractor for affine sources [Li16] required min-entropy $k \geq \log^C n$ for some large constant $C > 0$, while our new Theorem 6.1 can handle min-entropy $k \geq \log^{1+o(1)} n$. This comes close to the optimal entropy requirement of $k \geq O(\log n)$, and virtually matches the recent improvements that have been made in the *independent source* setting [CS16a, CL16a, Coh16b, Mek17, BDT19, Coh17, Li17, Li19]. Our constructions take inspiration from this exciting line of work, but require a serious amount of new machinery to make everything go through, due to the fact that we are working with just one (affine) source. The two key new ingredients we construct are (i) a linear seeded *somewhere extractor* for affine sources and (ii) a *correlation breaker* for linearly correlated sources - both of which can be viewed as a powerful twist on the classical seeded extractor.

Organization

We construct our affine extractor in Section 6.3, where we prove Theorem 6.1. Next, we demonstrate applications of new extractor in pseudorandomness (Section 6.4) and complexity theory (Section 6.5). Finally, we conclude with some open problems in Section 6.6.

6.3 The extractor

We are now ready to proceed with our extractor construction. In this section, we prove the following.

Theorem 6.2 (Affine extractors for almost logarithmic entropy (Theorem 6.1, restated))**.** *For any constant $\varepsilon > 0$, there exists a constant $C > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ *for affine sources of min-entropy $k \geq C \log n \cdot \log\log n \cdot (\log\log\log n)^6$, which has error ε.*

To construct this extractor, we build a framework out of several key ingredients, which we present below.

6.3.1 A simple construction via somewhere extractors and correlation breakers

Our framework for extracting for affine sources will follow the breakthrough framework of Chattopadhyay and Zuckerman for extracting from two sources with polylogarithmic min-entropy [CZ19]. At a very high level, this framework works as follows.

Creating a table Their general idea was as follows. Given two independent sources $\mathbf{X}, \mathbf{Y} \sim \{0,1\}^n$, each with some min-entropy, one can call a strong seeded extractor $\mathsf{sExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}$ on $\mathbf{X}$, enumerating over all the seeds, to obtain another source (called a "table"), $\mathbf{X}' \sim \{0,1\}^{2^d}$. Now, by the property of the strong seeded extractor, one is guaranteed that most $\mathbf{X}'_i$ are close to uniform. The other bits, on the other hand, can look like anything. Towards this end, one might intially think to attempt to extract from $\mathbf{X}'$ using an extractor for *non-oblivious bit-fixing (NOBF) sources*. Such sources look similar to $\mathbf{X}'$, in the sense that some of the bits in an NOBF source are guaranteed to be uniform, while the remaining bits can depend on these in any which way. However, the key difference, of course, is that the uniform bits in an NOBF are *independent*, while the uniform bits in $\mathbf{X}'$ can be arbitrarily correlated.

Breaking the correlation To fix this problem, one natural idea is to attempt to use the other source $\mathbf{Y}$ to *break the correlation*. And indeed, this is exactly what is done: using an object called a *correlation breaker*, it is possible to use $\mathbf{Y}$ to break the correlations in the uniform rows of $\mathbf{X}'$. So, does this mean that we're done? Indeed, if all the uniform bits in $\mathbf{X}'$ actually become uncorrelated, then we can certainly complete the extraction by applying an off-the-shelf extractor for NOBF sources. The problem, however, is that we *can't* use $\mathbf{Y}$ to break the correlation between *all* the uniform bits in $\mathbf{X}'$. This is because $\mathbf{X}'$ contains 2^d bits, which will be at least $\mathrm{poly}(n)$, given that $\mathbf{X}'$ was generated by a seeded extractor. In order to break this amount of correlation, $\mathbf{Y}$ would need to have at least $\mathrm{poly}(n)$ bits, which is of course impossible. Even worse, we are trying to extract from near *logarithmic* min-entropy, so this is nowhere in the ballpark.

A key idea baked into the framework of Chattopadhyay and Zuckerman [CZ19] is the fact that all the correlations between the good bits of $\mathbf{X}'$ need not be broken! Indeed, this is due to the fact that if we just break correlations between t-tuples of good bits in $\mathbf{X}'$, then it will still obtain a fair amount of structure. Furthermore, $\mathbf{Y}'$ won't need much more than t bits of randomness to make this happen. So, what does $\mathbf{X}'$ look like now?

The answer is that $\mathbf{X}'$ looks something like a "t-wise independent" NOBF source: i.e., one in which the good bits are (close to) t-wise independent, while the bad bits can still look like anything. But, can we even extract from such sources? What if all we have available is a vanilla extractor for NOBF sources? As it turns out, if the extractor is *simple enough* - i.e., computable in AC^0 - then it will, in fact, extract from such sources! Indeed, this follows almost immediately from Braverman's celebrated result that t-wise independence fools AC^0 circuits [Bra08]. Thus, in some sense, the extractor for NOBF sources simply cannot tell that it is in fact looking at a *t-wise independent* NOBF source, and it will extract anyway. This roughly completes the framework for extracting from two independent sources with polylogarithmic min-entropy.

Going beyond polylogarithmic min-entropy Here, the need for polylogarithmic min-entropy largely comes from the fact that AC^0 is only fooled by *polylogarithmic* independence. If one hopes to extract with near-logarithmic min-entropy, one must find some extractor for NOBF sources that is fooled by less independence. As it turns out, [Vio14] previously proved that the *majority* function is exactly such an extractor, and much of the efforts following [CZ19] have attempted to use such an extractor to achieve near-logarithmic min-entropy in the two-source setting. One difficult problem that arises, however, is that fact that while the majority function can work with good bits with *less independence* than the function from before, it in fact requires *a greater overall number of good bits*. Thus, it somehow shifts the hard work from the correlation breaker, to the original seeded extractor (which now must have very small error). Such extractors, however, are difficult to construct. To circumvent this, [BDT19] had the great idea that you don't actually need a seeded extractor to properly execute the first step. In fact, it is enough to just have a *somewhere* seeded extractor, which outputs a couple of results, only one of which is guaranteed to be good (such objects are much easier to construct). This is because one can simply request for the correlation breaker to also break the correlation between these few outputs, so we can safely XOR them together, and the correlation breaker promises that this output will look like it came from a bona fide seeded extractor.

This is a rough outline of the plan that we will follow. However, notably, we only have *one* (affine) source to extract from. Thus, this single source will need to play both of the roles of $\mathbf{X}$ and $\mathbf{Y}$ above. In order to make this happen, we will need some new equipment.

A formal presentation of the proof

For our formal proof, the first new device we need is a correlation breaker for *linearly correlated sources*.

Definition 6.3 (Correlation breaker for linearly correlated sources)**.** *A function* $\mathsf{CB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}^m$ *is called a* correlation breaker for linearly correlated sources *for min-entropy k, seed length d, advice length a, tampering degree t, and error ε if the following holds.*

- *Let* $\mathbf{A}, \mathbf{B} \sim \{0,1\}^n$ *and* $\mathbf{Y}, \mathbf{Y}^1, \dots, \mathbf{Y}^t \sim \{0,1\}^d$ *be random variables such that* $\mathbf{A}$ *is independent of* $(\mathbf{B}, \mathbf{Y}, \mathbf{Y}^1, \dots, \mathbf{Y}^t)$*. Furthermore, suppose that* $\mathbf{A}$ *has min-entropy at least* k *and* $\mathbf{Y}$ *is uniform.*
- *Define* $\mathbf{X} := \mathbf{A} + \mathbf{B}$.

Then for any pieces of "advice" $\alpha, \alpha^1, \dots, \alpha^t \in \{0,1\}^a$ *with* α *distinct from the rest,*

$$\begin{aligned}|&\mathsf{CB}(\mathbf{X}, \mathbf{Y}, \alpha) \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}^1, \alpha^1) \circ \dots \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}^t, \alpha^t)\\ &-\mathbf{U}_m \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}^1, \alpha^1) \circ \dots \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}^t, \alpha^t)| \leq \varepsilon.\end{aligned}$$

Indeed, this powerful object will play the role of the (standard) correlation breaker discussed earlier on. Next, to play the role of the somewhere seeded extractor, we will also need a version of this object that works well with linearly correlated sources. Such a device is called a *linear* seeded somewhere extractor.

Definition 6.4 (Linear seeded somewhere extractor)**.** *A function* $\mathsf{ssExt} : \{0,1\}^n \times \{0,1\}^d \times [A] \to \{0,1\}^m$ *is called a* linear seeded somewhere extractor for affine sources *of min-entropy k, with seed length d, advice size A, and error ε if the following holds. For any affine source* $\mathbf{X} \sim \{0,1\}^n$ *of min-entropy k,*

$$\Pr_{y \sim \mathbf{U}_d}[\nexists a \in A \text{ such that } \mathsf{ssExt}(\mathbf{X}, y, a) = \mathbf{U}_m] \leq \varepsilon,$$

and for every fixed $y \in \{0,1\}^d$ and $a \in A$ the function $\mathsf{ssExt}(\cdot, y, a) : \{0,1\}^n \to \{0,1\}^m$ *is linear over* $\mathbb{F}_2$.

Finally, at the very end of our construction, we will once again need an extractor for NOBF sources. Thus, as a reminder, let us briefly recall their definition (along with the definitions of their relaxations).

Definition 6.5 (Non-oblivious bit-fixing sources)**.** *A source* $\mathbf{X} \sim \{0,1\}^n$ *is called a* q-non-oblivious bit-fixing (NOBF) source *if there exists some set* $Q \subseteq [n]$ *of size* q *such that* $\mathbf{X}_{\overline{Q}}$ *is uniform. If there exists a set* $Q \subseteq [n]$ *of size* q *such that for any* $S \subseteq \overline{Q}$ *of size* t*,* $\mathbf{X}_S$ *is uniform, then it is called a* (q,t)*-NOBF source. Finally, if* $\mathbf{X}_S$ *is only guaranteed to be* γ*-close to uniform, then it is called a* (q,t,γ)*-NOBF source.*

In general, we will only be able to produce a (q,t,γ)-NOBF source, instead one where the good bits are truly t-wise independent (i.e., have the property that any t of them look perfectly uniform). However, this will not be a big problem, as it is known that (q,t,γ)-NOBF sources *look* like (q,t)-NOBF sources.

Lemma 6.1 (Local-to-global closeness for t-wise independence [AGM03])**.** *Let* $\mathbf{X} \sim \{0,1\}^n$ *be a source such that for any* $S \subseteq [n]$ *of size* t*, it holds that* $\mathbf{X}_S$ *is* γ*-close to uniform. Then* $\mathbf{X}$ *is* $n^t\gamma$*-close to a* t*-wise independent distribution.*

Corollary 6.1. *Any* (q,t,γ)*-NOBF source* $\mathbf{X} \sim \{0,1\}^n$ *is* $n^t\gamma$*-close to a* (q,t)*-NOBF source* $\mathbf{X}' \sim \{0,1\}^n$.

Finally, the last tool we need is the following classical way to working with linearly correlated sources.

Lemma 6.2 (Affine conditioning [Rao09b])**.** *Let* $\mathbf{X} \sim \{0,1\}^n$ *be an affine source with min-entropy* k*, and let* $L : \{0,1\}^n \to \{0,1\}^m$ *be an arbitrary linear function. Then there exist independent affine sources* $\mathbf{A}, \mathbf{B} \sim \{0,1\}^n$ *such that* $\mathbf{X} := \mathbf{A} + \mathbf{B}$*, and* $H_\infty(\mathbf{A}) \geq k - m$*, and* $L(\mathbf{A}) = 0$ *with probability* 1.

Given all of these ingredients, we present and prove our framework for extracting from affine sources, before finally instantiating it to yield affine extractors for almost logarithmic min-entropy.

Lemma 6.3 (Affine extractors from three ingredients)**.** *Suppose you have the following objects.*

- $\mathsf{ssExt} : \{0,1\}^n \times \{0,1\}^d \times [A] \to \{0,1\}^m$ *a linear seeded somewhere extractor for affine sources of min-entropy* k_1*, with seed length* d*, advice size* A*, and error* ε_1*.*
- $\mathsf{CB} : \{0,1\}^n \times \{0,1\}^m \times \{0,1\}^{d+\log A} \to \{0,1\}$ *a correlation breaker for linearly correlated sources for min-entropy* k_2*, seed length* m*, advice length* $d + \log A$*, tampering degree* tA*, and error* ε_2*.*
- $\mathsf{NOBF} : \{0,1\}^D \to \{0,1\}$ *an extractor for* $(\varepsilon_1 D, t)$*-NOBF sources with error* ε_3*.*

Then, consider the function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ *defined as:*

$$\begin{aligned}\mathsf{Ext}(\mathbf{X}) := \mathsf{NOBF}(\, &\oplus_{\alpha \in [A]} \mathsf{CB}(\mathbf{X}, \mathsf{ssExt}(\mathbf{X}, 1, \alpha), 1, \alpha),\\ &\oplus_{\alpha \in [A]} \mathsf{CB}(\mathbf{X}, \mathsf{ssExt}(\mathbf{X}, 2, \alpha), 2, \alpha),\\ &\vdots\\ &\oplus_{\alpha \in [A]} \mathsf{CB}(\mathbf{X}, \mathsf{ssExt}(\mathbf{X}, D, \alpha), D, \alpha))\end{aligned}$$

Ext *is an extractor for affine sources with min-entropy* $k \geq \max\{k_1, k_2 + mtA\}$ *and error* $\varepsilon = \varepsilon_3 + D^t t\varepsilon_2$.

Proof. For convenience, let us start by defining some random variables. For every $s \in \{0,1\}^d$ and $\alpha \in [A]$, we define $\mathbf{Y}_{s,\alpha} := \mathsf{ssExt}(\mathbf{X}, s, \alpha)$ and $\mathbf{Z}_{s,\alpha} := \mathsf{CB}(\mathbf{X}, \mathbf{Y}_{s,\alpha}, s, \alpha)$. Then, for every $s \in \{0,1\}^d$ we define $\mathbf{R}_s := \oplus_{\alpha \in [A]} \mathbf{Z}_{s,\alpha}$. Notice that we can now conveniently write

$$\mathsf{Ext}(\mathbf{X}) = \mathsf{NOBF}(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_D).$$

Let us call $(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_D)$ the *table*. We call a row $s \in [D]$ *good* if there is some $\alpha \in [A]$ such that the output of the somewhere seeded extractor $\mathbf{Y}_{s,\alpha}$ is uniform, and let us call a row *bad* otherwise. Since we fed the somewhere extractor an affine source of min-entropy at least k_1, the number of bad rows can be at most $\varepsilon_1 D$. Now, take a subset $T \subseteq [D]$ of t good rows. Our next goal is to argue $(\mathbf{R}_s)_{s \in T}$ is close to uniform, so that we can eventually say that the entire table looks close to an NOBF source.

Without loss of generality (just to make notation convenient), let us assume that $T = \{1, 2, \dots, t\} \subseteq D$. We wish to argue that $|\mathbf{R}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_t|$ is small, but let's start with trying to show

$$|\mathbf{R}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t| \tag{6.1}$$

is small. This is where our correlation breaker will come in handy. Now, since we are assuming that $\mathbf{R}_1$ is a good row, we know one of its components are good. Namely, there is some $\alpha \in [A]$ such that $\mathbf{Y}_{1,\alpha}$ is uniform. Without loss of generality (again just for convenient notation), let us assume $\alpha = 1$ has this property; namely, that $\mathbf{Y}_{1,1}$ is uniform. Now, using the data-processing inequality and the definition of $\mathbf{R}_s$, we can upper bound Equation (6.1) by

$$\begin{aligned}&|\mathbf{Z}_{1,1} \circ \cdots \circ \mathbf{Z}_{1,A} \circ \mathbf{Z}_{2,1} \circ \cdots \circ \mathbf{Z}_{2,A} \circ \cdots \circ \mathbf{Z}_{t,1} \circ \cdots \circ \mathbf{Z}_{t,A} \\ &- \mathbf{U}_1 \circ \cdots \circ \mathbf{Z}_{1,A} \circ \mathbf{Z}_{2,1} \circ \cdots \circ \mathbf{Z}_{2,A} \circ \cdots \circ \mathbf{Z}_{t,1} \circ \cdots \circ \mathbf{Z}_{t,A}|.\end{aligned}$$

Our new goal is to upper bound the above expression. Towards this end, let us make notation a little more convenient again. To do so, we identify $[t] \times [A]$ with the set $[tA]$ in the natural way, so that we can write the above expression as $|\mathbf{Z}_1 \circ \mathbf{Z}_2 \cdots \circ \mathbf{Z}_{tA} - \mathbf{U}_1 \circ \mathbf{Z}_2 \circ \cdots \circ \mathbf{Z}_{tA}|$. Unrolling the definition of $\mathbf{Z}_i$, this becomes:

$$\begin{aligned}&|\mathsf{CB}(\mathbf{X}, \mathbf{Y}_1, 1) \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}_2, 2) \circ \cdots \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}_{tA}, tA) \\ &\qquad - \mathbf{U}_1 \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}_2, 2) \circ \cdots \circ \mathsf{CB}(\mathbf{X}, \mathbf{Y}_{tA}, tA)|.\end{aligned}$$

Recall that we produced each $\mathbf{Y}_i$ with a *linear* somewhere extractor, and thus for every $i \in [tA]$ there is some linear function $L_i : \{0,1\}^n \to \{0,1\}^m$ such that $\mathbf{Y}_i = L_i(\mathbf{X})$, and thus the above expression becomes

$$\begin{aligned}&|\mathsf{CB}(\mathbf{X}, L_1(\mathbf{X}), 1) \circ \mathsf{CB}(\mathbf{X}, L_2(\mathbf{X}), 2) \circ \cdots \circ \mathsf{CB}(\mathbf{X}, L_{tA}(\mathbf{X}), tA) \\ &\qquad - \mathbf{U}_1 \circ \mathsf{CB}(\mathbf{X}, L_2(\mathbf{X}), 2) \circ \cdots \circ \mathsf{CB}(\mathbf{X}, L_{tA}(\mathbf{X}), tA)|.\end{aligned}$$

Consider now the linear function $L : \{0,1\}^n \to (\{0,1\}^m)^{tA}$ defined as $L(x) := (L_1(x), \dots, L_{tA}(x))$. By Rao's affine conditioning (Lemma 6.2), there exist independent affine sources $\mathbf{A}, \mathbf{B} \sim \{0,1\}^n$ such that $\mathbf{X} := \mathbf{A} + \mathbf{B}$, and $H_\infty(\mathbf{A}) \geq k - mtA$, and $L(\mathbf{B}) = 0$ with probability 1. This means that $L(\mathbf{X}) = L(\mathbf{A} + \mathbf{B}) = L(\mathbf{A}) + L(\mathbf{B}) = 0 + L(\mathbf{B}) = L(\mathbf{B})$. And by the definition of L, this means that for any $i \in [tA]$ it also holds that $L_i(\mathbf{X}) = L_i(\mathbf{B})$. We substitute this into our expression.

$$\begin{aligned}&|\mathsf{CB}(\mathbf{X}, L_1(\mathbf{B}), 1) \circ \mathsf{CB}(\mathbf{X}, L_2(\mathbf{B}), 2) \circ \cdots \circ \mathsf{CB}(\mathbf{X}, L_{tA}(\mathbf{B}), tA) \\ &\qquad - \mathbf{U}_1 \circ \mathsf{CB}(\mathbf{X}, L_2(\mathbf{B}), 2) \circ \cdots \circ \mathsf{CB}(\mathbf{X}, L_{tA}(\mathbf{B}), tA)|.\end{aligned}$$

But notice now that we have random variables $\mathbf{A}, \mathbf{B} \sim \{0,1\}^n$ and $L_1(\mathbf{B}), \dots, L_{tA}(\mathbf{B}) \sim \{0,1\}^m$ such that $\mathbf{A}$ is independent from $(\mathbf{B}, L_1(\mathbf{B}), \dots, L_{tA}(\mathbf{B}))$, and $\mathbf{A}$ has min-entropy at least $k - mtA \geq k_2$, while

$L_1(\mathbf{B})$ is uniform[13] and $\mathbf{X} = \mathbf{A} + \mathbf{B}$. This means that our correlation breaker works! In particular, we get an upper bound of ε_2 on the above expression. And thus we can finally upper bound Equation (6.1):

$$|\mathbf{R}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t| \leq \varepsilon_2. \tag{6.2}$$

Now, notice that if we wish to upper bound

$$|\mathbf{U}_1 \circ \mathbf{R}_2 \circ \mathbf{R}_3 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_1 \circ \mathbf{U}_1 \circ \mathbf{R}_3 \circ \cdots \circ \mathbf{R}_t|,$$

then it is equivalent to upper bound $|\mathbf{R}_2 \circ \mathbf{R}_3 \cdots \circ \mathbf{R}_t - \mathbf{U}_1 \circ \mathbf{R}_3 \circ \cdots \circ \mathbf{R}_t|$, since the first instance of $\mathbf{U}_1$ on each side of the expression are independent. But notice that we can follow the exact same argument as before to upper bound this by ε_2, and thus

$$|\mathbf{U}_1 \circ \mathbf{R}_2 \circ \mathbf{R}_3 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_1 \circ \mathbf{U}_1 \circ \mathbf{R}_3 \circ \cdots \circ \mathbf{R}_t| \leq \varepsilon_2.$$

Combining this with Equation (6.2) yields, via the triangle inequality,

$$|\mathbf{R}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_1 \circ \mathbf{U}_2 \circ \cdots \circ \mathbf{R}_t| \leq 2\varepsilon_2.$$

Continuing this argument will ultimately yield

$$|\mathbf{R}_1 \circ \mathbf{R}_2 \circ \cdots \circ \mathbf{R}_t - \mathbf{U}_t| \leq t\varepsilon_2,$$

as desired at the beginning of the proof. Returning to our original objective, we have now shown that every t good rows in the table $(\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_D)$ are $t\varepsilon_2$-close to uniform. Furthermore, recall that the number of *bad* rows was at most $\varepsilon_1 D$. In other words, the source $\mathbf{R} := (\mathbf{R}_1, \mathbf{R}_2, \ldots, \mathbf{R}_D)$ is precisely a $(\varepsilon_1 D, t, t\varepsilon_2)$-NOBF source! And by Corollary 6.1, this means that $\mathbf{R}$ is $(D^t t\varepsilon_2)$-close to some $(\varepsilon_1 D, t)$-NOBF source $\mathbf{R}' \sim \{0,1\}^D$. Thus $|\mathsf{NOBF}(\mathbf{R}') - \mathbf{U}_1| \leq \varepsilon_3$, which means that $|\mathsf{NOBF}(\mathbf{R}) - \mathbf{U}_1| \leq \varepsilon_3 + D^t t\varepsilon_2$, by an application of the triangle inequality and data-processing inequality. This completes the proof. □

Given the above framework, we are now ready to construct our affine extractors that achieve almost logarithmic min-entropy. In the full version of our paper [CGL21], we explicitly construct the following.

Lemma 6.4 (Linear seeded somewhere extractor)**.** *There exists a constant $C > 0$ which satisfies the following. For every constant $\delta > 0$, there exists a constant $C' > 0$ such that for every $n \in \mathbb{N}$ and $\xi(n) < 2^{(\log n)^{1/3}}$, the following holds. There exists an explicit linear seeded somewhere extractor* $\mathsf{ssExt} : \{0,1\}^n \times \{0,1\}^d \times [A] \to \{0,1\}^m$ *for affine sources of min-entropy $k \geq C \cdot \xi(n) \log n$, which has seed length $d \leq C' \log n$, advice size $A \leq C' \log^2(\xi(n))$, output length $m \geq \xi(n) \log n$, and error $\varepsilon \leq 2^{-(1-\delta)d}$.*

Lemma 6.5 (Correlation breaker for linearly correlated sources)**.** *There exists a constant $C > 0$ such that the following holds. There exists an explicit correlation breaker for linearly correlated sources* $\mathsf{CB} : \{0,1\}^n \times \{0,1\}^d \times \{0,1\}^a \to \{0,1\}$ *for min-entropy $k \geq Ct^2 \log(a) \log(na/\varepsilon)$, seed length $d \geq Ct^2 \log(a) \log(na/\varepsilon) + t^3 \log(na/\varepsilon)$, advice size a, tampering degree t, and error ε.*

Our correlation breaker is highly involved, and involves a white-box reconstruction of several ingredients that go into standard correlation breakers (for independent sources), to make them work for a single affine source, instead. In a recent work, Chattopadhyay and Liao [CL22] gave a simple and elegant way to *directly* (in a black-box manner) convert a standard correlation breaker into one that works for linearly correlated sources. Finally, to complete the construction of our extractor, all we need is the following simple extractor for (q,t)-NOBF sources, which is in fact just the *majority* function.

[13] Recall that $L_1(\mathbf{B})$ is uniform by our assumption (without loss of generality) that $\mathbf{Y}_{1,1}$ was uniform.

Lemma 6.6 (Extractor for (q,t)-NOBF sources [Vio14])**.** *There exists a constant $C > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{NOBF} : \{0,1\}^n \to \{0,1\}$ *for (q,t)-NOBF sources with error $\varepsilon \leq C \cdot (\frac{\log t}{\sqrt{t}} + \frac{q}{\sqrt{n}})$.*

Given all of these ingredients, it is easy to finish the construction of our affine extractor for almost logarithmic min-entropy (Theorem 6.2).

Proof of Theorem 6.2. Simply plug Lemmas 6.4 to 6.6 into the framework given by Lemma 6.3, setting parameters appropriately. □

6.4 Applications in pseudorandomness

In Part III of this thesis, we will see that affine extractors are surprisingly central figures in the world of seedless extraction. In particular, they have recently emerged as a unifying model among the family of *samplable sources*, in the sense that affine extractors can automatically extract from many of these seemingly unrelated sources. As a result, our new affine extractors give (slightly) improved extractors for the two of the most classic sources coming from this family. We go into detail below.

6.4.1 Extractors for small-space sources

Small-space sources are a classic type of samplable source, first introduced by Kamp, Rao, Vadhan and Zuckerman [KRVZ11]. Informally, a space s source $\mathbf{X} \sim \{0,1\}^n$ is just a source that can be generated by an algorithm with s bits of memory (for a formal definition, see Definition 7.2). In Chapter 7, we give new extractors for this model, improving the previous best entropy requirement from $k \geq s^{1.1} \cdot 2^{\log^{0.51} n}$ [CL16b] to $k \geq s \cdot \mathrm{polylog}(n)$ (Theorem 7.1). As it turns out, our new construction is based on a novel reduction from small-space sources to affine sources.

Lemma 6.7 (A reduction from small-space sources to affine sources - Lemma 7.1, restated)**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy at least k. Then $\mathbf{X}$ is $2^{-k'}$-close to a convex combination of affine sources of min-entropy k', where*

$$k' \geq \frac{ck}{s \cdot \log(2n/k)}$$

for some universal constant $c > 0$.

In order to obtain our small-space extractors for polylogarithmic min-entropy, we instantiate the above reduction with the affine extractor of Li [Li16] (which works for polylogarithmic min-entropy). Using our new extractors for affine sources that achieve almost logarithmic entropy, we can further improve this result to obtain the following.

Theorem 6.3 (Extractors for small-space sources)**.** *For any constant $\varepsilon > 0$, there exists a constant $C > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ *for space s sources of min-entropy $k \geq Cs \cdot \log^2 n \cdot \log\log n \cdot (\log\log\log n)^6$, which has error ε.*

Proof. Combine Theorem 6.1 with Lemma 6.7. □

6.4.2 Extractors for circuit sources

An equally popular model of samplable sources are *circuit sources*. If small-space sources model distributions samplable in limited *space*, then circuit sources model distributions samplable in limited *time*. Within the class of circuit sources, *local sources* have gained a lot of attention. Here, we take such sources to be of the form $\mathbf{X} = f(\mathbf{U}_\ell)$, where each output bit f_i of f depends on just a constant number of input bits (for a formal definition, see Definition 8.2). As it turns out, these sources, too, are reducible to affine sources.

Lemma 6.8 (A reduction from local sources to affine sources [DW12, Vio14] - Theorem 8.4, restated)**.** *Let* $\mathbf{X} \sim \{0,1\}^n$ *be a local source of min-entropy* k*. Then* $\mathbf{X}$ *is* $2^{-k'}$*-close to a convex combination of affine sources of min-entropy* k'*, where*

$$k' \geq ck^2/n$$

for some constant $c > 0$.

Instantiating the above reduction with the previous best affine extractor yields local source extractors for min-entropy $k \geq \sqrt{n} \cdot \mathrm{polylog}(n)$. Using our new affine extractors, we obtain the following improvement.

Theorem 6.4 (Extractors for local sources)**.** *For any constant* $\varepsilon > 0$*, there exists a constant* $C > 0$ *such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}$ *for local sources of min-entropy* $k \geq C\sqrt{n \log n \cdot \log\log n \cdot (\log\log\log n)^6}$*, which has error* ε.

Proof. Combine Theorem 6.1 with Lemma 6.8. □

6.5 Applications in complexity theory

Finally, recall from Section 6.1 that many state-of-the-art explicit lower bounds come from affine extractors. Given our improved extractors for affine sources, it is natural to ask if we also get improved lower bounds.

6.5.1 Explicit lower bounds against circuits and more

While it is true that our new affine extractors should give new lower bounds in every domain that they have been applied, it seems as though the *order of the improvement* obtained is not quite in the realm of interest. One exception, however is to model of *DNFs of parities*. These are depth 3 circuits of the form $\mathsf{OR} \circ \mathsf{AND} \circ \mathsf{XOR}$, and a natural goal is to find explicit functions $f : \{0,1\}^n \to \{0,1\}$ such that the size of the minimum such circuit computing f, denoted $\mathsf{DNF}_\oplus(f)$, is as large as possible. In a work by Cohen and Shinkar [CS16b], it was shown that affine extractors (in fact, *dispersers*) have exactly such a property.

Lemma 6.9 ([CS16b])**.** *Let* $f : \{0,1\}^n \to \{0,1\}$ *be an affine disperser for dimension* k*. Then*

$$\mathsf{DNF}_\oplus(f) \geq 2^{n-2k}.$$

By plugging in our new affine extractors, we obtain the following.

Theorem 6.5 (Explicit lower bounds against DNFs of parities)**.** *There exists a constant* $C > 0$ *such that there exists an explicit function* $f : \{0,1\}^n \to \{0,1\}$ *with*

$$\mathsf{DNF}_\oplus(f) \geq 2^n / n^{\log\log n \cdot (\log\log\log n)^6}.$$

Proof of Theorem 6.5. Combine Theorem 6.1 with Lemma 6.9. □

Indeed, this is a noticeable improvement over the prior best explicit lower bounds for this model (coming from the prior best affine extractor [Li16]), which achieved $\mathsf{DNF}_\oplus(f) \geq 2^n/n^{\mathrm{polylog}(n)}$. Furthermore ,it was shown in [CS16b] that $\mathsf{DNF}_\oplus(f) \leq O(2^n/n)$, and thus Theorem 6.5 is nearly tight.

6.6 Future directions

In this chapter, we constructed near-optimal extractors for affine sources, which can handle min-entropy $k \geq \log^{1+o(1)} n$. Two years following our result, a breakthrough paper of Li [Li23] managed to finally bring the min-entropy requirement down to $k \geq O(\log n)$, which is asymptotically optimal. Still, both of our extractors have high error, and many open questions around affine sources remain.

Low-error extractors for affine sources First, even though the extractor community has made great progress on the min-entropy requirement of affine extractors, there is still much work to be done on constructing *low-error* affine extractors. Indeed, to date, we still don't even know how to construct low-error affine extractors that achieve min-entropy $k \geq n^{0.99}$. Going forward, we view this as one of the most fundamental open questions in extractor theory, given the central role that affine extractors play in the field, and the fact that high-quality randomness is crucial in cryptography [DOPS04].

Simpler constructions of affine extractors The current techniques that go into building affine extractors (and independent source extractors, for that matter) are highly involved, and exhibit fundamental barriers to achieving low-error. It would be great to inject this problem with some brand new ideas, and construct much simpler affine extractors. The recent paper [HIV22] already has some interesting results in this direction.

Extractors for quadratic sources Finally, should the reader feel particularly adventurous, it would be interesting to attempt an explicit construction of extractors for *quadratic* sources. These are natural generalizations of affine sources, in the following sense: while affine sources are uniform over the solutions to a system of *linear* equations, quadratic sources are uniform over the solutions to a system of *quadratic* (i.e., degree 2) equations. It is known that any explicit construction of an extractor for such sources with min-entropy $0.01n$ would immediately imply new state-of-the-art circuit lower bounds [GK16].

Part III

Extractors for low-complexity sources

Chapter 7

Extractors for small-space sources

In Part I of this thesis, we constructed robust extractors for *independent sources*. In Part II of this thesis, we constructed extractors for *algebraic sources*. As we enter Part III of this thesis, we seek to construct extractors for *low-complexity sources*. This model represents the third and final key flavor of *structure* within the world of seedless extraction.

A low-complexity source $\mathbf{X}$ is simply a source that can be generated by a *low-complexity algorithm*. Also known as *samplable sources*, these sources were first introduced by Trevisan and Vadhan [TV00], under the suggestion that they are a reasonable model for the randomness one might actually find in nature. Beyond their natural definition, samplable sources are exceedingly general, and are deeply connected to emerging new areas of complexity theory. Within the family of samplable sources, *small-space sources* distinguish themselves as one of the oldest and most general models. With roots going back to the 1960s, these sources capture distributions that can be sampled by algorithms with *limited memory*.

In this chapter, we construct extractors for small-space sources that offer an *exponential improvement* over the previous state-of-the-art. Prior to our work, the best-known extractor for space s sources over n bits required min-entropy $k \geq s^{1.1} \cdot 2^{\log^{0.51} n}$. In our work, we construct extractors that require just $k \geq s \cdot \log^C n$ entropy for some constant C. Our new extractors are based on a surprising new structural result, which says that small-space sources are close to a convex combination of *affine sources*. As such, we can simply take any black-box affine extractor and use it to extract from small-space sources. In fact, using our new-and-improved affine extractors from Chapter 6, we can even extract from space s sources with min-entropy $k \geq s \cdot \log^{2+o(1)} n$, which is nearly optimal.

Finally, we provide an additional explicit extractor that can achieve *exponentially small error* (which is crucial for cryptographic applications). Here, we cannot simply apply an affine extractor, since all known low-error affine extractors require extremely high min-entropy. Instead, we invoke a known reduction from small-space sources to adversarial sources, which allows us to exploit our brand new low-error extractors from Chapter 3. As a result, for the challenging low-error setting, we obtain extractors for space $s \leq n^{0.01}$ sources that can handle min-entropy $k \geq n^{0.51}$. This significantly improves the previous state-of-the-art, which required min-entropy $k \geq n^{0.99}$.

The results presented in this chapter are based on the following joint work:

- [CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021

7.1 Introduction

In our journey through the world of seedless extraction, we have already explored two of the most classical flavors of *structure*: the *independent source* model (Part I), and the *algebraic source* model (Part II). Yet, one key flavor remains. As we embark upon the third and final leg of our journey, we turn to this third and final flavor of structure. In this part of the thesis (Part III), we set out to explore the model of *low-complexity sources*. As tradition would have it, let us begin with the classical definition of an extractor.

Definition 7.1 (Extractor). *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is called an* extractor for $\mathcal{X}$ with error ε *if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \varepsilon.$$

Low-complexity sources are a whole different beast than the models we've seen before. Previously, we've either assumed that each source $\mathbf{X} \in \mathcal{X}$ consists of several *independent chunks*, or that each source $\mathbf{X} \in \mathcal{X}$ exhibits some sort of *algebraic structure*. Now, we assume that each source $\mathbf{X} \in \mathcal{X}$ exhibits some amount of *computational structure*. This will turn out to be an exceedingly general notion, subsuming many previously studied models (like algebraic sources), and bearing deep ties to complexity theory.

Research on low-complexity sources has generally been divided into two complementary directions.[1] The first direction devotes itself to the study of *recognizable sources* [Sha11c, KVMS12, LZ19]. A recognizable source $\mathbf{X} \sim \{0,1\}^n$ is just a source that is uniform over the set $\{x : f(x) = 0\}$ for some function f coming from a "low-complexity" class $\mathcal{C}$. Extractors for recognizable sources capture a wide array of classical primitives,[2] and are notoriously difficult to construct. Despite this, there has been exciting progress in this direction, and we now have constructions (that work in the high entropy regime) for several classes $\mathcal{C}$, including *communication protocols* [Sha11c, LZ19], *linear threshold functions* [LZ19], and AC^0 *circuits* [Sha11c, LZ19].[3] Through this work, these extractors have also proven themselves to be closely related to the pursuit of *average-case lower bounds* [LZ19] and *derandomization* [Sha11c, KVMS12].

The second direction of study (within the setting of low-complexity sources) focuses on the model of *samplable sources* [TV00, KRVZ11, DW12, Vio14, Vio12b, CL16b, Li16, CGGL20, CG21, CGZ22, CL22, ACG$^+$22, Li23]. A samplable source $\mathbf{X} \sim \{0,1\}^n$ is just a source of the form $f(\mathbf{U})$ for some function f coming from a "low-complexity" class $\mathcal{C}$.[4] Thus, these sources correspond to distributions that can be *sampled efficiently*, and were originally introduced by Trevisan and Vadhan [TV00] under the suggestion that they are a good model for distributions that might actually arise in nature. More concretely, samplable sources generalize several previously studied models,[5] and are intimately connected to a burgeoning new area of complexity theory (Chapter 9).

Our focus In this thesis, we focus on the second (more well-studied) direction, and aim to construct new extractors for *samplable sources*. As these sources model distributions that can be sampled efficiently (i.e.,

[1] Here, by "complementary," we mean in the colloquial (non-mathematical) sense: in that the study of one direction often yields insight into the study of the other.

[2] For example, we can recover the definition of *affine extractors* (Chapter 6) or *extractors for variety sources* (Part II) by setting $\mathcal{C}$ to be the family of affine functions or low-degree polynomials, respectively.

[3] Of course, as suggested in the footnote above, we can also handle the class $\mathcal{C}$ of affine functions or low-degree polynomials.

[4] Thus, while recognizable sources capture distributions over the *preimage* of a low-complexity function, samplable sources capture distributions over the *image* of a low-complexity function.

[5] For example, we can recover the definition of *affine extractors* (Chapter 6) or *extractors for polynomial sources* (Part II) by setting $\mathcal{C}$ to be the family of affine functions or low-degree polynomials, respectively.

with limited resources), it may not be surprising to hear that research on samplable sources has *also* branched off into two different directions.

- The study of sources samplable with limited *time* (as modeled by *small circuits* or *Turing machines*) [TV00, DW12, Vio14, Vio12b, Li16, ACG$^+$22].
- The study of sources samplable with limited *memory* (as modeled by so-called *small-space sources*) [KRVZ11, CL16b, CGGL20, CG21, CGZ22, CL22, Li23].

In this thesis, we will carefully study both directions. We give a thorough treatment of the first direction in Chapter 8, and focus here on sources samplable with limited *memory*.

Sources samplable with limited memory Today, the most classical way to model sources samplable with limited memory is via *small-space sources* [KRVZ11]. To define this class of sources formally, one uses *branching programs* (which model the evolution of state in the memory-bounded algorithm). A branching program of width w and length n is just a directed acyclic graph with $n+1$ layers, where the first layer has 1 node, the remaining layers have w nodes (each), and every edge starting in layer i terminates in layer $i+1$. We assign an *output bit* and *transition probability* to each edge,[6] and define a small-space source as follows.

Definition 7.2 (Small-space source)**.** *A source* $\mathbf{X} \sim \{0,1\}^n$ *is called a* space s source *if it is generated by a random walk over a branching program of width* 2^s *and length* n*, which starts in the first layer, transitions according to the transition probabilities, and prints the output bits seen along the way.*

Staring at this definition, one may be concerned that it does not perfectly align with the definition of samplable sources put forth at the beginning of the chapter. Namely, it is not quite of the form $\mathbf{X} = f(\mathbf{U})$ for some low-complexity function f. However, have no fear: in Chapter 9, we show that Definition 7.2 is *equivalent* to passing uniform bits $\mathbf{U}$ into a (so-called) *read-once branching program* f.[7] Thus, small-space sources fit perfectly into the line of research on samplable sources, and (as hinted at before) their study may therefore shed new insight into an exciting new area of complexity theory (Chapter 9). Within the world of extractors, though, these sources are most interesting because they are extremely general, and capture several classical settings.

Relationships to other models As it turns out, small-space sources [KRVZ11] can actually be viewed as the culmination of a series of works on increasingly general ways to model sources samplable with limited memory [vN51, Sam68, Eli72, Blu86, Vaz87b, Vaz87a, KM04, KM05]. Because of this, small-space sources not only generalize classical models like von Neumann's independent coin flips [vN51], but they also capture sources samplable by two-state Markov chains [Sam68], constant-state Markov chains [Eli72, Blu86], and even *arbitary-state, time-dependent Markov chains* [Vaz87b, Vaz87a, KM04, KM05]. However, the generality of small-space sources does not stop there, and these sources can also simulate several models that do *not* have a "memory" component in their definition. In particular, small-space sources also generalize *oblivious bit-fixing sources* (Part II), *independent sources* (Part I), and even *adversarial sources* (Chapter 3).[8] In fact, it is suggested in [KRVZ11] that the only model of sources *unrelated* to small-space sources appears to be the class of *affine sources* (Chapter 6).

[6] Note that any given node can have as many as w outgoing edges, over which the transition probabilities should sum to 1.

[7] Read-once branching programs, which we will define in Chapter 9, are the canonical way to model small-space *computation*.

[8] As we will see shortly, this connection to adversarial sources also goes in the other way.

Prior constructions Needless to say, small-space sources are a very general model, making explicit extractors for this setting particularly valuable. Unfortunately, these extractors have remained elusive, and relatively few constructions are known. *Non-explicitly*, we know there exist extractors for space s sources of min-entropy $k \geq O(s + \log n)$ which have extremely low error $\varepsilon = 2^{-\Omega(k)}$ [KRVZ11]. *Explicitly*, on the other hand, the best known extractors for space s sources (prior to our work) are as follows.

- *The low-error setting*: There exist explicit extractors for min-entropy $k \geq O(n^{0.99} + n^{2/3}s^{1/3})$.
- *The high-error setting*: There exist explicit extractors for min-entropy $k \geq O(s^{1.1} \cdot 2^{\log^{0.51} n})$.

Let us briefly review the history on small-space extraction, which led to these explicit constructions.

The first explicit extractor for small-space sources is often credited to König and Maurer [KM04, KM05]. While these authors did not directly study small-space sources, their result implies that small-space sources can be reduced to *two independent sources*. As a result, they show that *inner product (mod 2)* is an extractor for space s sources of min-entropy $k \geq s + 0.51n$ with exponentially small error.

A year later at STOC 2006, this result was significantly improved by Kamp, Rao, Vadhan and Zuckerman [KRVZ11], who formally introduced the model of small-space sources and launched a systematic study on them. While the previous work reduced small-space sources to (two) independent sources, this new work considered a reduction to *total-entropy sources*. Total-entropy sources *generalize* the independent source model by enforcing an entropy requirement over the *total collection* of sources, instead of on each individual source. By carefully constructing extractors for this new model, Kamp, Rao, Vadhan and Zuckerman successfully extract from space s sources of min-entropy $k \geq O(n^{1-\gamma} + n^{2/3}s^{1/3})$ for some small constant $\gamma > 0$, while still achieving *exponentially small* error.

Ten years later, Chattopadhyay and Li [CL16b] dramatically improved the entropy requirement above, by reducing to an *even more general* model known as *sumset sources*. A sumset source is generated by taking the sum of several independent sources (mod 2), each with their own entropy guarantee. Using this generality, Chattopadhyay and Li showed how to successfully extract from space s sources of min-entropy $k \geq O(s^{1+\alpha} \cdot 2^{\log^{0.5+\alpha} n})$ for an arbitrarily small constant $\alpha > 0$, albeit with *polynomially small* error.

Our goal As can be seen, while there has been significant progress on constructing extractors for small-space sources, there is still much work to be done. In particular, even in the high-error setting, the best known explicit extractor requires min-entropy $k \geq O(s^{1.1} \cdot 2^{\log^{0.51} n})$, whereas non-explicitly we know it is possible to achieve $k \geq O(s + \log n)$. Naturally, this raises the question,

Can we construct improved extractors for small-space sources?

This is our primary goal.

7.2 Our results

In this chapter, we explicitly construct two significantly improved extractors for small-space sources. Our new constructions are based on two key new ingredients: namely, a new *structural result* for small-space sources, and a new extractor a *different*, equally important setting. Our small-space extractors follow easily from these ingredients, which may be of independent interest. We formally state these results, below.

Extractors for polylogarithmic entropy

In our first main result, we construct near-optimal small-space extractors in the polynomial error regime.

Theorem 7.1 (Small-space extractors for polylogarithmic entropy)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for space s sources of min-entropy $k \geq s \cdot \log^C n$, which has output length $m \geq (k/s)^\gamma$ and error $\varepsilon = n^{-\gamma}$.*

Thus, our extractor requires min-entropy $k \geq s \cdot \log^C n$, which is an exponential improvement over the previous best requirement [CL16b] of $k \geq s^{1.1} \cdot 2^{\log^{0.51} n}$. Furthermore, since the best *non-explicit* extractors for small-space sources require min-entropy $k \geq O(s + \log n)$, it follows that our new *explicit* extractor achieves a near-optimal min-entropy requirement (in the polynomial error regime). Most notably, however, Theorem 7.1 is just a simple consequence of the following new structural result, which establishes a surprising new connection between small-space sources and *affine sources* (Chapter 6).

Lemma 7.1 (Small-space sources are (close to) a convex combination of affine sources)**.** *There is a universal constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy k. Then $\mathbf{X}$ is 2^{-ck}-close to a convex combination of affine sources of min-entropy $\frac{ck}{s \log(2n/k)}$.*

Indeed, by combining this structural result with the affine extractor of Li [Li16] (which works for polylogarithmic min-entropy and has polynomially small error), we immediately obtain Theorem 7.1. In fact, if we are only interested in outputting *one bit* with *constant error*, then we can even apply our new affine extractor from before (Theorem 6.1) to obtain a small-space extractor that can handle even less min-entropy $k \geq s \cdot \log^{2+o(1)} n$. Beyond these *quantitative* improvements to the min-entropy requirement in small-space extraction, Lemma 7.1 is also interesting for one key *qualitative* reason, which we highlight below.

A fascinating takeaway of Lemma 7.1 is the deep connection it forges between *samplable sources* and *affine sources*. Indeed, it was previously known that *circuit sources* (the "other half" of samplable sources, which we study in Chapter 8) can be reduced to affine sources [DW12, Vio14]. Our result shows that *small-space sources* can *also* be reduced to affine sources, thereby cementing the role of affine sources as a *unifying figure* in samplable-source extraction. In fact, both flavors of samplable sources can actually be reduced to the *exact same specialization* of affine sources (known as 1*-local sources*),[9] whose connection to small-space sources (Lemma 7.7) is apparently *even stronger* than to circuit sources (Lemma 8.3).

Finally, it turns out that this new connection between small-space sources and affine sources has implications beyond the world of extractors. Namely, it can be leveraged to yield interesting new results on the *complexity of sampling*. In this exciting new subfield of complexity theory (Chapter 9), a classical result shows that AC^0 circuits can *sample* the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ for any linear map b [Bab87, BL87], and even some quadratic maps such as *inner product (mod* 2*)* [IN96]. Recently, however, Viola showed that there *do* exist *some* quadratic maps b that AC^0 circuits *cannot* sample [Vio16]. In Section 7.4, we show that the same is true for *limited memory algorithms* (and, more specifically, *read-once branching programs*), via a simple application of Lemma 7.1.

Extractors with exponentially small error

Returning to the world of extractors, recall that our first main result (Theorem 7.1) achieves near-optimal extractors for small-space sources in the *polynomial error regime*. While polynomially small error suffices for many applications, *cryptographic* applications often demand extractors with *negligibly small error* [DOPS04]. In our second main result, we construct new small-space extractors of exactly this type.

[9] A 1-local source (Definition 8.2) is just an affine source whose underlying subspace has a basis made out of "disjoint" vectors.

Theorem 7.2 (Low-error small-space extractors)**.** *For any fixed* $0 < \delta \leq 1/2$*, there exist universal constants* $C, \gamma > 0$ *such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for space* s *sources of min-entropy* $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$*, which has output length* $m \geq k^\gamma$ *and error* $\varepsilon = 2^{-n^\gamma}$*.*

Prior to our work, the best-known low-error extractor for small-space sources required min-entropy $k \geq O(n^{1-\gamma} + n^{2/3}s^{1/3})$ for some tiny constant $\gamma > 0$ [KRVZ11]. Thus, our new extractors enjoy an improved min-entropy requirement across all ranges of s,[10] with an especially pronounced improvement for space $s \leq n^{0.01}$ sources. Here, we improve the min-entropy requirement from $k \geq O(n^{0.99})$ to $k \geq O(n^{0.51})$.

In order to prove Theorem 7.2, a natural approach is to exploit our new reduction from small-space sources to affine sources (Lemma 7.1). However, since the best known *low-error* affine extractors [Bou07, Yeh11, Li11b] still require very high min-entropy $k \geq O(n/\sqrt{\log\log n})$, this unfortunately will just not work. Luckily, as hinted at above, the specific version of Lemma 7.1 that we *actually* prove (Lemma 7.7) reduces small-space sources to a *special type* of affine sources known as 1*-local sources*, which admit explicit low-error extractors for *much lower* min-entropy $k \geq O(n^{1/2+\delta})$ [Rao09b, DW12, Vio14] (see Section 8.1). Still, combining these extractors with our new reduction (Lemma 7.7) only yields small-space extractors for min-entropy $k \geq O(n^{1/2+\delta}s)$, which is much worse than Theorem 7.2 when s is large.

To get around this, we end up reducing small-space sources to a *different* family of sources: namely, *adversarial sources*. Recall that an (N, K, n, k)-adversarial source $\mathbf{X}$ consists of N independent sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N$, each over n bits, with the guarantee that at least K of them are *good* (i.e., have min-entropy at least k). In Chapter 3, we launched a systematic study into adversarial sources, and constructed significantly improved extractors for them. In particular, we proved the following.

Lemma 7.2 (Extractors for adversarial sources - Theorem 3.1, restated)**.** *For any fixed* $\delta > 0$*, there exist constants* $C, \gamma > 0$ *such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ *for* (N, K, n, k)*-adversarial sources with* $K \geq CN^\delta$ *good sources of min-entropy* $k \geq \log^C n$*, which has output length* $m \geq k^\gamma$ *and error* $\varepsilon = 2^{-k^\gamma}$*.*

Previously, the best-known low-error extractor for small-space sources [KRVZ11] was constructed by combining a classic reduction from small-space sources to adversarial sources (Lemma 7.3), with an explicit extractor for adversarial sources that worked for $K \geq O(\sqrt{N})$ good sources of min-entropy $k \geq O(n^{0.99})$. By plugging in our new-and-improved extractor for adversarial sources (Lemma 7.2), we immediately obtain our new-and-improved extractor for small-space sources (Theorem 7.2). Moreover, we will see that this is nearly the *best possible* low-error small-space extractor that one can hope to construct using this technique. Finally, while our new low-error small-space extractors are an immediate consequence of our new low-error adversarial source extractors, it is interesting to reflect upon the huge amount of machinery that went into the latter (including tools from coding theory, extractor theory, and communication complexity), and observe that this machinery is now finding important applications in small-space extraction.

Organization

We construct our new extractors for small-space sources in Section 7.3. Here, we start with Section 7.3.1, where we quickly recall the reduction from small-space sources to adversarial sources, and plug in our new adversarial source extractors (Lemma 7.2) to obtain our new low-error small-space extractors (Theorem 7.2). Then, in Section 7.3.2, we demonstrate our new reduction from small-space sources to *affine sources* (Lemma 7.1), and use it to obtain our new small-space extractors for polylogarithmic entropy (Theorem 7.1). We show an interesting bonus application of this reduction in *complexity theory* in Section 7.4, and conclude with some open problems in Section 7.5.

[10] It is important to note that we always have $s < n$, because otherwise the entropy bounds become trivial.

7.3 The extractors

In this section, we construct all of our new extractors for small-space sources. We start with our extractors that achieve exponentially small error (Theorem 7.2) in Section 7.3.1, as these precipitate almost immediately from other results in this thesis. Then, in Section 7.3.2, we turn to give our new extractors that achieve polylogarithmic entropy (Theorem 7.1), using a novel reduction from small-space sources to affine sources.

7.3.1 An old reduction from small-space sources to adversarial sources

As promised, we start by constructing our new low-error small-space extractors. We prove the following.

Theorem 7.3 (Low-error small-space extractors - Theorem 7.2, restated)**.** *For any fixed $0 < \delta \leq 1/2$, there exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources of min-entropy $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$, which has output length $m \geq k^{\gamma}$ and error $\varepsilon = 2^{-n^{\gamma}}$.*

As it turns out, given all of the tools we've developed thus far, it is not difficult to prove this result. Indeed, by taking a standard reduction from small-space sources to *adversarial sources* (Chapter 3), combined with our brand new adversarial source extractors (Theorem 3.1), the result is nearly immediate. In fact, to an avid reader of this thesis, the result may even look familiar - perhaps even *too* familiar. This is because we already proved this theorem in Section 3.6! However, we choose to include it again here, in order to collect all of our results on small-space sources in one place, and to present a slightly more streamlined argument.[11] We aim to keep things brief, and the reader is encouraged to revisit Section 3.6 should any confusions arise.

Now, let's begin by presenting a standard reduction from small-space sources to adversarial sources.

Lemma 7.3 (A reduction from small-space sources to adversarial sources - implicit in Lemmas 3.21 and 3.22)**.** *There is a universal constant $C > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy k. Then for any N_1, K_1, n_1, k_1 such that $n = N_1 n_1$ and*

$$k \geq C \cdot (N_1 s + N_1 k_1 + n_1 K_1),$$

it holds that $\mathbf{X}$ is 2^{-k_1}-close to a convex combination of (N_1, K_1, n_1, k_1)-adversarial sources.

Recall that in order to prove such a result, the main idea is to consider the random walk over the branching program that generates the small-space source, and condition on hitting a few nodes across several equally spaced layers. This will turn the small-space source into a sequence of independent sources, with some guarantee on the *total* amount of min-entropy that they contain. Then, using an averaging argument, one immediately obtains that a decent number of these sources contain a decent amount of entropy - or in other words, the sequence of independent sources is, in fact, an *adversarial source* (Definition 3.2).

Proof of Lemma 7.3. Let $\mathbf{W} = (\mathbf{W}_0, \mathbf{W}_1, \dots, \mathbf{W}_n) \sim (\{0,1\}^s)^{n+1}$ be a random variable holding the states reached in layers $0, 1, \dots, n$ of the branching program in the random walk that generates $\mathbf{X}$. Define $\mathbf{W}^* := (\mathbf{W}_{n_1}, \mathbf{W}_{2n_1}, \dots, \mathbf{W}_{N_1 n_1})$, and recall that the min-entropy chain rule (Lemma 2.1) tells us that

$$\Pr_{w \sim \mathbf{W}^*}[H_\infty(\mathbf{X} \mid \mathbf{W}^* = w) \geq k - N_1 s - k_1] \geq 1 - 2^{-k_1}.$$

[11] Recall that our proof of Theorem 7.3 in Section 3.6 went through an intermediate model, known as *total-entropy sources*, so that we could recover new extractors for that setting as well. Here, we will take no such detour.

Moreover, it is straightforward to verify that upon any such conditioning, $\mathbf{X}$ becomes a sequence of N_1 independent sources $\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{N_1}$ over n_1 bits each. And in the *good* conditionings (which enjoy the entropy lower bound presented above), an averaging argument tells us that at least K_1 of these sources have min-entropy at least k_1, as long as

$$N_1 k_1 + K_1 n_1 \leq k - N_1 s - k_1.$$

In other words, with probability $1 - 2^{-k_1}$ over fixing $w \sim \mathbf{W}^*$, we get that $\mathbf{X}$ becomes an (N_1, K_1, n_1, k_1)-adversarial source, provided that $k \geq N_1 s + N_1 k_1 + K_1 n_1 + k_1$. This completes the proof, via Fact 2.6. □

Equipped with this reduction from small-space sources to adversarial sources, now is the perfect time to invoke our new extractors for adversarial sources (Theorem 3.1) from Chapter 3.

Lemma 7.4 (Extractors for adversarial sources - Lemma 7.2, restated)**.** *For any fixed $\delta > 0$, there exist constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources with $K \geq CN^\delta$ good sources of min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = 2^{-k^\gamma}$.*

By combining the above reduction from small-space sources to adversarial sources (Lemma 7.3) with our low-error adversarial source extractors (Lemma 7.4), we immediately obtain our new low-error extractors for small-space sources (Theorem 7.3). We present a short proof of this, below.

Proof of Theorem 7.3. Define the parameters $N_1 = \sqrt{n/s}$ and $n_1 = \sqrt{sn}$ and $K_1 = N_1^\delta$ and $k_1 = n_1^\delta$. Without loss of generality, we assume that N_1 and n_1 are integers, as it is easy (but tedious) to extend the proof to handle when this is not the case. Now, invoke the reduction from small-space sources to adversarial sources (Lemma 7.3) with the above parameters, and combine it with our new extractors for adversarial sources (Lemma 7.4). The result is now immediate by resetting the constants C, δ appropriately.[12] □

This completes the proof of our new low-error extractors for small-space sources that can handle min-entropy $k \geq O(n^{1/2+\delta} s^{1/2-\delta})$. It is worth mentioning that such a min-entropy requirement is almost the best possible that one could hope for using a reduction to adversarial sources. Indeed, this min-entropy requirement is close to $k \geq \sqrt{n}$, and it is not hard to show that this is, in fact, a barrier for reductions of the above type. To see why, simply note that such a reduction must always either produce $N_1 \geq \sqrt{n}$ chunks or chunks of size $n_1 \geq \sqrt{n}$. If we start off with a small-space source $\mathbf{X} \sim \{0,1\}^n$ of min-entropy $k < \sqrt{n}$, then in the former case, the min-entropy chain rule will always destroy all the min-entropy in $\mathbf{X}$. And in the latter case, all of the min-entropy of $\mathbf{X}$ may get stuck in a single source, from which extraction will be impossible (Fact 1.1). Thus, if we want to break the $k = \sqrt{n}$ min-entropy barrier, we need a new idea.

7.3.2 A new reduction from small-space sources to affine sources

In order to break the above min-entropy barrier and construct significantly improved extractors for small-space sources, the key new idea is to reduce to *affine sources*. In this section, we will provide exactly such a reduction, and use it to obtain low-entropy extractors for small-space sources. Our extractors not only handle min-entropy below $\sqrt{n}$, but can even extract from *polylogarithmic* entropy. We record this result below (Theorem 7.1), which we view as the main contribution of this chapter.

[12] One should also notice (and use) the fact that it is okay to assume $s \leq n$, since otherwise the theorem statement is trivial.

Theorem 7.4 (Small-space extractors for polylogarithmic entropy - Theorem 7.1, restated)**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for space s sources of min-entropy $k \geq s \cdot \log^C n$, which has output length $m \geq (k/s)^\gamma$ and error $\varepsilon = n^{-\gamma}$.*

As promised, the key new ingredient behind these low-entropy extractors is a new, efficient reduction from small-space sources to affine sources. We prove the following, which says every small-space source is exponentially close to a convex combination of affine sources with just a little less min-entropy.

Lemma 7.5 (A reduction from small-space sources to affine sources - Lemma 7.1, restated)**.** *There is a universal constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy k. Then $\mathbf{X}$ is 2^{-ck}-close to a convex combination of affine sources of min-entropy $\frac{ck}{s\log(2n/k)}$.*

Given a reduction of this form, it is now trivial to obtain low-entropy extractors for small-space sources. Indeed, as we saw in Chapter 6, affine extractors have occupied a central role through the history of extractor theory, and many explicit constructions are already known. We record one such construction below.

Lemma 7.6 (Extractors for affine sources [Li16])**.** *There exist universal constants $C, \gamma > 0$ such that the following holds. There exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for affine sources with min-entropy $k \geq \log^C n$, which has output length $m \geq k^\gamma$ and error $\varepsilon = n^{-\gamma}$.*

Using Li's extractor, we immediately obtain our small-space extractor for polylogarithmic min-entropy.

Proof of Theorem 7.4. Simply combine Lemma 7.5 with Lemma 7.6. □

Needless to say, the key new ingredient that goes into the above proof is our new reduction from small-space sources to affine sources (Lemma 7.5), and this is all that remains to prove. However, before we do so, let us state a few other interesting ways to apply this reduction.

Extractors for small-space sources with almost logarithmic min-entropy First, we recall that in an earlier chapter of this thesis (Chapter 6), we actually obtained new affine extractors that can handle min-entropy $k \geq \widetilde{O}(\log n)$, which is strictly better than the min-entropy requirement in Lemma 7.6.[13] By combining our new affine extractors with our new reduction, we immediately obtain constant-error extractors for small-space sources with min-entropy $k \geq \widetilde{O}(s \cdot \log^2 n)$.

Super-low-error extractors for small-space sources On the other end of the extreme, one can combine a classical affine extractor of Bourgain [Bou07] with our new reduction to obtain extractors for small-space sources of min-entropy $k \geq \delta sn$ with error $\varepsilon = 2^{-\Omega(n)}$, for any constant $\delta > 0$. To the best of our knowledge, this is the only nontrivial small-space extractor that achieves super low error $\varepsilon = 2^{-\Omega(n)}$, as all previous constructions have error at least $\varepsilon = 2^{-\widetilde{\Omega}(n)}$ [KRVZ11]. This improvement in error is minor, but we include it as a nice demonstration that our *reduction* has very low error and thus has the capability to produce low-error small-space extractors. In our main application of it (Theorem 7.4), however, we use a *polynomial-error* affine extractor, whose error dominates the very low error of the reduction.

[13] Our extractors, however, have slightly worse error, which can be taken to be any constant $\varepsilon > 0$.

Semi-explicit low-error extractors for small-space sources Finally, on the topic of low-error small-space extractors, our new reduction can be used to construct *semi-explicit* extractors of this type. In more detail, a classical result of Cohen and Tal [CT15] (Theorem 8.7) shows that for any constant $\delta > 0$, there exists a (not necessarily fully explicit) constant-degree polynomial over $\mathbb{F}_2$ that extracts from affine sources of min-entropy $k \geq n^{\delta}$ with error $2^{-\Omega(k)}$. Combining this with our new reduction yields small-space extractors for min-entropy $k \geq sn^{\delta}$ with error 2, which would greatly improve upon Theorem 7.2 if the construction were fully explicit.

With all of these remarks out of the way, let us finally prove our reduction from small-space sources to affine sources (Lemma 7.5). We start with a proof overview, before diving into the proof in full formality.

An overview of the proof

As we saw in Section 7.3.1, it is impossible to extract from small-space sources $\mathbf{X} \sim \{0,1\}^n$ with entropy $k < \sqrt{n}$ using a reduction to adversarial sources. Recall that this is because if we try to split up $\mathbf{X}$ into t independent chunks by fixing the vertices hit across t equally-spaced layers, we run into a lose-lose scenario. In particular, if $t \geq \sqrt{n}$, then there will be $k - st \leq k - 1 \cdot \sqrt{n} < 0$ bits of entropy left after applying the min-entropy chain rule (Lemma 2.1), but if $t < \sqrt{n}$, then we will produce a chunk of size $n/t > \sqrt{n} > k$, which could hold all of the entropy and thus make extraction impossible (Fact 1.1). As promised, to circumvent this barrier, we provide a new reduction from small-space sources to *affine sources*. This reduction bypasses the $\sqrt{n}$ barrier by *adaptively* choosing vertices to fix. This was not possible above, because such adaptive fixings can produce independent sources of unknown and varying lengths, which cannot be captured by adversarial sources. We describe our new reduction in more detail below.

The reduction Our new reduction from small-space sources to affine sources starts the same way as before: by fixing t vertices in the random walk generating the space s source $\mathbf{X}$, to create t independent sources with roughly $k - st$ bits of total entropy. The key idea now is to use an extremely useful observation we proved in Chapter 3, which says that *any* source with entropy at least 1 is a convex combination of *affine sources* with entropy 1 (Lemma 3.20). Given this observation, we can say that as long as t' of the t independent sources have *just one bit of entropy*, then $\mathbf{X}$ currently looks like a convex combination of affine sources with min-entropy t'.

On the other hand, if *no* t' of the t independent sources have just one bit of entropy, then the $k - st$ remaining bits of entropy must be *very* highly concentrated on the $t' - 1$ most entropic independent sources. In this case, we can simply recursively apply the reduction on these $t' - 1$ independent sources. Because the entropy rate increases on each recursive call, we know the recursion must eventually stop, or else we will end up with a source with entropy rate exceeding 1, a contradiction. Thus, via a *win-win argument*, we are able to show that $\mathbf{X}$ is a convex combination of affine sources with entropy t'.

We show that even if $\mathbf{X}$ starts with entropy just $k \geq \operatorname{polylog} n$, our resulting affine source will have almost all of the entropy of the original source; namely, t' will barely be smaller than k. We are able to achieve such an efficient reduction for two reasons. First, our use of *affine sources* allows an *adaptive* and *recursive* reduction that bypasses the $k \geq \sqrt{n}$ discussed above. Second, our reduction to a sequence of t' *independent sources with entropy* 1 (which we argue is an affine source with entropy t' using Lemma 3.20) results in a *negligible* amount of lost entropy from each recursive step, whereas previously obtained recursive reductions to *a constant number of sources with relatively high entropy* [CL16b] are forced to lose much more entropy in each such step. As a result, we are able to bypass a $k \geq 2^{\sqrt{\log n}}$ entropy barrier arising in the work of Chattopadhyay and Li [CL16b].

Finally, we note that by carefully tracking the random variables that pop up in our recursion, we are able to describe all of the fixings that occur throughout the recursion *by the fixing of a single random variable*. As a result, we only need to apply the min-entropy chain rule (Lemma 2.1) *once*, which keeps the error of our reduction very low: $2^{-\Omega(k)}$, compared to an error of $2^{-k^{\Omega(1)}}$ in the recursive reduction of [CL16b].

A formal presentation of the proof

With our game plan in mind, we are now ready to provide a formal proof of our reduction from small-space sources to affine sources (Lemma 7.5). As it turns out, we will actually prove an *even stronger* reduction, from small-space sources to 1-*local sources*, which are a very special type of affine sources. Informally, a 1-local source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy k is just a source of the form $\mathbf{X} = f(\mathbf{U}_k)$, where $f : \{0,1\}^k \to \{0,1\}^n$ is a function where each output bit depends on just 1 input bit. For a formal definition, there are many equivalent options. We will use a definition that does the best job at highlighting the fact that 1-local sources are indeed special cases of affine sources. In more detail, we say that a matrix A has *row-weight* at most w if each row contains at most w 1s, and we define 1-local sources as follows.

Definition 7.3 (1-local source - specialization of Definitions 6.2 and 8.2). *A source $\mathbf{X} \sim \{0,1\}^n$ is called a* 1-local source *if there exists some matrix $A \in \mathbb{F}_2^{n\times k}$ with row-weight at most 1 and some $b \in \mathbb{F}_2^n$ such that*

$$\mathbf{X} = A\mathbf{U}_k + b,$$

where $\mathbf{U}_k$ is a uniform random variable over $\mathbb{F}_2^k$.

Now, before we proceed to formally prove our reduction, we have one surprise left: not only do reduce small-space sources to the special case of affine sources known as 1-*local sources*, but we actually reduce small-space sources to a special case of 1-local sources known as *simple* 1-local sources.

Definition 7.4 (Simple 1-local source). *Let $A \in \mathbb{F}_2^{n\times k}$ be a matrix with row-weight at most 1 and $b \in \mathbb{F}_2^n$ be some vector. We say that the 1-local source $\mathbf{X} = A\mathbf{U}_k + b$ is* simple *if for any $1 \leq i < j \leq n$, it holds that*

$$\max\{\alpha : A_{\alpha,i} = 1\} < \min\{\alpha : A_{\alpha,j} = 1\}.$$

In other words, a simple 1-local source is one which admits a generating matrix A whose columns have supports that are not *interleaved*. Given these definitions, we can finally state the actual reduction we obtain.

Lemma 7.7 (A reduction from small-space sources to 1-local sources). *There is a universal constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy k. Then $\mathbf{X}$ is 2^{-ck}-close to a convex combination of simple 1-local sources of min-entropy $\frac{ck}{s\log(2n/k)}$.*

As discussed, since simple 1-local sources are a specialization of affine sources, the above lemma immediately implies our reduction to affine sources Lemma 7.5, and we have already seen how this can be used to obtain our low-entropy extractors for small-space sources (Theorem 7.4). Thus, all that remains is to prove Lemma 7.7. But before we do so, let us make a few brief remarks.

An alternative construction of low-error small-space extractors First, now that we have a reduction from small-space sources to 1-local sources (instead of affine sources), we actually *do* have a way to exploit this new reduction in order to get new low-error extractors for small-space sources. In particular, while there are no known explicit general-purpose low-error affine extractors that can handle min-entropy significantly below $k = o(n)$, there *are* explicit low-error extractors for 1-local sources that can handle min-entropy slightly above $k = \sqrt{n}$ [Rao09b]. As a result, the above reduction actually provides an alternative construction of low-error small-space extractors. However, these extractors will require min-entropy $k \geq O(n^{1/2+\delta} s)$, which is worse than the parameters achieved in Theorem 7.3. Still, it seems useful to have a single, unifying way to extract from small-space sources using affine extractors at all error regimes.

A near-equivalence between small-space sources and 1-local sources Second, we observe that the reduction in Lemma 7.7 also works in the reverse direction, implying that small-space sources and simple 1-local sources are *roughly equivalent*, up to a factor of about s. In particular, using Definitions 7.2 and 7.4, it is relatively straightforward to show: a simple 1-local source $\mathbf{X}$ over n bits with min-entropy Γ is also a space $s = 1$ source over n bits with min-entropy Γ. Combining this with Lemma 7.7, we see that simple 1-local sources and space s sources are *roughly equivalent* (in the low-error convex combination sense), up to a factor of $\widetilde{O}(s)$.

We are now ready to prove Lemma 7.7. We will use an intermediate type of source, called an *independent source sequence*, which is a natural generalization of independent sources to allow for uneven (and unknown) length. We will show that small-space sources are (close to) a convex combination of independent source sequences, which are a convex combination of simple 1-local sources. We prove the latter first.

Definition 7.5 (Independent source sequence)**.** *A source $\mathbf{X}$ over $\{0,1\}^n$ is an (n, r, k)-*independent source sequence *if there exist some (unknown) lengths $\ell_1, \dots, \ell_r \in [n]$ that sum to n such that $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_r)$, where each $\mathbf{X}_i$ is an independent (ℓ_i, k)-source.*

While independent source sequences may not look like affine sources at first glance, it is actually easy to show that they are not only convex combinations of affine sources, but in fact simple 1-local sources!

Claim 7.1 (A reduction from independent source sequences to 1-local sources)**.** *If $\mathbf{X}$ be an $(n, \Gamma, 1)$-independent source sequence, it is a convex combination of simple 1-local sources with min-entropy Γ.*

Proof. We previously showed in Lemma 3.20 that *any* source with min-entropy at least 1 is a convex combination of 1-local sources with min-entropy exactly 1. Applying this to each independent source in the sequence $\mathbf{X}$ immediately yields the result. □

At last, we are finally ready to finish the proof of our reduction. We prove the following reduction from small-space sources to independent source sequences, which immediately yields our reduction from small-space sources to simple 1-local sources (Lemma 7.7) via Claim 7.1.

Lemma 7.8 (A reduction from small-space sources to independent source sequences)**.** *There is a universal constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy k. Then $\mathbf{X}$ is 2^{-ck}-close to a convex combination of $(n, \Gamma, 1)$-independent source sequences, where $\Gamma \geq \frac{ck}{s \log(2n/k)}$.*

Proof. Let $\mathbf{W}$ be the random walk over the width 2^s, length n branching program that generates $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n)$, and for each $i \in [n]$, let $\mathbf{L}_i$ be the vertex in layer i that is traversed by $\mathbf{W}$. In other words, $(\mathbf{L}_1, \mathbf{L}_2, \dots, \mathbf{L}_n)$ is a random variable over $[2^s]^n$ that lists the vertices visited by $\mathbf{W}$ in order (excluding the start vertex). Furthermore, for any $1 \leq i < j \leq n$, we define the *slice* $\mathbf{X}^{(i,j)} := (\mathbf{X}_{i+1}, \dots, \mathbf{X}_j)$.

For any indices $0 = i_0 < i_1 < \cdots < i_T = n$, it is straightforward to verify that the slices $\mathbf{X}^{(i_0,i_1)}, \mathbf{X}^{(i_1,i_2)}, \dots, \mathbf{X}^{(i_{T-1},i_T)}$ become mutually independent when conditioned on fixing $\mathbf{L}_{i_1}, \dots, \mathbf{L}_{i_{T-1}}$ to any $\ell_{i_1}, \dots, \ell_{i_{T-1}}$. Furthermore, given such a fixing, if we can guarantee that T' of these slices still have min-entropy at least 1 after this fixing, then the source $\mathbf{X}$ conditioned on this fixing must be an

$(n, T', 1)$-independent source sequence. This is simply because for each "good" slice with min-entropy 1, we can just concatenate to it all slices preceding it (until we reach another good slice or the start of the source), and for the last good slice with min-entropy 1, we can just concatenate to it all slices following it (until we reach the end of the source). Thus, the goal of this proof is to pick layers $\mathbf{L}_i$ to fix such that with high probability over these fixings, we can make the abovementioned guarantee for the largest T' possible. By the law of total probability, this will immediately imply $\mathbf{X}$ is close to a convex combination of $(n, T', 1)$-independent source sequences.

Let Γ, t be parameters that we will set later. Informally, we pick layers to fix in the following manner. We split up the branching program into 2Γ slices, and fix the layers in between them. Then, we will argue that with high probability, $\mathbf{X}$ still has most of its entropy, and so we must be in one of two situations:

(1) The Γ slices with the most entropy out of the 2Γ slices each have at least 1 bit of entropy, or

(2) They do not.

In case (1), $\mathbf{X}$ already looks like an $(n, \Gamma, 1)$-independent source sequence, and we are done. In case (2), we know the entropy must be highly concentrated in the Γ highest entropy slices, and so we can recurse on this sub-source that has half the size as the original source, but much more entropy. We will argue that it is impossible to forever avoid case (1) in this recursion, by showing that otherwise we would eventually (after at most t steps) find a sub-source with more entropy than its length, a contradiction. We will now describe our fixings more formally.

Fixings We pick layers to fix as follows.[14] We start by defining a set of indices $I^{(0)}$ that *split* the branching program into 2Γ slices of the same size. Namely, we define indices $0 = i_0^{(0)} < i_1^{(0)} < \cdots < i_{2\Gamma-1}^{(0)} < i_{2\Gamma}^{(0)} = n$ such that $i_j^{(0)} - i_{j-1}^{(0)} = \frac{n}{2\Gamma}$ for all $j \in [2\Gamma]$, and set $I^{(0)} := \{i_j^{(0)} : j \in [2\Gamma - 1]\}$. *We now fix* $(\mathbf{L}_i)_{i \in I^{(0)}}$ *to some string* $\ell^{(0)} \in [2^s]^{2\Gamma-1}$.

In order to decide what to fix next, we construct a set $B^{(0)}$ of the slices induced by $I^{(0)}$, and we construct a set $A^{(0)} \subseteq B^{(0)}$ of the Γ highest entropy slices indexed by $B^{(0)}$, conditioned on the fixing we just performed. More formally, we define $B^{(0)} = \{(i_0^{(0)}, i_1^{(0)}), (i_1^{(0)}, i_2^{(0)}), \ldots, (i_{2\Gamma-1}^{(0)}, i_{2\Gamma}^{(0)})\}$. We now pick the Γ *largest* elements from $B^{(0)}$ to create $A^{(0)}$, where *largest* is defined via the following total order: given $(a, b), (c, d) \in B^{(0)}$, we say $(a, b) > (c, d)$ if $H_\infty(\mathbf{X}^{(a,b)} \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)}) > H_\infty(\mathbf{X}^{(c,d)} \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)})$; or if these min-entropies are identical and $a > c$. We now check the min-entropies of the slices in $A^{(0)}$. If $H_\infty(\mathbf{X}^a \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)}) \geq 1$ for all $a \in A^{(0)}$, we stop our fixings here.

Otherwise, we proceed with more fixings. We initialize a counter $\tau = 1$, and use $(*)$ to refer to the current location of this text on this page (i.e., the beginning of a loop that we are creating). Then, we define a set of indices $I^{(\tau)}$ that *split* each of the good slices from the previous round of fixings. More formally, we define $I^{(\tau)} := \{\frac{a_1 + a_2}{2} : (a_1, a_2) \in A^{(\tau-1)}\}$. *We now fix* $(\mathbf{L}_i)_{i \in I^{(\tau)}}$ *to some string* $\ell^{(\tau)} \in [2^s]^\Gamma$.

In order to decide what to fix next, we construct a set $B^{(\tau)}$ of the new slices induced by $I^{(\tau)}$, and we construct a set $A^{(\tau)} \subseteq B^{(\tau)}$ of the Γ highest entropy slices indexed by $B^{(\tau)}$, conditioned on all of the fixings we have performed thus far. More formally, we define $B^{(\tau)} = \{(a_1, \frac{a_1 + a_2}{2}) : (a_1, a_2) \in A^{(\tau-1)}\} \cup \{(\frac{a_1 + a_2}{2}, a_2) : (a_1, a_2) \in A^{(\tau-1)}\}$. We now pick the Γ *largest* elements from $B^{(\tau)}$ to create $A^{(\tau)}$, where *largest* is defined via the following total order: given $(a, b), (c, d) \in B^{(\tau)}$, we say $(a, b) > (c, d)$ if $H_\infty(\mathbf{X}^{(a,b)} \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)}, (\mathbf{L}_i)_{i \in I^{(1)}} = \ell^{(1)}, \ldots, (\mathbf{L}_\tau)_{i \in I^{(\tau)}} =$

[14] To reduce notation, we assume throughout the proof that all divisions yield positive integers. It is easy to extend the arguments to handle when this is not the case.

$\ell^{(\tau)}) > H_\infty(\mathbf{X}^{(c,d)} \mid (\mathbf{L}_i)_{i\in I^{(0)}} = \ell^{(0)}, (\mathbf{L}_i)_{i\in I^{(1)}} = \ell^{(1)}, \dots, (\mathbf{L}_\tau)_{i\in I^{(\tau)}} = \ell^{(\tau)})$; or if these min-entropy are identical and $a > c$. We now check the min-entropies of the slices in $A^{(\tau)}$. If $H_\infty(\mathbf{X}^a \mid (\mathbf{L}_i)_{i\in I^{(0)}} = \ell^{(0)}, (\mathbf{L}_i)_{i\in I^{(1)}} = \ell^{(1)}, \dots, (\mathbf{L}_\tau)_{i\in I^{(\tau)}} = \ell^{(\tau)}) \geq 1$ for all $a \in A$, we stop our fixings here. Also, if $\tau = t$, we stop our fixings here. Otherwise, we increment $\tau \leftarrow \tau + 1$, and we go back to $(*)$.[15] This concludes our fixings.

Analysis For convenience, let $\mathbf{Q}$ denote a single random variable such that fixing $\mathbf{Q}$ is equivalent to performing all of the fixings described above. Note that $\mathbf{Q}$ is a deterministic function of $(\mathbf{L}_1, \dots, \mathbf{L}_n)$, and is of the form $(\mathbf{L}_i)_{i\in I}$, where I is not a single constant subset of $[n]$, but is chosen adaptively. Furthermore, observe that not all elements in the support of $\mathbf{Q}$ have the same length (depending on when the fixing of layers stopped); indeed, $\mathbf{Q}$ is a random variable over $[2^s]^{2\Gamma-1} \cup [2^s]^{2\Gamma-1+\Gamma} \cup \cdots \cup [2^s]^{2\Gamma-1+t\Gamma}$. However, notice that for every $q_1, q_2 \in \text{support}(\mathbf{Q})$, q_1 is cannot be a prefix of q_2; it is therefore straightforward to construct an injection from $\text{support}(\mathbf{Q}) \to [2^s]^{2\Gamma-1+t\Gamma}$, and so $|\text{support}(\mathbf{Q})| \leq 2^{s\cdot((t+2)\Gamma-1)} \leq 2^{(t+2)s\Gamma}$.

Recall that we currently have parameters Γ, t that we said we would fix later. We will add ε to the parameters that we will fix later. The goal now is to show that with probability $1-\varepsilon$ over fixing $\mathbf{Q}$ to q, the conditional distribution $(\mathbf{X} \mid \mathbf{Q} = q)$ is an $(n, \Gamma, 1)$-independent source sequence, since this would immediately imply $\mathbf{X}$ is ε-close to a convex combination of $(n, \Gamma, 1)$-independent source sequences. We would like to show this holds for the best possible choices of Γ, ε, t.

We start by invoking Lemma 2.1, which tells us that with probability at least $1-\varepsilon$ over fixing $\mathbf{Q}$ to q, we have $H_\infty(\mathbf{X} \mid \mathbf{Q} = q) \geq k - \log(|\text{support}(\mathbf{Q})|) - \log(1/\varepsilon) \geq k - (t+2)s\Gamma - \log(1/\varepsilon)$. Consider now some fixing $\mathbf{Q} = q$ where this holds. We know that there is some $\tau^* \in \{0, 1, \dots, t\}$ such that $q \in [2^s]^{2\Gamma-1+\tau^*\Gamma}$, where τ^* simply counts the number of times we iterated through the fixing loop from above. Recall that by definition, the fixing $\mathbf{Q} = q$ refers to the fixings $(\mathbf{L}_i)_{i\in I^{(0)}} = \ell^{(0)}, \dots, (\mathbf{L}_i)_{i\in I^{(\tau^*)}} = \ell^{(\tau^*)}$.

Thus, by definition of our fixing procedure, the source $\mathbf{X} \mid \mathbf{Q} = q$ is simply the concatenation of the slices $(\mathbf{X}^{(\beta,\beta')} | \mathbf{Q} = q)$, where (β, β') ranges over the set

$$(B^{(0)} \setminus A^{(0)}) \cup (B^{(1)} \setminus A^{(1)}) \cup \cdots \cup (B^{(\tau^*)} \setminus A^{(\tau^*)}) \cup A^{\tau^*},$$

and the concatenation happens in increasing order of β'. Also notice that the unions above are in fact disjoint. Furthermore, given our discussion at the very beginning of the proof, we know that these slices are mutually independent, because of the conditioning on the layers separating them. Now, we could be in one of two cases: either $\tau^* < t$, or $\tau^* = t$.

Case (1): $\tau^* < t$. In this case, by definition of our fixing procedure, we know that the Γ distinct slices in $(\mathbf{X} \mid \mathbf{Q} = q)$ that are indexed by $A^{(\tau^*)}$ each have min-entropy at least 1. Thus, $(\mathbf{X} \mid \mathbf{Q} = q)$ is a sequence of independent slices, with the guarantee that at least Γ of them have min-entropy at least 1. By our discussion at the very beginning of this proof, $(\mathbf{X} \mid \mathbf{Q} = q)$ is an $(n, \Gamma, 1)$-independent source sequence.

Case (2): $\tau^* = t$. In this case, observe that in our fixing procedure, we only proceed from iteration j to $j+1$ in the loop if some slice in $A^{(j)}$ has min-entropy < 1, which means that all slices in $B^{(j)} \setminus A^{(j)}$ have min-entropy < 1 (since $A^{(j)}$ contains the Γ slices with the highest min-entropy out of the 2Γ slices in $B^{(j)}$). Thus, for every $(\beta, \beta') \in (B^{(0)} \setminus A^{(0)}) \cup (B^{(1)} \setminus A^{(1)}) \cup \cdots \cup (B^{(\tau^*-1)} \setminus A^{(\tau^*-1)})$, we know $H_\infty(\mathbf{X}^{(\beta,\beta')} \mid \mathbf{Q} = q) < 1$. (This was also true in the previous case, but we did not need this observation there.) For all other $(\beta, \beta^*) \in (B^{(\tau^*)} \setminus A^{(\tau^*)}) \cup A^{(\tau)}$, it trivially holds that $H_\infty(\mathbf{X}^{(\beta,\beta')} \mid \mathbf{Q} = q) \leq \beta' - \beta$, since this slice is just a random variable over $\beta' - \beta$ bits. It is straightforward to show that $\beta' - \beta = \frac{n}{2\Gamma} \cdot 2^{-\tau^*}$, since our first slices $B^{(0)}$ divide the n bit source into 2Γ equal sized pieces, and $A^{(0)} \subseteq B^{(0)}$, and at each iteration

[15] In order for this process to be well-defined, we should stop if there is a slice $(a, b) \in A^{(\tau)}$ with $b - a = 1$. We will make sure to set our parameter t to guarantee this.

j of the loop we cut each slice from $A^{(j-1)}$ in half to get $B^{(j)}$. Thus, $H_\infty(\mathbf{X}^{(\beta,\beta')} \mid \mathbf{Q} = q) \leq \frac{n}{2\Gamma} \cdot 2^{-\tau^*}$ for all $(\beta, \beta^*) \in (B^{(\tau^*)} \setminus A^{(\tau^*)}) \cup A^{(\tau)}$.

Thus, we know an upper bound on the entropy of each slice in $(B^{(0)} \setminus A^{(0)}) \cup (B^{(1)} \setminus A^{(1)}) \cup \cdots \cup (B^{(\tau^*-1)} \setminus A^{(\tau^*-1)}) \cup (B^{(\tau^*)} \setminus A^{(\tau^*)}) \cup A^{\tau^*}$. Furthermore, the sets in the union are disjoint, and each set in this union contains Γ distinct slices, which we have already mentioned are mutually independent. Thus, we have:

$$\begin{aligned} H_\infty(\mathbf{X} \mid \mathbf{Q} = q) &< (1 + \tau^* - 1) \cdot \Gamma \cdot 1 + (1 + 1) \cdot \Gamma \cdot \left(\frac{n}{2\Gamma} \cdot 2^{-\tau^*} \right) \\ &= \Gamma\tau^* + n \cdot 2^{-\tau^*} \\ &= \Gamma t + n2^{-t}. \end{aligned}$$

Combining this with the assumption we made about q near the beginning of our analysis, we have:

$$k - (t+2)s\Gamma - \log(1/\varepsilon) \leq H_\infty(\mathbf{X} \mid \mathbf{Q} = q) < \Gamma t + n2^{-t}. \tag{7.1}$$

We finally arrive at our last goal: setting parameters Γ, t, ε. We know that for any setting of these parameters that contradicts Equation (7.1), Case (2) simply cannot occur. Thus, for any such setting, we know that with probability $1 - \varepsilon$ over fixing $\mathbf{Q} = q$, we have $H_\infty(\mathbf{X} \mid \mathbf{Q} = q) \geq k - (t+2)s\Gamma - \log(1/\varepsilon)$, and this implies Case (1) must occur. In other words, with probability $1 - \varepsilon$ over $q \sim \mathbf{Q}$, we have that $(\mathbf{X} \mid \mathbf{Q} = q)$ is an $(n, \Gamma, 1)$-independent source sequence, which immediately implies that $\mathbf{X}$ is ε-close to a convex combination of $(n, \Gamma, 1)$-independent source sequences.

So all that remains is to pick the best possible Γ, t, ε that contradict Equation (7.1), and in particular show that our selected Γ, ε matches the claimed parameters in the theorem statement. We only have one minor restriction in our freedom to pick these parameters. We briefly recall the footnote from our fixings procedure, and note that the only requirement we have is that t is set so the procedure remains valid; namely, so that for every $\tau \in [t]$ and $(\beta, \beta') \in B^{(\tau)}$ created by the fixing procedure, $\beta' - \beta \geq 1$, since this will ensure that we are creating valid slices. Above, we showed that $\beta - \beta = \frac{n}{2\Gamma} \cdot 2^{-\tau}$, and so the only restriction we have is that $\frac{n}{2\Gamma} \cdot 2^{-t} \geq 1$.

Thus, to complete the proof, we may pick any Γ, t, ε that satisfy the above restriction, while contradicting Equation (7.1). In particular, these parameters just need to satisfy

$$\begin{gathered} k - (t+2)s\Gamma - \log(1/\varepsilon) \geq \Gamma t + n2^{-t}, \text{ and} \\ \frac{n}{2\Gamma} \cdot 2^{-t} \geq 1. \end{gathered}$$

Combining these, we just require:

$$\Gamma \leq \min \left\{ \frac{k - n2^{-t} - \log(1/\varepsilon)}{t \cdot (s-1) + 2s}, \frac{n}{2^{t+1}} \right\}.$$

We take $t := \log(4n/k)$ and $\varepsilon := 2^{-k/2}$ and $\Gamma := \frac{k}{20s \cdot \log(2n/k)}$ to complete the proof. □

This completes the construction of our two new extractors for small-space sources. Next, we will discuss one interesting bonus application of these results.

7.4 Applications in complexity theory

Given that small-space sources fall into the category of *low-complexity sources*, it may not be surprising to learn that our new extractor constructions have applications in complexity theory. What may be surprising, however, is that these applications do not directly come from our extractors, but instead come from the *structural result* that we proved: namely, our reduction from small-space sources to 1-local sources (Lemma 7.7). Using our reduction, we are able to prove an interesting new result on the power of read-once branching programs (ROBPs) for the task of *sampling from distributions*. We go into more detail, below.

7.4.1 Complexity of sampling

Recently, a rich new subfield of complexity theory, called the *complexity of sampling*, has been the subject of intense study. While classical complexity theory seeks to understand the power of various computational models for the task of *computing* a Boolean function $f : \{0,1\}^\ell \to \{0,1\}^n$, the complexity of sampling seeks to understand the power of these same models for the alternative task of *sampling* a distribution $\mathbf{Q} \sim \{0,1\}^n$. More formally, we say that an instance of a computational model (e.g., a low-depth circuit) $F : \{0,1\}^\ell \to \{0,1\}^n$ *samples* the distribution $\mathbf{Q}$ if $F(\mathbf{U}_\ell) = \mathbf{Q}$. In Chapter 9, we give a comprehensive overview of this exciting new area of study, and contribute some of our own results.

A key motivating paradigm behind research into the complexity of sampling is the fact that sampling lower bounds are harder to obtain (and thus more desirable) than classical lower bounds. Indeed, for many computational models, there exist explicit functions $b : \{0,1\}^n \to \{0,1\}$ such that the map $x \mapsto (x, b(x))$ is hard to compute, yet the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ is easy to sample. This motivates researchers to seek out simple functions b for which $(\mathbf{U}_n, b(\mathbf{U}_n))$ is *hard* to sample.

Sampling lower bounds against low-degree polynomials

A well-known result of Viola [Vio16] shows that there exist *degree* 2 *polynomials* $b : \mathbb{F}_2^n \to \mathbb{F}_2$ such that AC^0 circuits cannot sample $(\mathbf{U}_n, b(\mathbf{U}_n))$. As it turns out, by exploiting our new reduction from small-space sources to 1-local sources (Lemma 7.7), we can obtain an analogous result for ROBPs (Definition 9.3): the canonical model for space-bounded computation. In fact, we prove something that is stronger in two notable ways. First, we prove sampling lower bounds against the entire spectrum of low-degree polynomials, achieving a tradeoff between the degree of the polynomial and the complexity required of an ROBP that can sample it. Second, our sampling lower bounds even work against distributions of the form $(\mathbf{X}, b(\mathbf{X}))$, where $\mathbf{X}$ can be taken as any random variable with sufficient min-entropy - including, of course, the uniform distribution. More formally, we prove the following.

Theorem 7.5 (Low-degree polynomials are hard to sample)**.** *There exists a universal constant $c > 0$ such that for any $n \geq r \geq 2 \in \mathbb{N}$, there exists a degree r polynomial $b : \mathbb{F}_2^n \to \mathbb{F}_2$ such that the following holds. For any random variable $\mathbf{X} \sim \{0,1\}^n$ with min-entropy k, there does not exist an ROBP of width*

$$w \leq \exp\left(\frac{ck}{r \cdot (n \log n)^{1/r} \cdot \log(n/k)}\right)$$

that can sample $(\mathbf{X}, b(\mathbf{X}))$.

For the setting of degree 2 polynomials $b : \mathbb{F}_2^n \to \mathbb{F}_2$ and full min-entropy, Theorem 7.5 shows that ROBPs of *exponential width* $w \leq 2^{\Omega(\sqrt{n/\log n})}$ cannot sample $(\mathbf{U}_n, b(\mathbf{U}_n))$. It is straightforward to show that ROBPs of width 2^n *can* sample any such distribution, and thus our lower bounds are quite strong.

Even more notably, however, is the fact that ROBPs can compute (and thus sample) any degree 1 (linear) polynomial $b : \{0,1\}^n \to \{0,1\}$ in width 2. Thus, it is fascinating to note that increasing the degree from $r = 1$ to $r = 2$ forces the width of the ROBP to jump from $w = 2$ all the way up to $w = 2^{\Omega(\sqrt{n/\log n})}$.

An overview of the proof

Let us now proceed to prove Theorem 7.5. In order to prove sampling lower bounds against distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ for a computational model $\mathcal{F}$, the classic intuition is to try to set b to an extractor for distributions that can be sampled by $\mathcal{F}$. Almost by definition, this will imply that the uniform distribution over $b^{-1}(0)$ or $b^{-1}(1)$ (whichever is bigger) cannot be sampled by $\mathcal{F}$. However, it does not necessarily imply that $(\mathbf{U}_n, b(\mathbf{U}_n))$ cannot be sampled by $\mathcal{F}$. Indeed, if one tries to run this argument, the main issue they will run into is the fact that even if a distribution $\mathbf{Y} \sim \{0,1\}^{n+1}$ can be generated by $\mathcal{F}$, the same is not necessarily true of the random variable $\mathbf{Y}' := (\mathbf{Y} \mid \mathbf{Y}_{n+1} = y)$. Thus, there is little hope that one can pursue a contradiction by fixing the last bit of $\mathbf{Y}$ to the more likely value y, and arguing that if $\mathbf{Y}$ truly sampled $(\mathbf{U}_n, b(\mathbf{U}_n))$, then the existence of $\mathbf{Y}'$ contradicts the fact that b extracts from $\mathcal{F}$.

In order to circumvent the above issue, the key idea is to first reduce distributions samplable by $\mathcal{F}$ into distributions corresponding to a simpler class, say $\mathcal{F}'$. If the class $\mathcal{F}'$ is simple enough, then it actually is possible that the random variable $\mathbf{Y}$ remains in $\mathcal{F}'$, even upon fixing its last bit. Now, in order to execute this plan to prove Theorem 7.5, we naturally set $\mathcal{F}$ to be the class of ROBPs, but what should we set b and $\mathcal{F}'$ to? Well, we absolutely need b to be a low-degree polynomial, since these are the distributions claimed to be hard in Theorem 7.5. As for $\mathcal{F}'$, we set it to the family of 1-local sources, as there is some hope that distributions samplable by ROBPs *look like* small-space sources, which we know to be reducible to 1-local sources via the key lemma from this chapter (Lemma 7.7). To make this plan work, all we need is to show that ROBPs do indeed look like small-space sources, and that low-degree polynomials b can extract from 1-local sources. As it turns out, we actually prove both such results in this thesis, and can thereby successfully execute the above plan. This completes the proof overview of Theorem 7.5, which interestingly uses one new ingredient from every chapter of this part of the thesis (Chapters 7 to 9).

A formal presentation of the proof

We can now proceed with the formal proof of Theorem 7.5. As suggested above, we use one new ingredient from each chapter in Part III. The first ingredient we need is a reduction showing that any distribution $\mathbf{X} \sim \{0,1\}^n$ samplable by an ROBP looks like a small-space source of roughly the same size.

Lemma 7.9 (A reduction from ROBP samplers to small-space sources - Theorem 9.0, restated)**.** *For any $\mathbf{X} \sim \{0,1\}^n$, if there is an ROBP of width w that samples $\mathbf{X}$, then $\mathbf{X}$ is a space $s = \log(2w)$ source.*

The second key ingredient we need is the main reduction underpinning the current chapter, which shows that any small-space source is close to a convex combination of 1-local sources. By composing this with the previous ingredient, we get that any distribution samplable by an ROBP looks like a 1-local source.

Lemma 7.10 (A reduction from small-space sources to 1-local sources - Lemma 7.7, restated)**.** *There is a universal constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a space s source of min-entropy k. Then $\mathbf{X}$ is 2^{-ck}-close to a convex combination of 1-local sources of min-entropy $\frac{ck}{s\log(2n/k)}$.*

Finally, the third key ingredient shows that (non-explicit) low-degree polynomials are great extractors for 1-local sources. Such a result is almost implied by a classic paper of Cohen and Tal [CT15]. However, they prove that low-degree polynomials are good extractors for affine sources, and thus obtain slightly weaker parameters (see Section 8.4.1 for a comparison).

Lemma 7.11 (Low-degree polynomials extract from 1-local sources - Theorem 8.2, specialized)**.** *There exist constants $C, c > 0$ such that for all $n \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. There exists a degree r polynomial $b : \mathbb{F}_2^n \to \mathbb{F}_2$ that is an extractor for 1-local sources of min-entropy*

$$k \geq Cr \cdot (n \log n)^{1/r},$$

which has error $\varepsilon = 2^{-ck/r}$.

Equipped with these three ingredients, we can finally prove that low-degree polynomials are hard to sample by ROBPs (Theorem 7.5).

Proof of Theorem 7.5. Let $b : \{0,1\}^n \to \{0,1\}$ be a degree r polynomial that is an extractor for 1-local sources of min-entropy $\Gamma \geq Cr \cdot (n \log n)^{1/r}$ which has error $\varepsilon = 2^{-c\Gamma/r}$, as promised by Lemma 7.11, and let $\mathbf{X} \sim \{0,1\}^n$ be an arbitrary distribution of min-entropy k, where k is to be decided later. Let $\mathbf{Q} \sim \{0,1\}^{n+1}$ be a distribution that is generated by an ROBP of width w, and parse $\mathbf{Q}$ as $\mathbf{Q} = (\mathbf{R}, \mathbf{S}) \sim \{0,1\}^n \times \{0,1\}$. Suppose for contradiction that $\mathbf{Q} \equiv (\mathbf{X}, b(\mathbf{X}))$, and notice that this implies $H_\infty(\mathbf{Q}) = k$.

Now, by combining Lemmas 7.9 and 7.10, we know that $\mathbf{Q} = (\mathbf{R}, \mathbf{S})$ is $2^{-k'}$-close to a convex combination of 1-local sources $\mathbf{Q}' = (\mathbf{R}', \mathbf{S}')$ that have min-entropy at least

$$k' := \frac{ck}{\log(2w) \cdot \log(2(n+1)/k)}.$$

Furthermore, by the min-entropy chain rule (Lemma 2.1) and the fact that 1-local sources remain 1-local even upon fixing a bit, we know that each $\mathbf{Q}' = (\mathbf{R}', \mathbf{S}')$ is $2^{-k'/2}$-close to a convex combination of 1-local sources $\mathbf{Q}'' = (\mathbf{R}'', \mathbf{S}'')$ such that $\mathbf{S}''$ is fixed and $\mathbf{R}''$ has min-entropy at least $k'' := k'/2 - 1$. Putting this all together, we get that $\mathbf{Q}$ is $2^{-k''}$-close to a convex combination of 1-local sources $\mathbf{Q}'' = (\mathbf{R}'', \mathbf{S}'') \sim \{0,1\}^{n+1}$ such that $\mathbf{S}''$ is fixed and $\mathbf{R}''$ has min-entropy at least

$$k'' = \frac{ck}{2\log(2w) \cdot \log(2(n+1)/k)} - 1.$$

Now, define k so that $k'' \geq Cr \cdot (n \log n)^{1/r}$. Then we know that b in fact extracts from each $\mathbf{R}''$ defined above with error $\varepsilon = 2^{-ck''/r}$, and thus since $\mathbf{S}''$ is constant, we have

$$\Pr[b(\mathbf{R}'') \neq \mathbf{S}''] \geq \frac{1}{2} - \varepsilon.$$

By the closeness of $\mathbf{Q} = (\mathbf{R}, \mathbf{S})$ to this convex combination of such sources, we get that

$$\Pr[b(\mathbf{R}) \neq \mathbf{S}] \geq \frac{1}{2} - \varepsilon - 2^{-k''} \geq \frac{1}{2} - 2^{-c_1 k''/r}$$

for some constant $c_1 > 0$. But at the very beginning we assumed that $(\mathbf{R}, \mathbf{S}) \equiv (\mathbf{X}, b(\mathbf{X}))$, which means that $\Pr[b(\mathbf{R}) \neq \mathbf{S}] = 0$. Thus, we arrive at a contradiction because $c_1 k''/r > 1$ and thus $\frac{1}{2} - 2^{-c_1 k''/r} > 0$ by our setting of k''. Finally, recall that this setting of k'' (which yielded the contradiction) came about by fixing k so that

$$\frac{ck}{2\log(2w) \cdot \log(2(n+1)/k)} - 1 \geq Cr \cdot (n \log n)^{1/r}.$$

Solving for w and resetting constants appropriately yields

$$w \leq \exp\left(\frac{ck}{r \cdot (n \log n)^{1/r} \cdot \log(n/k)}\right),$$

as desired.

□

7.5 Future directions

In this chapter, we gave two new-and-improved extractors for small-space sources: one in the high-error regime (Theorem 7.1) and one in the low-error regime (Theorem 7.2). While our new *low-error extractors* for small-space sources were a direct consequence of the adversarial source extractors we constructed in Chapter 3, our new high-error extractors (which achieve *polylogarithmic min-entropy*) precipitated from a brand new reduction from small-space sources to affine sources (Lemma 7.1). Following our work, further exciting progress on small-space sources has been achieved. In a recent paper [CL22], Chattopadhyay and Liao improved the above structural result by showing that there is an *even more efficient* reduction from small-space sources to *sumset sources*, a model which naturally generalizes affine sources. Even more recently [Li23], Li gave a breakthrough construction of sumset extractors that can handle truly logarithmic min-entropy. Thus, combining these two results, one immediately obtains high-error small-space extractors for truly logarithmic min-entropy. Despite this amazing progress, many natural questions around small-space sources remain open. We outline two of them, below.

Improved low-error extractors for simple 1-local sources While we now know how to extract from small-space sources with truly logarithmic min-entropy, these extractors have high error. In the low-error regime, the best extractors still require over $k \geq \sqrt{n}$ bits of min-entropy. Since low-error extractors are crucial for most applications in cryptography [DOPS04], it is important that we make further progress on this min-entropy requirement. Using our new reduction to affine sources, any significant improvement to the state-of-the-art in low-error affine extraction would immediately yield new low-error extractors for small-space sources. Notably, however, such an affine extractor isn't even needed! Indeed, recall that our structural result actually reduces small-space sources to *simple* 1*-local sources* (Lemma 7.7), which are much more basic than full-blown affine sources. Surely, it seems like it should be easier to construct low-error extractors for this model. However, to date, no progress has been made on this tantalizing problem.

Extractors for read-k small-space sources Another natural direction is to construct extractors for sources that more realistically correspond to small-space computation. In particular, recall that one of the original motivating factors for extracting from samplable sources (of which *small-space sources* are a special case) was the belief that they might actually model distributions found in nature [TV00]. However, as we will see in Chapter 9, small-space sources only correspond to distributions samplable by *read-once branching programs* (ROBPs), which may not be a realistic model for distributions in nature. In particular, while these programs remain one of the most classical ways to model space-bounded computation, they are inherently limited: as per their name, ROBPs are only allowed to read each bit in their input *once*. Because of this, it is natural to study a generalization of ROBPs that can read each bit in their input k *times*. These objects are known as *read-k branching programs (RkBPs)*, and have received a substantial amount of attention in classical complexity theory [BRS93, Tha98, BJS01]. To date, however, there has yet to be a study on extracting from distributions generated by RkBPs. It would be great to develop such a theory, given that small-space sources already capture a variety of well-studied models.

Chapter 8

Extractors for circuit sources

We now turn from small-space sources to *circuit sources*. While both of these models belong to the family of *low-complexity sources*, circuit sources were the original model studied in the seminal paper of Trevisan and Vadhan [TV00]. The definition of these sources is simple: a circuit source $\mathbf{X}$ is just a distribution that can be generated by feeding uniform bits into a Boolean circuit. By restricting the power (size, depth, etc.) of the circuit in various ways, one can study a variety of different flavors of circuit sources. The most popular flavor, however, is the model of *local sources*.

A local source $\mathbf{X}$ is simply a source where each bit depends on a small number of (hidden) uniformly random input bits. This model was first introduced by De and Watson [DW12] and Viola [Vio14], and has since established its role as a central figure in the world of circuit sources. Indeed, local sources not only generalize sources samplable by NC^0 circuits, but they also capture sources generated by AC^0 circuits and even read-once branching programs. In other words, extractors for local sources also work for sources generated by these classical computational models.

Despite being introduced over a decade ago, little progress has been made on improving the entropy requirement for extracting from local sources. The current best explicit extractors require min-entropy $k \geq \sqrt{n}$, and follow via a reduction to affine extractors. To start, we prove a barrier result showing that one cannot hope to improve this entropy requirement via a black-box reduction of this form. Given this impossibility result for *black-box* affine extractors, we turn our attention to a simple family of functions that are known to perform as excellent *white-box* affine extractors: low-degree polynomials (over $\mathbb{F}_2$). We ask,

Can this simple family of functions help break the $\sqrt{n}$ barrier?

In this chapter, we answer the above question in the positive, and *fully characterize the power of low-degree polynomials* as extractors for local sources. More precisely, we show that a random degree r polynomial is a low-error extractor for n-bit local sources with min-entropy $k \geq O(r(n \log n)^{1/r})$, and we show that this is tight. Our result leverages several new ingredients, which may be of independent interest. Our positive result relies on a new explicit reduction from local sources to a more structured family, known as *local non-oblivious bit-fixing sources*. To show its tightness, we prove a new type of *Chevalley-Warning theorem*, which guarantees the existence of *low-weight* solutions to systems of low-degree polynomials.

The results presented in this chapter are based on the following joint work:

- [ACG^{+}22] Omar Alrabiah, Eshan Chattopadhyay, Jesse Goodman, Xin Li, and João Ribeiro. Low-degree polynomials extract from local sources. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022

8.1 Introduction

As we have seen, the model of *samplable sources* (within the broader family of *low-complexity sources*) is a rich setting in which to study seedless extraction.[1] In the previous chapter, we explored one half of this setting, by studying sources samplable in limited *space*. In this chapter, we explore the *other* half of this setting, by studying sources samplable in limited *time*. As always, the ultimate goal is to construct an *extractor*, so let's kick things off with this familiar definition.

Definition 8.1 (Extractor)**.** *Let $\mathcal{X}$ be a family of distributions over $\{0,1\}^n$. A function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is called an* extractor for $\mathcal{X}$ with error ε *if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \varepsilon.$$

In order to model a source $\mathbf{X} \in \mathcal{X}$ that is samplable in limited *time*, the most traditional approach might be to use a *Turing machine*. Indeed, this has historically been the most classical way to model computation. In the world of complexity theory, however, *Boolean circuits* have gradually emerged as the model of choice, as their simple definition and combinatorial nature make them much easier to reason about [AB09]. The same is true in the world of extractors, and more specifically, in the study of *samplable sources*. In particular, while some work (on sources samplable in limited time) *does* study Turing machines [Vio12b], most of the work on this subject focuses on *circuits* [TV00, DW12, Vio14, Li16, ACG+22]. In this chapter, we continue this trend, and study *circuit sources*.

Circuit sources A circuit source $\mathbf{X} \sim \{0,1\}^n$ is simply a source of the form $f(\mathbf{U}_\ell)$ for some function f coming from a "low-complexity" circuit class $\mathcal{C}$.[2] As it turns out, this was the original model considered by Trevisan and Vadhan in their founding paper on samplable sources [TV00]. There, they prove that such sources are, in fact, *extremely difficult* to extract from. In particular, they show that any explicit extractor for sources generated by *relatively small circuits* would break longstanding barriers in circuit lower bounds. Given this result, they focus instead on constructing extractors under complexity-theoretic assumptions. Under these assumptions, they successfully extract from sources of min-entropy $k \geq 0.99n$ that are sampled by circuits of polynomial size.

If one hopes to *unconditionally* extract from circuit sources, it seems all-but-necessary to consider more restricted circuit classes $\mathcal{C}$. In particular, while the circuits studied above had a restriction on their *size*, one might also consider restricting their *depth*. This leads us to two of the most well-known circuit classes in complexity theory.

- NC^0 *circuits*, which have constant depth, polynomial size, and consist of $\mathsf{AND}, \mathsf{OR}$, and NOT gates of fan-in at most 2.
- AC^0 *circuits*, which have constant depth, polynomial size, and consist of $\mathsf{AND}, \mathsf{OR}$, and NOT gates of unbounded fan-in.

Given these circuit classes, we can now define two interesting families of circuit sources that may actually yield (unconditional) explicit extractors. In particular, we can define NC^0 *sources* to be of the form $\mathbf{X} = f(\mathbf{U}_\ell)$ for an NC^0 circuit f, and we can define AC^0 *sources* analogously. In the years following the original paper of Trevisan and Vadhan [TV00], all work on circuit sources has focused on (unconditionally) extracting from these families [DW12, Vio14, Li16, ACG+22].[3]

[1] We remind the reader that a full backstory on this model is provided at the beginning of Chapter 7.

[2] Here and throughout, there won't be any restriction on the number ℓ of random bits fed as *input* to the circuit. Instead, the "complexity" of the circuit class will be in terms of the number n of bits *output* by the circuit.

[3] This is not to mention recently heard whispers of an exciting follow-up to [TV00] coming soon [BDSGM23].

From circuit sources to local sources The first to consider extracting from NC^0 and AC^0 sources were De and Watson [DW12] and Viola [Vio14]. While the former work focuses on NC^0 sources [DW12] and the latter studies both NC^0 *and* AC^0 sources [Vio14], they both construct (unconditional) explicit extractors via the *same key idea*. Namely, they introduce a *fundamental new model* known as *local sources*, and focus on extracting from that, instead. This new model is defined as follows.

Definition 8.2 (Local source)**.** *A source* $\mathbf{X} \sim \{0,1\}^n$ *is called a* d-local source *if there exists some* $\ell \in \mathbb{N}$ *and a function* $g : \{0,1\}^\ell \to \{0,1\}^n$ *such that each output bit depends on at most* d *input bits, and* $\mathbf{X} = g(\mathbf{U}_\ell)$*.*

Despite this exceptional simplicity, local sources *generalize* both NC^0 *and* AC^0 sources. For NC^0 sources, this is not hard to see: indeed, upon staring at Definition 8.2, it becomes clear that $O(1)$-local sources are *identical* to NC^0 sources. For AC^0 sources, the connection is much less obvious. However, Viola provided a beautiful argument [Vio14] showing that AC^0 sources can *also* be reduced to $O(1)$-local sources, and this reduction comes at a *very small* loss in min-entropy (Lemma 8.14). Moreover, if you're willing to lose just a little more (Lemma 8.3), these reductions can actually be pushed all the way to 1-*local sources* [DW12, Vio14] - meaning that even a humble *extractor for* 1-*local sources* works for NC^0 and AC^0! All of this is to say:

If you wish to extract from NC^0 *and* AC^0 *circuit sources,*
it suffices to extract from local sources*, instead.*

In fact, given the simplicity of Definition 8.2 and the connections highlighted above, it becomes apparent that *local sources* may actually be the more fundamental model. Indeed, even for the case of 1-*local sources*, extractors for this model can not only handle NC^0 sources and AC^0 sources (as discussed above), but also *small-space sources* (Lemma 7.7), *oblivious bit-fixing sources* (Part II), and - of course - the *classical sources of von Neumann* [vN51]. For d-local sources of higher locality $d > 1$, the connection to circuit sources only grows stronger (Lemma 8.14), along with their ability to unlock exciting new applications in complexity theory (Section 8.6). Thus, following in the footsteps of the work that comes before us [DW12, Vio14, Li16], this chapter will focus on extracting from *local sources*.[4]

Prior constructions As always, the holy grail is to construct a great explicit extractor. Unfortunately, it is *deceptively hard* to extract from local sources. Even for the most basic setting of 1-*local sources*, surprisingly little is known. Indeed, while any *affine extractor* (Chapter 6) can be used to extract from 1-local sources, this is essentially the only known approach for doing so.[5] This is not an issue in the *high-error* regime, as we now have affine extractors for min-entropy $k \geq \log^{1+o(1)}(n)$ with constant error (Theorem 6.1). In the *low-error* regime, however, the best known affine extractors [Bou07, Yeh11, Li11b] still require very high min-entropy $k \geq O(n/\sqrt{\log \log n})$. Luckily, Rao's *low-weight* affine extractor [Rao09b, DW12, Vio14] can partially come to the rescue here, and extract (with low error) from 1-local sources of slightly less min-entropy $k \geq O(n^{0.51})$.[6] Beyond this, however, nothing else is known.

The situation only gets worse when we move to higher locality. Even for the setting of 2-*local sources*, there is a giant gap between what is possible existentially and explicitly. *Non-explicitly*, it is not hard to show that there exist extractors for extremely low entropy $k \geq O(\log n)$ with extremely low error $\varepsilon = 2^{-\Omega(k)}$. *Explicitly*, on the other hand, the current best extractors for 2-local sources are as follows.

[4] For posterity, we will also provide in Section 8.5 the consequences of our results in the worlds of NC^0 and AC^0.

[5] This is concerning, as 1-local sources seem *significantly simpler* than general affine sources.

[6] This is because a 1-local source of min-entropy k is a convex combination of affine sources of min-entropy $k/2$ and weight $2n/k$. Thus, Rao's low-weight affine extractor (Chapter 6) will work as long as $(2n/k) \leq (k/2)^{0.99}$, or rather $k \geq O(n^{0.51})$.

- *The low-error setting*: There exist explicit extractors for min-entropy $k \geq O(n^{0.67})$ [DW12, Vio14].
- *The high-error setting*: There exist explicit extractors for min-entropy $k \geq O(n^{0.51})$ [Li16].

Indeed, unlike the situation for 1-*local sources*, the best-known *high-error* explicit extractor for 2-*local sources* still has a huge min-entropy requirement of $k \geq O(n^{0.51})$.[7] In fact, this is one of the worst min-entropy requirements in the entire world of seedless extraction,[8] despite the fact that 2-local sources seem so simple, and were introduced over a decade ago. What gives?

A barrier for known techniques If one were to venture into the actual *techniques* that underlie any explicit extractor for local sources, they would come to find that they all rely on the same key tool: *a reduction to affine sources*. This reduction, first obtained by De and Watson [DW12] and Viola [Vio14], shows that any $O(1)$-local source of min-entropy k is close to a convex combination of affine sources of min-entropy $\Omega(k^2/n)$ (Theorem 8.4). Given this result (and the best-known affine extractors), it begins to become clear *why* we know how to extract from local sources with min-entropy just above $k \geq \sqrt{n}$. However, it should also be clear that this reduction cannot possibly lead to local source extractors for min-entropy $k < \sqrt{n}$. It is natural to wonder whether this "$\sqrt{n}$ barrier" is an artifact of *this specific reduction*, or whether *every possible reduction* to affine sources will encounter this same barrier. This raises the question,

Can affine extractors be used to extract from local sources with min-entropy $k < \sqrt{n}$?

One reason to believe this may be true is our result from the previous chapter (Lemma 7.1), which shows that affine extractors can handle *small-space sources* with min-entropy as low as $k = \operatorname{polylog} n$. Thus, one may hope a similar result can be established for *local sources*, which represent the "other half" of samplable sources. Unfortunately, we start this chapter by proving (in Section 8.3) that this is not the case.

Theorem 8.0 (Barrier result)**.** *It is not possible to extract from* 2*-local sources with min-entropy $k < \sqrt{n}$ by applying an affine extractor in a black-box manner.*[9]

Given this result, it becomes clear that if we wish to break this $\sqrt{n}$ barrier, new techniques will be needed.

Towards breaking the barrier with new techniques While Theorem 8.0 tells us that *black-box* affine extractors have reached their limit in local source extraction, it does not say the same about *white-box* affine extractors. That is, Theorem 8.0 just claims that an affine extractor Ext cannot be proven to extract from local sources of min-entropy $k < \sqrt{n}$ *using the extractor property alone*.[10] However, it is still totally possible that the *specific function* Ext used above actually *can* extract from local sources of min-entropy $k < \sqrt{n}$, but that this must be proven using a *completely different argument* (making no mention of affine sources). While this might sound ridiculous, it actually happens all the time in extractor theory.[11] Indeed, it seems that certain functions exhibit strong enough *pseudorandom* properties that they can extract from multiple unrelated models at once. So, where should we look for such functions? Even if we stay within the realm

[7] Technically, this can be reduced to $k \geq O(n^{1/2} \log n)$ by combining Theorem 8.4 with our new affine extractor (Theorem 6.1).

[8] To the best of our knowledge, the only other (well-studied) models with worse min-entropy requirements in the high-error regime are *polynomial sources*, *variety sources* (Part II), and other challenging *recognizable sources* (Part III).

[9] More formally, we show that there exists a 2-local source of min-entropy $k \geq \sqrt{n}$ that is *extremely far* from a convex combination of affine sources of min-entropy ≥ 6 (Theorem 8.5). This is fueled by a simple observation that the set of symmetric matrices in $\mathbb{F}_2^{n \times n}$ of rank at most one form a perfect *Sidon set* (Lemma 8.4).

[10] Here, by *extractor property*, we mean the guarantee that $\mathsf{Ext}(\mathbf{X})$ is close to uniform for any big enough affine source $\mathbf{X}$.

[11] As an example, consider the *inner product (mod 2)* function, which extracts from both independent sources (Chapter 3) *and* affine sources (Chapter 6), despite the fact that there is no black-box reduction in either direction.

of affine extractors, we saw in Chapter 6 that there are a *huge number* of different constructions. Which one of these should we try?

As it turns out, many of these constructions share a common theme: the use of *low-degree polynomials*. Indeed, while some of them *heavily depend* on the use of low-degree polynomials [Li11b], others are low-degree polynomials *themselves* [BSHR$^+$01, Bou07, BSK12]. In fact, Cohen and Tal [CT15] even proved that a *random* low-degree polynomial (over $\mathbb{F}_2$) is, with high probability, an excellent affine extractor. Looking at these constructions, a clear candidate emerges for a specific family of functions that might help us break the $\sqrt{n}$ barrier in local source extraction. Inspired by the work of Cohen and Tal, we ask the following.

Question 8.1. *Can (random) low-degree polynomials extract from local sources with min-entropy $k < \sqrt{n}$?*

This is the main question we seek to answer.

Why random low-degree polynomials?

While *explicit* extractors that break the $\sqrt{n}$ barrier for local sources are the end goal, answering Question 8.1 could very well be the next best thing. In fact, this question seems interesting *in its own right*, as it not only has consequences in the world of *extractors*, but also in *complexity theory* and *algebra*.

Extractors Within the world of *extractors*, a positive answer to Question 8.1 could be viewed as a *partial derandomization* or a *semi-explicit construction*. In particular, while a straightforward application of the probabilistic method can produce excellent local source extractors using 2^n random bits, a positive answer to Question 8.1 would provide a way to do so using just $n^{O(1)}$ random bits. Furthermore, a random construction based specifically on *low-degree polynomials* seems particularly conducive towards *further* derandomization, as these objects can naturally be interpreted as *hypergraphs*, which may be open to analysis via the tools and intuition of combinatorics. Finally, as we will soon see, the pursuit of Question 8.1 likely requires the development of new machinery that could be useful in future explicit constructions.

Complexity theory Moving to the world of *complexity theory*, Question 8.1 may eventually help us establish better *fine-grained lower bounds* (as advocated for in, e.g., [HHTT23]). In more detail, since this question asks whether *low-degree polynomials* can extract from local sources, it is effectively asking whether local-source extractors belong to a specific *low-complexity class* $\mathcal{C}$. Since extractors are known to witness the best-known lower bounds for several complexity classes $\mathcal{C}'$ [CS16b, LY22], a positive answer to Question 8.1 (combined with an efficient derandomization) could lead to more *fine-grained* lower bounds of the form $\mathcal{C}' \nsubseteq \mathcal{C}$ instead of $\mathcal{C}' \nsubseteq \mathsf{P}$ (where P denotes the class of poly-time algorithms). More generally, however, the existence of *low-complexity extractors* is a fascinating question in its own right, which has received a lot of interest over the years [Lu04, Vad04, Vio05, BG13, CT15, GVW15, CL18, CZ18, DY21, HIV22, ACG$^+$22].

Algebra Finally, from an *algebraic* point of view, Question 8.1 could provide deep insight into the *structure of low-degree polynomials*. Indeed, by asking whether low-degree polynomials extract from local sources, we are effectively asking about the structure of their solution sets. While there is an entire field of mathematics dedicated to this question (algebraic geometry), two beautiful works have considered it through the lens of *extractors* [BEHL12, CT15]. Most recently, Cohen and Tal [CT15] showed that a *random degree r polynomial*[12] extracts from *affine sources* with min-entropy $k \geq O(rn^{\frac{1}{r-1}})$, and that this is *tight*.[13] In our quest to answer Question 8.1, we seek out a *local source* analog of this result.

[12] Formally, a random degree r polynomial $p : \mathbb{F}_2^n \to \mathbb{F}_2$ includes each monomial of size at most r with probability $1/2$.

[13] In other words, they characterized the size of the largest *affine subspace* in the solution set of such polynomials.

8.2 Our results

In this chapter, we fully characterize the power of *low-degree polynomials* as extractors for *local sources*, answering Question 8.1 and providing a local source analog of Cohen and Tal [CT15]. Our characterization relies on *two new structural results*: one for local sources, and one for low-degree polynomials. These key new ingredients may be of independent interest, as the former can be used in future explicit constructions, while the latter provides interesting insight into the solutions of systems of low-degree polynomials.

The main result

As hinted at above, our main result gives a tight characterization of the power of low-degree polynomials as extractors for local sources. We prove the following.

Theorem 8.1 (Low-degree polynomials extract from local sources)**.** *For every $d \in \mathbb{N}$ there exist constants $C, c > 0$ such that for all $n \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/r}$-extractor for d-local sources of min-entropy*

$$k \geq Cr \cdot (n \log n)^{1/r},$$

except with probability at most $2^{-\binom{ck}{\leq r}}$. On the other hand, for every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists a d-local source of min-entropy

$$k \geq cr \cdot (n \log n)^{1/r}$$

on which it is constant.

More concisely, Theorem 8.1 shows that a random degree r polynomial extracts from local sources of min-entropy $k \geq O(r(n \log n)^{1/r})$, and that this is tight. In particular, this means that *degree 3 polynomials* are already enough to extract from min-entropy $k \geq O((n \log n)^{1/3})$, answering Question 8.1 and (non-explicitly) breaking the $\sqrt{n}$ min-entropy barrier of previous techniques (Theorem 8.0). Furthermore, given the known reduction from AC^0 sources to local sources [Vio14], it also follows that low-degree polynomials break the $\sqrt{n}$ barrier in extracting from this classical circuit family. In more detail, Theorem 8.1 immediately implies that degree r polynomials extract from AC^0 sources with min-entropy $k \geq O(n^{\frac{1}{r}+\delta})$ for any constant $\delta > 0$, which is tight up to the removal of δ (Theorem 8.15). Finally, it turns out that our main result has interesting consequences in the *complexity of sampling*, a rich new subfield of complexity theory. Here, we show (in Section 8.6) that our main theorem yields an alternative proof and generalization of a classical result of Viola [Vio16].

The upper bound

While Theorem 8.1 is stated for constant locality d, we actually prove stronger results that work for *superconstant* d. In particular, Theorem 8.1 immediately follows from two separate (and more technical) theorems we prove, which provide an *upper bound* and *lower bound* on the entropy required to extract from d-local sources using degree r polynomials (where d need not be constant). We start with the upper bound.

Theorem 8.2 (Theorem 8.1, upper bound)**.** *There exist constants $C, c > 0$ such that for all $n, d \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/(r2^d d^2)}$-extractor for d-local sources of min-entropy*

$$k \geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r},$$

except with probability at most $2^{-\binom{ck/(2^d d^2)}{\leq r}}$.

In order to prove this result, the key new ingredient we exploit is a new *structural result* for d-local sources. In particular, we show that such sources are close to a convex combination of sources coming from a more structured family, called *d-local non-oblivious bit-fixing (NOBF) sources*. Informally, a d-local NOBF source $\mathbf{X} \sim \{0,1\}^n$ of min-entropy k is simply a source that contains k uniform independent bits, with the remaining $n-k$ bits depending on up to d of these k "good" bits each. We prove the following.

Lemma 8.1 (A reduction from d-local sources to d-local NOBF sources)**.** *There exists a constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a d-local source of min-entropy k. Then $\mathbf{X}$ is $2^{-ck'}$-close to a convex combination of d-local NOBF sources of min-entropy k', where*

$$k' \geq \frac{ck}{2^d d^2}.$$

Previously, it was known that d-local sources can actually be reduced to 1-*local NOBF sources* [DW12, Vio14], but this reduction suffers a prohibitively large loss in min-entropy (Lemma 8.3). Indeed, it was this exact reduction that was responsible for the $\sqrt{n}$ min-entropy barrier that we lamented above. In Lemma 8.1, we show that by reducing to a slightly richer class of sources, we can save a huge amount of min-entropy, and circumvent this barrier. Indeed, by combining our reduction with a classical result on the bias of random low-degree polynomials [BEHL12], our upper bounds (Theorem 8.2) almost immediately follow.

The lower bound

We now turn towards our *lower bound* on the entropy required to extract from local sources using low-degree polynomials, which provides the second half of Theorem 8.1. We prove the following.

Theorem 8.3 (Theorem 8.1, lower bound)**.** *There exists a constant $c > 0$ such that for all $n, d \in \mathbb{N}$ and $2 \leq r \leq c \log n$ such that $d \leq n^{\frac{1}{r-1} - 2^{-10r}} / \log n$, the following holds. For every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists a d-local source of min-entropy*

$$k \geq cr \cdot (dn \log n)^{1/r}$$

on which it is constant.

In order to prove this result, we actually show that the lower bound holds for a *special class* of d-local sources known as *d-local affine sources*. A d-local affine source $\mathbf{X} \sim \mathbb{F}_2^n$ is simply an affine source whose underlying subspace admits a basis $v_1, \dots, v_k \in \mathbb{F}_2^n$ where each coordinate $i \in [n]$ is set to 1 in at most d of these vectors. For this special class of sources, our lower bound is actually completely tight (Theorem 8.9), and can be viewed as a "local" version of the classical result (Theorem 8.7) of Cohen and Tal [CT15]. Finally, in order to prove our lower bound, the key new ingredient we exploit is the following "low-weight" *Chevalley-Warning theorem*, which shows that any small system of low-degree polynomials admits a (nontrivial) solution of low Hamming weight.

Lemma 8.2 (A low-weight Chevalley-Warning theorem)**.** *Let $\{f_i : \mathbb{F}_2^n \to \mathbb{F}_2\}$ be a set of polynomials of linear degree*[14] *at most D and nonlinear degree*[15] *at most Δ, such that 0 is a common solution and $D + \Delta < n$. Then there is a common solution $x \neq 0$ of Hamming weight*

$$w \leq 8D / \log(n/D) + 8\Delta + 8.$$

[14] The *linear degree* is the sum of the degrees of the f_i's which have degree 1.

[15] The *nonlinear degree* is the sum of the degrees of the f_i's which have degree at least 2.

To help digest this result, we note that proving Theorem 8.3 requires showing that a system of low-degree polynomials admits a low-weight solution. If the degrees of these polynomials sum to D, then the classical *Chevalley-Warning theorem* [War36] guarantees the existence of a weight $w \leq O(D)$ solution. On the other hand, if all of the polynomials in the system are guaranteed to be *linear*, then the *Hamming bound* (from coding theory) can be applied to obtain the improvement $w \leq O(D/\log(n/D))$. Our *low-weight Chevalley-Warning theorem* (Lemma 8.2) shows that you can still get some of these improvements, even if some of the polynomials in the system are nonlinear. This is crucial in proving Theorem 8.3, and we hope it might find applications elsewhere.

Organization

We start by revisiting the power and limitations of *black-box* affine extractors in *local source extraction*, in Section 8.3. There, we recall the well-known *reduction from local sources to affine sources* of De and Watson [DW12] and Viola [Vio14], and we prove our barrier result (Theorem 8.0) showing this black-box reduction cannot be improved. In Section 8.4, we turn to consider *white-box* affine extractors. First, in Section 8.4.1, we motivate the use of *low-degree polynomials* as extractors, by recalling how they perform in other classical settings. Then, in Section 8.4.2, we turn to prove our *upper bounds* on the entropy required to extract from local sources using low-degree polynomials. We start with our reduction from d-local sources to d-local NOBF sources (Lemma 8.1), from which our upper bounds (Theorem 8.2) easily follow. After that, we turn to Section 8.4.3. There, we prove our low-weight Chevalley-Warning theorem (Lemma 8.2), and show that it yields our lower bounds (Theorem 8.3) without too much additional trouble. By combining our lower bounds and upper bounds, we immediately obtain our main theorem (Theorem 8.1). Finally, we demonstrate the impact of this theorem on NC^0 and AC^0 sources in Section 8.5, discuss applications in the *complexity of sampling* in Section 8.6, and conclude with some open problems in Section 8.7.

8.3 Do black-box affine extractors suffice?

As we saw in Chapter 6, affine extractors are powerful objects in extractor theory, capable of extracting in a black-box manner from a host of seemingly unrelated models. Thus, to begin our quest of constructing better extractors for *local sources*, it is only natural to revisit the capacity of affine extractors in this setting.

8.3.1 Entropy upper bounds: a reduction from d-local sources to 1-local NOBF sources

When local sources were first introduced [DW12, Vio14], it was observed that they cannot easily be broken into *independent chunks*, unlike many of the models in seedless extraction at that time. This was a key piece of evidence that local sources were a completely different beast than what had been dealt with before. Thus, it came as a big surprise that, nevertheless, local sources can be broken down into *a different* fundamental model from the field: *affine sources*. In their founding papers on local sources, De and Watson [DW12] and Viola [Vio14] showed that every $O(1)$-local source $\mathbf{X} \sim \{0,1\}^n$ of min-entropy k can be broken down into affine sources with min-entropy $\Omega(k^2/n)$. They proved the following.

Theorem 8.4 (Black-box affine extractors can extract from local sources [DW12, Vio14])**.** *For every $d \in \mathbb{N}$ there exists a constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a d-local source of min-entropy k. Then $\mathbf{X}$ is $2^{-k'}$-close to a convex combination of affine sources of min-entropy k', where*

$$k' \geq ck^2/n.$$

In fact, De, Watson, and Viola actually showed that local sources can be reduced to an even more basic model: local sources, themselves! They obtained the following fine-grained result, which shows that any d-local source can be turned into a 1-local NOBF source, by paying a relatively modest price in min-entropy.

Lemma 8.3 (A reduction from d-local sources to 1-local NOBF sources [DW12, Vio14])**.** *There exists a constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a d-local source of min-entropy k. Then $\mathbf{X}$ is $2^{-k'}$-close to a convex combination of 1-local NOBF sources of min-entropy k', where*

$$k' \geq \max\left\{\frac{ck}{(2^d d^2)\cdot(n/k)}, \frac{ck}{(d^3/\log d)\cdot(n/k)^2\log(n/k)}\right\}.$$

This reduction is extremely powerful, and used to construct all known extractors for local sources. However, staring at the parameters above, one might notice that a fundamental *barrier* is baked into Lemma 8.3. In particular, even for 2-local sources, the above reduction cannot possibly be used to extract from min-entropy below $k = \sqrt{n}$, since it may turn such a source into a 1-local source with no min-entropy at all. It is natural to ask if this is an artifact of the reduction, or whether this is truly a fundamental barrier in black-box affine extraction from local sources. Indeed, the case gets even more mysterious when one considers our reduction from small-space sources to 1-local sources in the previous chapter (Lemma 7.7), which exhibits no such barrier. Thus, one might think there is hope to further improve Lemma 8.3.

8.3.2 Entropy lower bounds: a big 2-local Sidon set from low-rank symmetric matrices

As it turns out, the reduction of De, Watson, and Viola is, unfortunately for us, almost completely tight. As our first small result of this chapter, we prove the following, which shows that black-box affine extractors *cannot* be used to extract from local sources with min-entropy below $k = \sqrt{n}$.

Theorem 8.5 (Black-box affine extractors cannot extract from local sources - Theorem 8.0, formal)**.** *There is a 2-local source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy $k \geq \sqrt{n}$ that is $(1/2)$-far from a convex combination of affine sources with min-entropy 6.*

In order to prove this result, we show that 2-local sources can perfectly sample a big *Sidon set*. A Sidon set $S \subseteq \mathbb{F}_2^n$ is a fundamental object from additive combinatorics, which the property that for any $u \neq v \in S$ and $x \neq y \in S$ with $\{u,v\} \neq \{x,y\}$, it holds that $u+v \neq x+y$. In other words, all pairs of elements in S that *can* sum to distinct elements *do* sum to distinct elements, and the set S *blows up* under addition. At a high-level, this means that a Sidon set has the opposite behavior of a subspace, which is *closed* under addition, and thus does not blow up at all. This is the key idea in proving Theorem 8.5. So without further ado, let us prove it. We start by presenting a simple "local-looking" object that is a Sidon set.

Lemma 8.4. *The set $\mathcal{M} := \{vv^T : v \in \mathbb{F}_2^n\}$ of symmetric matrices of rank at most one is a Sidon set.*

Proof. By definition, to show that $\mathcal{M}$ is a Sidon set, we must show that distinct pairs of elements sum to distinct values. In other words, we must show that for any $u \neq v \in \mathbb{F}_2^n$ and $x \neq y \in \mathbb{F}_2^n$,

$$uu^T + vv^T = xx^T + yy^T \implies \{u,v\} = \{x,y\}.$$

Towards this end, define the matrices $A := uu^T + vv^T$ and $B := xx^T + yy^T$. Let's now look at the matrix A. Let support$(u) := \{i \in [n] : u_i = 1\}$ denote the support of u and let support(v) denote the same quantity for v. Clearly support$(u) \neq$ support(v), since $u \neq v$. Using this observation, it can be verified that the column space of A is precisely $\mathsf{span}(u,v)$. For the same reason, the column space of B is $\mathsf{span}(x,y)$.

By the argument above, we know that $uu^T + vv^T = xx^T + yy^T \implies A = B \implies \mathsf{span}(u, v) = \mathsf{span}(x, y)$, since two identical matrices must clearly have the same column space. All that remains is to show that this further implies $\{u, v\} = \{x, y\}$. First, notice that this is easy when one of the elements in u, v, x, y is 0: In this case, suppose, without loss of generality, that $u = 0$. Then $\mathsf{span}(u, v) = \{0, v\}$. Now, in order to maintain the condition that $\mathsf{span}(u, v) = \mathsf{span}(x, y)$, it must clearly hold that $\{x, y\}$ has one element that is zero (so as to not contradict dimension), while the other element must be v (so that the spaces are equal). In other words, we have $\{u, v\} = \{0, v\} = \{x, y\}$, as desired.

We may now assume that none of the elements in u, v, x, y are zero. Since $\mathsf{span}(u, v) = \mathsf{span}(x, y)$ and none of the elements are zero, we know that $\{x, y\}$ can either be $\{u, v\}$, $\{u, u + v\}$, or $\{v, u + v\}$. Consider the case $\{x, y\} = \{u, u + v\}$. Since we know that $uu^T + vv^T = xx^T + yy^T$, this implies

$$uu^T + vv^T = uu^T + (u + v)(u + v)^T \implies vv^T = (u + v)(u + v)^T.$$

But since we know that $u \neq 0$, the vectors v and $u + v$ must be different, and thus the matrices vv^T and $(u + v)(u + v)^T$ must also be different. Thus we cannot have $\{x, y\} = \{u, u + v\}$. Similarly, we can show that $\{x, y\} = \{v, u + v\}$ is also impossible. The only possibility left is that $\{x, y\} = \{u, v\}$, as desired. □

Given the above, it is not hard to show that the set of symmetric matrices of rank at most one have tiny intersections with affine subspaces - in other words, they form a *subspace-evasive set*, which is a fundamental object from linear-algebraic pseudorandomness [DL12]. Indeed, if the intersection were big, then the Sidon property of the these matrices guarantees that the intersection blows up under addition, while the fact that the intersection belongs to an affine subspace means that there simply isn't enough room to grow.

Corollary 8.1. *Let $\mathcal{M} := \{vv^T : v \in \mathbb{F}_2^n\}$ be the set of symmetric matrices of rank at most one. Then for any nonempty affine subspace $S \subseteq \mathbb{F}_2^{n\times n}$,*

$$\frac{|\mathcal{M} \cap S|}{|S|} \leq \frac{4}{\sqrt{|S|}}$$

Proof. Let $\mathcal{M}' := \mathcal{M} \cap S$, and let S' be the smallest subspace containing S. Observe that S' has size at most $2|S|$, since adding the shift vector of S to its basis will always create a subspace containing S. Now, note that since $\mathcal{M}$ is a Sidon set (by Lemma 8.4), we also have that $\mathcal{M}'$ is a Sidon set. Thus, $\mathcal{M}'$ grows maximally under addition. At the same time, we also have $\mathcal{M}' \subseteq S \subseteq S'$, which means that the sumset[16] $\mathcal{M}' + \mathcal{M}'$ must be contained in S' (since S' is a subspace). Putting these observations together, we get

$$1 + \binom{|\mathcal{M}'|}{2} \leq |\mathcal{M}' + \mathcal{M}'| \leq |S'| \leq 2|S|,$$

which implies $|\mathcal{M}'|^2/4 \leq 2|S|$. The result follows. □

With these results in hand, it is now easy to obtain our barrier result, Theorem 8.5.

Proof of Theorem 8.5. Fix any $\ell \in \mathbb{N}$, and define $n := \ell^2$. Let $\mathbf{Y} \sim \{0, 1\}^\ell$ be a uniform random variable, and define $\mathbf{X} := \mathbf{Y}\mathbf{Y}^T$, where we identify $\{0, 1\}^\ell$ with the vector space $\mathbb{F}_2^\ell$ in the natural way. It is easy to verify that $\mathbf{X}$ is a 2-local source over $\{0, 1\}^n$ with min-entropy $k = \ell = \sqrt{n}$, since every bit in $\mathbf{X}$ is a product of two bits in $\mathbf{Y}$, and since every fixing of $\mathbf{Y}$ forces $\mathbf{X}$ to a different value. Now, let $\mathbf{S} \sim \{0, 1\}^n$

[16] Given a set A, the *sumset* $A + A$ is simply defined as the set of all sums, namely $\{a + a' : a \in A, a' \in A\}$.

be an arbitrary convex combination of affine sources $\{\mathbf{S}^{(i)}\}_i$, each with min-entropy at least t. If we let $\mathcal{M}$ denote the support of $\mathbf{X}$, then the definition of statistical distance and Corollary 8.1 yield the following:

$$\begin{aligned}|\mathbf{X}-\mathbf{S}| &\geq \Pr[\mathbf{X}\in\mathcal{M}]-\Pr[\mathbf{S}\in\mathcal{M}]\\ &\geq 1-\max_i \Pr[\mathbf{S}^{(i)}\in\mathcal{M}]\\ &= 1-\max_i \frac{|\mathcal{M}\cap\mathsf{support}(\mathbf{S}^{(i)})|}{|\mathsf{support}(\mathbf{S}^{(i)})|}\\ &\geq 1-\max_i \frac{4}{\sqrt{|\mathsf{support}(\mathbf{S}^{(i)})|}} = 1-\frac{4}{2^{t/2}}.\end{aligned}$$

Plugging in $t \geq 6$ yields the result. □

Thus, if we ever hope to extract from local sources with min-entropy beyond $\sqrt{n}$, we must look beyond the world of black-box affine extractors. But where exactly should we look? The answer is simple: towards the world of *white-box* affine extractors.

8.4 Do white-box affine extractors suffice?

While *black-box reductions* form the bread and butter of extractor theory, sometimes such reductions are simply impossible to obtain. In such cases, it is not uncommon to attempt to extract using a *white-box extractor*: namely, a specific function (or class of functions) that are known to extract from a host of different source classes. While surprising, it turns out that this heavy-handed approach can be surprisingly effective. Indeed, certain specific functions just happen to be good at extracting from a variety of different sources.[17] So, which specific function should we use to deal with our unruly local sources?

8.4.1 On the power of (random) low-degree polynomials

We turn to the family of (random) low-degree polynomials for help. The reason for this is simple: in a classical work of Cohen and Tal [CT15], it was shown that this family of functions are excellent extractors for affine sources. Thus, given that black-box affine extractors cannot extract from local sources beyond min-entropy $k=\sqrt{n}$ (Theorem 8.5), it is only natural to see if *white box* affine extractors can help.

The basics of random low-degree polynomials

Before we can dive into the above direction, let us review a few basic definitions and facts about random low-degree polynomials. First, recall that every function $f:\mathbb{F}_2^n\to\mathbb{F}_2$ has a unique representation as a multilinear $\mathbb{F}_2$-polynomial. This means there exist (unique) coefficients $\{c_S\}_{S\subseteq[n]}$, each from $\mathbb{F}_2$, so that

$$f(x)=\sum_{S\subseteq[n]} c_S x^S.$$

One consequence of this is the simple fact for *every* family of sources (from which extraction is possible), there is an polynomial (over $\mathbb{F}_2$) that extracts from that family. The interesting question, then, is whether there exists a *low-degree* polynomial that does the trick. Here, the *degree* of the polynomial f (from above)

[17] For example, *inner product (mod 2)* extracts from both independent sources (Chapter 3) and affine sources (Chapter 6).

is denoted $\deg(f)$ and defined as the largest S such that $c_S = 1$. If no such S exists, we simply say that $\deg(f) = -\infty$. Now, with all of these definitions in mind, we define a *random degree r polynomial* $f : \mathbb{F}_2^n \to \mathbb{F}_2$ to be one that sets each c_S to 1 with probability $1/2$ if $|S| \leq r$, and otherwise sets $c_S = 0$.

It will also sometimes be useful to think about *sets* of polynomials. Here, we let $\mathbb{F}_2[x_1, \ldots, x_n]$ denote the set of *all* (multilinear) polynomials over $\mathbb{F}_2$, and we let $\{f_i\} \subseteq \mathbb{F}_2[x_1, \ldots, x_n]$ refer to a *subset* (of unspecified size) of such polynomials. Given such a subset, we say that the *linear degree* of $\{f_i\}$ is the sum of the degrees of f_i's that have degree 1, while the *nonlinear degree* of $\{f_i\}$ is the sum of the degrees of the f_i's which have degree > 1. The *degree* of $\{f_i\}$, then, is simply the sum of the degree across all the f_i's. Finally, we say that x is a common solution to $\{f_i\}$ if $f_i(x) = 0$ for all i, and we say it is *nontrivial* if $x \neq 0$.

With this in mind, let us return to the task at hand: seeing if low-degree polynomials are good extractors.

Known extractors from random low-degree polynomials

As it turns out, there is an increasing amount of evidence that random low-degree polynomials are great extractors for a number of different source families. The first result in this direction was obtained by Ben-Eliezer, Hod, and Lovett [BEHL12], who showed that random low-degree polynomials have extremely low bias.[18] More formally, they proved the following.

Lemma 8.5 (Low-degree polynomials have low bias [BEHL12])**.** *There exists a constant $c > 0$ such that for all $n \in \mathbb{N}$ and $1 \leq r \leq 0.99n$, a random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ has*

$$\Pr_f\left[|\operatorname{bias}(f)| > 2^{-cn/r}\right] \leq 2^{-c\binom{n}{\leq r}}.$$

It may be a bit silly to think of it this way, but the above can actually be viewed as a result on the power of low-degree polynomials as extractors. In particular, it shows that they extract from the trivial source family $\mathcal{X} = \{\mathbf{U}_n\}$. Combining this with the classical extractor impossibility result (Fact 1.1), we get:

Theorem 8.6 (Low-degree polynomials extract from general sources [BEHL12])**.** *There exist constants $C, c > 0$ such that for all $n \in \mathbb{N}$ and $1 \leq r \leq 0.99n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/r}$-extractor for sources of min-entropy*

$$k \geq n,$$

except with probability at most $2^{-c\binom{k}{\leq r}}$. On the other hand, for every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists a source of min-entropy

$$k \geq n - 1$$

on which it is constant.

As it turns out, it is sometimes easy to bootstrap the above result to show that random low-degree polynomials extract from other families of sources. In particular, Cohen and Tal [CT15] made the observation that by doing so, one can show that random low-degree polynomials can extract from *affine sources* that have relatively low min-entropy. Their main result, however, is a lower bound showing the exact min-entropy at which such extractors fail. In total, they prove the following.

[18] Recall that the *bias* of a fixed function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is $\operatorname{bias}(f) := |\mathbb{E}_x[(-1)^{f(x)}]| = |\Pr_x[f(x) = 0] - \Pr_x[f(x) = 1]|$, while the *correlation* between two functions f, g is $\operatorname{corr}(f, g) := |\mathbb{E}_x[(-1)^{f(x)+g(x)}]| = |\Pr_x[f(x) = g(x)] - \Pr[f(x) \neq g(x)]|$.

Theorem 8.7 (Low-degree polynomials extract from affine sources [CT15])**.** *There exist constants $C, c > 0$ such that for all $n \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/r}$-extractor for affine sources of min-entropy*

$$k \geq Cr \cdot n^{1/(r-1)},$$

except with probability at most $2^{-c\binom{k}{\leq r}}$. On the other hand, for every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists an affine source of min-entropy

$$k \geq cr \cdot n^{1/(r-1)}$$

on which it is constant.

Given the above results, it is natural to ask from what other sources can random low-degree polynomials extract. As advertised in Section 8.2, we contribute a new result in this direction, further demonstrating the power of low-degree polynomials. The main contribution of this chapter is the following theorem.

Theorem 8.8 (Low-degree polynomials extract from local sources - Theorem 8.1, restated)**.** *For every $d \in \mathbb{N}$ there exist constants $C, c > 0$ such that for all $n \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/r}$-extractor for d-local sources of min-entropy*

$$k \geq Cr \cdot (n \log n)^{1/r},$$

except with probability at most $2^{-\binom{ck}{\leq r}}$. On the other hand, for every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists a d-local source of min-entropy

$$k \geq cr \cdot (n \log n)^{1/r}$$

on which it is constant.

Comparing all the theorems above, we see that random low-degree polynomials gradually become better extractors as we move from the family of *all sources* (Theorem 8.6) to the family of *affine sources* (Theorem 8.7) to the family of *local sources* (Theorem 8.8). However, upon further reflection, one might notice that the entropy requirement in the latter two results are eerily similar. Given this, a natural question is whether can prove a result that interpolates between them. As it turns out, we prove exactly such a result along the way to proving Theorem 8.8. More formally, we show the following.

Theorem 8.9 (Low-degree polynomials extract from local affine sources)**.** *There exist constants $C, c > 0$ such that for all $n \in \mathbb{N}$ and $2 \leq r \leq c \log n$ and $d \leq n^{\frac{1}{r-1} - 2^{-10r}} / \log n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/r}$-extractor for d-local affine sources of min-entropy*

$$k \geq Cr \cdot (dn \log n)^{1/r},$$

except with probability at most $2^{-c\binom{k}{\leq r}}$. On the other hand, for every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists a d-local affine source of min-entropy

$$k \geq cr \cdot (dn \log n)^{1/r}$$

on which it is constant.

In the remainder of this section, we focus on showing the above results, providing a formal proof for all of our main theorems discussed above and in Section 8.2. We start by proving *upper bounds* on the entropy required to extract from local sources with low-degree polynomials in Section 8.4.2, thereby obtaining Theorem 8.2. Then, in Section 8.4.3, we turn to prove our lower bounds on min-entropy, Theorem 8.3. Together, these immediately imply our main contribution of the chapter: a tight characterization of the power of low-degree polynomials as extractors for local sources (Theorem 8.1, stated as Theorem 8.8 above). Along the way, as promised, we prove our result that interpolates between affine and local sources (Theorem 8.9).

8.4.2 Entropy upper bounds: a reduction from d-local sources to d-local NOBF sources

To kick things off, we will prove our *upper bounds* on the entropy required to extract from d-local sources using random degree r polynomials (Theorem 8.10). In particular, we prove the following.

Theorem 8.10 (Entropy upper bounds - Theorem 8.2, restated)**.** *There exist constants $C, c > 0$ such that for all $n, d \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. A random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/(r2^d d^2)}$-extractor for d-local sources of min-entropy*

$$k \geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r},$$

except with probability at most $2^{-\binom{ck/(2^d d^2)}{\leq r}}$.

We start with a proof overview, before diving into the proof in full formality.

An overview of the proof

A natural first attempt at proving Theorem 8.10 would use a standard application of the probabilistic method, which looks something like the following. First, let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a random degree r polynomial, meaning that each monomial of size $\leq r$ is included in f with probability $1/2$. Then, let $\mathcal{X}$ be the family of d-local sources over $\{0,1\}^n$, each with min-entropy at least k. To prove that most of these polynomials are low-error extractors for $\mathcal{X}$, a standard application of the probabilistic method would suggest that we:

1. Prove that f is an extractor for a single $\mathbf{X} \in \mathcal{X}$ with extremely high probability.

2. Show that the family $\mathcal{X}$ does not contain too many sources.

3. Conclude, via the union bound, that f is an extractor for *every* $\mathbf{X} \in \mathcal{X}$ with high probability.

It is not too hard to complete Steps 2 and 3 in the above framework, but Step 1 turns out to be much more challenging. To see why, let us consider an arbitrary d-local source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy at least k. By definition of d-local source, there exists some $m \in \mathbb{N}$ and functions $g_1, \ldots, g_n : \{0,1\}^m \to \{0,1\}$ such that each g_i depends on just d of its inputs, and such that given a uniform $\mathbf{Y} \sim \{0,1\}^m$, we have

$$\mathbf{X} = (g_1(\mathbf{Y}), g_2(\mathbf{Y}), \ldots, g_n(\mathbf{Y})).$$

Now, we want to argue that a random degree r polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a low-error extractor for $\mathbf{X}$. To do so, consider the function $F : \{0,1\}^m \to \{0,1\}$ defined as

$$F(y_1, \ldots, y_m) = f\left(g_1(y_1, \ldots, y_m), \ldots, g_n(y_1, \ldots, y_m)\right).$$

Notice that by the definition of $\mathbf{X}$ and by Definition 1.1 of extractor, f is an extractor for $\mathbf{X}$ with error ε if

$$|\operatorname{bias}(F)| := \left| \Pr_{y \sim \mathbf{U}_m}[F(y) = 1] - \Pr_{y \sim \mathbf{U}_m}[F(y) = 0] \right| \leq 2\varepsilon.$$

Thus, to argue that a random degree r polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ is a low-error extractor for $\mathbf{X}$, it suffices to argue that the function F has low bias (with high probability over the selection of f).

Of course, the question now becomes: how can we ensure that F has low bias? We can start by noticing some properties of F. First, we know $F = f(g)$, where f is a random degree r polynomial and g is a fixed function where each output bit depends on $\leq d$ input bits. Thus, it is not hard to argue that F will have degree $\leq rd$. Furthermore, since f is random and g is fixed, one may hope to argue that F is a *uniformly random* polynomial of degree $\leq rd$: in this case we would be done, since we know that uniformly random low-degree polynomials have extremely low bias, with extremely high probability (Lemma 8.5).

Unfortunately, it is too much to hope that F is a uniformly random low-degree polynomial. Indeed, it is not hard to see that the distribution of F over degree $\leq rd$ polynomials depends heavily on the exact selection of g. Furthermore, for most selections of g, the random function F is *not* uniformly distributed over degree $\leq t$ polynomials for *any* t.

Thus, there is no obvious way to apply Lemma 8.5 in order to argue that F will have low bias. To proceed, it seems like we will somehow need to argue that the distribution of F over low-degree polynomials is guaranteed to have some specific *structure*, and then somehow argue that a random polynomial from any such structured distribution is guaranteed to have low-bias. Each of these steps seems quite challenging.

Reductions to the rescue. As it turns out, there is a simple trick we can use to greatly simplify the above approach. The key idea is to *reduce* local sources to a simpler class of sources. Given two familes $\mathcal{X}, \mathcal{Y}$ of distributions over $\{0,1\}^n$, we say that $\mathcal{X}$ *reduces* to $\mathcal{Y}$ if each $\mathbf{X} \in \mathcal{X}$ is (close to) a convex combination of $\mathbf{Y} \in \mathcal{Y}$.[19] Reductions are extremely useful, because of the following well-known fact (Fact 2.8): if $\mathcal{X}$ reduces to $\mathcal{Y}$, and $f : \{0,1\}^n \to \{0,1\}$ is an extractor for $\mathcal{Y}$, then f *is also an extractor for* $\mathcal{X}$.

Thus, in order to show that low-degree polynomials extract from d-local sources, a key new ingredient we use is a reduction from d-local sources to a simpler class of sources called *d-local non-oblivious bit-fixing (NOBF) sources*. Using the above discussion, it then suffices to show that low-degree polynomials extract from d-local NOBF sources. Thus, we proceed by:

1. Defining local NOBF sources, and showing how we can appropriately tailor our previous attempt at the probabilistic method so that it works for local NOBF sources.

2. Providing a new reduction from local sources to local NOBF sources.

Low-degree extractors for local NOBF sources. A *d-local NOBF source* $\mathbf{X} \sim \{0,1\}^n$ is a natural specialization of a d-local source where the entropic bits of the source must show up "in plain sight" somewhere in the source.[20] More formally, a d-local NOBF source with min-entropy k is a random variable $\mathbf{X} \sim \{0,1\}^n$ for which there exist functions $g_1, \ldots, g_n : \{0,1\}^k \to \{0,1\}$ such that the following holds:

[19] By this we mean that each $\mathbf{X} \in \mathcal{X}$ can be written in the form $\mathbf{X} = \sum_i p_i \mathbf{Y}_i$, where each $\mathbf{Y}_i \in \mathcal{Y}$, $\sum_i p_i = 1$, and $\mathbf{X}$ samples from $\mathbf{Y}_i$ with probability p_i.

[20] The relationship between local sources and local *NOBF* sources is not dissimilar to the relationship between error-correcting codes and *systematic* error-correcting codes.

each $g_i, i \in [n]$ depends on $\leq d$ input bits; for every $i \in [k]$ there is some $i' \in [n]$ such that $g_{i'}(y) = y_i$; and for uniform $\mathbf{Y} \sim \{0,1\}^k$ we have

$$\mathbf{X} = (g_1(\mathbf{Y}), g_2(\mathbf{Y}), \ldots, g_n(\mathbf{Y})).$$

In other words, some k "good" bits in $\mathbf{X}$ are uniform, and the remaining $n - k$ "bad" bits are d-local functions of the good bits.

We must now show that a random degree r polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ extracts from d-local NOBF sources with entropy k. As in our strawman application of the probabilistic method, consider an arbitary d-local NOBF source $\mathbf{X} = (g_1(\mathbf{Y}), \ldots, g_n(\mathbf{Y}))$ and let $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ be a uniformly random degree r polynomial. To show that f extracts from all d-local NOBF sources, recall that we just need to show that the function $F : \{0,1\}^k \to \{0,1\}$ defined as

$$F(y) := f\left(g_1(y), \ldots, g_n(y)\right)$$

has extremely low bias with extremely high probability. Furthermore, recall that if we can show that F itself is a uniformly random low-degree polynomial, then we know via Lemma 8.5 that this is true.

It is still too much to hope that F is a uniform low-degree polynomial, but F is now "close enough in structure" to one so that we can make this work. To see why, we can first assume without loss of generality (by definition of local NOBF source) that $g_1(y) = y_1, \ldots, g_k(y) = y_k$. Thus, we can define $c_S \sim \{0,1\}$ as an independent uniform bit (for each $S \subseteq [n]$ of size $\leq r$) and write

$$F(y) = \sum_{S \subseteq [k]:|S| \leq r} c_S \prod_{i \in S} g_i(y) + \sum_{S \subseteq [n]:|S| \leq r, S \not\subseteq [k]} c_S \prod_{i \in S} g_i(y) = A(y) + B(y),$$

where $A \in \mathbb{F}_2[y_1, \ldots, y_k]$ is a uniformly random polynomial of degree $\leq r$, and $B \in \mathbb{F}_2[y_1, \ldots, y_k]$ is a random polynomial whose selection of monomials is *not uniformly random*, but is nevertheless *independent* of the selections made by A.

Thus, to show that F has extremely low bias, it suffices to show that $A + B$ has extremely low bias. And to show that $A + B$ has extremely low bias, it suffices to show that $A + B'$ has low bias for any fixed polynomial B' induced by fixing the random monomials selected by B. As it turns out, this follows immediately from (a slight tweaking of the proof of) Lemma 8.5.

Thus, a random low-degree polynomial f extracts from the d-local NOBF source $\mathbf{X}$ with min-entropy k with very high probability. In other words, it fails to do so with some very small probability $\delta = \delta(k)$ which decreases rapidly as k grows. By applying the union bound, we get that f extracts from the entire family $\mathcal{X}$ of d-local NOBF sources, provided $\delta(k) \cdot |\mathcal{X}| \ll 1$. All that remains is to upper bound the size of $\mathcal{X}$, which can easily be done using the d-locality of the sources.

A reduction to local NOBF sources. Above, we saw that random low-degree polynomials extract from *local NOBF sources*. To complete the proof that they also extract from more general *local sources*, recall that we need to provide a reduction from local sources to local NOBF sources. In other words, we need to show that every d-local source with min-entropy k is (close to) a convex combination of d-local NOBF sources with min-entropy $k' \approx k$. This is the main key ingredient in our result that low-degree polynomials extract from local sources (Lemma 8.1).

Our reduction works as follows. First, pick an arbitrary d-local source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy k. Let k' be a parameter which is slightly smaller than k, which will be picked later. We start by arguing that $\mathbf{X}$ is (close to) a convex combination of d-local NOBF sources where there are k' good bits, but the good bits may be biased (but not constant).

Towards this end, recall that $\mathbf{X} = (g_1(\mathbf{Y}), \dots, g_n(\mathbf{Y}))$ for some d-local functions $g_1, \dots, g_n : \{0,1\}^m \to \{0,1\}$ and uniform $\mathbf{Y} \sim \{0,1\}^m$. The key idea is to consider the *largest possible set* $T \subseteq [n]$ of "good bits," i.e., such that $\{\mathbf{X}_i\}_{i \in T}$ are independent (and none are constants). Then, we let $T' \subseteq [m]$ be the bits of $\mathbf{Y}$ on which $\{\mathbf{X}_i\}_{i \in T}$ depend. The key observation is that *every* bit in $\mathbf{X}$ depends on *some* bit in $\{\mathbf{Y}_i\}_{i \in T'}$, by the maximality of T. Using this observation, there are *two possible cases*, over which we perform a *win-win analysis*.

First, it is possible that T contains $\geq k'$ bits. In this case, we consider fixing all bits $\{\mathbf{Y}_i\}_{i \notin T'}$. It is then not too hard to show that $\mathbf{X}$ becomes a source which contains $\geq k'$ good bits (which are mutually independent and not constants), and the remaining bad bits in $\mathbf{X}$ are deterministic d-local functions of these good bits.[21] Thus in this case, we get that $\mathbf{X}$ is a convex combination of NOBF sources of the desired type.

Second, it is possible that T contains $< k'$ bits. In this case, we consider fixing all bits $\{\mathbf{Y}_i\}_{i \in T'}$. But since all bits in $\mathbf{X}$ depend on *some* bit in this set, this fixing *decrements* the locality $d \to d-1$. And furthermore, since this fixes $|T'| \leq d|T| < dk'$ bits, the entropy only decreases from $k \to k - dk'$ by the entropy chain rule. We then recurse until we hit the first case, or until we hit $d = 1$. If we eventually hit the first case, we already know that $\mathbf{X}$ is a convex combination of NOBF sources of the desired type. On the other hand, it is easy to show that a 1-local source is actually a 1-local NOBF source! Thus we will always arrive at a (biased) d'-local NOBF source with $d' \leq d$, proving that $\mathbf{X}$ is always convex combination of NOBF sources of the desired type. Depending on when this recursion stops, we will arrive at an NOBF source with the number of good bits equal to at least

$$\min\{k', k - dk', k - d(d-1)k', \dots, k - k' \prod_{i \in [d]} i\} \geq \min\{k', k - d^2 k'\},$$

which is always at least k' provided $k' \leq \frac{k}{2d^2}$.

Thus, we see that any d-local source with min-entropy k can be written as a convex combination of d-local NOBF source with $\Omega(k/d^2)$ good bits, where the good bits are mutually independent (and nonconstant), but they may be heavily biased. So all that remains is to show that such biased d-local NOBF sources can be written as a convex combination of *unbiased* d-local NOBF sources (as they were originally defined). This step is not difficult, by applying a standard Chernoff bound. However, since each good bit depends on up to d bits, each such good bit $\mathbf{X}_i$ may have $|\operatorname{bias}(\mathbf{X}_i)| = 1 - 2 \cdot 2^{-d}$. As a result, we end up with $\Omega(\frac{k}{d^2 2^d})$ unbiased good bits.

This completes the reduction from local to local NOBF sources. Given our earlier proof sketch that low-degree polynomials extract from local NOBF sources, we have finally finished our proof sketch that low-degree polynomials also extract from local sources, as desired.

A formal presentation of the proof

Given the plan above, we are now ready to formally prove our entropy upper bounds for extracting from local sources using low-degree polynomials (Theorem 8.10). Recall that the key idea is to reduce d-local NOBF sources to d-local NOBF sources, and subsequently show that random low-degree polynomials extract from the latter. Towards this end, let us start with the reduction, which is the key ingredient in our proof.

Lemma 8.6 (A reduction from d-local sources to d-local NOBF sources - Lemma 8.1, restated)**.** *There exists a constant $c > 0$ such that the following holds. Let $\mathbf{X} \sim \{0,1\}^n$ be a d-local source of min-entropy k. Then*

[21] Technically, we need to fix a little more randomness to make this happen, but this can be done without much trouble by invoking some standard tricks from the extractor literature.

$\mathbf{X}$ *is* $2^{-ck'}$*-close to a convex combination of* d*-local NOBF sources of min-entropy* k'*, where*

$$k' \geq \frac{ck}{2^d d^2}.$$

We will first focus on proving this reduction, and then see that Theorem 8.10 follows without much additional work. Towards this end, in order to reduce d-local sources to d-local NOBF sources, we use an intermediate model called *biased* d*-local NOBF sources*. These are just like standard d-local NOBF sources, except the good bits can be a little biased (but they are still independent).

Definition 8.3. *A random variable* $\mathbf{X} \sim \{0,1\}^n$ *is a* (δ, k)-biased d-local NOBF source *if there exists a set* $S \subseteq [n]$ *of size* k *such that both of the following hold:*

- *The bits in* $\mathbf{X}_S$ *are mutually independent (but need not be identically distributed), and each* $\mathbf{X}_i, i \in S$ *has bias* $|\Pr[\mathbf{X}_i = 1] - \Pr[\mathbf{X}_i = 0]| \leq \delta$.
- *Every other bit* $\mathbf{X}_j, j \notin S$ *is a deterministic function of at most* d *bits in* $\mathbf{X}_S$.

Notice that this intermediate model generalizes d-local NOBF sources of min-entropy k, which are just $(0, k)$-biased d-local NOBF sources. Now, given this intermediate model, we prove our reduction from d-local to d-local NOBF sources (Lemma 8.6) by combining the following two lemmas.

Lemma 8.7 (A reduction from d-local sources to biased d-local NOBF sources)**.** *Let* $\mathbf{X} \sim \{0,1\}^n$ *be a* d*-local source with min-entropy* $\geq k$*. Then* $\mathbf{X}$ *is a convex combination of* (δ, k')*-biased* d*-local NOBF sources, where* $\delta \leq 1 - 2^{-d}$ *and* $k' \geq k/(2d^2)$.

Lemma 8.8 (A reduction from biased d-local NOBF sources to d-local NOBF sources)**.** *Let* $\mathbf{X} \sim \{0,1\}^n$ *be a* (δ, k)*-biased* d*-local NOBF source. Then* $\mathbf{X}$ *is* ε*-close to a convex combination of* $(0, k')$*-biased* d*-local NOBF sources, where* $k' \geq (1-\delta)k/4$ *and* $\varepsilon = 2^{-(1-\delta)k/4}$.

A straightforward calculation shows that by combining these two lemmas, Lemma 8.6 follows immediately. Thus, all that remains is to prove each lemma. We start with the former, which gives a reduction from d-local sources to biased d-local NOBF sources.

Proof of Lemma 8.7. Let $\mathbf{X} \sim \{0,1\}^n$ be a d-local source with min-entropy $\geq k$. We wish to show that $\mathbf{X}$ is a convex combination of (δ, k')-biased d-local NOBF sources, where $\delta \leq 1 - 2^{-d}$ and $k' \geq k/(2d^2)$.

The key observation that we will prove is that for any t, one of the following *must* hold: either

- $\mathbf{X}$ is a convex combination of (δ, t)-biased d-local NOBF sources, for $\delta \leq 1 - 2^{-d}$; or
- $\mathbf{X}$ is a convex combination of $(d-1)$-local sources with min-entropy $> k - td$.

Before we prove this key observation, let us see how we can use it to prove the desired result. First, recall that convex combinations "stack" in the following sense: if a source $\mathbf{X}$ is a convex combination of convex combinations of sources from a family $\mathcal{X}$, then $\mathbf{X}$ is just a convex combination of sources from $\mathcal{X}$. Thus, by repeatedly applying the key observation until either the first item becomes true or we arrive at a 1-local source (the "base case"), we see that $\mathbf{X}$ is a convex combination of sources $\{\mathbf{Z}_i\}$, where each $\mathbf{Z}_i$ is either:

- A (δ, t)-biased d-local NOBF source, for $\delta \leq 1 - 2^{-d}$; or
- A 1-local source with min-entropy $> k - t \cdot (d + (d-1) + \cdots + 2) = k - t \cdot (d^2 + d - 2)$.

However, it is clear from the definitions that a 1-local source with min-entropy k' is a 1-local NOBF source with min-entropy k'. Furthermore, it is easy to see that a 1-local NOBF source with min-entropy $\geq k'$ is a convex combination of 1-local NOBF sources with min-entropy exactly k', by fixing any additional random "good" bits. Thus, for any $t \leq k'$, we know that a 1-local source with min-entropy $\geq k'$ is a convex combination of (δ, t)-biased d-local NOBF sources, for $\delta \leq 1 - 2^{-d}$.

By the above discussion, we see that for any $t \leq k - t \cdot (d^2 + d - 2)$, $\mathbf{X}$ is a convex combination of (δ, t)-biased d-local NOBF sources, where $\delta \leq 1 - 2^{-d}$. Setting $t = \frac{k}{2d^2}$ yields the result.

Thus, all that remains is to prove the key observation stated at the beginning of the proof. Towards this end, let $\mathbf{X} \sim \{0,1\}^n$ be a d-local source with min-entropy $\geq k$. By definition of d-local source, there exists some ℓ and $f : \{0,1\}^\ell \to \{0,1\}^n$ such that $\mathbf{X} = f(\mathbf{Y})$ for uniform $\mathbf{Y} \sim \mathbf{U}_\ell$, such that each bit $\mathbf{X}_i$ is a deterministic function of at most d bits in $\mathbf{Y}$. In other words, there exist sets $S_1, \dots, S_n \subseteq [\ell]$ of size d and functions $f_1, \dots, f_n : \{0,1\}^d \to \{0,1\}^n$ such that

$$\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_n) = (f_1(\mathbf{Y}_{S_1}), f_2(\mathbf{Y}_{S_2}), \dots, f_n(\mathbf{Y}_{S_n})).$$

Now, let $T \subseteq [n]$ be any set of coordinates of *maximal size* such that:

- $H_\infty(\mathbf{X}_i) > 0$ for all $i \in T$; and
- $S_i \cap S_j = \emptyset$ for any distinct $i, j \in T$.

Suppose T has size τ. Without loss of generality, assume $T = [\tau]$. We conclude with two cases.

Case (i): $\tau < t$. In this case, we fix the random variable $\mathbf{Y}_{S_1}, \dots, \mathbf{Y}_{S_\tau}$. We know that with probability 1 over this fixing, all bits $\mathbf{X}_i, i \in [n]$ become deterministic functions of at most $d-1$ unfixed variables in $\mathbf{Y}$, by the maximality of T and its intersection property. In other words, $\mathbf{X}$ becomes a $(d-1)$-local source. Furthermore, by Lemma 2.2, we know that with probability 1 over this fixing, $\mathbf{X}$ loses $\sum_{i \in [\tau]} |S_i| = d\tau < dt$ bits of min-entropy. Thus in this case, $\mathbf{X}$ is a convex combination of $(d-1)$-local sources of min-entropy $> k - dt$.

Case (ii): $\tau \geq t$. In this case, define $\overline{S} := [n] - (\bigcup_{i \in [\tau]} S_i)$ and notice that $S_1, S_2, \dots, S_\tau, \overline{S}$ partition the coordinates of $\mathbf{Y}$. Next, define the random variables $\mathbf{Z}_i := \mathbf{Y}_{S_i}$ for each $i \in [\tau]$, and define $\overline{\mathbf{Z}} := \mathbf{Y}_{\overline{S}}$. Notice that $\mathbf{X}_i = f_i(\mathbf{Z}_i)$ for each $i \in [\tau]$. Furthermore, it is straightforward to verify that for all $j > \tau$, there exists a set $Q_j \subseteq [\tau]$ of size at most d and a deterministic function f'_j such that $\mathbf{X}_j = f'_j(\mathbf{Z}_{Q_j}, \overline{\mathbf{Z}})$. In other words, we can rewrite $\mathbf{X}$ as

$$\begin{aligned}\mathbf{X} &= (\mathbf{X}_1, \dots, \mathbf{X}_\tau, \mathbf{X}_{\tau+1}, \dots, \mathbf{X}_n) \\ &= (f_1(\mathbf{Z}_1), \dots, f_\tau(\mathbf{Z}_\tau), f'_{\tau+1}(\mathbf{Z}_{Q_{\tau+1}}, \overline{\mathbf{Z}}), \dots, f'_n(\mathbf{Z}_{Q_n}, \overline{\mathbf{Z}})).\end{aligned}$$

Now, for each $i \in [\tau]$, define $\mathbf{A}_i := f_i(\mathbf{Z}_i)$. Furthermore, it is straightforward to show that we can define a new random variable $\mathbf{B}$ independent of $\mathbf{Y}$, and for each $i \in [\tau]$ a deterministic function g_i such that $g_i(\mathbf{A}_i, \mathbf{B}) = \mathbf{Z}_i$ for all $i \in [\tau]$. Thus, for any subset $Q \subseteq [\tau]$ we have $\mathbf{Z}_Q = g'_Q(\mathbf{A}_Q, \mathbf{B})$ for some deterministic function g'_Q. And finally, for each $j > \tau$ there must be some deterministic function ψ_j such that

$$f'_j(\mathbf{Z}_{Q_j}, \overline{\mathbf{Z}}) = \psi_j(\mathbf{A}_{Q_j}, \mathbf{B}, \overline{\mathbf{Z}}).$$

Thus we can rewrite $\mathbf{X}$ as:

$$\mathbf{X} = (\mathbf{A}_1, \dots, \mathbf{A}_\tau, \psi_{\tau+1}(\mathbf{A}_{Q_{\tau+1}}, \mathbf{B}, \overline{\mathbf{Z}}), \dots, \psi_n(\mathbf{A}_{Q_n}, \mathbf{B}, \overline{\mathbf{Z}})).$$

Notice that the collection $\{\mathbf{A}_i\}_{i \in [\tau]}$ are mutually independent, and each has bias at most $1 - 2^{-d}$ since it is a non-constant deterministic function of d uniform bits. Thus no matter how $\mathbf{B}, \overline{\mathbf{Z}}$ are fixed, $\mathbf{X}$ becomes a (δ, t)-biased d-local NOBF source, for $\delta \leq 1 - 2^{-d}$. □

We now prove the second part of our reduction (Lemma 8.8), which shows that every biased d-local NOBF source is (close to) a convex combination of (unbiased) d-local NOBF sources.

Proof of Lemma 8.8. Let $\mathbf{X} \sim \{0,1\}^n$ be a (δ, k)-biased d-local NOBF source. We wish to show that $\mathbf{X}$ is ε-close to a convex combination of $(0, k')$-biased d-local NOBF sources, where $k' \geq (1-\delta)k/4$ and $\varepsilon = 2^{-(1-\delta)k/4}$.

Without loss of generality, assume that the first k bits in $\mathbf{X}$ are the "good bits": that is, there exist d-local functions $g_{k+1}, \dots, g_n : \{0,1\}^k \to \{0,1\}$ such that

$$\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_k, g_{k+1}(\mathbf{X}_1, \dots, \mathbf{X}_k), \dots, g_n(\mathbf{X}_1, \dots, \mathbf{X}_k),$$

where each $\mathbf{X}_i$ is independent and has bias at most δ. To remove the bias from this source, the key idea will be to simulate each $\mathbf{X}_i$ by two independent coins: one which is biased, and one which is not.

In more detail, for every $i \in [k]$ we construct a pair of independent random variables $\mathbf{B}_i, \mathbf{A}_i \sim \{0,1\}$ as follows. First, let $\gamma_i \in \{0,1\}$ be the value favored by $\mathbf{X}_i$, breaking ties arbitrarily. Then, define $p_i := \Pr[\mathbf{X}_i = \gamma_i]$, and notice that $\frac{1}{2} \leq p_i \leq \frac{1+\delta}{2}$, where the lower bound holds because $\mathbf{X}_i$ favors γ_i over $1 - \gamma_i$, and the upper bound holds because the good bits have bias at most δ. Next, define $\mathbf{B}_i$ such that

$$\Pr[\mathbf{B}_i = 0] = 2p_i - 1,$$
$$\Pr[\mathbf{B}_i = 1] = 2 - 2p_i.$$

Finally, let $\mathbf{A}_i \sim \{0,1\}$ simply be a uniform bit, and define the function $h_i : \{0,1\} \times \{0,1\} \to \{0,1\}$ as

$$h_i(b, a) := \begin{cases} \gamma_i & \textbf{if } b = 0, \\ a & \textbf{if } b = 1. \end{cases}$$

Given these definitions, it is straightforward to verify that $\mathbf{X}_i$ has the same distribution as $\mathbf{H}_i := h_i(\mathbf{B}_i, \mathbf{A}_i)$, and thus we may rewrite $\mathbf{X}$ as

$$\mathbf{X} = (\mathbf{H}_1, \dots, \mathbf{H}_k, g_{k+1}(\mathbf{H}_1, \dots, \mathbf{H}_k), \dots, g_n(\mathbf{H}_1, \dots, \mathbf{H}_k)).$$

Now, define the random variable $\mathbf{B} = (\mathbf{B}_1, \dots, \mathbf{B}_k)$. Given the above description of $\mathbf{X}$, it is not too difficult to see that for any $b \in \{0,1\}^k$, the conditional distribution $(\mathbf{X} \mid \mathbf{B} = b)$ is a d-local (unbiased) NOBF source, which has min-entropy equal to the Hamming weight of b. Thus, we may write $\mathbf{X}$ as the convex combination

$$\mathbf{X} = \sum_{b \in \{0,1\}^k} \Pr[\mathbf{B} = b] \cdot (\mathbf{X} \mid \mathbf{B} = b).$$

This means that if $\mathbf{B}$ has Hamming weight $\geq k'$ with probability $\geq 1 - \varepsilon$, then $\mathbf{X}$ is ε-close to a convex combination of d-local NOBF sources with min-entropy $\geq k'$. Such a claim will follow almost immediately from a Chernoff bound.

In more detail, define a random variable $\mathbf{Z} := \sum_{i \in [k]} \mathbf{B}_i$ and notice that the value of $\mathbf{Z}$ is exactly the Hamming weight of $\mathbf{B}$. Furthermore, recall that $\mathbf{B}_i = 1$ with probability $2 - 2p_i$ for some $p_i \in [\frac{1}{2}, \frac{1+\delta}{2}]$. Thus

$$\mu := \mathbb{E}[\mathbf{Z}] = \sum_{i \in [k]} \mathbb{E}[\mathbf{B}_i] \geq k \cdot (2 - 2 \cdot ((1+\delta)/2)) = (1-\delta)k.$$

Thus, by a standard Chernoff bound on the lower tail, we get that

$$\Pr[\mathbf{Z} \leq \mu/4] \leq e^{-\mu \cdot (3/4)^2/2} \leq 2^{-\mu/4},$$

which means that $\mathbf{B}$ has Hamming weight $\geq \mu/4$ with probability $\geq 1 - 2^{-\mu/4}$. Thus, $\mathbf{X}$ is $2^{-\mu/4}$-close to a convex combination of d-local NOBF sources with min-entropy $\geq \mu/4$, as desired. □

Given our reduction from d-local sources to d-local NOBF sources, not much work remains to show that a random low-degree polynomial extracts from the former (and thereby prove Theorem 8.10). The final missing ingredient we need is the following generalization of Lemma 8.5, which shows that a random degree r polynomial not only has low bias, but has low *correlation* with any (single) fixed function.

Lemma 8.9 (Implicit in [BEHL12]). *There exists a constant $c > 0$ such that for all $n \in \mathbb{N}$ and $1 \leq r \leq c \log n$, and any fixed function $g : \mathbb{F}_2^n \to \mathbb{F}_2$, a random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ has*

$$\Pr_f[|\operatorname{corr}(f, g)| > 2^{-cn/r}] \leq 2^{-c\binom{n}{\leq r}}$$

By combining our reduction (Lemma 8.6) with the above correlation bounds (Lemma 8.9), we can finally show that a random low-degree polynomial extracts from local sources (Theorem 8.10).

Proof of Theorem 8.10. We aim to show that a random degree r polynomial extracts from d-local NOBF sources, and will apply our reduction (Lemma 8.6) at the end, to show the same is true for d-local sources. We use the probabilistic method to establish the former. Towards this end, let $\mathbf{X} \sim \{0,1\}^n$ be an arbitrary d-local NOBF source of min-entropy k, and let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be a random degree r polynomial. Assume (without loss of generality) that $\mathbf{X}_1, \dots, \mathbf{X}_k$ are the "good bits" in the NOBF source.

Now, let f' be the (sum of the) monomials in f that do not use any variables outside $\mathbf{X}_1, \dots, \mathbf{X}_k$, and let g be the (sum of the) monomials in f that use at least one variable outside $\mathbf{X}_1, \dots, \mathbf{X}_k$. This gives us $f(\mathbf{X}) = f'(\mathbf{X}) + g(\mathbf{X})$, where f' is a random degree r polynomial over $\mathbf{X}_1, \dots, \mathbf{X}_k$, and g is an (independent) random polynomial over $\mathbf{X}_1, \dots, \mathbf{X}_n$. By definition of NOBF source, each of the bits $\mathbf{X}_{k+1}, \dots, \mathbf{X}_n$ is a deterministic function of the good bits $\mathbf{X}_1, \dots, \mathbf{X}_k$, and thus we have

$$f(\mathbf{X}) = f'(\mathbf{X}_1, \dots, \mathbf{X}_k) + g'(\mathbf{X}_1, \dots, \mathbf{X}_k),$$

where f' is a random degree r polynomial over $\mathbf{X}_1, \dots, \mathbf{X}_k$ and g' is an independent (but not necessarily uniform) random polynomial over the same set of variables. Now, since $\mathbf{X}_1, \dots, \mathbf{X}_k$ are independent uniform bits, we have that f is an extractor with error

$$\varepsilon = \left| \Pr_{x \sim \mathbf{U}_k}[f(x) = 1] - \frac{1}{2} \right| = \left| \Pr_{x \sim \mathbf{U}_k}[f'(x) \neq g'(x)] - \frac{1}{2} \right| = \frac{1}{2} \operatorname{corr}(f', g').$$

Thus

$$\begin{aligned}\Pr_f[f \text{ is } \textit{not} \text{ an extractor for } \mathbf{X} \text{ with error } \varepsilon] &= \Pr_{f',g'}[\operatorname{corr}(f', g') > 2\varepsilon] \\ &\leq \Pr_{f'}[\operatorname{corr}(f', g^*) > 2\varepsilon],\end{aligned}$$

where $g^* = \operatorname{argmax}_{g'} \Pr_{f'}[\operatorname{corr}(f', g') > 2\varepsilon]$. Thus, $g^* : \mathbb{F}_2^k \to \mathbb{F}_2$ is an arbitrary fixed function and $f' : \mathbb{F}_2^k \to \mathbb{F}_2$ is a random degree r polynomial. As such, the correlation bounds from Lemma 8.9 yield

$$\begin{aligned}\Pr_f[f \text{ is } \textit{not} \text{ an extractor for } \mathbf{X} \text{ with error } 2^{-ck/r}] &\leq \Pr_{f'}[\operatorname{corr}(f', g^*) > 2 \cdot 2^{-ck/r}] \\ &\leq \Pr_{f'}[\operatorname{corr}(f', g^*) > 2^{-c'k/r}] \\ &\leq 2^{-c\binom{k}{\leq r}},\end{aligned}$$

for some small constants $c, c' > 0$. Thus, we have established that a random degree r polynomial f is a $2^{-ck/r}$-extractor for a single d-local NOBF source with min-entropy k, except with probability at most $2^{-c\binom{k}{\le r}}$. Now, notice that if we let $\mathcal{X}$ be the family of all d-local NOBF sources with min-entropy k, a standard counting argument yields

$$|\mathcal{X}| \le \binom{n}{k} \cdot \left(\binom{k}{d} \cdot 2^{2^d}\right)^{n-k}.$$

To show that f is an extractor for this entire family at the same time, we just need $|\mathcal{X}| \cdot 2^{-c\binom{k}{\le r}} \le 2^{-0.5c\binom{k}{\le r}}$, or rather $|\mathcal{X}| \le 2^{0.5c\binom{k}{\le r}}$. It is straightforward to verify that this is true if

$$k \ge Cr \cdot (2^d n \log n)^{1/r}$$

for a big enough constant C. Thus, we get that a random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is a $2^{-ck/r}$-extractor for the family of d-local NOBF sources of min-entropy $k \ge Cr \cdot (2^d n \log n)^{1/r}$, except with probability at most $2^{-0.5c\binom{k}{\le r}}$. Thus, using our reduction (Lemma 8.6), we get that f is a $2^{-c'k/(r2^dd^2)}$-extractor for the family of d-local sources of min-entropy

$$k \ge C'r \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r},$$

except with probability at most $2^{-c'\binom{c'k/(2^d d^2)}{\le r}}$ for some constant $c' > 0$, as desired. □

8.4.3 Entropy lower bounds: a low-weight Chevalley-Warning theorem

We now move towards proving our *lower bounds* on the entropy required to extract from d-local sources using random degree r polynomials (Theorem 8.10). In particular, we prove the following.

Theorem 8.11 (Entropy lower bounds - Theorem 8.3, restated)**.** *There exists a constant $c > 0$ such that for all $n, d \in \mathbb{N}$ and $2 \le r \le c \log n$ such that $d \le n^{\frac{1}{r-1} - 2^{-10r}} / \log n$, the following holds. For every degree r polynomial $g : \mathbb{F}_2^n \to \mathbb{F}_2$ there exists a d-local source of min-entropy*

$$k \ge cr \cdot (dn \log n)^{1/r}$$

on which it is constant.

As before, we shall start with a proof overview, before providing the proof in its full formality.

An overview of the proof

In order to prove Theorem 8.11, we must show that for every degree $\le r$ polynomial $f \in \mathbb{F}_2[x_1, \dots, x_n]$, we can find a d-local source $\mathbf{X} \sim \{0,1\}^n$ with relatively high min-entropy k on which f is constant. In other words, we show that in order to disperse (and thus extract) from d-local sources, they must have min-entropy exceeding this value k. Towards this end, we actually prove a slightly stronger result: we show that we can find a d-local source $\mathbf{X} \sim \{0,1\}^n$ with the above parameters such that it is also *affine*.

Our starting point is the tight result of Cohen and Tal (Theorem 8.7), which shows that any degree $\le r$ polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ admits a subspace $V \subseteq \mathbb{F}_2^n$ of dimension $\Omega(rn^{1/(r-1)})$ on which it is constant. Here, we obtain a (tight) *local version* of their result, and show that any degree $\le r$ polynomial f admits a

d-local subspace $X \subseteq \mathbb{F}_2^n$ of dimension $k = \Omega(r(dn \log n)^{1/r})$ on which it is constant. Here, we say that V is *d-local* if V has a basis $v_1, \ldots, v_k \in \mathbb{F}_2^n$ such that for any index $i \in [n]$, at most d of these basis vectors equal 1 at this index. It is straightforward to verify that the uniform distribution $\mathbf{X}$ over V is a d-local source with min-entropy k, so we focus now on proving the existence of such a V.

At a high level, the proof of Cohen and Tal proceeds by iteratively growing a subspace V on which f is constant. At each phase, they define a set $A \subseteq \mathbb{F}_2^n$ such that f is constant over $\mathsf{span}(V, x)$ for every $x \in A$. They note that if $|A|$ has size $> 2^{\mathsf{dim}(V)}$, then of course there is some $x \in A \setminus V$ and furthermore we already know that f is constant on $\mathsf{span}(V, x)$. Thus, they can grow their monochromatic subspace by one dimension.

In order to get a lower bound on $|A|$, they note that this set can be defined as the common solutions to a small collection of low-degree polynomials. A classical result known as the *Chevalley-Warning theorem* (Theorem 8.12) then shows that $|A| \geq 2^{n-t}$, where t is the sum of degrees across the collection of polynomials. To complete their proof, they grow their subspace V until they are no longer able to show $|A| > 2^{\mathsf{dim}(V)}$.

In our lower bound, we show that f is monochromatic on a *d-local subspace*. To prove this, we start with the same approach as Cohen and Tal. However, at each phase, we add extra constraints to A which guarantee the following: if we take any $x \in A$ and add it to our current subspace V (with basis, say, $v_1, \ldots, v_{|\dim(V)|}$), then the location of the 1s appearing in vectors $x, v_1, \ldots, v_{|\dim(V)|}$ satisfy the d-locality constraint defined above. Again, as long as A is large enough, we can find some $x \in A$ that grows the dimension of our d-local subspace.

In order to ensure that A remains large for as many iterations as possible, we would like to minimize the impact of the new "locality" constraints that we have added to A. Given the description of these constraints above, we observe that these constraints are minimized if we grow V by carefully selecting vectors that have the lowest possible Hamming weight. However, we now need an upper bound on the Hamming weight of the lightest (nontrivial) common solution to a system of polynomial equations. Thus, our key new ingredient will be a result of this type, which we call a *low-weight Chevalley-Warning theorem.*

A low-weight Chevalley-Warning theorem. Above, we saw how the classical Chevalley-Warning theorem is critical in lower bounding the size of A, thereby showing that there is some (nontrivial) vector $v \in A$ by which we can grow our monochromatic subspace. Now, we need an additional guarantee that there is such a $v \in A$ that also has low Hamming weight. We prove such a result, and call it a *low-weight Chevalley-Warning theorem*. Our theorem roughly says the following. Given a collection $\{f_i\}$ of polynomials that have cumulative degree D (and a common solution 0), if most of these polynomials have degree ≤ 1 then they admit a nontrivial solution of Hamming weight at most

$$w = O(D/\log(n/D)).$$

In order to prove our result, our key observation is that for any large enough subset A of common solutions to $\{f_i\}$, it holds that $A + A$ also contains a (nontrivial) common solution to $\{f_i\}$. As it turns out, this observation follows quite readily from the CLP lemma [CLP17] - a result which was instrumental in the recent resolution of the cap set conjecture. Furthermore, given the above observation, we obtain an elementary proof of our low-weight Chevalley-Warning theorem, as follows. First, we let Q denote the set of common solutions to our collection of polynomials, and then:

1. We assume for contradiction that there is no nontrivial solution $q \in Q$ of weight $\leq w$.

2. We use our key observation to conclude that every Hamming ball of radius $w/2$ cannot have too many elements of Q in it, which implies that Q is a list-decodable code with small list size at radius $w/2$.

3. Using the list-decoding properties of Q, we use the Hamming bound (for list-decodable codes) to get an *upper bound* on its size.

4. Using the fact that Q holds the common solutions to a small set of low-degree polynomials, we use the classical Chevalley-Warning theorem to get a *lower bound* on its size.

5. We observe that the lower bound is greater than the upper bound, which yields a contradiction.

Equipped with our low-weight Chevalley-Warning theorem, our entropy lower bound for low-degree extraction from d-local affine spaces follows immediately via the proof sketch described above.

A formal presentation of the proof

Now that we have a plan for how we will prove our entropy lower bounds (Theorem 8.11), let us proceed with its formal proof. Recall that in order to prove our lower bounds, we obtain a new type of *Chevalley-Warning theorem*, which we view as our main new ingredient (Lemma 8.2). The Chevalley-Warning theorem is a classical result from number theory, which guarantees that a small set of low-degree polynomials admits a nontrivial common solution. More formally, it states the following.

Theorem 8.12 (Chevalley-Warning theorem [War36])**.** *Let $\{f_i : \mathbb{F}_2^n \to \mathbb{F}_2\}$ be a set of polynomials with degree at most D such that 0 is a common solution. Then there are at least 2^{n-D} common solutions to $\{f_i\}$. In particular, if $D < n$, then there must be a nontrivial common solution.*

To start things off, we prove a "low-weight" version of this theorem, which will be instrumental in our proof of Theorem 8.11. In more detail, we believe it is natural to ask not only if $\{f_i\}$ contains a nontrivial common solution, but if $\{f_i\}$ contains a nontrivial common solution of *low Hamming weight*.

In the case where all the f_i are linear, this is a question about the distance-dimension tradeoff of linear codes. Here, this question is answered by the *Hamming bound* (Theorem 2.2), which says that there must be a nontrivial common solution of weight $w \leq O(D/\log(n/D))$.[22] On the other hand, for general collections $\{f_i\}$ that may have nonlinear polynomials, it is straightforward to use classical Chevalley-Warning (Theorem 8.12) to show that there is always a nontrivial common solution of weight $w \leq D + 1$,[23] and one can show that this is tight in general.[24]

Thus, the upper bound on Hamming weight is much better when we know that all the polynomials in $\{f_i\}$ are linear. This begs the question: if we know that *most* of the polynomials in $\{f_i\}$ are linear, can we get an upper bound on Hamming weight that is almost as strong as in the purely linear case? And, more generally, can we get a granular bound that takes into account exactly how many f_i are linear and nonlinear?

Our *low-weight Chevalley-Warning theorem*, which we state next, provides a result of exactly this form.

Theorem 8.13 (Low-weight Chevalley-Warning theorem)**.** *Let $\{f_i\} \subseteq \mathbb{F}_2[x_1, \dots, x_n]$ be a set of polynomials with linear degree at most D and nonlinear degree at most Δ, such that 0 is a common solution and $D + \Delta < n$. If all nontrivial common solutions to $\{f_i\}$ have Hamming weight $> w$, then*

$$\binom{n}{\leq \lfloor w/2 \rfloor} \leq 2^{D+\Delta+1} \cdot \binom{n}{\leq \lfloor \Delta/2 \rfloor}.$$

As we will see, a relatively straightforward (but tedious) calculation yields the following corollary.

[22] More formally, it says that if all nontrivial solutions have weight $> w$, then it must hold that $\binom{n}{\leq \lfloor w/2 \rfloor} \leq 2^D$.

[23] Given a collection $\{f_i\} \subseteq \mathbb{F}_2[x_1, \dots, x_n]$ of degree D, apply Theorem 8.12 to the collection $\{f_i\} \cup \{x_1, x_2, \dots, x_{n-(D+1)}\}$.

[24] Consider the singleton set $\{f\} \subseteq \mathbb{F}_2[x_1, \dots, x_n]$, where $f(x) = \sum_{\emptyset \subsetneq S \subseteq [n] : |S| \leq D} x^S$.

Corollary 8.2 (Lemma 8.2, restated)**.** *Let $\{f_i : \mathbb{F}_2^n \to \mathbb{F}_2\}$ be a set of polynomials with linear degree at most D and nonlinear degree at most Δ such that 0 is a common solution and $D + \Delta < n$. Then there is a nontrivial common solution with Hamming weight*

$$w \le 8\Delta + 8D/\log(n/D) + 8.$$[25]

Notice that the weight upper bound in Corollary 8.2 tightly interpolates (up to constant factors) between the linear and nonlinear cases discussed above. We will also see that it is not difficult to extend Corollary 8.2 to obtain the following, which is the result that we will actually end up using in our proof of our Theorem 8.9.

Corollary 8.3. *Let $\{f_i : \mathbb{F}_2^n \to \mathbb{F}_2\}$ be a set of polynomials with linear degree at most D and nonlinear degree at most Δ such that 0 is a common solution. Then for any $S \subseteq [n]$ such that $D + \Delta < |S|$, there is a nontrivial common solution supported on S with Hamming weight*

$$w \le 8\Delta + 8D/\log(|S|/D) + 8.$$

We start by proving Theorem 8.13 and Corollaries 8.2 and 8.3. The proof of Theorem 8.13 is more interesting, and presented first. The proofs of Corollaries 8.2 and 8.3 are less interesting and somewhat tedious, and presented after. Once those proofs are complete, we will finally turn to prove the lower bounds in Theorem 8.11. We encourage the reader to skip the proofs of the corollaries, but provide them for completeness.

A proof of our low-weight Chevalley-Warning theorem

We now prove our low-weight Chevalley-Warning theorem (Theorem 8.13). The key ingredient we need is the following lemma, which says that for any collection of polynomials $\{f_i\}$ and any big enough set of common solutions A, it holds that $A + A$ contains a nontrivial common solution.

Lemma 8.10. *Let $\{f_i\} \subseteq \mathbb{F}_2[x_1, \ldots, x_n]$ be a set of polynomials with nonlinear degree at most Δ such that 0 is a common solution. Then for any set $A \subseteq \mathbb{F}_2^n$ of common solutions of size*

$$|A| > 2\binom{n}{\le \lfloor \Delta/2 \rfloor},$$

it holds that $A + A$ contains a nontrivial common solution.

Before proving this result, we show how it can be combined with the Hamming bound and the classical Chevalley-Warning theorem to get Theorem 8.13, our low-weight Chevalley-Warning theorem.

Proof of Theorem 8.13. Let $Q \subseteq \mathbb{F}_2^n$ be the set of common solutions to $\{f_i\}$, and suppose that all $v \in Q - \{0\}$ have Hamming weight $> w$. Then for any ball $\mathcal{B}$ of radius $\lfloor w/2 \rfloor$, it must hold that

$$|Q \cap \mathcal{B}| \le 2\binom{n}{\le \lfloor \Delta/2 \rfloor},$$

since otherwise Lemma 8.10 (combined with the triangle inequality) implies that that there is a nontrivial solution of weight at most $2\lfloor w/2 \rfloor \le w$. Thus, we see that Q is a (ρ, L)-list-decodable code (Definition 2.9),

[25] We define the expression $8D/\log(n/D)$ to be 0 if $D = 0$.

where

$$\rho = \lfloor w/2 \rfloor / n,$$
$$L = 2\binom{n}{\le \lfloor \Delta/2 \rfloor}.$$

Now, the Hamming bound (Theorem 2.2) implies that $|Q| \le 2^n L / \binom{n}{\le \rho n}$, whereas the Chevalley-Warning theorem (Theorem 8.12) implies that $2^{n-(D+\Delta)} \le |Q|$. Combining these inequalities yields

$$2^{n-(D+\Delta)} \le \frac{2^n L}{\binom{n}{\le \rho n}} \le \frac{2^n \cdot 2\binom{n}{\le \lfloor \Delta/2 \rfloor}}{\binom{n}{\le \lfloor w/2 \rfloor}},$$

which immediately implies the result. □

All that remains is to prove our key ingredient, Lemma 8.10. As it turns out, it follows quite readily from the following well-known *CLP Lemma* [CLP17], which was instrumental in the recent resolution of the cap set conjecture [EG17].

Lemma 8.11 (CLP Lemma [CLP17])**.** *Let $f \in \mathbb{F}_2[x_1, \dots, x_n]$ be a polynomial of degree at most r, and let M denote the $2^n \times 2^n$ matrix with entries $M_{x,y} = f(x+y)$ for $x, y \in \mathbb{F}_2^n$. Then*

$$\mathsf{rank}(M) \le 2\binom{n}{\le \lfloor r/2 \rfloor}.$$

Given the CLP Lemma, we are ready to prove Lemma 8.10, which will conclude our proof of Theorem 8.13.

Proof of Lemma 8.10. First, let $\{g_i\} \subseteq \{f_i\}$ be the set of polynomials in $\{f_i\}$ that have degree > 1. Notice that if $A + A$ contains a nontrivial common solution to the system $\{g_i\}$, then it also contains a nontrivial common solution to $\{f_i\}$: this follows from the linearity of the polynomials of $\{f_i\} - \{g_i\}$ and the fact that every $a \in A$ is a common solution (by definition of A). Thus, it suffices to show the result for the set $\{g_i\}$.

Next, consider the polynomial $g \in \mathbb{F}_2[x_1, \dots, x_n]$ defined as

$$g(x) := \prod_i (1 + g_i(x)).$$

It is straightforward to verify that g has degree at most Δ, and that $g(x) = 1$ if and only if x is a common solution to $\{g_i\}$. Now, suppose for contradiction that $A + A$ contains no nontrivial common solution to $\{g_i\}$: that is, for every distinct $x, y \in A$ it holds that $g(x+y) = 0$. Then, consider the $2^n \times 2^n$ matrix M with entries $M_{x,y} = g(x+y)$ for every $x, y \in \mathbb{F}_2^n$. Define $k := |A|$, and let $M[A, A]$ denote the $k \times k$ submatrix of M obtained by taking the rows and columns of M indexed by A. Since 0 is a common solution to $\{g_i\}$, we get that $M[A, A] = I_k$ and thus

$$\mathsf{rank}(M) \ge \mathsf{rank}(M[A, A]) = \mathsf{rank}(I_k) = k > 2\binom{n}{\lfloor \Delta/2 \rfloor},$$

which directly contradicts Lemma 8.11. □

Proofs of the corollaries to our low-weight Chevalley-Warning theorem

Now that we have proven our general low-weight Chevalley-Warning theorem (Theorem 8.13), we are ready to prove its corollaries. These proofs are relatively straightforward, but somewhat tedious. We start with the proof of Corollary 8.2, which says that we can always find a nontrivial common solution of Hamming weight $w \leq 8\Delta + 8D/\log(n/D) + 8$.

Proof of Corollary 8.2. First, note that the result is true if $8\Delta + 8D/\log(n/D) + 8 \geq D + \Delta + 1$, as the existence of a nontrivial common solution with Hamming weight $\leq D + \Delta + 1$ is immediate from Chevalley-Warning (see the discussion following Theorem 8.12). Thus we henceforth assume, without loss of generality, that

$$8\Delta + 8D/\log(n/D) + 8 < \Delta + D + 1 \leq n, \tag{8.1}$$

where the last inequality comes from the given hypothesis $D + \Delta < n$. This string of inequalities will come in handy later.

Now, by Theorem 8.13, we know that for any W satisfying both

$$\binom{n}{\leq \lfloor W/2 \rfloor} > 2^{(D+\Delta+1)\cdot 2}, \tag{8.2}$$

$$\binom{n}{\leq \lfloor W/2 \rfloor} > \binom{n}{\leq \lfloor \Delta/2 \rfloor}^2, \tag{8.3}$$

it holds that there is a nontrivial common solution of Hamming weight $\leq W$. We seek to find the smallest W satisfying both Equations (8.2) and (8.3).

We start with Equation (8.3). First, by relatively straightforward binomial inequalities, observe

$$\binom{n}{\leq \lfloor \Delta/2 \rfloor}^2 \leq \binom{2n}{\leq 2\lfloor \Delta/2 \rfloor} \leq \binom{n}{\leq 4\lfloor \Delta/2 \rfloor},$$

where the last inequality uses the fact that $8\Delta < n$ (as implied by Equation (8.1)). Thus, any W satisfying

$$\binom{n}{\leq \lfloor W/2 \rfloor} > \binom{n}{\leq 4\lfloor \Delta/2 \rfloor}$$

also satisfies Equation (8.3). And the above inequality is satisfied by any W satisfying

$$\lfloor W/2 \rfloor > 4\lfloor \Delta/2 \rfloor,$$

where we have again used the assumption that $8\Delta < n$ to ensure $4\lfloor \Delta/2 \rfloor < n$. Thus, any W satisfying

$$W > 4\Delta + 1.$$

also satisfies Equation (8.3).

We now consider Equation (8.2). By Equation (8.1), we may assume $D > 7\Delta + 7$, which means that $\Delta < D/7 - 1$ and thus $(D + \Delta + 1) \cdot 2 < 16D/7$. As a result, any W satisfying

$$\binom{n}{\leq \lfloor W/2 \rfloor} > 2^{16D/7} \tag{8.4}$$

also satisfies Equation (8.2). We consider two cases.

Case (i): $D \leq \log n$: By Equation (8.1), we may assume $n > 8$. It is now easy to verify that for any $W \geq 8$,

$$\binom{n}{\leq \lfloor W/2 \rfloor} \geq \binom{n}{\leq 4} > n^{16/7} \geq 2^{16D/7},$$

thereby satisfying Equation (8.4).

Case (ii): $D > \log(n)$: Let $1 \leq k \leq n$ be a parameter we will pick later, and notice that for any $W \geq 2k$ we have

$$\binom{n}{\leq \lfloor W/2 \rfloor} \geq \binom{n}{\leq \lfloor k \rfloor} \geq \binom{n}{\lfloor k \rfloor} \geq \left(\frac{n}{\lfloor k \rfloor}\right)^{\lfloor k \rfloor}.$$

Thus if $1 \leq k \leq n$ is a parameter satisfying $\lfloor k \rfloor \log(n/\lfloor k \rfloor) > 16D/7$ then any $W \geq 2k$ satisfies Equation (8.4). Now, define $k := 4D/\log(n/D)$. By the case condition and Equation (8.1), we have $4 \leq k < n$, and thus

$$3D/\log(n/D) \leq \lfloor k \rfloor \leq 4D/\log(n/D).$$

By Equation (8.1), we may assume $\log(n/D) \geq 8$, and thus

$$\lfloor k \rfloor \log(n/\lfloor k \rfloor) \geq \frac{3D}{\log(n/D)} \log\left(\frac{n\log(n/D)}{4D}\right) \geq \frac{3D}{\log(n/D)} \log\left(\frac{2n}{D}\right) \geq 3D,$$

which is strictly greater than $16D/7$. Thus, every $W \geq 2k = 8D/\log(n/D)$ satisfies Equation (8.4).

To conclude, we get that every

$$W \geq \max\{4\Delta + 2, 8, 8D/\log(n/D)\}$$

satisfies both Equations (8.2) and (8.3), which completes the proof. □

We now turn towards proving Corollary 8.3, which says that, not only is it possible to find a low weight common solution, but it is even possible to force this solution to be supported on a target set of coordinates S. The proof follows quite directly by combining Corollary 8.2 with the idea of function restrictions.

Proof of Corollary 8.3. For any $y \in \mathbb{F}_2^{|S|}$, let $y^+ \in \mathbb{F}_2^n$ denote the unique string where $y_S^+ = y$ and $y_{\overline{S}}^+ = 0^{n-|S|}$, the string of $n-|S|$ zeroes. Now, for each f_i let $g_i \in \mathbb{F}_2[x_1, \ldots, x_{|S|}]$ denote the polynomial $g_i(x) := f_i(x^+)$. Notice that each $\deg(g_i) \leq \deg(f_i)$. Now, for each i create a polynomial $h_i \in \mathbb{F}_2[x_1, \ldots, x_{|S|}]$ as follows. If $\deg(f_i) > 1$ but $\deg(g_i) = 1$: assume without loss of generality that $g_i(x) \neq x_1$, and set $h_i(x) := 1 + (1 + x_1)(1 + g_i(x))$; otherwise, just define $h_i(x) := g_i(x)$.

Consider the set of polynomials $\{h_i\}$. Notice that the nonlinear degree of $\{h_i\}$ is at most Δ, and the linear degree of $\{h_i\}$ is at most D. Thus, by Corollary 8.2, there is a nontrivial common solution y to $\{h_i\}$ with weight $w \leq 8\Delta + 8D/\log(|S|/D) + 8$. It is then straightforward to verify that y must be a common solution to $\{g_i\}$, and that y^+ must be a common solution to $\{f_i\}$. Furthermore, y^+ is clearly supported on S and has the same Hamming weight as y. □

A proof of our entropy lower bounds

At last, we turn towards proving Theorem 8.11. As it turns out, we will not only exhibit a d-local source that witnesses the lower bounds, but a *d-local affine source*. Our proof will combine our new low-weight Chevalley-Warning theorem (Corollary 8.3) with an ingredient from Cohen and Tal [CT15] that involves *directional derivatives*. Given a polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ and a set of vectors $S \subseteq \mathbb{F}_2^n$, we let f_S denote the *derivative of f in the directions of S*, where

$$f_S(x) := \sum_{T \subseteq S} f\left(x + \sum_{v \in T} v\right).$$

One useful observation is that $\deg(f_S) \leq \max\{0, \deg(f) - |S|\}$. In addition to this, the key ingredient about directional derivatives that we import from Cohen and Tal is the following.

Lemma 8.12 ([CT15]). *For any degree $\leq r$ polynomial $f \in \mathbb{F}_2[x_1, \ldots, x_n]$ and $B \subseteq \mathbb{F}_2^n$,*

$$f\left(x + \sum_{v \in S} v\right) = 0 \text{ for all } S \subseteq B \iff f_S(x) = 0 \text{ for all } S \subseteq B \text{ of size } |S| \leq r.$$

This shows that f is 0 on the entire affine space $x + \mathsf{span}(B)$ if and only if 0 is a common solution to all $(\leq r)$-wise directional derivatives of f across B. Now, given Lemma 8.12 and our low-weight Chevalley-Warning theorem (Corollary 8.3), we are ready to prove our lower bound for extracting from d-local sources.

Proof of Theorem 8.11. Assume without loss of generality that $f(0) = 0$, for otherwise we can work with the polynomial $1 + f$. We will iteratively build a basis $B \subseteq \mathbb{F}_2^n$ of a d-local subspace $X \subseteq \mathbb{F}_2^n$ on which f is constantly 0.

Towards this end, we start by initializing several sets:

- $B \leftarrow \emptyset$.
- $\mathsf{UNIQUE} \leftarrow \emptyset$.
- $\mathsf{SATURATED} \leftarrow \emptyset$.
- $P \leftarrow \{f\}$.

Now, **while** there exists a common nontrivial solution $b \in \mathbb{F}_2^n$ to the set $P \subseteq \mathbb{F}_2[x_1, \ldots, x_n]$ such that b is supported on $[n] \setminus (\mathsf{UNIQUE} \cup \mathsf{SATURATED})$, **do** the following:

- Let b^* be the lowest (Hamming) weight vector of this type (breaking ties arbitrarily).
- Set $B \leftarrow B \cup \{b^*\}$.
- Define $\alpha \in [n]$ as the smallest index such that $b^*_\alpha = 1$.
- Set $\mathsf{UNIQUE} \leftarrow \mathsf{UNIQUE} \cup \{\alpha\}$.
- Set $\mathsf{SATURATED} \leftarrow \{i \in [n] :$ there are d vectors $v \in B$ with $v_i = 1\}$.
- For each $S \subseteq B$ of size $|S| \leq r$, set $P \leftarrow P \cup \{f_S\}$.

This completes the construction of B.

We now proceed with the analysis. First, we argue that at the end of the above construction, B will hold a basis for a d-local subspace $X := \mathsf{span}(B)$. To see why, first define T to be the number of times the above loop executes, and for each $i \in [T]$ let $b^i \in \mathbb{F}_2^n$ denote the vector added to B in iteration i. Then, let $\alpha(i) \in [n]$ denote the smallest index such that $b^i_{\alpha(i)} = 1$, and observe that $b^j_{\alpha(i)} = 0$ for all $j > i$, because of bullets 3-4 above (combined with the loop condition). Thus $B = \{b^1, \ldots, b^T\}$ is a collection of linearly independent vectors: in other words, a basis. Furthermore, bullet 5 above (combined with the loop condition) ensures that $X := \mathsf{span}(B)$ is d-local.

Thus we have argued that X is a d-local subspace of dimension T. Now, by combining bullet 6 above with the loop condition and Lemma 8.12, we immediately get that $f(x) = 0$ for all $x \in X$. Thus, all that remains is to get a lower bound on T, the number of times that the loop executes.

We call the first iteration of the loop *iteration* 1, and we call the pseudocode that precedes the while loop *iteration* 0. Now, for all $t = 0, \ldots, T$, we define the following variables, for convenience:

- Let D_t denote the degree of the system P upon completing iteration t.
- Let Δ_t denote the nonlinear degree of the system P upon completing iteration t.
- Let u_t denote the size of $|\mathsf{UNIQUE}|$ upon completing iteration t.
- Let s_t denote the size of $|\mathsf{SATURATED}|$ upon completing iteration t.
- Let w_t denote the Hamming weight of the vector b^* selected in iteration t.

Note that w_0 is undefined above, and for convenience we define $w_0 := 0$. Furthermore, for any $t > T$, it will be convenient to define $D_t := D_T$, $\Delta_t := \Delta_T$, and so on. Now, before we turns towards getting a lower bound on T, we *upper bound* the above quantities.

- $D_t \leq \sum_{i=0}^{r-1} \binom{t}{i}(r-i) \leq r\binom{t}{\leq r-1}$. This is because, at the end of iteration t we have: $|B| = t$; and P holds the set of polynomials f_S for all $S \subseteq B$ of size $|S| \leq r$; and $\deg(f_S) \leq \max\{0, \deg(f) - |S|\}$.
- $\Delta_t \leq \sum_{i=0}^{r-2} \binom{t}{i}(r-i) \leq r\binom{t}{\leq r-2}$. This holds for the same reasons as above, except that we are now ignoring the polynomials f_S for all $S \subseteq B$ of size $|S| \geq r-1$ (since these have degree at most 1).
- $u_t = t$, since we grow UNIQUE by a (unique) coordinate in every iteration.
- $s_t \leq \frac{1}{d}\sum_{i=1}^{t} w_i$, since otherwise the sum of the Hamming weights of the vectors b^* selected in iterations $1, \ldots, t$ is at least $ds_t > \sum_{i=1}^{t} w_i$, a contradiction.
- $w_t \leq 8\Delta_{t-1} + 8D_{t-1}/\log(\frac{n-u_{t-1}-s_{t-1}}{D_{t-1}}) + 8$, by Corollary 8.3.

We now turn towards getting a lower bound on T. Here, the key observation is that it *must* hold that

$$D_T \geq n - (u_T + s_T),$$

because otherwise the degree of the set of polynomials is strictly less than the size of the set on which it must be supported, so Corollary 8.3 trivially implies that there is *some* nontrivial common solution supported on S (thereby forcing the next execution of the while condition to pass). A straightforward consequence of this is $T \geq 1$ since $D_0 = r < n = n - (u_0 + s_0)$.

Instead of getting a lower bound on T, it will facilitate the analysis to instead lower bound some $\tau \leq T$. In particular, let $\tau \in \{0, 1, \ldots, T-1\}$ be the smallest integer such that

$$D_{\tau+1} + u_{\tau+1} + s_{\tau+1} \geq n/2. \tag{8.5}$$

Such a τ must exist, since we saw above that $D_T + u_T + s_T \geq n$. We now seek to lower bound τ, which will in turn lower bound T.

We start with a basic (but useful) lower bound on τ. In particular, we claim that $\tau \geq 2r$. To see why, suppose for contradiction $\tau < 2r$. Then by definition of τ it holds that $D_{2r} + u_{2r} + s_{2r} \geq n/2$. But by our upper bounds on these quantities (and the fact that $r \geq 1$) we have

$$\begin{aligned} D_{2r} + u_{2r} + s_{2r} &\leq D_{2r} + u_{2r} + 8 \sum_{i=0}^{2r-1} (\Delta_i + D_i + 1) \\ &\leq D_{2r} + u_{2r} + 16r \cdot (\Delta_{2r} + D_{2r} + 1) \\ &\leq r \cdot \binom{2r}{\leq r-1} + 2r + 16r \cdot (r \cdot \binom{2r}{\leq r-2} + r \cdot \binom{2r}{\leq r-1} + r) \\ &\leq r \cdot 2^{2r} + 2r + 16r \cdot (r \cdot 2^{2r} + r \cdot 2^{2r} + r) \\ &\leq 2^{12r}. \end{aligned}$$

Thus we get that $n/2 \leq D_{2r} + u_{2r} + s_{2r} \leq 2^{12r}$, which implies that $r \geq \log(n)/13$. But the theorem hypothesis claims that $r \leq c\log(n)$ for some universal constant c; thus, we can just pick c later to ensure that $c < \frac{1}{13}$, which would yield a contradiction and conclude the proof of the claim that $\tau \geq 2r$. And since $r \geq 1$ by the theorem hypothesis, we also know that $\tau \geq 2$.

Given the above, we now proceed to get a more general lower bound on τ. We will do so by sandwiching the quantity $D_{\tau+1} + u_{\tau+1} + s_{\tau+1}$ between two inequalities. Towards this end, the key observation is that for any $i \in [\tau]$, it holds that $u_i + s_i \leq u_i + s_i + D_i < n/2$.[26] Combining this with Equation (8.5) and our

[26] This useful inequality is the reason we work with τ instead of T, as this is not necessarily true for any $i \in [T]$.

earlier upper bounds on s_t, w_t, we have

$$\begin{aligned}
n/2 &\leq D_{\tau+1} + u_{\tau+1} + s_{\tau+1} \\
&\leq D_{\tau+1} + u_{\tau+1} + \frac{1}{d}\sum_{i=1}^{\tau+1} w_i \\
&\leq D_{\tau+1} + u_{\tau+1} + \frac{8}{d}\sum_{i=0}^{\tau}\left(\Delta_i + \frac{D_i}{\log\left(\frac{n-u_i-s_i}{D_i}\right)} + 1\right) \\
&\leq D_{\tau+1} + u_{\tau+1} + \frac{8}{d}\sum_{i=0}^{\tau}\left(\Delta_\tau + \frac{D_\tau}{\log\left(\frac{n-u_\tau-s_\tau}{D_\tau}\right)} + 1\right) \\
&\leq D_{\tau+1} + (\tau+1) + \frac{8}{d}(\tau+1)\left(\Delta_\tau + \frac{D_\tau}{\log\left(\frac{n-u_\tau-s_\tau}{D_\tau}\right)} + 1\right) \\
&\leq D_{\tau+1} + (\tau+1) + \frac{8}{d}(\tau+1)\left(\Delta_\tau + \frac{D_\tau}{\log\left(\frac{n}{2D_\tau}\right)} + 1\right) \\
&\leq D_{\tau+1} + 2\tau + \frac{16\tau}{d}\left(\Delta_\tau + \frac{D_\tau}{\log\left(\frac{n}{2D_\tau}\right)} + 1\right) \\
&\leq D_{\tau+1} + 18\tau + \frac{16\tau}{d}\left(\Delta_\tau + \frac{D_\tau}{\log\left(\frac{n}{2D_\tau}\right)}\right).
\end{aligned}$$

Thus we know that if we define

$$\begin{aligned}
K_1 &:= D_{\tau+1} + 18\tau + \frac{16\tau}{d}\Delta_\tau, \\
K_2 &:= \frac{16\tau D_\tau}{d\log(\frac{n}{2D_\tau})},
\end{aligned}$$

it must hold that $K_1 + K_2 \geq n/2$, or rather we know that either $K_1 \geq n/4$ or $K_2 \geq n/4$ must hold. We analyze these cases separately, and get a lower bound on τ in each case. But before we do so, it will be useful to recall that for any integers $b > a \geq 0$ it holds that $\binom{b}{a} \leq \binom{b}{a+1}$ if $a+1 \leq b/2$. Since we saw earlier that $\tau \geq 2r$, we have $\binom{2\tau}{\leq r-1} \leq r \cdot \binom{2\tau}{r-1}$ and $\binom{\tau}{\leq r-2} \leq r \cdot \binom{\tau}{r-2}$ and $\binom{\tau}{\leq r-1} \leq r \cdot \binom{\tau}{r-1}$. We now proceed with the case analysis.

Case (1): $K_1 \geq n/4$: By applying the upper bounds on D_t, Δ_t obtained earlier in the proof, the standard binomial inequality $\binom{n}{k} \leq (en/k)^k$, and the fact that $r \geq 2$, we have:

$$\begin{aligned}
n/4 \leq K_1 &= D_{\tau+1} + 18\tau + \frac{16\tau}{d}\Delta_\tau \\
&\leq r \cdot \binom{\tau+1}{\leq r-1} + 18\tau + \frac{16\tau}{d} r \cdot \binom{\tau}{\leq r-2} \\
&\leq r \cdot \binom{2\tau}{\leq r-1} + 18\tau + \frac{16\tau}{d} r \cdot \binom{\tau}{\leq r-2} \\
&\leq r^2 \cdot \binom{2\tau}{r-1} + 18\tau + \frac{16\tau r^2}{d} \cdot \binom{\tau}{r-2} \\
&\leq r^2 \cdot (4e)^{r-1} \cdot \left(\frac{\tau}{r}\right)^{r-1} + 18r\left(\frac{\tau}{r}\right)^{r-1} + \frac{16}{d} r^3 \cdot (3e)^r \cdot \left(\frac{\tau}{r}\right)^{r-1} \\
&\leq 3 \cdot 18 \cdot (4e)^r \cdot r^3 \cdot \left(\frac{\tau}{r}\right)^{r-1} \\
&\leq 2^6 \cdot 2^{7r} \cdot \left(\frac{\tau}{r}\right)^{r-1},
\end{aligned}$$

which implies that

$$n \leq 2^8 \cdot 2^{7r} \cdot \left(\frac{\tau}{r}\right)^{r-1} \leq 2^{15r} \cdot \left(\frac{\tau}{r}\right)^{r-1},$$

which gives

$$\tau \geq r \cdot \left(n \cdot 2^{-15r}\right)^{\frac{1}{r-1}} \geq 2^{-30} \cdot rn^{\frac{1}{r-1}}.$$

Case (2): $K_2 \geq n/4$: In this case we have

$$n/4 \leq K_2 = \frac{16\tau D_\tau}{d \log\left(\frac{n}{2D_\tau}\right)},$$

which of course implies

$$64\tau D_\tau \geq dn \log\left(\frac{n}{2D_\tau}\right).$$

Now, notice that our earlier upper bounds on D_τ yield

$$D_\tau \leq r \cdot \binom{\tau}{\leq r-1} \leq r^2 \cdot \binom{\tau}{r-1} \leq r^2 \cdot (2e)^{r-1} \cdot \left(\frac{\tau}{r}\right)^{r-1} \leq 2^{5r} \cdot \left(\frac{\tau}{r}\right)^{r-1},$$

and combining this with the previous inequality yields

$$64\tau \cdot 2^{5r} \cdot \left(\frac{\tau}{r}\right)^{r-1} \geq 64\tau D_\tau \geq dn \log\left(\frac{n}{2D_\tau}\right) \geq dn \log\left(\frac{n}{2^{5r+1} \cdot \left(\frac{\tau}{r}\right)^{r-1}}\right).$$

Then, applying straightforward bounds on both sides of the above chain of inequalities yields

$$2^{12r} \cdot \left(\frac{\tau}{r}\right)^r \geq dn \log\left(\frac{n}{2^{6r} \cdot \left(\frac{\tau}{r}\right)^{r-1}}\right). \tag{8.6}$$

We now seek to get a lower bound on all τ that satisfy the above inequality. Towards this end, notice that for any $\alpha > 0$ such that

$$2^{12r} \cdot \left(\frac{\alpha}{r}\right)^r < dn \log \left(\frac{n}{2^{6r} \cdot \left(\frac{\alpha}{r}\right)^{r-1}}\right), \tag{8.7}$$

it holds that all τ that satisfy Equation (8.6) must also satisfy $\tau \geq \alpha$: this is because as α decreases, the left hand side of the above inequality decreases, while its right hand side increases.

It is now straightforward to verify that

$$\alpha = 2^{-30} \cdot r \cdot (dn \log n)^{1/r}$$

satisfies Equation (8.7), as long as $d \log n \leq n^{\frac{1}{r-1} - 2^{-10r}}$. Thus in this case we have

$$\tau \geq \alpha = 2^{-30} \cdot r \cdot (dn \log n)^{1/r},$$

provided $d \log n \leq n^{\frac{1}{r-1} - 2^{-10r}}$.

To conclude, since one of the cases must hold, we get that

$$\tau \geq 2^{-30} \cdot \min\{rn^{\frac{1}{r-1}}, r(dn \log n)^{1/r}\}$$

if $d \log n \leq n^{\frac{1}{r-1} - 2^{-10r}}$. But given this condition on $d \log n$, it always holds that $r(dn \log n)^{1/r} \leq rn^{\frac{1}{r-1}}$. Thus, as long as $d \log n \leq n^{\frac{1}{r-1} - 2^{-10r}}$, we get that

$$T \geq \tau \geq 2^{-30} r (dn \log n)^{1/r},$$

as desired. □

A tighter result for local affine sources

Notice that our proof to our lower bound (Theorem 8.11) actually provided a d-local source that is simultaneously an *affine source*. On the other hand, our upper bounds (Theorem 8.10) worked for *any* d-local source. It is natural to ask whether we can get better (perhaps even tight) upper bounds for the restricted family of d-local affine sources. We show the answer is yes, with the following lemma.

Lemma 8.13 (Upper bound of Theorem 8.9)**.** *There exist universal constants $C, c > 0$ such that a random degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is an extractor for d-local affine sources with min-entropy*

$$k \geq Cr \cdot (dn \log n)^{1/r}$$

and error $\varepsilon = 2^{-ck/r}$, except with probability at most $2^{-c\binom{k}{\leq r}}$ over the selection of f.

Notice that combining this with (our proof to) Theorem 8.11 immediately yields Theorem 8.9. Now, in order to prove Lemma 8.13, recall that our upper bounds for the more genereal family of d-local sources (Theorem 8.10) were proven by reducing d-local sources to d-local NOBF sources (Lemma 8.6), followed by applying the fact that a random low-degree polynomial has low correlation with any fixed function (Lemma 8.9). For d-local *affine sources*, we can completely skip the first step, which is where the entropy was lost. Furthermore, for the second step, we can apply the slightly simpler result that a random low-degree polynomial has low *bias* (Lemma 8.5).

Proof of Lemma 8.13. Let $\mathbf{X} \sim \mathbb{F}_2^n$ be a d-local affine source of dimension k. It is straightforward to show, via Gaussian elimination, that the following holds: there exists a subset $S \subseteq [n]$ of size k such that $\mathbf{X}_S$ is uniform over $\mathbb{F}_2^k$, and for every other $j \in [n]$ there exists an affine function $\ell_j : \mathbb{F}_2^k \to \mathbb{F}_2$ such that $\mathbf{X}_j = \ell_j(\mathbf{X}_S)$. Without loss of generality, assume that $S = [k]$. Thus if we consider the function $\ell : \mathbb{F}_2^k \to \mathbb{F}_2^n$ defined as

$$\ell(y_1, \dots, y_k) := (y_1, \dots, y_k, \ell_{k+1}(y_1, \dots, y_k), \dots, \ell_n(y_1, \dots, y_k))$$

we of course have $\mathbf{X} = \ell(\mathbf{U}_k)$, where $\mathbf{U}_k \sim \mathbb{F}_2^k$ denotes the uniform random variable.

Now, let $f \sim \mathbb{F}_2[x_1, \dots, x_n]$ be a random polynomial of degree $\leq r$. Notice that

$$|f(\mathbf{X}) - \mathbf{U}_1| = |f(\ell(\mathbf{U}_k)) - \mathbf{U}_1| = \mathsf{bias}(f(\ell))/2$$

So we want to upper bound the probability that $\mathsf{bias}(f(\ell))$ is large, over our random selection of f. Well, consider fixing whether each monomial $x^S, S \not\subseteq [k]$ exists. Notice that under any such fixing, $f(\ell)$ actually turns into a polynomial of the form $f' + g$, where $f' \sim \mathbb{F}_2[x_1, \dots, x_n]$ is a uniformly random degree $\leq r$ polynomial, and g is a sum of monomials, each of size at most r, where each term in each monomial is linear (i.e., degree 1). In other words, g is a fixed degree $\leq r$ polynomial, and thus $f' + g$ is a uniformly random degree $\leq r$ polynomial over k variables.

Thus, by Lemma 8.5, we get that under *every* such fixing, $\mathrm{bias}(f(\ell)) \leq 2^{-c_1 k/r}$ except with probability $\leq 2^{-c_2\binom{k}{\leq r}}$. Thus for a given d-local affine source $\mathbf{X} \sim \mathbb{F}_2^n$ of min-entropy k, it holds that a random degree $\leq r$ polynomial extracts from $\mathbf{X}$ with error $\leq 2^{-c_1 k/r}$, except with probability $\leq 2^{-c_2\binom{k}{\leq r}}$.

Now, it is not hard to show that there are at most $\binom{k}{\leq d}^n \cdot 2^n$ d-local affine sources of dimension k. Thus, by a union bound, we know that a random degree $\leq r$ polynomial extracts from *all* d-local affine sources $\mathbf{X} \sim \mathbb{F}_2^n$ of min-entropy k, except with probability at most

$$2^{-c_2\binom{k}{\leq r}} \cdot \binom{k}{\leq d}^n \cdot 2^n.$$

Thus as long as $\binom{k}{\leq d}^n \cdot 2^n \leq 2^{c_2\binom{k}{\leq r}/2}$, it holds that a random degree $\leq r$ polynomial extracts from d-local affine sources with min-entropy k with error $\leq 2^{-c_1 k/r}$, with probability $\geq 1 - 2^{-c_2\binom{k}{\leq r}/2}$. It is now easy to verify that the inequality $\binom{k}{\leq d}^n \cdot 2^n \leq 2^{c_2\binom{k}{\leq r}/2}$ holds for the stated lower bound on k. □

8.5 Extensions to more powerful circuits

In Section 8.1, we made the bold claim that local sources are, perhaps, an even more fundamental model than circuit sources. For the case of NC^0 sources, this is clear, as $O(1)$-local sources are *identical* to such circuit sources. As a result, the main theorem of this chapter (Theorem 8.1) immediately yields the following.

Theorem 8.14 (Low-degree polynomials extract from NC^0 sources)**.** *For every fixed $r \in \mathbb{N}$, there exist constants $C, c > 0$ such that the following holds. For every $n \in \mathbb{N}$, there exists a (not necessarily explicit) degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that is an extractor for NC^0 sources of min-entropy at least*

$$k \geq C \cdot (n \log n)^{1/r},$$

which has error $\varepsilon = 2^{-ck/r}$. Furthermore, this is tight up to the constant C.

For AC^0 sources, the connection to local sources is not nearly as obvious. Nevertheless, Viola provided an elegant argument in [Vio14] that local sources do, in fact, capture AC^0 sources (which may, at first glance, appear *more powerful*). As a result, as claimed in Section 8.1, extractors for local sources give extractors for AC^0 sources. In particular, the main theorem of this chapter (Theorem 8.1) immediately yields the following.

Theorem 8.15 (Low-degree polynomials extract from AC^0 sources)**.** *For every fixed $r \in \mathbb{N}, \delta > 0$, and $t \in \mathbb{N}$, there exist constants $C, c > 0$ such that the following holds. For every $n \in \mathbb{N}$ there exists a (not necessarily explicit) degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that is an extractor for AC^0 sources of depth t and size 2^{n^c} of min-entropy at least*

$$k \geq C \cdot n^{\frac{1}{r}+\delta},$$

which has error $\varepsilon = n^{-c\sqrt{\log\log n}}$. Furthermore, this is tight up to the small polynomial factor n^δ.

In particular, notice that in terms of the min-entropy requirement k, low-degree polynomials extract from AC^0 sources with almost the exact same parameters they can achieve on the much weaker NC^0 sources. Now, while the proof of Theorem 8.14 is immediate, the same is certainly not true about Theorem 8.15. Thus, for completeness, let us briefly sketch its proof. Towards this end, the first key ingredient is the following reduction from AC^0 sources to d-local sources, as provided by Viola [Vio14]. Its proof makes use of random restrictions, switching lemmas, and the fact that AC^0 sources keep their min-entropy under such restrictions.

Lemma 8.14 (A reduction from AC^0 sources to d-local sources [Vio14])**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be an AC^0 source of size s, depth t, and min-entropy k. Then for any $d \in \mathbb{N}$ and $\varepsilon > 0$, it holds that $\mathbf{X}$ is ε-close to a convex combination of d-local sources of min-entropy k', where*

$$k' \geq \frac{k}{(C\log s)^{t-1} \cdot (n/\varepsilon)^{1/\log d} \cdot \log(1/\varepsilon)}$$

for some universal constant $C > 0$.

The next ingredient, of course, is our new result on extracting from d-local sources.

Lemma 8.15 (Low-degree polynomials extract from d-local sources (Theorem 8.2, restated))**.** *There exist constants $C, c > 0$ such that for all $n, d \in \mathbb{N}$ and $2 \leq r \leq c\log n$, the following holds. There exists a degree r polynomial $f : \mathbb{F}_2^n \to \mathbb{F}_2$ that is an extractor for d-local sources of min-entropy*

$$k \geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r},$$

which has error $\varepsilon = 2^{-ck/(r2^d d^2)}$.

By leveraging these two key ingredients, the proof of Theorem 8.15 is now immediate.

Proof of Theorem 8.15. Set $d = \sqrt{\log n}$ and $\varepsilon = n^{-\sqrt{\log d}}$, and combine Lemmas 8.14 and 8.15. □

8.6 Applications in complexity theory

Finally, just like the small-space sources studied in Chapter 7, circuit sources also fall into the category of *low-complexity sources*. Thus, our new results on extracting from circuit sources also have applications in complexity theory. In particular, just as in Section 7.4, we gain interesting new insight into the power of computational models for the task of *sampling* - this time, for AC^0 circuits. We go into more detail, below.

8.6.1 Complexity of sampling

The *complexity of sampling* is an exciting new area of complexity theory, where the goal is to understand how various classical computational models perform when they are tasked with *sampling from a distribution*. Lately, the complexity of sampling (especially with AC^0 circuits) has received a lot of attention, and has uncovered a wide range of unexpected applications. The primary motivation behind this new field is the fact that *sampling lower bounds* are harder to obtain than classical (computing) lower bounds, and thus require a whole new suite of tools to tackle. For a detailed introduction to the complexity of sampling, see Chapter 9.

To see why sampling lower bounds are harder to obtain (for a given computational model of interest), it suffices to demonstrate a function $f : \{0,1\}^n \to \{0,1\}^m$ which cannot be computed by the model, but whose output distribution *can* be sampled. In the context of AC^0 circuits, the prototypical example is the function $f(x) := (x, b(x))$, where $b(x) := \oplus_i x_i$. Indeed, classical hardness results in complexity theory imply that AC^0 cannot compute f, but it turns out that small AC^0 circuits *can*, in fact, sample the distribution $f(\mathbf{U}_n)$![27] This set researchers on a quest to find sampling lower bounds for AC^0 circuits against distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$, where b is some simple explicit function.

Sampling lower bounds against low-degree polynomials

A classical result in this direction is due to Viola [Vio16], who showed that there exist degree 2 polynomials $b : \mathbb{F}_2^n \to \mathbb{F}_2$ such that AC^0 circuits cannot sample $(\mathbf{U}_n, b(\mathbf{U}_n))$. As it turns out, by exploiting our new result that low-degree polynomials extract from d-local sources (Theorem 8.2), we can not only recover this result, but in fact prove something that is stronger in two notable ways. First, we prove sampling lower bounds against the entire spectrum of low-degree polynomials, achieving a tradeoff between the degree of the polynomial and the complexity required of the AC^0 circuit that can sample it. Second, our sampling lower bounds even work against distributions of the form $(\mathbf{X}, b(\mathbf{X}))$, where $\mathbf{X}$ can be taken to be anything with sufficiently high min-entropy (including, of course, the uniform distribution). We prove the following.

Theorem 8.16 (Low-degree polynomials are hard to sample)**.** *For every constant $\delta > 0$ there exists some $C > 0$ such that for any $n \geq r \geq 2 \in \mathbb{N}$, there exists a degree r polynomial $b : \mathbb{F}_2^n \to \mathbb{F}_2$ such that the following holds. For any random variable $\mathbf{X} \sim \{0,1\}^n$ with min-entropy k, there does not exist an AC^0 circuit of size s and depth t satisfying*

$$(C \log s)^{t-1} \leq k/rn^{\frac{1}{r}+\delta}$$

that can sample $(\mathbf{X}, b(\mathbf{X}))$.

For the setting of degree 2 polynomials $b : \mathbb{F}_2^n \to \mathbb{F}_2$, Theorem 8.16 shows that AC^0 circuits of constant depth and *exponential size* $2^{n^{\Omega(1)}}$ cannot sample $(\mathbf{U}_n, b(\mathbf{U}_n))$. It is well-known that AC^0 circuits of size 2^n can sample any distribution, highlighting the strength of our lower bounds. More notably, even though AC^0 circuits cannot *compute* linear functions, they sure can *sample* them with ease (as demonstrated in Chapter 9). Thus, just as with the case of ROBPs (Section 7.4), there is a massive jump in the required complexity of an AC^0 circuit that samples low-degree polynomials: from linear size for degree $d = 1$, to *exponential* size for degree $d = 2$. Now, let us turn towards proving this result.

[27] For an explanation why, see Chapter 9.

A formal presentation of the proof

The proof of Theorem 8.16 will follow the same general game plan as the proof of our analogous result for ROBPs (Theorem 7.5) as presented in Section 7.4. Thus, for fear of sounding too repetitive, we refer the reader to that section for a detailed overview of how the proof is executed, and simply dive right into the technical details here. Towards this end, we will again need three key ingredients. The first ingredient claims that any distribution sampled by an AC^0 circuit of size s and depth t is an AC^0 source. This is in fact trivially true, and indeed the definition of AC^0 sources. Nevertheless, we include it below, so as to better mirror the proof of our corresponding result for ROBPs.

Lemma 8.16 (A reduction from AC^0 samplers to AC^0 sources)**.** *For any $\mathbf{X} \sim \{0,1\}^n$, if there is an AC^0 circuit of size s and depth t that samples $\mathbf{X}$, then $\mathbf{X}$ is an AC^0 source of size s and depth t.*

Proof. This is immediate from the definition of AC^0 sources, as given in Section 8.1. □

Next, we will use the elegant reduction of Viola from AC^0 sources to d-local sources, as presented in Section 8.5. We include it again here, for convenience.

Lemma 8.17 (A reduction from AC^0 sources to d-local sources - Lemma 8.14, restated)**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be an AC^0 source of size s, depth t, and min-entropy k. Then for any $d \in \mathbb{N}$ and $\varepsilon > 0$, it holds that $\mathbf{X}$ is ε-close to a convex combination of d-local sources of min-entropy k', where*

$$k' \geq \frac{k}{(C \log s)^{t-1} \cdot (n/\varepsilon)^{1/\log d} \cdot \log(1/\varepsilon)}$$

for some universal constant $C > 0$.

Finally, we'll need the following main result from this chapter.[28]

Lemma 8.18 (Low-degree polynomials extract from d-local sources - Theorem 8.2, restated)**.** *There exist constants $C, c > 0$ such that for all $n, d \in \mathbb{N}$ and $2 \leq r \leq c \log n$, the following holds. There exists a degree r polynomial $p : \mathbb{F}_2^n \to \mathbb{F}_2$ that is an extractor for d-local sources of min-entropy*

$$k \geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r},$$

which has error $\varepsilon = 2^{-ck/(r2^d d^2)}$.

Equipped with these three ingredients, we can finally prove that low-degree polynomials are hard to sample by AC^0 circuits (Theorem 8.16).

Proof of Theorem 8.16. Let $b : \{0,1\}^n \to \{0,1\}$ be a degree r polynomial that is an extractor for d-local sources of min-entropy $\Gamma \geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r}$ and error $\varepsilon = 2^{-c\Gamma/(r2^d d^2)}$, as promised by Lemma 8.18. Let $\mathbf{X} \sim \{0,1\}^n$ be an arbitrary distribution of min-entropy k, where k is to be decided later. Let $\mathbf{Q} \sim \{0,1\}^{n+1}$ be a distribution sampled by an AC^0 circuit of size s and depth t, and parse $\mathbf{Q}$ as $\mathbf{Q} = (\mathbf{R}, \mathbf{S}) \sim \{0,1\}^n \times \{0,1\}$. Suppose for contradiction that $\mathbf{Q} \equiv (\mathbf{X}, b(\mathbf{X}))$, and notice this implies $H_\infty(\mathbf{Q}) = k$.

[28] Previously, it was known that low-degree polynomials extract from affine sources, and thus 1-local sources. However, due to the significant loss in entropy that arises from reducing AC^0 sources all the way down to 1-local sources (via Lemmas 8.3 and 8.14), this cannot be used to obtain Theorem 8.16.

Now, by combining Lemmas 8.16 and 8.17, we know that $\mathbf{Q} = (\mathbf{R}, \mathbf{S})$ is δ-close to a convex combination of d-local sources $\mathbf{Q}' = (\mathbf{R}', \mathbf{S}')$ that have min-entropy at least

$$k' := \frac{k}{(C \log s)^{t-1} \cdot (n/\delta)^{1/\log d} \cdot \log(1/\delta)}.$$

Furthermore, by fixing the $\leq d$ input bits on which $\mathbf{S}'$ depends, the min-entropy chain rule (Lemma 2.1) tells us that each $\mathbf{Q}' = (\mathbf{R}', \mathbf{S}')$ is $2^{-k'/2}$-close to a convex combination of d-local sources $\mathbf{Q}'' = (\mathbf{R}'', \mathbf{S}'')$ such that $\mathbf{S}''$ is fixed and $\mathbf{R}''$ has min-entropy at least $k'' := k'/2 - d$. Putting this all together, we get that $\mathbf{Q}$ is $(2^{-k''} + \delta)$-close to a convex combination of d-local sources $\mathbf{Q}'' = (\mathbf{R}'', \mathbf{S}'') \sim \{0,1\}^{n+1}$ such that $\mathbf{S}''$ is fixed and $\mathbf{R}''$ has min-entropy at least

$$k'' := \frac{k}{2 \cdot (C \log s)^{t-1} \cdot (n/\delta)^{1/\log d} \cdot \log(1/\delta)} - d.$$

Now, define k so that $k'' \geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r}$. Then we know that b in fact extracts from each $\mathbf{R}''$ defined above with error $\varepsilon = 2^{-ck''/(r2^d d^2)}$, and thus since $\mathbf{S}''$ is constant, we have

$$\Pr[b(\mathbf{R}'') \neq \mathbf{S}''] \geq \frac{1}{2} - \varepsilon.$$

By the closeness of $\mathbf{Q} = (\mathbf{R}, \mathbf{S})$ to this convex combination of sources, we get that

$$\Pr[b(\mathbf{R}) \neq \mathbf{S}] \geq \frac{1}{2} - \varepsilon - \delta - 2^{-k''} = \frac{1}{2} - 2^{-ck''/(r2^d d^2)} - \delta$$

for some constant $c > 0$. But at the very beginning we assumed that $(\mathbf{R}, \mathbf{S}) \equiv (\mathbf{X}, b(\mathbf{X}))$, which means that $\Pr[b(\mathbf{R}) \neq \mathbf{S}] = 0$. Thus even if we set $\delta := 1/4$, we arrive at a contradiction because we may assume $ck''/(r2^d d^2)$ is an arbitrarily large constant, by our setting of k''. Finally, recall that this setting of k'' (which yielded the contradiction) came about by setting k so that (when we fix $\delta := 1/4$) it holds that

$$\begin{aligned}&\frac{k}{2 \cdot (C \log s)^{t-1} \cdot (n/\delta)^{1/\log d} \cdot \log(1/\delta)} - d \\ &= \frac{k}{4 \cdot (C \log s)^{t-1} \cdot (4n)^{1/\log d}} - d \\ &\geq Cr \cdot 2^d d^2 \cdot (2^d n \log n)^{1/r}.\end{aligned}$$

Solving for $(C \log s)^{t-1}$ and setting $d = \sqrt{\log n}$ immediately yields the result. □

8.7 Future directions

In this chapter, we gave a full characterization of the power of low-degree polynomials for extracting from local sources (Theorem 8.1). This not only opens the door to construct better explicit extractors for this extremely challenging model (Lemma 8.1), but provides valuable insight into the structure of one of the most fundamental algebraic objects (Lemma 8.2). Of course, as the ultimate goal in extractor theory is to obtain explicit constructions, many open questions remain.

Improved explicit extractors for 2-local sources Naturally, the most obvious open question is to construct better explicit extractors for local sources. In particular, we still don't even have explicit *dispersers* for 2-local sources that can handle min-entropy below $k = \sqrt{n}$! This is in stark contrast to nearly *every other* simple model studied in the world of seedless extraction, for which we now have dispersers (in fact, high error extractors) that can handle $k = O(\log n)$ min-entropy [Li23]. Thus, there seems to be something fundamentally different about extracting from local sources, and any new explicit constructions for this setting would significantly push forward the frontier of extractor theory.

An improved reduction from local sources to local NOBF sources A key tool we obtained in this chapter was a new reduction from local sources to local NOBF sources (Lemma 8.1). We believe that this reduction should help facilitate future explicit constructions of local source extractors, as it now suffices to simply construct an explicit extractor for local NOBF sources. However, recall that our reduction transforms a d-local source $\mathbf{X} \sim \{0,1\}^n$ with min-entropy k into a d-local NOBF source with min-entropy $\Omega(k/2^d d^2)$. While this entropy loss is no big deal if we are working with $O(1)$-local sources, it becomes a much bigger problem if we wish to extract from sources of higher locality. It would be great to achieve a more efficient reduction, so that we can safely deal with the latter case. We believe it may be possible to keep around $k/\operatorname{poly}(d)$ bits of min-entropy using a more careful argument.

An improved low-weight Chevalley-Warning theorem In order to obtain our entropy lower bounds, we proved a new type of Chevalley-Warning theorem (Lemma 8.2), which guarantees that a small set of low-degree polynomials not only has a common solution, but has a common solution *of low (Hamming) weight*. Recall that for a collection $\{f_i\}$ of polynomials of linear degree D and nonlinear degree Δ with zero as a common solution, our result shows the existence of a solution that has weight $w \leq O(\Delta + \frac{D}{\log(n/D)})$. It is not too hard to show that this is tight if $\Delta = 0$ or $D = 0$, simply by using the Hamming bound (Theorem 2.2) or the polynomial $\sum_{1 \leq |S| \leq \Delta} x^S$, respectively. However, it is an interesting question whether the tightness remains for collections of polynomials that don't exhibit such an extreme "degree profile" - for example, taking $\{f_i\}$ to be a collection of $\Delta/2$ quadratics.

Chapter 9

The space complexity of sampling

In the previous two chapters, we explored the question of constructing extractors for sources generated by *limited-memory algorithms* and *low-depth circuits*. While these models are central to the field of randomness extraction, their study gives rise to another, perhaps even more fundamental question: what do these sources even *look like*? In other words,

What distributions can (and cannot) be sampled by algorithms with bounded computational resources?

This question lies at the heart of a burgeoning new area of complexity theory called the *complexity of sampling*. While classical complexity theory seeks to quantify the resources required to compute some function $f : \{0,1\}^n \to \{0,1\}$, this new area seeks to quantify the resources required to *sample from some distribution* $\mathbf{Q} \sim \{0,1\}^n$. Proving lower bounds in this new setting is more challenging than in the classical setting, and has yielded interesting new techniques and surprising applications. However, while a flurry of recent work has developed a rich theory around the complexity of sampling with *low-depth circuits*, little is known about the complexity of sampling with *limited memory*.

In this chapter, we aim to fill this gap, and initiate a study of the *space complexity of sampling*. We model space-bounded algorithms using oblivious read-once branching programs (ROBPs), and start by answering several basic questions about the complexity of sampling with this classical model. Among these, we show:

- Sampling with ROBPs is equivalent to sampling with small-space sources.
- Sampling is easier than computing (sampling lower bounds are harder than classical lower bounds).
- There exist simple distributions that are extremely difficult to sample with ROBPs.

These results are easy to prove, and lay the foundation for the theory of sampling with limited memory. With these foundations in place, we move towards establishing more challenging sampling lower bounds.

In our first main result, we obtain near-optimal sampling lower bounds for a classical family of distributions. In particular, we show that any distribution sampled by an ROBP of width $2^{\Omega(n)}$ has statistical distance $1-2^{-\Omega(n)}$ from any distribution that is uniform over a good error-correcting code. More generally,

The results presented in this chapter are based on the following joint work:

- [CGZ22] Eshan Chattopadhyay, Jesse Goodman, and David Zuckerman. The space complexity of sampling. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *LIPIcs*, pages 40:1–40:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022 we obtain sampling lower bounds for any list-decodable code, which are nearly tight. Previously, such a result was only known for sampling with AC^0 circuits. As an application of our result, a known connection implies new data structure lower bounds for storing codewords.

In our second main result, we provide a generic way to bootstrap weak sampling lower bounds into extremely strong sampling lower bounds. In particular, we obtain a direct product theorem for sampling with ROBPs. Previously, no direct product theorems were known for the task of sampling, for any computational model. A key ingredient in our proof is a simple new lemma about amplifying statistical distance between sequences of somewhat-dependent random variables. Using this lemma, we also obtain a simple new proof of a known lower bound for sampling disjoint sets using two-party communication protocols.

9.1 Introduction

Part III of this thesis has largely focused on extracting from *samplable sources*. As we have seen, these sources take the form $\mathbf{X} = F(\mathbf{U})$, where F is just a function coming from some "low-complexity" class $\mathcal{C}$. Thus far in our journey, we have sought out explicit extractors for the class $\mathcal{C}$ of *limited-memory algorithms* (Chapter 7) and *low-depth circuits* (Chapter 8). In this chapter, we do not wish to extract from yet another class $\mathcal{C}$ of samplable sources. Instead, we seek out an answer to an *even more fundamental* question:

What do these samplable sources even "look like"*?*

At first glance, this question appears innocent enough, as it could certainly help us construct better extractors. Indeed, as we have already seen (in Lemmas 7.1 and 8.1), insight into the *structure* of samplable sources can greatly assist in extracting from them. However, the true motivation behind this question runs much deeper than that. In particular, if one were to reflect upon the actual definition of samplable sources for just a minute, one might realize that the above question is really asking,

What distributions can (and cannot) be sampled using limited computational resources?

With this interpretation in mind, it becomes clear that the pursuit of this question may unlock new applications and revelations that extend far beyond the reach of extractor theory. In fact, it is this exact question that gave rise to an exciting new area of *complexity theory*, called the *complexity of sampling*.

The complexity of sampling

A central goal in complexity theory has always been to *quantify the resources* required to perform certain tasks. Traditionally, complexity theory has focused on the task of *computing*. Here, one fixes a function $f : \{0,1\}^\ell \to \{0,1\}^n$ and computational model $\mathcal{C}$ (e.g., low-depth circuits), and asks for lower bounds on the size of any $F \in \mathcal{C}$ that *computes* f.[1] Recently, a growing body of work has instead sought to understand the power of these same computational models for the alternative task of *sampling*. Now, instead of fixing a function f, one picks a target distribution $\mathbf{Q} \sim \{0,1\}^n$. Then, one asks for lower bounds on the size of any $F \in \mathcal{C}$ that *samples* $\mathbf{Q}$.[2] In other words, the goal is to understand the *complexity of sampling* $\mathbf{Q}$.

While the origins of this question can be traced all the way back to the 80s [JVV86, AST$^+$03, GGN10], Viola was the first to advocate for a systematic study into the complexity of sampling [Vio12a]. The primary motivation is easy to appreciate: While many computational tasks can be formalized as *computing a specific function* f, the need to *sample from a specific distribution* $\mathbf{Q}$ also frequently arises in computer science

[1] We say that F *computes* f if $F(x) = f(x)$ for all x.

[2] We say that F *samples* $\mathbf{Q}$ if $F(\mathbf{U}) = \mathbf{Q}$.

[Liu23, Chapter 1]. Thus, it is only natural to seek out a theory that can help us understand how our favorite computational models perform in this common alternative task. But beyond this basic motivation, a number of other compelling reasons have emerged to further motivate research into the complexity of sampling.

- **Extractor theory**: As hinted at before, understanding the complexity of sampling certain distributions can provide valuable insight into their *structure*. This, in turn, can be used to greatly facilitate the construction of extractors for such sources [DW12, Vio14, Vio12b, CG21, ACG$^+$22].

- **Complexity theory**: Surprisingly, *sampling lower bounds* (as defined above) are more challenging to obtain than classical lower bounds.[3] Thus, their pursuit may unveil exciting new techniques and applications [Vio12a]. Indeed, this phenomenon has already been observed in several fascinating works that exploit sampling lower bounds to obtain *data structure lower bounds* [Vio12a, LV12, BIL12], and in a recent paper of Guruswami, Lyu, and Wang [GLW22] which suggests that sampling lower bounds could help make progress on the influential *range avoidance problem* [KKMP21, Kor21].

- **Quantum computing**: Finally, it is believed that *sampling problems* are great candidates for demonstrating *quantum advantage* [AA13, BFNV19].[4] Thus, by proving new sampling lower bounds against classical models of computation, one can make important progress towards this landmark goal in quantum computing. In fact, this program has already been successfully executed to demonstrate quantum advantage for a variety of restricted models of computation [AST$^+$03, WP23].

Needless to say, a better understanding of the complexity of sampling distributions could have a huge impact on complexity theory and beyond. As a result, ever since the seminal work of Viola [Vio12a], the complexity of sampling has garnered a massive amount of interest [LV12, Aar14, DW12, Vio14, Vio12b, BIL12, JSWZ13, Wat13, BCS16, Wat16, Vio16, Wat20, Vio20, GW20, CG21, BDF$^+$22, CGZ22, ACG$^+$22, Vio23, FLRS23, WP23, YZ24], unveiling deep new insights into classical computational models, and unearthing a swath of unexpected applications. However, despite this exciting progress, results are still only known for a few computational models like AC^0 circuits and communication protocols. In particular, little is known about the complexity of sampling with *limited memory*, while this remains a fundamental model in other areas of complexity theory.

Our goal In this chapter, we aim to fill this gap, and initiate a study of the *space* complexity of sampling. In order to do so, our goals are threefold. First, we need a reasonable model for sampling in limited memory. Second, we must identify key questions whose answers would form a solid foundation for this new direction. Third, and most importantly, we'd like to *answer these key questions*.

Key questions

Before we officially introduce our model (and risk boring the reader with unmotivated formalisms), let us start by identifying some interesting questions about sampling in limited memory. We will focus on *five key questions*, motivated by some fascinating observations and results that have already been discovered in the complexity of sampling. Towards this end, let us go on a brief tour of what is known about sampling with AC^0 *circuits* (the most well-studied model in this new field), and place our key questions in this context.

[3] A formal proof of this will appear in the coming pages.

[4] Here, the goal is to find a problem that can't be solved by a classical computer, but can easily be solved by a quantum computer.

Sampling is easier than computing A motivating paradigm in the complexity of sampling is the surprising fact that a computational model $\mathcal{C}$ may be more powerful at sampling than computing. In more detail, consider fixing a function $f : \{0,1\}^n \to \{0,1\}^m$ and comparing the tasks of *computing* f on every input x, with the task of *sampling* its output distribution $f(\mathbf{U}_n)$ [GGN10]. Intuitively, the latter task should be easier. Indeed, any $F \in \mathcal{C}$ that computes f is also guaranteed to have $F(\mathbf{U}_n) = f(\mathbf{U}_n)$. On the other hand, if f is the inverse of a *one-way permutation*, then f is very difficult to compute, but $f(\mathbf{U}_n)$ is trivial to sample [Vio12a].

Amazingly, we also have examples of simple *explicit* functions that demonstrate this separation.[5] The canonical example is the function $f : \{0,1\}^n \to \{0,1\}^{n+1}$ defined as $f(x) = (x, \oplus_{i\in[n]} x_i)$. Here, some of the most celebrated results in complexity theory imply that f cannot be computed in AC^0 [FSS84, Ajt83, Yao85, Hås86], yet Babai [Bab87], Boppana and Lagarias [BL87] gave an extremely simple way to *sample* its output distribution in AC^0. In particular, just consider the function $g : \{0,1\}^n \to \{0,1\}^{n+1}$ defined as

$$g(x) := (x_1, x_1 \oplus x_2, x_2 \oplus x_3, \dots, x_{n-1} \oplus x_n, x_n),$$

and note that g is both (1) implementable in AC^0, and (2) has the same output distribution as f. Thus, while AC^0 cannot *compute* the function f, it can certainly *sample* its output distribution $f(\mathbf{U}_n)$.[6] In general, this means that sampling is strictly *easier* than computing, which implies that

Sampling lower bounds are more challenging *to obtain than classical lower bounds,*
and their pursuit may unveil exciting new techniques and applications.

As discussed at the beginning of Section 9.1, this observation has been one of the key driving forces behind research into the complexity of sampling, and (in particular) into the pursuit of *sampling lower bounds*. However, we actually only saw above that sampling is easier than computing in the context of AC^0 *circuits*. In this chapter, we wish to initiate a systematic study into the complexity of sampling with *limited memory*, and thus it is natural to ask whether the above motivation still holds in this new setting.

Question 9.1. *Does there exist an explicit Boolean function $b : \{0,1\}^n \to \{0,1\}$ such that $(x, b(x))$ is hard to compute with limited memory, but $(\mathbf{U}_n, b(\mathbf{U}_n))$ is easy to sample with limited memory?*

Sampling lower bounds against input-output pairs In the discussion leading up to our first question, we saw that for the parity function $b : \{0,1\}^n \to \{0,1\}$, it holds that $(x, b(x))$ is hard to compute in AC^0, yet $(\mathbf{U}_n, b(\mathbf{U}_n))$ is easy to sample in AC^0. Given this observation, Viola raised the challenge of finding an explicit distribution of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ that is *hard* to sample in AC^0 [Vio12a]. The following year, he demonstrated exactly such a distribution [Vio14], showing that AC^0 circuits cannot sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ if b is an extractor for 1-*local sources* (Chapter 8). Nearly a decade later, Viola greatly strengthened this result [Vio20], proving that any distribution generated by an AC^0 circuit is, in fact, *extremely far in statistical distance* from the above distribution.[7] More formally, he showed that if $b : \{0,1\}^n \to \{0,1\}$ is a good enough extractor for 1-local sources, then for any AC^0 circuit $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$, it holds that

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{-n^{\Omega(1)}}.$$

[5] We don't actually know if one-way functions (not to mention one-way *permutations*) exist: this would imply $\mathsf{P} \neq \mathsf{NP}$ [Gol01].

[6] In fact, this can be extended to show that AC^0 can sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ for *any symmetric function b* (including $\mathsf{majority}$) up to an exponentially small error [Vio12a], in addition to some other classically hard functions such as *inner product (mod 2)* [IN96].

[7] One might expect that a low-error extractor for 1-local sources would immediately yield such a result, but several technical issues arise with such an approach. Instead, Viola's argument relies on a sophisticated method of *polarizing min-entropy* [Vio20].

Assuming Question 9.1 can be answered positively, it is natural to ask whether a similar result can be established for sampling in the limited memory setting. In particular, our second question is as follows.

Question 9.2. *Does there exist an explicit Boolean function* $b : \{0,1\}^n \to \{0,1\}$ *such that* $(\mathbf{U}_n, b(\mathbf{U}_n))$ *is hard to sample with limited memory?*

Sampling lower bounds against general distributions It is easy to see that sampling lower bounds for distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ cannot exceed $1/2$, since $(\mathbf{U}_n, 0)$ or $(\mathbf{U}_n, 1)$ will always yield an an upper bound of $1/2$, and both are trivial to sample (using any reasonable model of computation). A complementary question, suggested by Viola [Vio12a], is to find other natural distributions $\mathbf{Q} \sim \{0,1\}^n$ with much stronger sampling lower bounds - perhaps even approaching 1. In 2012, Lovett and Viola demonstrated an explicit distribution $\mathbf{Q}$ of exactly this type, showing that for any AC^0 circuit $F : \{0,1\}^\ell \to \{0,1\}^n$, it holds that

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - \varepsilon,$$

where $\varepsilon = n^{-\Omega(1)}$ [LV12]. In a subsequent work, Beck, Impagliazzo, and Lovett greatly improved this lower bound on statistical distance, achieving $\varepsilon = 2^{-n^{\Omega(1)}}$ for the exact same explicit distribution $\mathbf{Q}$ [BIL12]. Given this huge success in obtaining extremely strong sampling lower bounds against general distributions for AC^0 circuits, we must ask whether the same can be done for sampling in limited memory.

Question 9.3. *Does there exist an explicit distribution* $\mathbf{Q} \sim \{0,1\}^n$ *that is extremely hard to sample with limited memory?*

Sampling lower bounds against codes As it turns out, the distribution $\mathbf{Q} \sim \{0,1\}^n$ that yields the strong sampling lower bounds against AC^0 circuits (discussed above) is not only explicit, but also extremely *natural*. In particular, Lovett and Viola [LV12] and Beck, Impagliazzo, and Lovett [BIL12] achieve the abovementioned results by setting $\mathbf{Q}$ to be uniform over the elements of an *asymptotically good error-correcting code*.[8] In doing so, they obtain a number of wonderful consequences, which would not have been possible had they used a more artificial distribution.

- They provide valuable new insight into the *structure of error-correcting codes*, one of the most fundamental objects spanning coding theory, pseudorandomness, complexity theory, and beyond [GRS22].
- Using an observation of Viola [Vio12a], they obtain new *data structure lower bounds* for storing codewords: a classical real-world task that arises whenever data must be written to an unreliable disk.
- They obtain a *complexity separation* between sampling with AC^0 circuits [Vio12a] and sampling with communication protocols [GW20], since the latter can trivially sample (certain) good codes.

In addition to these consequences, proving *sampling lower bounds against good codes* has turned out to be a surprisingly fundamental task, as the quantum analog of this question played a crucial role in the recent breakthrough resolution of the *NLTS conjecture* [ABN23].[9] Given all of this, even if we can manage to find *some* explicit distribution $\mathbf{Q}$ that is extremely hard to sample in limited memory (and thereby positively answer Question 9.3), we would really like to know if this is true when $\mathbf{Q}$ is uniform over a good code.

Question 9.4. *Are good codes hard to sample with limited memory?*

[8] An error-correcting code (Definition 2.8) is said to be *asymptotically good* if it has constant relative distance and constant rate.

[9] The NLTS conjecture, short for *No Low-energy Trivial State* conjecture, was a fundamental barrier blocking a resolution of the *quantum PCP conjecture* [AAV13], which remains one of the most important open problems in quantum complexity theory.

Sampling lower bounds via direct product theorems Thus far, our questions have asked for small-space analogs of several key results known for the complexity of sampling in AC^0. For our final question, we ask for a type of result that has yet to be studied in the complexity of sampling, but which has been well-explored in the context of classical complexity theory. In particular, we ask for a *direct product theorem*.

In classical complexity, direct product theorems (e.g., Yao's XOR Lemma [Yao82]) are key tools that are used for *hardness amplification*: such a result roughly says that if a function f is somewhat hard to compute for a given computational model, then t independent copies of f are *very* hard to compute for that same model. Direct product theorems offer a simple and concrete way to (i) construct simple functions with strong (average-case) lower bounds, and thus (ii) establish strong (average-case) complexity separations between complexity classes. As a result, direct product theorems have been heavily examined throughout the history of classical complexity theory [Yao82, IRW94, Raz98, Sha04b, LSŠ08, Ema08, Kla10, Jai15], and have become a fundamental tool in every theorist's toolbox. Despite this, prior to our work,

No direct product theorems were known for the task of *sampling*.

In the context of sampling, a direct product theorem can be defined as a result which asserts the following: If a distribution $\mathbf{Q} \sim \{0,1\}^n$ has statistical distance δ from any distribution sampled by some computational model, then t independent copies of $\mathbf{Q}$ (concatenated together) have statistical distance $\gg \delta$ from any distribution sampled by that same model. While no direct product theorems were known for sampling with *any* computational model, we focus here on sampling with *limited memory*. Thus, our final question is:

Question 9.5. *Can a direct product theorem be established for distributions sampled with limited memory?*

This completes our brief tour of some of the most well-known results on sampling with AC^0 circuits, and finishes our presentation of the *five key questions* we wish to answer about sampling with limited memory. In this chapter, we make significant progress on each of these five questions. But before we present our results, we must first discuss our model for sampling distributions with limited memory.

The model

To model sampling with limited memory, we use the classic model of *oblivious read-once branching programs* (ROBPs). This model corresponds to the *streaming* model of computation, and thus a better understanding of the power of ROBPs for sampling may provide new insights and tools for streaming algorithms. In what follows, we give a formal definition of this model, and relate it to *small-space sources* (Chapter 7).

Computing with ROBPs Read-once branching programs (ROBPs) may very well be the most popular model for computing with limited memory. An ROBP of width w and length ℓ is just a directed acyclic graph $G = (V, E)$ with $\ell + 1$ layers $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$, each holding at most w vertices. For every $i \in [\ell]$, each vertex $v \in V_{i-1}$ has two outgoing edges into the next layers: one labeled 0, and one labeled 1. Some vertex in V_0 is labeled v_{start}, and some vertex in V_ℓ is labeled v_{accept}. The ROBP computes a function $F : \{0,1\}^\ell \to \{0,1\}$ in the natural way: given an input $x \in \{0,1\}^\ell$, it follows the path with edge labels corresponding to $x_1, x_2, \ldots, x_\ell$, and outputs 1 if and only if the path arrives at v_{accept}.

Sampling with ROBPs To model sampling with limited memory, we would like to use the same computational model as above, and just replace its input with uniform bits. However, this ROBP computes a function $F : \{0,1\}^\ell \to \{0,1\}$, and so the distribution $F(\mathbf{U}_\ell)$ it samples will be over $\{0,1\}$. To sample general distributions $\mathbf{Q} \sim \{0,1\}^n$, we need an ROBP that can compute multi-output functions $F : \{0,1\}^\ell \to \{0,1\}^n$.

Towards this end, perhaps the most natural way to extend an ROBP to output multiple bits is to simply allow it to output a *sequence of bits* upon reading any input bit. More formally, we can assign each edge in the ROBP an additional label consisting of a *string of output bits*. Then, given an input $x \in \{0,1\}^\ell$, the ROBP can traverse a path in the usual manner, but now output all the output labels seen along the way.[10] Now, while this may be the most obvious way to define *multi-output ROBPs*, one problem arises: with this definition alone, different paths through the ROBP may yield a different number of output bits. To guarantee that the ROBP will compute a function of the form $F : \{0,1\}^\ell \to \{0,1\}^n$, we make one additional simplifying assumption: just as the inputs in an ROBP are "layered," we will assume that the outputs are also layered. That is, we require any two edges traversing between the same two layers to be labeled with the same number of output bits. This completes our definition of *multi-output ROBP* (see Definition 9.3 a more formal definition). To sample a distribution using it, we simply replace its input with uniformly random bits.

Just as a standard ROBP models an algorithm that *reads from an input stream*, multi-output ROBPs also allow the algorithm to *write to an output stream*: this is because a multi-output ROBP can write an arbitrary number of bits at each time step, without storing any of them in its memory. Furthermore, note that for functions with one bit of output, our definition is equivalent to the classic single-bit-output ROBP definition. Thus, we will henceforth refer to multi-output ROBPs simply as ROBPs.

An equivalence theorem Finally, as described at the beginning of Section 9.1, a fundamental goal in studying the *complexity of sampling* is to understand the different types of distributions that can be sampled if we take one of our favorite computational models $F \in \mathcal{C}$, and replace its input with uniformly random bits. As such, the model we use for *sampling in limited memory* is just an ROBP fed with uniformly random bits. However, a dedicated reader of this thesis might recall that in Chapter 7, we studied a *different* model of distributions that can be sampled in limited memory, known as *small space sources* (Definition 7.2). And while ROBPs are the classical way to model limited-memory computation in *complexity theory*, small-space sources are the classical way to model limited memory distributions in *extractor theory*.

Thus, we have a sticky situation where there are two different, equally canonical, ways to model sampling with limited memory: one that is favored by complexity theorists (feeding uniform bits into an ROBP), and one that is favored by extractor theorists (small-space sources). Is there any way to reconcile these two notions? As it turns out, the answer is a resounding *yes*. In fact, up to a negligible loss in parameters, the two notions are *equivalent*. We ultimately prove the following, which will be instrumental to our main results.[11]

Theorem 9.0 (Equivalence theorem)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$ and $\varepsilon > 0$, the following hold. If there is an ROBP of width w that samples $\mathbf{X}$, then $\mathbf{X}$ is a small-space source of width $2w$. If $\mathbf{X}$ is a small-space source of width w, then there is an ROBP of width $7w$ that samples a distribution ε-close to $\mathbf{X}$.*

Notice that the above theorem makes *no reference* to the input length ℓ of the ROBP. Indeed, like most results in the complexity of sampling, the sampling power of a computational model is not significantly affected by its *input length*, only its *size* (which, here, corresponds to the *width* of the ROBP). Furthermore, we remark that the error ε appearing in Definition 7.2 is just an artifact from the fact that small-space sources can assign *irrational* probabilities to elements in their support, while ROBPs can only sample distributions that are more *granular*. Notably, $\varepsilon > 0$ can taken to be arbitrarily small at no cost, and can even be eliminated in many cases. In Section 9.3, we give a detailed exposition of Theorem 9.0, and prove several other equivalence theorems between various models for sampling and computing with limited memory.

[10] This is exactly a *read-once* version of multi-output branching programs studied in the classical works [BFK+81, BC82, Bea91].

[11] In Theorem 9.0, a *small-space source of width w* simply refers to a *space $s := \log(w)$ source* (Definition 7.2).

9.2 Our results

With our questions in mind and our model formally defined, we are ready to state our results. As promised, we launch a systematic study of the *space complexity of sampling*. In order to do so, we provide positive answers to all five questions from Section 9.1. Question 9.1, Question 9.2, and Question 9.3 ("*the basics*") are straightforward to answer, by applying easy-to-prove lower bounds from communication complexity. On the other hand, Question 9.4 (*sampling lower bounds against codes*) and Question 9.5 (*sampling lower bounds via direct product theorems*) are more challenging to resolve, and we view these as our main contributions.

The basics

To start things off, let us present our formal, positive answers to the first three questions from Section 9.1. In our first main theorem, we demonstrate an explicit function that is very difficult to compute using ROBPs, but extremely easy to sample. In other words, we provide a positive answer to Question 9.1.

Theorem 9.1 (Sampling is easier than computing)**.** *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any $\varepsilon > 0$ and ROBP $F : \{0,1\}^n \to \{0,1\}$ of width at most $2^{\frac{n\varepsilon^2}{9} - \log(1/\varepsilon)}$,*

$$\Pr_x[F(x) = b(x)] < \frac{1}{2} + \varepsilon,$$

but there exists an ROBP $G : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $2n$ and length $\ell \le n + \log n + 2$ such that

$$G(\mathbf{U}_\ell) = (\mathbf{U}_n, b(\mathbf{U}_n)).$$

The explicit function we use above is the notorious *address function*,[12] which has a rich history in proving lower bounds against various computational models [Sav98]. Given Theorem 9.1, we see that just like with AC^0, sampling with ROBPs is strictly easier than computing. As a result, sampling lower bounds for ROBPs will be strictly more challenging to obtain than classical lower bounds. Next, we demonstrate an explicit function that is not only hard to compute, but also *hard to sample*, answering Question 9.2.

Theorem 9.2 (Sampling lower bounds against input-output pairs)**.** *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width at most $2^{n/16}$,*

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \ge \frac{1}{2} - 128 \cdot 2^{-n/16}.$$

The explicit function we use above is the *inner product (mod 2)* function IP, and the proof that it works simply follows from the fact that it is a good low-error *two-source extractor* [CG88]. Surprisingly, however, $(\mathbf{U}_n, \mathsf{IP}(\mathbf{U}_n))$ *can* be sampled in AC^0 [IN96], and thus the above result gives a simple distribution that can be sampled by AC^0 circuits, but cannot be sampled by ROBPs of width $2^{\Omega(n)}$. Furthermore, we prove an equivalence theorem (Theorem 9.8) which shows that sampling lower bounds against distributions of the form $(\mathbf{U}_n, b(\mathbf{U}_n))$ immediately yield correlation bounds against b for ROBPs. By instantiating this theorem with Theorem 9.2, we obtain as a corollary an alternate proof of the known result that *inner product (mod 2)* has exponentially small correlation with ROBPs of exponential width. Finally, for our last basic theorem, we show how to boost the above sampling lower bounds to be exponentially close to 1, answering Question 9.3.

[12] The address function $b : \{0,1\}^k \times [k] \to \{0,1\}$ is simply defined as $b(x, i) := x_i$. The address function is also known as the *index function* or the *multiplexer function*, and is a specialization of the well-known *Andreev's function* [And87].

Theorem 9.3 (Sampling lower bounds against general distributions)**.** *There exists an explicit distribution $\mathbf{Q} \sim \{0,1\}^n$ such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width at most $2^{n/12}$,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 16 \cdot 2^{-n/12}.$$

As it turns out, the distribution $\mathbf{Q} \sim \{0,1\}^n$ that witnesses this lower bound is remarkably simple: it's just the concatenation of two copies of the same uniform random variable $(\mathbf{U}_{n/2}, \mathbf{U}_{n/2})$. Since AC^0 can clearly sample this distribution, Theorem 9.3 gives an even easier way to separate AC^0 circuits and ROBPs of width $2^{\Omega(n)}$ for the task of sampling. This concludes the discussion of our three basic theorems, which answer Questions 9.1 to 9.3. Together, they provide a solid foundation to study the space complexity of sampling, and set the stage for our two main results.

Sampling lower bounds against codes

In our first main theorem, we show it is hard to sample good codes using ROBPs. More generally, we obtain the following sampling lower bounds against *any* (n, k, d) code, which are nearly tight.

Theorem 9.4 (Sampling lower bounds against codes)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n, k, d) code. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 12w \cdot 2^{-\frac{kd}{4n}}.$$

Remark 9.1 (Tightness - Theorem 9.15, informal)**.** *We show Theorem 9.4 is nearly tight, in the sense that for almost all n, k, d, there exists an (n, k, d) code that* can *be sampled by an ROBP of width $2^{\widetilde{O}(\frac{kd}{n})}$.*

As a corollary, we immediately get that any distribution sampled by an ROBP of exponential width has statistical distance exponentially close to 1 from a good code, answering Question 9.4. We remark that these sampling lower bounds against codes for ROBPs are stronger than the best known sampling lower bounds against codes for AC^0. In particular, the best sampling lower bounds against good codes for AC^0 are of the form $1 - 2^{-n^{\Omega(1)}}$ [BIL12], and the authors leave as an open problem whether similar lower bounds for AC^0 can be obtained against (n, k, d) codes with $kd \geq n^{1+\Omega(1)}$. On the other hand, our sampling lower bounds against good codes are of the form $1 - 2^{-\Omega(n)}$ for ROBPs of width $2^{\Omega(n)}$, and we obtain lower bounds of the form $1 - 2^{-n^{\Omega(1)}}$ against (n, k, d) codes with $kd \geq n^{1+\Omega(1)}$ for ROBPs of width $2^{n^{\Omega(1)}}$. Finally, we note that we actually obtain a more general version of Theorem 9.4, which works for any *list-decodable code* (Theorem 9.16). We refer the reader to Section 9.5 for more details.

Applications to data structure lower bounds By applying a known connection between sampling lower bounds and data structure lower bounds [Vio12a], our sampling lower bounds immediately yield tight data structure lower bounds for storing codewords succinctly and retrieving them using ROBPs (Section 9.7).

Corollary 9.1 (Data structure lower bounds)**.** *For any good code $Q \subseteq \{0,1\}^n$ of dimension k, there is a constant $c > 0$ such that if we can store codewords of Q using $k + r$ bits so that a codeword can be computed by an ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width at most 2^{cn}, then we must have redundancy $r \geq \lfloor cn \rfloor$.*

The above corollary shows that if one wishes to store codewords that are retrievable by a width $2^{\Omega(n)}$ ROBP, they must use $\Omega(n)$ extra bits of redundancy. This is tight up to constant factors, since (1) It is easy to store codewords that are retrievable by a width 2^n ROBP using 0 extra bits of redundancy, and (2) It is easy to store codewords that are retrievable by a width 1 ROBP using $n - k$ extra bits of redundancy. We remark that these data structure lower bounds for storing codes and retrieving them using ROBPs is stronger than the best known data structure lower bounds for storing codes and retrieving them using AC^0 circuits. In particular, for ROBPs of exponential width $2^{\Omega(n)}$, we show that $r \geq \Omega(n)$ bits of redundancy are necessary, whereas the best known result for AC^0 requires $r \geq n^{\Omega(1)}$ bits of redundancy [BIL12].

Applications to separating ROBPs and communication protocols Finally, it is worth noting that our sampling lower bounds provide a simple family of distributions that separate ROBPs and two-party communication protocols for the task of sampling. In particular, if we let $\mathbf{X} \sim \{0,1\}^{n/2}$ be any good code, and let $\mathbf{0} \in \{0,1\}^{n/2}$ denote the all zeroes vector, it is straightforward to show that two-party communication protocols can exactly sample $(\mathbf{X}, \mathbf{0})$ with 0 bits of communication, whereas Theorem 9.4 implies that any ROBP of width $2^{\Omega(n)}$ samples a distribution with statistical distance $1 - 2^{-\Omega(n)}$ from $(\mathbf{X}, \mathbf{0})$. This highlights the fact that the sampling lower bounds established in Theorem 9.4 cannot be obtained via results from communication complexity, unlike the bounds obtained in Theorems 9.1 to 9.3.

Sampling lower bounds via direct product theorems

Our second main theorem is a direct product theorem, answering Question 9.5. This gives a generic way to construct distributions with strong sampling lower bounds against ROBPs. Informally, we show that if a distribution $\mathbf{Q}$ is even a little hard to sample for ROBPs, then the distribution $\mathbf{Q}^{\otimes t}$ (defined as a sequence of t independent copies of $\mathbf{Q}$) is *extremely hard* to sample for ROBPs. More formally, we prove the following.

Theorem 9.5 (Direct product theorem)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w, it holds that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$. Then for any $t \in \mathbb{N}$ and ROBP $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width w, it holds that*

$$|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}.$$

In particular, Theorem 9.5 gives a way to boost statistical distance lower bounds of the form $\delta > 0$ (some tiny constant) to lower bounds of the form $1 - 2^{-\Omega(t)}$. Moreover, a notable feature of our direct product theorem is that it is *strong*, in the sense that the statistical distance blows up even if the ROBP keeps all of its width (one usually needs to sacrifice some computational power in order to get such hardness amplification).

A simple new lemma on amplifying statistical distance A key ingredient in the proof of our direct product theorem is a simple new lemma on amplifying statistical distance between sequences of somewhat-dependent random variables. To the best of our knowledge, no such lemma was previously known, and we believe it may be of independent interest.

Lemma 9.1. *Let $\mathbf{X} \sim V^n$ and $\mathbf{Y} \sim V^n$ each be a sequence of random variables over V, where elements in the sequence need not be independent. Suppose that for every $i \in [n]$ and $v \in V^{i-1}$,*

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)| \geq \delta.$$

Then

$$|\mathbf{X} - \mathbf{Y}| \geq 1 - e^{-n\delta^2/2}.$$

The proof of Lemma 9.1 is not difficult: we simply prove an analogous result over a more amenable notion of distance, known as the *Bhattacharyya coefficient*, and use estimates on statistical distance (in terms of the Bhattacharyya coefficient) to obtain the desired result. Despite its simple proof, we believe that it could be a useful tool for proving lower bounds. In particular, we discuss one such application, below.

Applications to sampling with two-party communication protocols As an application of the above lemma, we obtain a simple new proof of a known result on sampling lower bounds for two-party communication protocols [GW20]. In particular, we provide in Section 9.8 a short, self-contained proof that for any distribution $\mathbf{X} \sim \{0,1\}^n \times \{0,1\}^n$ sampled by two-party communication protocols with $\Omega(n)$ bits of communication, it holds that $\mathbf{X}$ has statistical distance $1 - 2^{-\Omega(n)}$ from the distribution $\mathbf{Q} \sim \{0,1\}^n \times \{0,1\}^n$ that is uniform over pairs of disjoint strings. This concludes the presentation of our main results.

Organization

The remainder of this chapter is organized as follows. We start by providing formal definitions for all our models of computing and sampling in Section 9.3, where we also prove a host of useful *equivalence theorems* (including Theorem 9.0). In Section 9.4, we prove our three basic results on the space complexity of sampling (Theorems 9.1 to 9.3), providing a solid foundation on which to build our two main results. We prove our two main results in the following two sections. In particular, we obtain our sampling lower bounds against codes (Theorem 9.4) in Section 9.5, and we prove our direct product theorem (Theorem 9.5) in Section 9.6. Finally, we give some applications of our results in data structure lower bounds (Section 9.7) and communication complexity (Section 9.8), and conclude with some open problems (Section 9.9).

9.3 The models

We are now ready to begin the technical portion of the chapter. This section will be dedicated to formalizing our models and establishing useful relationships between them. We start with our models for *computing* in small space in Section 9.3.1, and define our models for *sampling* in small space in Section 9.3.2. Finally, in Section 9.3.3 we prove a variety of *equivalence theorems* (including Theorem 9.0), which demonstrate important relationships these models. These theorems are very useful, and exploited throughout the chapter.

9.3.1 Computing in small space

Let us now proceed to define our formal models for computing in small space. Here, read-once branching programs (ROBPs) may very well be the simplest and most popular model. We gave an informal description of them in Section 9.1, and provide a formal definition of this model below.

Definition 9.1 (ROBP)**.** *An* ROBP $\mathcal{B}$ *of width* w *and length* ℓ *is a directed acyclic graph* $G = (V, E)$ *consisting of* $\ell + 1$ *disjoint layers* $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$*, each holding* w *vertices. For every* $i \in [\ell]$*, each vertex* $v \in V_{i-1}$ *has exactly two outgoing edges into* V_i*, one of which is labeled* 0*, and the other labeled* 1*. There is a designated* start vertex $v_{\mathsf{start}} \in V_0$*, and a designated* accept vertex $v_{\mathsf{accept}} \in V_\ell$*. The ROBP* $\mathcal{B}$ *computes a function* $f_{\mathcal{B}} : \{0,1\}^\ell \to \{0,1\}$ *as follows. On input* $x \in \{0,1\}^\ell$*, the program starts at* v_{start} *and traverses the unique path* $P(x)$ *whose edges are labeled with input bits* $x_1, x_2, \ldots, x_\ell$*. The program outputs* 1 *if* $P(x)$ *terminates on* v_{accept}*, and* 0 *otherwise.*

ROBPs are useful for computing Boolean functions $f : \{0,1\}^\ell \to \{0,1\}$. But if we wish to use ROBPs to *sample distributions* over $\{0,1\}^n$, we need to extend the definition of ROBPs to model the computation of functions $f : \{0,1\}^\ell \to \{0,1\}^n$ with multi-bit outputs. A first attempt might look like the following.

Definition 9.2 (Σ-ROBP)**.** *A* Σ*-ROBP* $\mathcal{B}$ *of width* w *and length* ℓ *is identical to an ROBP with the same parameters (as per Definition 9.1), except instead of having a designated accept vertex* $v_{\mathsf{accept}} \in V_\ell$*, each vertex* $v \in V_\ell$ *is labeled with an element of* Σ*. The* Σ*-ROBP* $\mathcal{B}$ *computes a function* $f_{\mathcal{B}} : \{0,1\}^\ell \to \Sigma$ *as* *follows. On input $x \in \{0,1\}^\ell$, the program starts at v_{start} and traverses the unique path $P(x)$ whose edges are labeled with input bits $x_1, x_2, \dots, x_\ell$. Then, it outputs the label (in Σ) of the final vertex on this path.*

By setting Σ to $\{0,1\}^n$, a Σ-ROBP can certainly compute functions of the form $f : \{0,1\}^\ell \to \{0,1\}^n$. However, upon further inspection, such ROBPs have an inherent flaw: the number of possible strings they can output is naturally limited by their width. Thus, while Σ-ROBPs have certain use cases, we would really like a better way to extend Definition 9.1 to an even more general definition of multi-output ROBPs, whose output length is not limited in such a way. We use the following definition, which can be thought of as a *read-once* version of classical multi-output branching programs [BFK+81, BC82, Bea91].

Definition 9.3 (Multi-output ROBP)**.** *A* multi-output ROBP *$\mathcal{B}$ of width w and (input) length ℓ is a directed acyclic graph $G = (V, E)$ consisting of $\ell + 1$ disjoint layers $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$, each holding w vertices. For every $i \in [\ell]$, each vertex $v \in V_{i-1}$ has exactly two outgoing edges into V_i, one of which is labeled with the input bit $\mathsf{0}$, and the other labeled with the input bit $\mathsf{1}$. Each edge e is also labeled with* output bits *$\Gamma(e) \in \{0,1\}^*$, and we assume that all edges e between the same two layers V_{i-1}, V_i have the same output length $|\Gamma(e)| = \gamma_i \geq 0$. The* output length *of $\mathcal{B}$ is $n = \sum_i \gamma_i$. Finally, there is a designated* start vertex *$v_{\mathsf{start}} \in V_0$. The ROBP $\mathcal{B}$ computes a function $f_{\mathcal{B}} : \{0,1\}^\ell \to \{0,1\}^n$ as follows. On input $x \in \{0,1\}^\ell$, the program starts at v_{start} and traverses the unique path $P(x)$ whose edges are labeled with input bits $x_1, x_2, \dots, x_\ell$. The program outputs the concatenation of all output bits seen along this path, so that $f_{\mathcal{B}}(x) = (\Gamma(e))_{e \in P(x)}$.*

It is straightforward to verify that Definition 9.3 generalizes Definition 9.2, which generalizes Definition 9.1. Going forward, we will use Definition 9.3 for our definition of multi-output ROBPs, and will omit the qualifier "multi-output" when referring to them. For brevity, we will also sometimes call a function $f : \{0,1\}^\ell \to \{0,1\}^n$ an ROBP, when we really mean that f is the function computed by an ROBP.

9.3.2 Sampling in small space

Now that we have our models for *computing* in small space formally defined, we are ready to discuss our formal models for *sampling* in small space. We start with the main motivating model that we study in this chapter, which simply involves feeding uniform bits into an ROBP.

Definition 9.4 (ROBP sampler)**.** *An* ROBP sampler *$\mathcal{B}$ of width w and input length ℓ and output length n is just an ROBP with the same parameters (as per Definition 9.3). The distribution $\mathbf{X} \sim \{0,1\}^n$ sampled by the ROBP $\mathcal{B}$ is simply $\mathbf{X} = f_{\mathcal{B}}(\mathbf{U}_\ell)$, where $f_{\mathcal{B}}$ is the function computed by $\mathcal{B}$.*

While Definition 9.4 may be the most natural way to define sampling in small-space to a complexity theorist, an avid reader of this thesis may recall that we have already defined an *alternative* model for small-space samplers, which has received much more attention from the extractor community. In particular, in Chapter 7, we studied the model of *small-space sources*, first introduced by Kamp, Rao, Vadhan and Zuckerman [KRVZ11]. We will also study small-space sources in this chapter, but will refer to them as *KRVZ samplers*, to emphasize that we are treating them as samplers instead of sources.

Definition 9.5 (KRVZ sampler - Definition 7.2, restated)**.** *A* KRVZ sampler *$\mathcal{B}$ of width w and output length n is a directed acyclic graph $G = (V, E)$ consisting of $n + 1$ disjoint layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$, each holding w vertices. For every $i \in [n]$, each vertex $v \in V_{i-1}$ has an* arbitrary number of outgoing edges *into V_i, some of which are labeled $\mathsf{0}$, and the rest labeled $\mathsf{1}$. There is a designated start vertex $v_{\mathsf{start}} \in V_0$, and each vertex $v \in V$ has a probability distribution p_v over its outgoing edges. The distribution $\mathbf{X} \sim \{0,1\}^n$ sampled by $\mathcal{B}$ is the one generated by taking a random walk over $\mathcal{B}$, which starts at v_{start}, transitions according to $\{p_v\}$, and outputs the edge labels seen along the way.*

In Section 9.3.3, we will show that ROBP samplers and KRVZ samplers are *equivalent* (up to a small loss in parameters), thereby unifying the two most natural models from the worlds of complexity and extractor theory. Next, throughout this chapter, we also prove sampling *upper bounds* to show the tightness of our results. To obtain our upper bounds, we will construct KRVZ samplers that come close to sampling certain distributions. As it turns out, our upper bounds will actually be even stronger, and we will show it often suffices to use a weaker model of sampling, which we call a *simple sampler*. It is identical to the KRVZ sampler, except that each vertex in the branching program is restricted to have out-degree exactly two.

Definition 9.6 (Simple sampler)**.** *A* simple sampler *$\mathcal{B}$ of width w and output length n is a directed acyclic graph $G = (V, E)$ consisting of $n + 1$ disjoint layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$, each holding w vertices. For every $i \in [n]$, each vertex $v \in V_{i-1}$ has* exactly two outgoing edges *into V_i, one of which is labeled* 0*, and the other labeled* 1*. There is a designated start vertex $v_{\mathsf{start}} \in V_0$, and each vertex $v \in V$ has a probability distribution p_v over its outgoing edges. The distribution $\mathbf{X} \sim \{0, 1\}^n$ sampled by $\mathcal{B}$ is the one generated by taking a random walk over $\mathcal{B}$, which starts at v_{start}, transitions according to $\{p_v\}$, and outputs the edge labels seen along the way.*

For brevity, we sometimes call a distribution $\mathbf{X} \sim \{0, 1\}^n$ an (ROBP, KRVZ, simple) sampler, when we really mean that $\mathbf{X}$ is the distribution sampled by an (ROBP, KRVZ, simple) sampler. For our final definition of this section, we introduce the following variant of KRVZ and simple samplers, which will be important whenever we need to show that these can be simulated by ROBP samplers.

Definition 9.7 (Granular sampler)**.** *A KRVZ or simple sampler is called α-*granular *if each edge probability is an integer multiple of α, and if $\alpha = 2^{-t}$ for some $t \in \mathbb{N}$, we simply refer to such a sampler as* granular.

Now that we have all of our models for sampling in small space defined, we record some basic but useful facts about KRVZ and simple samplers. First, we show a big enough simple sampler can sample anything.

Fact 9.1. *Any distribution $\mathbf{X} \sim \{0, 1\}^n$ can be sampled by a simple sampler of width $w = |support(\mathbf{X})|$.*

Proof. We construct the simple sampler $\mathcal{B}$, consisting of graph $G = (V, E)$, as follows. Let $V = V_0 \cup V_1 \cup \cdots \cup V_n$, where V_0 consists of the single start vertex v_{start} and $V_i := \text{support}(\mathbf{X}_{1\to i}) \subseteq \{0, 1\}^i$ for each $i \in [n]$. Then for each $u \in V_{i-1}$ and $(u, b) \in V_i$, draw an edge from u to (u, b), label it with bit b, and give it probability $\Pr[\mathbf{X}_{1\to i} = (u, b) \mid \mathbf{X}_{1\to i-1} = u]$. □

Since KRVZ samplers strictly generalize simple samplers, it immediately follows from Fact 9.1 that any KRVZ sampler can also sample any distribution. However, we record this result anyway, since we will give a different proof that allows for more flexibility when designing *granular* KRVZ samplers.

Fact 9.2. *Any distribution $\mathbf{X} \sim \{0, 1\}^n$ can be sampled by a KRVZ sampler of width $w = |support(\mathbf{X})|$.*

Proof. We construct the simple sampler $\mathcal{B}$, consisting of graph $G = (V, E)$, as follows. Let $V = V_0 \cup V_1 \cup \cdots \cup V_n$, where V_0 consists of the single start vertex v_{start}, and each $V_i, i \in [n]$ is a fresh copy of $\text{support}(\mathbf{X}) \subseteq \{0, 1\}^n$. For each $v \in V_1$, draw an edge from v_{start} to v, label it with the bit v_1, and give it probability $\Pr[\mathbf{X} = v]$. Then, for every $i \in [n - 1]$ and $v \in V_i$, draw an edge from v to its copy in V_{i+1}, label it v_{i+1}, and give it probability 1. □

For our final fact about KRVZ and simple samplers, we note that if two distributions can be sampled by one of these samplers in width w, then so can the concatenation of these two distributions.

Fact 9.3. *Let $\mathbf{X} \sim \{0,1\}^n, \mathbf{Y} \sim \{0,1\}^k$ be independent distributions that can each be sampled by a KRVZ (simple) sampler of width w. Then the distribution $(\mathbf{X}, \mathbf{Y})$ can be sampled by a KRVZ (simple) sampler of width w. Moreover, if the samplers for $\mathbf{X}$ and $\mathbf{Y}$ were α-granular, then so is the sampler for $(\mathbf{X}, \mathbf{Y})$.*

Proof. This is almost immediate from Definition 9.6 and Definition 9.5. In particular, simply wire the two samplers for $\mathbf{X}$ and $\mathbf{Y}$ in a series configuration, and merge their boundaries. □

This concludes the formal introduction of the models that will be used in this chapter for both computing and sampling. Next, we prove a host of theorems that establish useful relationships between them.

9.3.3 Equivalence theorems

Throughout this chapter, we make use of several *equivalence theorems*, which establish important relationships between the various models introduced in Sections 9.3.1 and 9.3.2 for sampling and computing in limited memory. At a high level, we prove three different types of equivalence theorems.

- **Equivalence theorems between ROBP samplers and KRVZ samplers:** We prove that ROBP samplers and KRVZ samplers are equivalent, up to a small loss in parameters. That is, we prove the formal version of Theorem 9.0 (Theorem 9.6), along with a few minor variants that achieve better parameters in certain settings. We will heavily exploit these theorems through this chapter, as they allow us to focus on proving sampling results for the most friendly model (usually KRVZ samplers), and immediately obtain equivalent results for less friendly models (usually ROBP samplers).

  Equivalence theorems between simple samplers and ROBP computers for input-output pairs: We prove that sampling a distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ with a simple sampler is equivalent to computing the function b with an ROBP. We obtain both worst-case and average-case versions of this equivalence (Theorems 9.7 and 9.8), which provide a way to convert sampling lower bounds into hard functions for ROBPs.

- **Equivalence theorems between simple samplers and ROBP computers for flat distributions:** We prove that sampling a distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over some set S with a simple sampler is equivalent to computing the indicator of S with an ROBP. We obtain both worst-case and average-case versions of this equivalence (Theorems 9.9 and 9.10), which provide another way to convert sampling lower bounds into hard functions for ROBPs.

We hope that these equivalence theorems are useful in future work, and we exploit them heavily throughout this chapter. However, we remark that they are a bit tedious to prove, and are not our main focus. Thus, after briefly glancing at the statements of theorems referenced above, the reader should feel free to skip their proofs and proceed to Section 9.4, where we begin a formal exposition of the main results quoted in Section 9.2. With that being said, let's continue onwards with the proofs of our equivalence theorems.

Equivalence theorems between ROBP samplers and KRVZ samplers

In our first and most important equivalence theorem, we show that ROBP samplers and KRVZ samplers are *equivalent*, up to a small loss in parameters. As a reminder from Section 9.3.2, KRVZ samplers are the same thing as small-space sources, and thus this equivalence theorem is exactly the one advertised in Section 9.1.

Theorem 9.6 (Equivalence theorem between ROBP samplers and KRVZ samplers - Theorem 9.0, formal)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$,*

- *If there is an ROBP of width w and input length ℓ that samples $\mathbf{X}$, then there exists a KRVZ sampler of width $2w$ that samples $\mathbf{X}$.*
- *If there exists a KRVZ sampler of width w that samples $\mathbf{X}$, then for any $\varepsilon > 0$, there exists an ROBP of width $7w$ and input length $\ell = 8nw\log(nw/\varepsilon)$ that samples a distribution that is ε-close to $\mathbf{X}$.*

In many settings, the second bullet of the above theorem will be applied to a KRVZ sampler that is α-granular (Definition 9.7), for some $\alpha = 2^{-t}$ and $t \in \mathbb{N}$. In this case, we can strengthen this result to get an ROBP of width $7w$ and input length $\ell = 4nwt$ that *exactly* samples $\mathbf{X}$. Furthermore, we will also provide an alternative version of the second bullet that saves on input length at the cost of a greater blow-up in width.

Let us now turn towards proving Theorem 9.6. For convenience, we isolate its first bullet as follows.

Lemma 9.2 (ROBP sampler $\Longrightarrow$ KRVZ sampler)**.** *For any $\mathbf{X} \sim \{0,1\}^n$, if there exists an ROBP of width w and input length ℓ that samples $\mathbf{X}$, then there exists a KRVZ sampler of width $2w$ that samples $\mathbf{X}$.*

To prove this, we will need to show that KRVZ samplers can efficiently simulate ROBP samplers. In other words, we need to transform an ROBP sampler into a KRVZ sampler that generates the same distribution. This will require some basic local modifications to the ROBP sampler, but not any complex machinery.

The second bullet in Theorem 9.6 will be more challenging to prove. Here, we need to show that ROBP samplers can efficiently simulate KRVZ samplers; or rather, that any KRVZ sampler can be transformed into an ROBP sampler that generates the same distribution. This transformation will proceed in **two stages**.

The first stage addresses the following issue: a KRVZ sampler can generate distributions that assign certain elements arbitrarily precise probabilities (since its edges may be assigned probabilities that are arbitrary reals), whereas the distribution generated by an ROBP sampler has some fundamental limit to its precision (the probability assigned to any element will be an integer multiple of $2^{-\ell}$, where ℓ is the input length of the ROBP). Thus, the distributions generated by ROBP samplers are inherently "granular," so if we would like to transform a KRVZ sampler into an ROBP sampler, we would first like to transform the KRVZ sampler into a *granular* KRVZ sampler (as per Definition 9.7), without introducing too much error. As it turns out, Kamp, Rao, Vadhan, and Zuckerman proved a lemma of exactly this type (in the language of *small space sources*), which we use as the **first stage** of our transformation.

Lemma 9.3 (KRVZ sampler $\Longrightarrow$ granular KRVZ sampler [KRVZ11, Lemma 8.4])**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a KRVZ sampler of width w. For any $\alpha = 1/A$ with $A \in \mathbb{N}$, there exists an α-granular KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^n$ of width w that is (αnw)-close to $\mathbf{X}$.*

After we use the above lemma to make our KRVZ sampler into a granular KRVZ sampler, the next step will be to transform this granular KRVZ sampler directly into an ROBP sampler. We will prove the following, which is the **second stage** of the transformation needed to prove the second bullet of Theorem 9.6.

Lemma 9.4 (Granular KRVZ sampler $\Longrightarrow$ ROBP sampler)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a (2^{-t})-granular KRVZ sampler of width w that samples $\mathbf{X}$, then there exists an ROBP of width $7w$ and input length $\ell = 4nwt$ that samples $\mathbf{X}$.*

Indeed, Lemma 9.4 is the result mentioned earlier, which strengthens the second bullet of Theorem 9.6 when the KRVZ sampler is granular. Given Lemmas 9.2 to 9.4, we can easily prove Theorem 9.6 as follows.

Proof of Theorem 9.6. The first bullet is clearly true by Lemma 9.2. For the second bullet, let $\mathbf{X} \sim \{0,1\}^n$ be any distribution that can be generated by a KRVZ sampler of width w, and pick any $\varepsilon > 0$. Now, set $\alpha = 2^{-t}$ and $t = \lceil \log(nw/\varepsilon) \rceil$. By Lemma 9.3, there is a 2^{-t}-granular KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^n$ of width w that is $(2^{-t}nw \leq \varepsilon)$-close to $\mathbf{X}$. By Lemma 9.4, there is an ROBP of width $7w$ and input length $\ell = 4nwt = 4nw\lceil \log(nw/\varepsilon) \rceil \leq 8nw\log(nw/\varepsilon)$ that exactly samples $\mathbf{X}^*$. The result follows. □

Thus, if we can show Lemmas 9.2 and 9.4, we are done. We start with the proof of Lemma 9.2, which is much easier, but also less interesting. After that, we prove Lemma 9.4, completing the proof of Theorem 9.6.

ROBP sampler $\implies$ KRVZ sampler Let's show that KRVZ samplers can simulate ROBP samplers.

Proof of Lemma 9.2. As a reminder, we must show that if there is an ROBP $\mathcal{B}$ of width w and length ℓ that samples $\mathbf{X} \sim \{0,1\}^n$, then there is a KRVZ sampler $\mathcal{B}'$ of width $2w$ that samples $\mathbf{X}$.

Step 1 The first step is transforming $\mathcal{B}$ into an ROBP where each edge is labeled by 0 or 1 output bits. Towards this end, let $G = (V, E)$ be the underlying graph of $\mathcal{B}$, with layer $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$. Fix any $i \in [n]$, and consider the edges between layers V_{i-1} and V_i. They are each labeled by γ_i output bits. If γ_i is already 0 or 1, we do not change anything about layers V_{i-1} and V_i. So henceforth assume $\gamma_i > 1$.

The idea is to simply add γ_{i+1} new layers $L_i^0, L_i^1, \ldots, L_i^{\gamma_i}$ in between V_{i-1} and V_i. For each vertex $v \in V_{i-1}$, we do the following: suppose that v originally had an edge to $u \in V_i$ with input label 0 and output label $s \in \{0,1\}^{\gamma_i}$. Now, delete that edge and simulate it as follows: add a new vertex v^j to each new layer L_i^j. Draw an edge from v to v^0, and give it input label 0 and no output label. Then, draw two new edges from v^0 to v^1, with input labels $0, 1$, and output label s_1. Then, draw two new edges from v^1 to v^2, with input labels $0, 1$, and output label s_2. Continue this process until we have drawn edges up to vertex v^{γ_i}. Finally, draw two new edges from v^{γ_i} to u with input labels $0, 1$ and an empty output label.

Now suppose that v originally had an edge to $w \in V_i$ with input label 1 and output label $t \in \{0,1\}^{\gamma_i}$. Do the exact same process as before, except give the first new edge that is drawn the input label 1 (instead of 0). Finally, recall that we needed to do this for every $v \in V_{i-1}$. After this is done, repeat the process for any V_{i-1}, V_i with $\gamma_i > 1$. Now, it is straightforward to verify that for any $v \in V_{i-1}, u \in V_i$, if we plug random bits into our new branching program and arrive at v, then the probability of then transitioning from v to u and outputting any given string s of bits will be the same as it was before. Thus the new ROBP samples the same distribution $\mathbf{X}$ as before, has width $2w$, and every edge is labeled with at most 1 output bit.

Step 2 The second step is to transform this new ROBP into a KRVZ sampler for $\mathbf{X}$. Let $\mathcal{B}$ now denote the new ROBP we have sampling $\mathbf{X}$, which has each edge labeled with at most 1 output bit. Let its underlying graph $G = (V, E)$ have layers $V = V_0 \cup V_1 \cup \cdots \cup V_\ell$, where we are now guaranteed $\ell \geq n$. Define a collection of indices $0 = a_0 < a_1 < \cdots < a_n$ as follows: let a_j be the smallest integer greater than a_{j-1} such that V_{a_j} has incoming edges labeled with 1 output bit. To translate our ROBP sampler into a KRVZ sampler for $\mathbf{X}$, we will do a transformation for each pair of layers $V_{a_{j-1}}, V_{a_j}$.

Fix some $v \in V_{a_{j-1}}, u \in V_{a_j}$. Notice that all paths between v, u are labeled with exactly 1 output bit. For every $b \in \{0,1\}$, compute the following probability, $p_{v,u,b}$: plug random bits into the ROBP, condition on reaching v, then let $p_{v,u,b}$ be the probability of traversing from v to u and outputting b. Given this probability, draw a *KRVZ sampler* edge from v to u, and give it label b and probability $p_{v,u,b}$. Now do the same for every $v \in V_{a_{j-1}}, u \in V_{a_j}$. Then, repeat this process for all $j \in [n]$.

At the end of the above process, delete all edges that are not KRVZ sampler edges, and delete all vertices that are not in $V_{a_1}, V_{a_2}, \ldots, V_{a_n}$. Thus we obtain a KRVZ sampler of width $2w$. Furthermore, for any $v \in V_{a_{j-1}}, u \in V_{a_j}$, it is straightforward to verify that the probability of transitioning from v to u and outputting any single bit b (conditioned on reaching v in the first place) is the same in the original ROBP and the new KRVZ sampler. Thus we have a KRVZ sampler $\mathcal{B}'$ of width $2w$ that samples $\mathbf{X}$. □

Granular KRVZ sampler $\implies$ ROBP sampler We've arrived at the much harder direction of our equivalence: Lemma 9.4. We warm up with an easier lemma, which is independently useful, and whose proof contains all the intuition needed to ultimately prove Lemma 9.4 (and thus finish the proof of Theorem 9.6).

Lemma 9.5 (Granular KRVZ sampler $\implies$ ROBP sampler, but larger width, less randomness)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a (2^{-t})-granular KRVZ sampler of width w that samples $\mathbf{X}$, then there exists an ROBP of width $4w^2$ and input length $\ell = nt$ that samples $\mathbf{X}$.*

This lemma is similar to Lemma 9.4, except that the ROBP has *larger width* but uses *less randomness*. Since the main parameter we care about when sampling using ROBPs is width, Lemma 9.4 is usually more useful. Still, in some applications, squaring the width is a trivial loss in parameters, and it is preferable to use as little randomness as possible, favoring Lemma 9.5. For this reason, we also record a general version of Lemma 9.5, which can be obtained by combining it with Lemma 9.3 (set $\alpha := 2^{-t}, t := \lceil \log(nw/\varepsilon) \rceil$).

Corollary 9.2 (KRVZ sampler $\implies$ ROBP sampler, but larger width, less randomness)**.** *For any distribution $\mathbf{X} \sim \{0,1\}^n$, if there exists a KRVZ sampler of width w that samples $\mathbf{X}$, then for any $\varepsilon > 0$, there exists an ROBP of width $4w^2$ and input length $\ell = 2n\log(nw/\varepsilon)$ that samples a distribution that is ε-close to $\mathbf{X}$.*

Note this is an alternative version of the second bullet in Theorem 9.6, using more width but less randomness.

The plan now is to prove Lemma 9.5, and then show how we can extend the intuitions developed in this proof to prove Lemma 9.4. In order to prove Lemma 9.5, we will transform a granular KRVZ sampler into an ROBP sampler in a vertex-by-vertex fashion. In particular, we will replace each vertex v in the KRVZ sampler with a small ROBP sampler gadget. The goal of the gadget will roughly be to simulate the edge probabilities coming out of v. The exact gadget that we will use will look something like an (efficient) ROBP that computes a "multi-threshold" function.

To formally define this multi-threshold function, we need a few definitions. First, given distinct strings $x, y \in \{0,1\}^n$, recall the definition of the lexicographic order. In this order, $x < y$ if $x_i < y_i$ at the smallest index $i \in [n]$ where $x_i \neq y_i$. Given this ordering, we define (open and closed) intervals in the natural way. For example, for $x, y \in \{0,1\}^n$, we let $(x, y] := \{s \in \{0,1\}^n : x < s \leq y\}$. It will also be convenient to let $\vec{0} \in \{0,1\}^n$ denote the all zeroes bitstring, $\vec{1} \in \{0,1\}^n$ denote the all ones bitstring, and $-\vec{1}$ denote an imaginary bitstring that is strictly less than all $x \in \{0,1\}^n$. This lets us write $(-\vec{1}, x] = [\vec{0}, x]$ for any x.

We are now ready to define the multi-threshold function. For any bitstring "thresholds" $-\vec{1} = \tau_0 < \tau_1 < \cdots < \tau_t = \vec{1}$, we define the *$t$-threshold function* $f : \{0,1\}^n \to [t]$ over these thresholds to output the label of the "bucket" into which the input falls. Formally, $f(x)$ is defined as the unique $i \in [t]$ such that $x \in (\tau_{i-1}, \tau_i]$. Given this definition, we are ready to state the key ingredient that goes into proving Lemma 9.5, Lemma 9.4, and Theorem 9.6. In particular, we construct a (near-optimal) Σ-ROBP for computing t-threshold functions (recall Definition 9.2 for the definition of Σ-ROBPs).

Lemma 9.6 (Key ingredient for Theorem 9.6)**.** *For any t-threshold function $f : \{0,1\}^n \to [t]$, there exists a Σ-ROBP of width $2t$ that computes f. Furthermore, this is almost tight: There exist many t-threshold functions that cannot be computed in width $< 2t - 1$.*

The tightness of this result will imply that some constant blow-up in width is necessary when simulating KRVZ samplers with ROBP samplers via this gadget. It is natural to ask whether this gadget can be replaced by a different gadget (computing a different function) that only requires width t. We answer this question in the positive in Section 9.6, where the different gadget is used to keep our direct product theorem strong. While the different gadget will have very low width, it will only be able to help us approximately sample distributions. It will therefore be useful for the direct product theorem, but less useful for the exact sampling required by Lemmas 9.4 and 9.5. We will prove Lemma 9.6 at the very end of this section. But first, we show how it can be used to prove Lemmas 9.4 and 9.5.

Proof of Lemma 9.5. We must show that if there is a 2^{-t}-granular KRVZ sampler $\mathcal{B}$ of width w that samples $\mathbf{X} \sim \{0,1\}^n$, then there exists an ROBP of width $4w^2$ and input length $\ell = nt$ that samples $\mathbf{X}$.

Step 1 The first step is just developing a single gadget. Let $\mathcal{A}$ be a 2^{-t}-granular KRVZ sampler of width w for just one bit $\mathbf{Y} \sim \{0,1\}$. Let its underlying graph be $G = (V, E)$ with layers $V = V_0 \cup V_1$. Label the vertices in the last layer $V_1 = \{v_1, \dots, v_w\}$. For each $i \in [w]$ and $b \in \{0,1\}$, let $p_{i,b}$ denote the probability that the KRVZ sampler transitions from its start state to v_i and outputs b. Assume without loss of generality that each $p_{i,b} > 0$ (it is straightforward, but notationally inconvenient, to handle when this is not the case). We would like to construct an ROBP sampler $\mathcal{A}'$ of width $4w$ (and length t) that exactly simulates this.

Since $\mathcal{A}$ is 2^{-t}-granular, we know that each $p_{i,b} = P_{i,b} \cdot 2^{-t}$ for some nonnegative integer $P_{i,b}$. Furthermore, since we must have $\sum_{i,b} p_{i,b} = 1$ it must hold that $\sum_{i,b} P_{i,b} = 2^t$. Recalling our discussion of thresholding functions before Lemma 9.6, it is straightforward to define thresholds in $\{0,1\}^t$

$$-\vec{1} = \tau_\emptyset < \tau_{1,0} < \tau_{2,0} < \cdots < \tau_{w,0} < \tau_{1,1} < \tau_{2,1} < \cdots < \tau_{w,1} = \vec{1}$$

so that for any threshold $\tau_{i,b}$, if we consider the threshold τ' immediately preceding it, then the set $(\tau', \tau_{i,b}] \subseteq \{0,1\}^t$ has exactly $P_{i,b}$ elements.

Now, let $f : \{0,1\}^t \to [2w]$ be the multi-threshold function over the above thresholds. Identify $[2w]$ with the set $[w] \times \{0,1\}$. By definition of our thresholds, note that for any $i \in [w], b \in \{0,1\}$, if we uniformly draw $x \sim \{0,1\}^t$, then the output $f(x) = (i, b)$ with probability $P_{i,b} \cdot 2^{-t} = p_{i,b}$. By Lemma 9.6, there is an ROBP $\mathcal{C}$ of width $4w$ (and length t) that exactly computes this function. We label the nodes in the last layer of this ROBP with the set $\{u_{i,b}\}_{i \in [w], b \in \{0,1\}}$. We can assume without loss of generality that, upon feeding random bits into this ROBP, the computation path reaches $u_{i,b}$ with probability $p_{i,b}$.

Finally, we can construct $\mathcal{A}'$ from $\mathcal{C}$, as follows. Add a final layer consisting of w nodes, which we will call $\{q_i\}_{i \in [w]}$. Now, for every $i \in [w], b \in \{0,1\}$, draw two edges from $u_{i,b}$ to q_i, with input labels $0, 1$ respectively, but both with the same *output* label b. This completes the construction of our ROBP gadget $\mathcal{A}'$. It is straightforward to verify that for any $i \in [w], b \in \{0,1\}$, it holds that if we feed random bits into $\mathcal{A}'$, then we arrive at q_i and output b with probability $p_{i,b}$. Notice that $\mathcal{A}'$ has length $t+1$. In fact, since the transitions into the last layer are trivial, it is easy to redirect edges from the third-to-last layer to bypass the second-to-last layer and give $\mathcal{A}'$ length t.

Step 2 The second step is to use the above gadget to transform our 2^{-t}-granular KRVZ sampler $\mathcal{B}$ into an ROBP $\mathcal{B}'$. Let $G = (V, E)$ be the underlying graph of the KRVZ sampler, with layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. For each $i \in [n]$ and $v \in V_{i-1}$, replace its outgoing edges with a new gadget described above. For any $u \in V_i$ and $b \in \{0,1\}$, note that the probability of traversing from v to u and outputting b remains the same, by the gadget construction. Thus we have constructed an ROBP $\mathcal{B}$ that *exactly* samples the same distribution, as desired. Since we construct a fresh gadget (which has width $4w$) for each vertex in each layer, $\mathcal{B}'$ will have width $w \cdot 4w = 4w^2$. And since each gadget has length t, and we are concatenating n gadgets in a series configuration, $\mathcal{B}$ will have length $\ell = nt$. □

Using the ideas in the above proof, we finally turn towards proving Lemma 9.4.

Proof of Lemma 9.4. We must show that if there is a 2^{-t}-granular KRVZ sampler $\mathcal{B}$ of width w that samples $\mathbf{X} \sim \{0,1\}^n$, then there is an ROBP of width $7w$ and input length $\ell = 4nwt$ that samples $\mathbf{X}$. By the proof of

Lemma 9.5, we know for any 2^{-t}-granular KRVZ sampler $\mathcal{A}$ of width w that samples just 1 bit $\mathbf{Y} \sim \{0,1\}$, there is an ROBP sampler $\mathcal{A}'$ of width $4w$ and length t that exactly samples $\mathbf{Y}$. We call $\mathcal{A}'$ a gadget.

The next step is to use the above gadget to transform the granular KRVZ sampler $\mathcal{B}$ that samples $\mathbf{X} \sim \{0,1\}^n$ into an ROBP $\mathcal{B}'$ that generates the same distribution. Let $G = (V, E)$ be the underlying graph of the KRVZ sampler, with layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. In the proof to Lemma 9.5, we transformed $\mathcal{B}$ into $\mathcal{B}'$ by looking at each boundary V_{i-1}, V_i, and replacing each vertex $v \in V_{i-1}$ and its outgoing edges (and neighbors) with a gadget $\mathcal{A}'$. However, the gadgets corresponding to each $v \in V_{i-1}$ were stacked on top of each other in a *parallel configuration*, meaning that the width of $\mathcal{B}$ was forced to grow by a factor of w. To prevent this from happening, our goal will be to arrange the gadgets corresponding to each $v \in V_{i-1}$ in a *series configuration*.

In more detail, we will transform $\mathcal{B}$ into $\mathcal{B}'$ via a layer-by-layer process as follows. For each $i \in [n]$, consider the boundary between layers V_{i-1} and V_i. We will replace the edges that cross this boundary with a large ROBP $\mathcal{Z}^*$ that consists of three medium-sized ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$ stacked atop one another. We call $\mathcal{S}$ the "source" or "pre-processing" ROBP, we call $\mathcal{W}$ the "working" or "processing" ROBP, and we call $\mathcal{T}$ the "sink" or "post-processing" ROBP. Intuitively, the source ROBP will take inputs coming from each $v \in V_{i-1}$ and keep them in a "holding pattern." Then, the source ROBP will send these inputs into the working ROBP, which is used to simulate the appropriate probabilities coming out of the edges of each $v \in V_{i-1}$ in the KRVZ sampler. In particular, $\mathcal{W}$ will consist of several gadgets of the form $\mathcal{A}'$ arranged in a series configuration. Finally, the working ROBP will send its inputs to the sink ROBP, which will keep its inputs in a holding pattern before finally passing them off to the proper vertices in V_i.

We now formalize the above intuition. The plan is to start by formally describing each of the medium-sized ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$.[13] Then, we will describe how to interface these ROBPs together to create the large ROBP $\mathcal{Z}^*$. Then, we will describe how to interface $\mathcal{Z}^*$ with the layers V_{i-1}, V_i (i.e., replace the edges crossing between V_{i-1}, V_i with $\mathcal{Z}^*$) and argue that this transformation preserves the desired edge probabilities. For convenience, we will label the vertices in V_{i-1} as $u_1, \ldots, u_w$ and the vertices in V_i as $v_1, \ldots, v_w$.

Construction of the source ROBP $\mathcal{S}$ The directed acyclic graph $G_S = (S, E_S)$ underlying this ROBP will consist of $1 + tw$ layers $S = S_0 \cup S_1 \cup \cdots \cup S_{tw}$, each holding w vertices. Label the vertices in S_i as $s_1^{(i)}, \ldots, s_w^{(i)}$. Then, for every $i \in [tw]$ and $j \in [w]$, draw two edges from $s_j^{(i-1)}$ to $s_j^{(i)}$, one of which is given the input label 0 and the other is given the input label 1. This completes the construction of $\mathcal{S}$. Notice that $\mathcal{S}$ should appear as w parallel lines (of length $1 + tw$) drawn atop one another.

Construction of the sink ROBP $\mathcal{T}$ The directed acyclic graph $G_T = (T, E_T)$ underlying this ROBP will consist of $1 + tw$ layers $T = T_1 \cup T_2 \cup \cdots \cup T_{tw+1}$, each holding $2w$ vertices. Label the vertices in T_i as $\{t_{j,b}^{(i)}\}_{j \in [w], b \in \{0,1\}}$. Then, for every $i \in [tw], j \in [w], b \in \{0,1\}$, draw two edges from $t_{j,b}^{(i)}$ to $t_{j,b}^{(i+1)}$, one of which is given the input label 0 and the other is given the input label 1. This completes the construction of $\mathcal{T}$. Notice that $\mathcal{T}$ should appear as $2w$ parallel lines (of length $1 + tw$) drawn atop one another.

Construction of the working ROBP $\mathcal{W}$ The directed acyclic graph $G_W = (W, E_W)$ underlying this ROBP will consist of tw layers $W = W_1 \cup W_2 \cup \cdots \cup W_{tw}$, each holding $4w$ vertices. We break the construction of $\mathcal{W}$ into the construction of w smaller ROBPs $\mathcal{A}_1, \ldots, \mathcal{A}_w$. Each $\mathcal{A}_i$ will have width $4w$ and length t, and they will be arranged in a series configuration (i.e., consecutively) in order to create $\mathcal{W}$.

Each $\mathcal{A}_i$ will be constructed as follows. First, let us return to thinking about the KRVZ sampler, and for every $j \in [w], b \in \{0,1\}$, let $p_{i,j,b}$ be the probability assigned to the edge (u_i, v_j) with label b by the KRVZ

[13] These "ROBPs" will actually just be layered DAGs with edge labels, and won't perfectly match the formal definition of ROBP.

sampler. Using the proof of Lemma 9.5, we construct an ROBP $\mathcal{A}_i$ that has width $4w$ and length t (with the last layer having just $2w$ vertices) such that the following holds: if the vertices in the last layer of $\mathcal{A}_i$ are called $\{a_{j,b}\}_{j\in[w],b\in\{0,1\}}$, then vertex $a_{j,b}$ is hit with probability $p_{i,j,b}$ when a random string $x \in \{0,1\}^t$ is fed as input into $\mathcal{A}_i$.

This completes the construction of $\mathcal{W}$. Notice that $\mathcal{W}$ should appear as w gadgets (each of width $4w$ and length t) $\mathcal{A}_1, \dots, \mathcal{A}_w$ arranged in a series configuration. That is, the start vertex of gadget $\mathcal{A}_i$ will belong to layer $W_{(i-1)t+1}$ and the final layer of $\mathcal{A}_i$ will belong to layer W_{it}.

Combining ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$ into the final ROBP $\mathcal{Z}^*$ The final ROBP $\mathcal{Z}^*$ will be combined by stacking the ROBPs $\mathcal{S}, \mathcal{W}, \mathcal{T}$ one atop another (i.e., arranging them in a parallel configuration). In more detail, the directed acyclic graph $G_Z = (Z, E_Z)$ underlying this ROBP will consist of $tw+2$ layers $Z = Z_0 \cup Z_1 \cup \cdots \cup Z_{tw+1}$. We will have $Z_0 = S_0, Z_1 = S_1 \cup W_1 \cup T_1, Z_2 = S_2 \cup W_2 \cup T_2, \dots, Z_{tw} = S_{tw} \cup W_{tw} \cup T_{tw}, Z_{tw+1} = T_{tw+1}$. Thus, $\mathcal{Z}^*$ has width $w + 4w + 2w = 7w$ and length $tw + 2$.

Next, we add and rearrange a few edges. These modifications will focus on connecting the gadgets $\mathcal{A}_i$ in the working ROBP $\mathcal{W}$ to the source ROBP $\mathcal{S}$ and sink ROBP $\mathcal{T}$. In particular, for each $i \in [w]$, consider the gadget $\mathcal{A}_i$. Define $\beta' < \beta'' \in [tw]$ such that $Z_{\beta'}$ holds the start vertex of $\mathcal{A}_i$ and $Z_{\beta''}$ holds the last layer of $\mathcal{A}_i$. Then, consider the i^{th} vertex in layer $S_{\beta'-1}$ of the source ROBP $\mathcal{S}$. Recall it is called $s_i^{(\beta'-1)}$. Furthermore, recall that it had two edges going into the next layer of the source ROBP $\mathcal{S}$. Delete these edges, and replace them with two edges from $s_i^{(\beta'-1)}$ into the start vertex of $\mathcal{A}_i$, and give them input labels 0 and 1, respectively. This completes the connection of the source ROBP $\mathcal{S}$ to the working ROBP $\mathcal{W}$.

Next, recall that the vertices in the final layer of $\mathcal{A}_i$ are labeled $\{a_{j,b}\}_{j\in[w],b\in\{0,1\}}$, and are located in layer $Z_{\beta''}$. Currently, they have no edges leaving them. Now, for each $j \in [w], b \in \{0,1\}$, draw two edges from $a_{j,b}$ to $t_{j,b}^{(\beta''+1)} \in Z_{\beta''+1}$, and give them input labels 0 and 1, respectively. This completes the connection of the working ROBP $\mathcal{W}$ to the sink ROBP $\mathcal{T}$.

We have therefore completed the connection of source ROBP $\mathcal{S}$ to working ROBP $\mathcal{W}$, and the connection from working ROBP $\mathcal{W}$ to sink ROBP $\mathcal{T}$. Thus we have fully completed the construction of ROBP $\mathcal{Z}^*$. The most important property of $\mathcal{Z}^*$ is as follows: for any $i \in [w]$, if we start at vertex $s_i^{(0)}$ and feed a random string $x \in \{0,1\}^{tw+1}$ as input into the ROBP $\mathcal{Z}^*$, then for any $j \in [w], b \in \{0,1\}$, we arrive at vertex $t_{j,b}^{(tw+1)}$ with probability $p_{i,j,b}$. This is straightforward to verify via the above construction and the guaranteed properties of each gadget $\mathcal{A}_i$.

All that remains now is to interface the ROBP $\mathcal{Z}^*$ with the layers V_{i-1}, V_i.

Inserting ROBP $\mathcal{Z}^*$ between layers V_{i-1}, V_i This is the final and easiest step of our construction. Recall that the vertices in V_{i-1} are labeled as $u_1, \dots, u_w$ and the vertices in V_i are labeled as $v_1, \dots, v_w$. For each $i \in [w]$, we do the following: first, delete the edges leaving u_i in the KRVZ sampler. Then, draw two edges from u_i to $s_i^{(0)}$, and give them input labels 0 and 1, respectively. Then, for each $j \in [w], b \in \{0,1\}$, draw two edges from $t_{j,b}^{(tw+1)}$ to v_j, and give them input labels 0 and 1, respectively, and give them both *output label* b. This completes the interfacing of ROBP $\mathcal{Z}^*$ with layers V_{i-1}, V_i.

It is now straightforward to verify that for any $u \in V_{i-1}$ and $v \in V_i$ and $b \in \{0,1\}$, the probability of transitioning from u to v and outputting b is the same in the original KRVZ sampler as it is in the new ROBP sampler. Thus if we replace the boundary between every pair of layers V_{i-1}, V_i with an appropriate ROBP $\mathcal{Z}^*$ as constructed above, we obtain an ROBP $\mathcal{B}'$ that samples the exact same distribution as the original KRVZ sampler, as desired. Furthermore, each $\mathcal{Z}^*$ used in this construction has width $7w$ and length $tw + 2$. Since the overall ROBP $\mathcal{B}'$ will contain a $\mathcal{Z}^*$ between each consecutive layers V_{i-1}, V_i for $i \in [n]$, the overall ROBP $\mathcal{B}'$ will have length $\ell = n \cdot (tw + 2) + n \leq 4ntw$ and width $7w$, as desired. □

At last, all that remains is to prove our result on computing multi-threshold functions in low width.

Proof of Lemma 9.6. Let $f : \{0,1\}^n \to [t]$ be a t-threshold function over the thresholds

$$-\vec{1} = \tau^0 < \tau^1 < \cdots < \tau^t = \vec{1},$$

where each $\tau^\alpha \in \{0,1\}^t$ (in particular, the superscripts are labels, not powers). We wish to show that there is an ROBP $\mathcal{B}$ of width $2t$ that computes f. To specify $\mathcal{B}$, we must specify its underlying graph $G = (V, E)$, which will have t layers $V = V_0 \cup V_1 \cup \cdots \cup V_t$. We first specify the construction, and then explain why it works.

For each layer $i \in [t]$, we label the nodes $V_i = \{v_i^1, v_i^2, \ldots, v_i^t, \widetilde{v}_i^1, \widetilde{v}_i^2, \ldots, \widetilde{v}_i^t\}$. The nodes of the form $\widetilde{v}_i^\alpha$ can be thought of as "short-circuit" nodes. In particular, for every short circuit node $\widetilde{v}_i^\alpha$, the both edges leaving it (with input labels $0, 1$ respectively) will simply connect to $\widetilde{v}_{i+1}^\alpha$. The edges leaving the nodes of the form v_i^α will be a little more complex. For each $b \in \{0,1\}$, we draw an edge leaving v_i^α into the next layer, and give the edge the input label b, based on the following logic:

- If $b < \tau_{i+1}^\alpha$, connect the edge to $\widetilde{v}_{i+1}^\alpha$.
- If $b = \tau_{i+1}^\alpha$, connect the edge to v_{i+1}^α.
- If $b > \tau_{i+1}^\alpha$, let $\beta > \alpha$ be the smallest integer such that $(\tau_1^\alpha, \ldots, \tau_i^\alpha, b) \leq (\tau_1^\beta, \ldots, \tau_i^\beta, \tau_{i+1}^\beta)$, and:
  - If the above (rightmost) inequality is strict, connect the edge to $\widetilde{v}_{i+1}^\beta$.
  - Otherwise, connect the edge to v_{i+1}^β.

Now, we just need to specify the edges leaving $v_{\mathsf{start}} \in V_0$. For each $b \in \{0,1\}$, we draw an edge from v_{start} into V_1 with the label b using the same logic as the third bullet above. In particular, let β be the smallest integer such that $b \leq \tau_1^\beta$, and: if this inequality is strict, connect the edge to $\widetilde{v}_1^{(\beta)}$; otherwise, connect the edge to v_1^β. Finally, we give each v_n^α and $\widetilde{v}_n^\alpha$ the output label α.

To see why this ROBP computes the threshold function f, consider its computation path as it reads the input $x = (x_1, \ldots, x_t) \in \{0,1\}^t$. Note that as it reads each input bit x_i, it follows the branching instructions described by our itemized list (think of x_i as b). We make a few observations:

1. Suppose that the branching program reaches node v_i^α after reading $x_1, \ldots, x_i$. Then it must hold that $(x_1, \ldots, x_i) = (\tau_1^\alpha, \ldots, \tau_i^\alpha)$ and $x > \tau^{\alpha-1}$. This follows easily by induction on i.
2. Suppose the branching program reaches short circuit node $\widetilde{v}_i^\alpha$ after reading $x_1, \ldots, x_i$. Then it must hold that $\tau^{\alpha-1} < x < \tau^\alpha$. This is straightforward to show using the above observation.

Thus by combining the above observations, if a string x leads to v_n^α or $\widetilde{v}_n^\alpha$, it must hold that $\tau^{\alpha-1} < x \leq \tau^\alpha$.

Now, consider any $x = (x_1, \ldots, x_n)$ that the ROBP will read. By the definition of our thresholds, there must be some α such that $\tau^{\alpha-1} < x \leq \tau^\alpha$. Of course, x must lead to some final state in the branching program. In order to not contradict the above, this state must be either v_n^α or $\widetilde{v}_n^\alpha$, both of which have the output label α. So the ROBP will output α, and therefore compute the multi-threshold function f, as desired. This completes the proof that any t-threshold function f can be computed by a Σ-ROBP of width $2t$.

We now show that many t-threshold functions $f : \{0,1\}^n \to [t]$ cannot be computed in width $< 2t-1$. In particular, pick any thresholds

$$-\vec{1} = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_t = \vec{1}$$

in $\{0,1\}^n$ such that both of the following hold:

- For every $i \in [t-1]$ the last bit of τ_i is 0.
- For every $i \in [t]$ the interval $(\tau_{i-1}, \tau_i]$ contains at least 3 strings.

We will show that for any such thresholds $\tau_0, \tau_1, \ldots, \tau_t$, the corresponding t-threshold function $f : \{0,1\}^n \to [t]$ cannot be computed in width $< 2t-1$.

To see why, consider any Σ-ROBP $\mathcal{B}$ of width $< 2t-1$. Let $g : \{0,1\}^n \to [t]$ denote the function it computes. We will show that $g \neq f$. First, notice that the lower bound on the size of each $(\tau_{i-1}, \tau_i]$ implies that for every $i \in [t]$ there exists some $\tau_{i-1} < \alpha_i < \tau_i$ such that the last bit of α_i is 0. Next, note that since $\mathcal{B}$ has width $< 2t-1$, there must exist two distinct strings $x < y$ in the sequence

$$\alpha_1 < \tau_1 < \alpha_2 < \tau_2 < \cdots < \tau_{t-1} < \alpha_t$$

that lead to the same state in the second-to-last layer of $\mathcal{B}$. Now, define x^0 to be x with its last bit replaced by 0, and x^1 to be x with its last bit replaced by 1. Similarly, define y^0 to be y with its last bit replaced by 0, and y^1 to be y with its last bit replaced by 1. Since the ROBP is agnostic to which of x, y it read once it reaches the second to last layer, we know $g(x^0) = g(y^0)$ and $g(x^1) = g(y^1)$.

Suppose now that there is no $i \in [t]$ such that x^0, y^0 both belong to the interval $(\tau_{i-1}, \tau_i]$. Then by definition of thresholding functions, we clearly have $f(x^0) \neq f(y^0)$, and thus $g \neq f$. Thus assume that there is some $i \in [t]$ such that x^0, y^0 both belong to $(\tau_{i-1}, \tau_i]$. Note that this is only possible if $x = \alpha_i$ and $y = \tau_i$ for some $i \in [t-1]$. But then $\tau_{i-1} < x^1 \leq \tau_i$ and $y^1 > \tau_i$, meaning that there is no $j \in [t]$ such that x^1, y^1 both belong to the interval $(\tau_{j-1}, \tau_j]$. In other words, $f(x^1) \neq f(y^1)$, and thus $g \neq f$, as desired. □

This fully completes the proof of our first main equivalence theorem (Theorem 9.6), which establishes that ROBP samplers and KRVZ samplers are equivalent, up to a very small loss in parameters.

Equivalence theorems between simple samplers and ROBP computers for input-output pairs

We now turn to a completely different kind of equivalence theorem. Namely, we will show that a simple sampler can generate the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ if and only if an ROBP can compute the function b. More formally, we prove the following theorem.

Theorem 9.7 (Equivalence theorem between simple samplers and ROBPs for input-output pairs)**.** *For any function $b : \{0,1\}^n \to \{0,1\}$, there exists a simple sampler of width w that samples $(\mathbf{U}_n, b(\mathbf{U}_n))$ if and only if there exists an ROBP of width w that computes b.*

Notice that one direction of this theorem offers a way to translate sampling lower bounds against simple (and thus KRVZ and ROBP) samplers into hard functions for ROBPs.

Proof of Theorem 9.7. We start by proving that a simple sampler implies an ROBP. Let $\mathcal{B}$ be the simple sampler of width w that can sample $(\mathbf{U}_n, b(\mathbf{U}_n)$, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_{n+1}$. We now define an ROBP $\mathcal{B}'$ that computes b as follows. Define its underlying graph $G' = (V', E')$ to have layers $V'_0 \cup V'_1 \cup \cdots \cup V'_n$, where each V'_i is an exact copy of V_i. Furthermore, for each

$e \in E$ that goes between layers V_{i-1}, V_i for some $i \in [n]$, and which is not assigned probability 0, copy this edge (with its label) into E'. Finally, for all $v' \in V'_n$ whose corresponding vertex in $v \in V_n$ has an outgoing edge labeled 1, label the vertex v' as an *accepting state*.[14]

Now, notice that for every $x \in \{0,1\}^n$ with $b(x) = 1$, the simple sampler $\mathcal{B}$ must output $(x, 1)$ with nonzero probability, and so x must lead to an accepting state in $\mathcal{B}'$. And for every $x \in \{0,1\}^n$ with $b(x) = 0$, it must hold that x does *not* lead to an accepting state in $\mathcal{B}'$, because otherwise this would imply that $\mathcal{B}$ samples $(x, 1) = (x, \neg b(x))$ with nonzero probability (meaning that it does not sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ exactly). Thus $\mathcal{B}'$ computes b, and has width w.

We now prove the reverse direction. Let $\mathcal{B}$ be an ROBP of width w that computes b, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. We define a simple sampler $\mathcal{B}'$ that samples $(\mathbf{U}_n, b(\mathbf{U}_n))$ as follows. Define its underlying graph $G' = (V', E')$ to have layers $V'_0 \cup V'_1 \cup \cdots \cup V'_n \cup V'_{n+1}$, where each V'_i, for $i \in [n]$, is an exact copy of V_i. Furthermore, copy the entire edge set of E into E' (including its labels).

Let V'_{n+1} consist of a single vertex, which we call v^*. For each $v' \in V'_n$, check if the corresponding vertex $v \in V_n$ is labeled v_{accept}: if so, draw an edge from $v' \in V'_n$ to $v^* \in V'_{n+1}$ and label it 1; otherwise, draw the same edge but label it 0. Finally, for each vertex $v' \in V'$, let $p_{v'}$ be the uniform probability distribution over its outgoing edges.

It is straightforward to verify that for any fixed $x \in \{0,1\}^n$, the first n bits produced by the simple sampler $\mathcal{B}'$ are exactly x with probability 2^{-n}, and if this is true then the final bit produced by $\mathcal{B}'$ is $b(x)$ with probability 1. Thus $\mathcal{B}'$ exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n)$ and has width w. □

Theorem 9.7 provides a way to convert worst-case lower bounds against sampling $(\mathbf{U}_n, b(\mathbf{U}_n))$ into worst-case lower bounds against computing b. We now strengthen this direction and provide a way to convert average-case sampling lower bounds into average-case computing lower bounds. For this, we need the definition of the *correlation* between two functions.

Definition 9.8 (Correlation)**.** *The* correlation *between two functions* $f, g : \{0,1\}^n \to \{0,1\}$ *is defined as*

$$\operatorname{corr}(f, g) := \mathbb{E}_x[(-1)^{f(x)}(-1)^{g(x)}] = \Pr_x[f(x) = g(x)] - \Pr_x[f(x) \neq g(x)].$$

With this definition in hand, we now prove the following *average-case* extension of Theorem 9.7.

Theorem 9.8 (Theorem 9.7, extension to correlation bounds)**.** *Fix any function* $b : \{0,1\}^n \to \{0,1\}$*, and suppose that for any simple sampler* $\mathbf{X} \sim \{0,1\}^{n+1}$ *of width* w*, it holds that* $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| > \frac{1-\varepsilon}{2}$*. Then for any ROBP* $f : \{0,1\}^n \to \{0,1\}$ *of width* w*, it holds that* $|\operatorname{corr}(f, b)| < \varepsilon$*.*

Proof. We show the contrapositive: that if there exists an ROBP $\mathcal{B}$ of width w computing a function $f : \{0,1\}^n \to \{0,1\}$ with $|\operatorname{corr}(f, b)| \geq \varepsilon$, then there is a simple sampler $\mathcal{B}'$ of width w sampling a distribution $\mathbf{X} \sim \{0,1\}^n$ such that $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))| \leq \frac{1-\varepsilon}{2}$. We start by assuming, without loss of generality, that $\operatorname{corr}(f, b) \geq \varepsilon$. To see why, simply note that $\operatorname{corr}(f, b) = -\operatorname{corr}(1 - f, b)$ and that if f is computable by a width w ROBP then so is $1 - f$ (by swapping the accept and reject states). Thus if we started with $\operatorname{corr}(f, b) \leq -\varepsilon$, we could instead consider the ROBP computing $f' := 1 - f$ which has $\operatorname{corr}(f', b) \geq \varepsilon$.

So we now assume $\mathcal{B}$ is a width w ROBP computing a function f with $\operatorname{corr}(f, b) \geq \varepsilon$. By definition of correlation, $\operatorname{corr}(f, b) = \Pr[f = b] - \Pr[f \neq b] = 2\Pr[f = b] - 1 \geq \varepsilon$, which implies that $\Pr_{x \sim \mathbf{U}_n}[f(x) = b(x)] \geq \frac{1+\varepsilon}{2}$. We will use this in a moment.

[14] Technically, the definition of ROBP requires a single vertex in V_n to be labeled v_{accept}, but we can easily adjust for this by selecting one of the accepting states to be designated v_{accept}, and redirecting all edges that go into an accepting state to go into v_{accept}, instead.

Now, let $\mathcal{B}'$ be a simple sampler that is constructed from the ROBP $\mathcal{B}$ in the exact same way as in the proof to Theorem 9.7. It is easy to verify that the first n bits produced by $\mathcal{B}'$ are equal to any given $x \in \{0,1\}^n$ with probability 2^{-n}, and if this is true then the final bit produced by $\mathcal{B}'$ is $f(x)$ with probability 1. Thus if $\mathbf{X} \sim \{0,1\}^{n+1}$ is the distribution produced by $\mathcal{B}'$, we have that $|\mathbf{X} - (\mathbf{U}_n, b(\mathbf{U}_n))|$

$$
\begin{aligned}
&= \frac{1}{2} \sum_{x \in \{0,1\}^n, y \in \{0,1\}} |\Pr[\mathbf{X} = (x,y)] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) = (x,y)]| \\
&= \frac{1}{2} \sum_x (|\Pr[\mathbf{X} = (x, b(x))] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) = (x, b(x))]| + \\
&\qquad |\Pr[\mathbf{X} = (x, \neg b(x))] - \Pr[(\mathbf{U}_n, b(\mathbf{U}_n)) = (x, \neg b(x))]|) \\
&= \frac{1}{2} \left(\sum_{x : f(x) = b(x)} (|2^{-n} - 2^{-n}| + |0 - 0|) + \sum_{x : f(x) \neq b(x)} (|0 - 2^{-n}| + |2^{-n} - 0|) \right) \\
&= \frac{1}{2} \sum_{x : f(x) \neq b(x)} 2 \cdot 2^{-n} \\
&= \Pr[f(x) \neq b(x)] = 1 - \Pr[f(x) = b(x)] \\
&\leq 1 - \frac{1+\varepsilon}{2} = \frac{1-\varepsilon}{2}.
\end{aligned}
$$

Thus $\mathcal{B}'$ achieves the claimed sampling bound, and has width w. □

This completes the discussion of our second flavor of equivalence theorem, which shows that sampling the distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ with a simple sampler is equivalent to computing the function b with an ROBP.

Equivalence theorems between simple samplers and ROBP computers for flat distributions

For our third and final flavor of equivalence theorem, we consider distributions $\mathbf{Q} \sim \{0,1\}^n$ uniform over some subset $S \subseteq \{0,1\}^n$. We let $1_S : \{0,1\}^n \to \{0,1\}$ denote the indicator for S, and prove the following.

Theorem 9.9 (Equivalence theorem between simple samplers and ROBPs for flat distributions)**.** *For any distribution $\mathbf{Q} \sim \{0,1\}^n$ that is uniform over some subset $S \subseteq \{0,1\}^n$:*

- *If there exists a simple sampler of width w that samples $\mathbf{Q}$, then there exists an ROBP of width $w+1$ that computes $1_S : \{0,1\}^n \to \{0,1\}$.*
- *If there exists an ROBP of width w that computes $1_S : \{0,1\}^n \to \{0,1\}$, then there exists a simple sampler of width w that samples $\mathbf{Q}$.*

The second bullet of the above theorem is quite useful: for example, it gives an easy way to sample from affine spaces (by computing parity checks in low-width), and furthermore it offers a way to translate lower bounds against simple samplers (and thus KRVZ and ROBP) samplers into hard functions for ROBPs.

Proof of Theorem 9.9. We start by proving that a simple sampler implies an ROBP. Let $\mathcal{B}$ be the simple sampler of width w that can sample $\mathbf{Q}$, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. We now define an ROBP $\mathcal{B}'$ that computes 1_S as follows. Define its underlying graph $G' = (V', E')$ to have layers $V_0' \cup V_1' \cup \cdots \cup V_n'$, where each V_i' is an exact copy of V_i. Furthermore, copy each edge $e \in E$, which is not assigned probability 0, into E' (with its label).

Now, for each $i \in [n]$, we add an additional vertex to V'_i, which we call $\widetilde{v}_i$. And for every $i \in [n-1]$, draw two edges from $\widetilde{v}_i$ to $\widetilde{v}_{i+1}$: one labeled with 0 and the other labeled 1. Intuitively, this new row of vertices might be considered as the "reject gutter." Next, for each $i \in [n]$ and $v' \in V'_{i-1}$, if v' has no outgoing edge labeled 0, draw an edge from v' to $\widetilde{v}_i$ and label it 0. And if v' has no outgoing edge labeled 1, draw an edge from v' to $\widetilde{v}_i$ and label it 1. Finally, for all $v' \in V'_n$ *except* $\widetilde{v}_n$, label v' as an accepting state.

Now, notice that for every $x \in \{0,1\}^n$ such that $1_S(x) = 1$, it holds by definition that $x \in \text{support}(\mathbf{Q})$, which means that the simple sampler $\mathcal{B}$ of course outputs x with nonzero probability. Thus, x must lead to an accepting state in $\mathcal{B}'$. And for every $x \in \{0,1\}^n$ such that $1_S(x) = 0$, it holds by definition that $x \notin \text{support}(\mathbf{Q})$, which means that the simple sampler $\mathcal{B}$ of course outputs x with zero probability. This means that the unique path in $\mathcal{B}$ labeled with x must have some edge assigned zero probability, which means that x must enter the "reject gutter" at some point in $\mathcal{B}'$, and ultimately arrive at $\widetilde{v}_n$, the only reject state in V_n. Thus $\mathcal{B}'$ computes 1_S, and has width $w+1$.

We now prove the reverse direction. Let $\mathcal{B}$ be an ROBP of width w that computes 1_S, with underlying graph $G = (V, E)$ and layers $V = V_0 \cup V_1 \cup \cdots \cup V_n$. We define a simple sampler $\mathcal{B}'$ that samples $\mathbf{Q}$ as follows. First, set its underlying graph to be identical to $G = (V, E)$.

Now, for each $v' \in V'$, let $P(v')$ count the number of paths from v' to an accept state. In particular, $P(v') = 0$ for vertices in the last layer that are not accept states. Then, we assign edge transition probabilities as follows. For any $u' \in V'$ with $P(u') = 0$, assign any arbitrary distribution over its outgoing edges (it won't matter). For all other edges (u', v'), assign it probability $P(v')/P(u')$. Note that this is indeed a valid probability distribution: suppose u' has outgoing edges to v' and w': then it will always be true that $P(u') = P(v') + P(w')$, and thus the probabilities we assigned over its outgoing edges, namely $P(v')/P(u')$ and $P(w')/P(u')$, must add up to 1.

Suppose now that x is not accepted by $\mathcal{B}$. Then the last vertex v' on its computation path in $\mathcal{B}$ will not be an accept state. Thus $P(v') = 0$, which means that the probability on the last edge before hitting v' is 0, and thus the overall probability assigned to this path in $\mathcal{B}'$ is 0.

Suppose now that x *is* accepted by $\mathcal{B}$, and that its computation path uses edges $e_1 = (v_0, v_1), e_2 = (v_1, v_2), \ldots, e_n = (v_1, v_n)$. Then of course $P(v_i) > 0$ for each vertex on this path, and the overall probability of the path is

$$\frac{P(v_1)}{P(v_0)} \cdot \frac{P(v_2)}{P(v_1)} \cdots \frac{P(v_n)}{P(v_{n-1})} = \frac{P(v_n)}{P(v_0)}.$$

But $P(v_n)$ is just 1, and v_0 must be the start vertex of the program, so $P(v_0)$ must be exactly the number of strings accepted by $\mathcal{B}$. Thus our simple sampler $\mathcal{B}'$ samples each accepting string of $\mathcal{B}$ with the same probability $1/P(v_0)$. In other words, the distribution it outputs is uniform over $1_S^{-1}(1) = S$, meaning that it exactly samples $\mathbf{Q}$. □

Theorem 9.9 shows how to convert worst-case lower bounds against sampling a flat distribution $\mathbf{Q} \sim \{0,1\}^n$ into worst-case lower bounds against computing the indicator function 1_S of its support. We now strengthen this direction and show how to convert average-case sampling lower bounds into covariance bounds. Towards this end, we define the *bias* of a function $f : \{0,1\}^n \to \{0,1\}$ as $\text{bias}(f) := \mathbb{E}_x[(-1)^{f(x)}]$, and provide the definition of covariance, below.

Definition 9.9 (Covariance)**.** *The* covariance *between two functions $f, g : \{0,1\}^n \to \{0,1\}$ is defined as*

$$\text{cov}(f,g) := \text{corr}(f,g) - \text{bias}(f)\,\text{bias}(g),$$

Note that we must use the more general notion of covariance (instead of correlation, as in Theorem 9.8) since it is possible to get strong sampling lower bounds against $\mathbf{Q}$ even if 1_S is very biased. With this in mind, let us proceed to prove our average-case version of Theorem 9.9.

Theorem 9.10 (Theorem 9.9, extension to covariance bounds)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be any distribution that is uniform over some subset $S \subseteq \{0,1\}^n$, and suppose that for any simple sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, it holds that $|\mathbf{X} - \mathbf{Q}| > 1 - \frac{\varepsilon}{4}$. Then for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width w, it holds that $|\operatorname{cov}(f, 1_S)| < \varepsilon$.*

Proof. We show the contrapositive: that if there exists an ROBP $\mathcal{B}$ of width w computing $f : \{0,1\}^n \to \{0,1\}$ such that $|\operatorname{cov}(f, 1_S)| \geq \varepsilon$, then there is a simple sampler $\mathcal{B}'$ of width w sampling a distribution $\mathbf{X} \sim \{0,1\}^n$ such that $|\mathbf{X} - \mathbf{Q}| \leq 1 - \varepsilon/4$. We start by assuming, without loss of generality, that $\operatorname{cov}(f, 1_S) \geq \varepsilon$. We can do this because $\operatorname{cov}(f, 1_S) = -\operatorname{cov}(1 - f, 1_S)$, and because $1 - f$ must also be computable by an ROBP of width w (by swapping the accept and reject states).

So now we assume $\mathcal{B}$ is a width w ROBP computing a function f with $\operatorname{cov}(f, 1_S) \geq \varepsilon$. Let $\mathcal{B}'$ be a simple sampler that is constructed from the ROBP $\mathcal{B}$ in the exact same way as in the proof to Theorem 9.9. We know it outputs a distribution $\mathbf{X} \sim \{0,1\}^n$ that is uniform over $f^{-1}(1)$. We want to show that $|\mathbf{X} - \mathbf{Q}| \leq 1 - \varepsilon/4$.

It will now be notationally convenient to define the following quantities, taking uniform $x \sim \{0,1\}^n$:

$$\begin{aligned} a &:= \Pr[f(x) = 1] \\ b &:= \Pr[1_S(x) = 1] \\ c &:= \Pr[f(x) = 1 \text{ and } 1_S(x) = 1]. \end{aligned}$$

Without loss of generality, we may assume both $a, b > 0$. Now, by definition of covariance, we have $\operatorname{cov}(f, 1_S) = \operatorname{corr}(f, 1_S) - \operatorname{bias}(f)\operatorname{bias}(1_S)$, and using the definitions of correlation and bias, it is a straightforward calculation to obtain

$$\operatorname{cov}(f, 1_S) = 4c - 4ab.$$

Now, notice that for any $x \in \operatorname{support}(\mathbf{X})$, it holds that $\Pr[\mathbf{X} = x] = 2^{-n}/a$. Similarly, for any $q \in \operatorname{support}(\mathbf{Q})$, it holds that $\Pr[\mathbf{Q} = q] = 2^{-n}/b$. So using the (half ℓ_1 norm) definition of statistical distance, it is straightforward to compute:

$$\begin{aligned} 2 \cdot |\mathbf{X} - \mathbf{Q}| &= \frac{1}{a} \cdot \Pr_x[f(x) = 1, 1_S(x) = 0] + \frac{1}{b} \cdot \Pr_x[f(x) = 0, 1_S(x) = 1] \\ &\quad + |\frac{1}{a} - \frac{1}{b}| \cdot \Pr[f(x) = 1, 1_S(x) = 1] \\ &= (1/a)(a - c) + (1/b)(b - c) + |1/a - 1/b| \cdot c. \end{aligned}$$

Without loss of generality assume $1/a \geq 1/b$, and notice this quantity is $2 - 2c/b$. Thus we have:

$$\begin{aligned} |\mathbf{X} - \mathbf{Q}| &= 1 - c/b, \\ \operatorname{cov}(f, 1_S) &= 4(c - ab). \end{aligned}$$

Notice we have $c/b \geq c \geq c - ab$. Thus $c/b \geq \operatorname{cov}(f, 1_S)/4$, and thus

$$|\mathbf{X} - \mathbf{Q}| = 1 - c/b \leq 1 - \operatorname{cov}(f, 1_S)/4 \leq 1 - \varepsilon/4,$$

as desired. □

With our equivalence theorems in hand, we are ready to go out and explore sampling with limited memory.

9.4 The basics

As it turns out, many natural questions about sampling with ROBPs can be answered using easy-to-prove results from *communication complexity*. Using this connection, it is straightforward to prove that sampling is easer than computing for ROBPs (Theorem 9.1), and to obtain sampling lower bounds against input-output pairs (Theorem 9.2) and general distributions (Theorem 9.3). In this part of the chapter, we focus on proving these three basic results, and do so in Section 9.4.1, Section 9.4.2, and Section 9.4.3, respectively. Afterward, we will have built a strong enough foundation for the *space complexity of sampling*, and will be ready to prove the two main contributions of this chapter (Theorems 9.4 and 9.5). But before we proceed with all of this, let us start by reviewing the connection between ROBPs and communication protocols, and see why communication lower bounds imply lower bounds against ROBPs.

Two-party communication protocols can simulate ROBP computers It is well known that ROBPs of width 2^s can be simulated by two-party protocols that communicate s bits [RY20]. In fact, *one-way* communication protocols suffice. In a one-way two-party communication protocol, Alice and Bob are given inputs $x \in \{0,1\}^a$ and $y \in \{0,1\}^b$, respectively, and they try to compute some function $f : \{0,1\}^a \times \{0,1\}^b \to \{0,1\}$. Alice is allowed to send a single message to Bob, after which Bob must immediately determine the value $f(x, y)$, using just his input y and the message from Alice.

Observe that for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width 2^s, and any a, b that sum to n, it holds that a one-way communication protocol (where Alice is given a bits, Bob is given b bits, and s bits are communicated) can exactly compute f. To see why, simply note that Alice can use her a bits to simulate the first a steps of the ROBP, send the state at which she arrives to Bob (using s bits), which he can then use to complete the computation.

Thus, if a function $g : \{0,1\}^n \to \{0,1\}$ cannot be computed by one-way protocols that communicate s bits, then it also cannot be computed by ROBPs of width 2^s. More generally, if all one-way protocols over s bits fail to compute g on $1/2 - \varepsilon$ fraction of its inputs, the same must be true for all ROBPs of width 2^s.

Two-party communication protocols can simulate ROBP samplers Analogously, it is not difficult to show that if a distribution $\mathbf{Q} \sim \{0,1\}^n$ can be sampled by an ROBP of width 2^s, then it can also be sampled by a two-party communication protocol that communicates $s+1$ bits. In such a protocol [AST$^+$03, GW20], Alice is given private randomness $\mathbf{A}$, Bob is given private randomness $\mathbf{B}$, and they communicate (back and forth) $s + 1$ bits about their private randomness to one another. At the end of the protocol, Alice outputs some $\mathbf{X} \sim \{0,1\}^{n/2}$, Bob outputs some $\mathbf{Y} \sim \{0,1\}^{n/2}$, and the overall output is defined as $(\mathbf{X}, \mathbf{Y})$.[15]

To see why such communication protocols can sample the same distributions as ROBPs, first recall that any distribution $\mathbf{Q} \sim \{0,1\}^n$ sampled by a width 2^s ROBP can also be sampled by a KRVZ sampler of width 2^{s+1}, as per Theorem 9.6. It is then easy to simulate a KRVZ sampler using two-party protocols: Alice can use her randomness $\mathbf{A}$ to simulate the first $n/2$ steps of the random walk over the KRVZ sampler, send the state at which she arrives to Bob (using $s + 1$) bits, at which point Bob can simulate the remaining $n/2$ steps of the random walk using his randomness $\mathbf{B}$. At the end, Alice can output the bits she saw on her random walk, and Bob can do the same.

Thus, if a distribution $\mathbf{Q} \sim \{0,1\}^n$ cannot be sampled by two-party protocols that communicate $s + 1$ bits, then it also cannot be sampled by ROBPs of width 2^s. More generally, if any distribution $\mathbf{Q}' \sim \{0,1\}^n$ sampled by such a protocol has statistical distance $\geq \delta$ from $\mathbf{Q}$, then the same must be true for any distribution generated by an ROBP of width 2^s.

[15] For a more formal description of sampling with two-party communication protocols, see Section 9.8.

9.4.1 Sampling is easier than computing

Using the above connection, it is not hard to prove that sampling is easier than computing for ROBPs (Theorem 9.1). As a reminder, this theorem asserts that there is some explicit function $b : \{0,1\}^n \to \{0,1\}$ such that ROBPs of exponential width cannot compute b (even on average), yet they can easily sample $(\mathbf{U}_n, b(\mathbf{U}_n))$ using just linear width. The latter is straightforward to show, and the former follows immediately from known communication lower bounds. We thank an anonymous reviewer for a concise proof of these communication lower bounds, which is included and slightly optimized in the proof below.

Theorem 9.11 (Sampling is easier than computing - Theorem 9.1, restated)**.** *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any $\varepsilon > 0$ and ROBP $F : \{0,1\}^n \to \{0,1\}$ of width at most $2^{\frac{n\varepsilon^2}{9} - \log(1/\varepsilon)}$,*

$$\Pr_x[F(x) = b(x)] < \frac{1}{2} + \varepsilon,$$

but there exists an ROBP $G : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $2n$ and length $\ell \le n + \log n + 2$ such that

$$G(\mathbf{U}_\ell) = (\mathbf{U}_n, b(\mathbf{U}_n)).$$

Proof. Let $k \in \mathbb{N}$ be any power of 2, and define $n := k + \log k$. We take $b : \{0,1\}^n \to \{0,1\}$ as the well-known *address* (or *index*) function $\mathsf{address} : \{0,1\}^k \times [k] \to \{0,1\}$, defined as

$$\mathsf{address}(x, i) := x_i.$$

Upper bounds for sampling We first prove the second part of the theorem. Towards this end, define $\mathbf{X} \sim \{0,1\}^k$ and $\mathbf{Y} \sim [k]$ as independent random variables that are uniform over their respective domain, and note that $(\mathbf{U}_n, b(\mathbf{U}_n)) = (\mathbf{X}, \mathbf{Y}, \mathbf{X}_\mathbf{Y})$. Now, for any $y \in [k]$ and $b \in \{0,1\}$, define the random variable $\mathbf{X}^{y \leftarrow b} \sim \{0,1\}^k$ to have all of its bits independent and uniform, except for the y^{th} bit, which is fixed to b. Furthermore, define the random variable $\mathbf{Y}^{y \leftarrow b} := (\mathbf{X}^{y \leftarrow b}, y, b)$. It is straightforward to verify that $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be written as the following convex combination:

$$(\mathbf{U}_n, b(\mathbf{U}_n)) = (\mathbf{X}, \mathbf{Y}, \mathbf{X}_\mathbf{Y}) = \sum_{y \in [k], b \in \{0,1\}} \frac{1}{2k} \cdot \mathbf{Y}^{y \leftarrow b}. \tag{9.1}$$

It is not difficult to show that $\mathbf{Y}^{y \leftarrow b}$ can be sampled by an ROBP of width 1 and length $k + \log k + 1 = n + 1$. Let $\mathcal{B}^{y \leftarrow b}$ be the ROBP that samples it. Then, to sample the entire convex combination written above, we can create an ROBP $\mathcal{B}$ as follows. First, construct a complete binary tree of depth d satisfying $2^d = 2k$ (recall that k is a power of 2). For any node in the tree, assign its outgoing edges the input labels $0, 1$, and assign to them no output labels. Identify the $2k$ leaves of the tree with the set $[k] \times \{0,1\}$, and for each $y \in [k], b \in \{0,1\}$, attach $\mathcal{B}^{y \leftarrow b}$ to the leaf (y, b): more formally, identify leaf (y, b) with the start vertex of $\mathcal{B}^{y \leftarrow b}$. This completes the construction of $\mathcal{B}$.

It is straightforward to verify that $\mathcal{B}$ exactly samples the convex combination from Equation (9.1), and thus it exactly samples $(\mathbf{U}_n, b(\mathbf{U}_n))$. Furthermore, it has width $2k \le 2n = O(n)$ and length $\ell = d + (n + 1) = \log(2k) + n + 1 \le n + \log n + 2 = O(n)$, as desired.

Lower bounds for computing We now prove the first part of the theorem, using a slightly optimized version of a proof suggested by an anonymous reviewer. This part follows immediately from known lower bounds on computing the address function using one-way communication protocols. We prove these lower bounds, below.

Suppose there is a one-way communication protocol that computes a function $f : \{0,1\}^k \times [k] \to \{0,1\}$ such that $f(x,i) = \mathsf{address}(x,i)$ on at least $\frac{1}{2} + \varepsilon$ fraction of the possible pairs x, i. In other words,

$$\Pr_{\substack{x\sim\{0,1\}^k\\ i\sim[k]}}[f(x,i) = \mathsf{address}(x,i)] \geq \frac{1}{2} + \varepsilon.$$

Furthermore, suppose that in this protocol, Alice is given $x \in \{0,1\}^k$ and Bob is given $i \in [k]$. Let s be the maximum number of bits that Alice sends to Bob on any input. The goal is to lower bound s.

First, note that Alice's message to Bob will always be in the set $\mathcal{M} := \{0,1\} \cup \{0,1\}^2 \cup \cdots \cup \{0,1\}^s$, and thus there are $< 2^{s+1}$ possible messages she could send to Bob. For each message $m \in \mathcal{M}$, let $S_m \subseteq \{0,1\}^k$ denote the set of all strings that cause Alice to send m to Bob. Note that $\{S_m\}_{m\in\mathcal{M}}$ partitions Alice's input space $\{0,1\}^k$.

Now, consider any $m \in \mathcal{M}$ such that

$$\Pr_{\substack{x\sim S_m\\ i\sim[k]}}[f(x,i) = \mathsf{address}(x,i)] > \frac{1+\varepsilon}{2}. \tag{9.2}$$

We will argue that $t := |S_m|$ must be quite small. To see why, first notice that by definition of one-way communication protocols, there must be some deterministic function $g : [k] \to \{0,1\}$ such that $f(x,i) = g(i)$ for all $x \in S_m$: this is because Bob's output only depends on the message he receives and his own input i, and all inputs $x \in S_m$ end up in Bob receiving the same message. Thus we have

$$\Pr_{\substack{x\sim S_m\\ i\sim[k]}}[g(i) = \mathsf{address}(x,i)] > \frac{1+\varepsilon}{2}. \tag{9.3}$$

Now, define $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_k) \sim \{0,1\}^k$ to be uniform over S_m, and define for each $i \in [k]$ the value

$$\mathsf{bias}(\mathbf{Y}_i) := |\Pr[\mathbf{Y}_i = 1] - \Pr[\mathbf{Y}_i = 0]|.$$

For any fixed i, it is easy to verify that $\Pr_{x\sim S_m}[g(i) = \mathsf{address}(x,i)] \leq (1 + \mathsf{bias}(\mathbf{Y}_i))/2$, since $g(i)$ is a fixed value. Thus we have

$$\frac{1+\varepsilon}{2} < \Pr_{\substack{x\sim S_m\\ i\sim[k]}}[g(i) = \mathsf{address}(x,i)] \leq \sum_{i\in[k]} \frac{1}{k} \cdot \frac{1 + \mathsf{bias}(\mathbf{Y}_i)}{2} = \frac{1 + \sum_i \mathsf{bias}(\mathbf{Y}_i)/k}{2},$$

which implies

$$\sum_{i\in[k]} \mathsf{bias}(\mathbf{Y}_i) > k\varepsilon. \tag{9.4}$$

Now, letting $H(\cdot)$ denote Shannon entropy, it is a straightforward calculation to show that for any random variable $\mathbf{Z} \sim \{0,1\}$, it holds that $H(\mathbf{Z}) \leq 1 - \mathsf{bias}(\mathbf{Z})^2/2$. Using this fact, we have

$$\log t = H(\mathbf{Y}) \leq \sum_{i\in[k]} H(\mathbf{Y}_i) \leq \sum_{i\in[k]} \left(1 - \frac{\mathsf{bias}(\mathbf{Y}_i)^2}{2}\right) = k - \frac{1}{2}\sum_{i\in[k]} \mathsf{bias}(\mathbf{Y}_i)^2. \tag{9.5}$$

Now, note that by Cauchy-Schwarz, it holds that $(\sum_i \mathsf{bias}(\mathbf{Y}_i))^2 \leq k \sum_i \mathsf{bias}(\mathbf{Y}_i)^2$. Combining this with Equations (9.4) and (9.5), we have:

$$\log t \leq k - \frac{1}{2} \sum_{i \in [k]} \mathsf{bias}(\mathbf{Y}_i)^2 \leq k - \frac{1}{2k} \left(\sum_i \mathsf{bias}(\mathbf{Y}_i) \right)^2 < k - \frac{1}{2k} (k\varepsilon)^2 = k \cdot (1 - \varepsilon^2/2).$$

At last, we can conclude that for any $m \in \mathcal{M}$ such that Equation (9.2) holds, it must also be true that $|S_m| = t < 2^{k \cdot (1-\varepsilon^2/2)}$. Now, let $p_m := \Pr_{x \sim S_m, i \sim [k]}[f(x,i) = \mathsf{address}(x,i)]$, and observe that

$$\begin{aligned}
\frac{1}{2} + \varepsilon \leq & \Pr_{x \sim \{0,1\}^k, i \sim [k]} [f(x,i) = \mathsf{address}(x,i)] \\
&= \sum_{m \in \mathcal{M}} p_m \cdot \Pr_{x \sim \{0,1\}^k} [x \in S_m] \\
&= \sum_{m \in \mathcal{M} : p_m \leq \frac{1+\varepsilon}{2}} p_m \cdot \Pr_{x \sim \{0,1\}^k} [x \in S_m] + \sum_{m \in \mathcal{M} : p_m > \frac{1+\varepsilon}{2}} p_m \cdot \Pr_{x \sim \{0,1\}^k} [x \in S_m] \\
&\leq \frac{1+\varepsilon}{2} \cdot \sum_{m \in \mathcal{M} : p_m \leq \frac{1+\varepsilon}{2}} \Pr_{x \sim \{0,1\}^k} [x \in S_m] + \sum_{m \in \mathcal{M} : p_m > \frac{1+\varepsilon}{2}} 1 \cdot 2^{-k} \cdot 2^{k \cdot (1-\varepsilon^2/2)}. \\
&\leq \frac{1+\varepsilon}{2} \cdot 1 + |\mathcal{M}| \cdot 2^{-k\varepsilon^2/2} \\
&< \frac{1+\varepsilon}{2} + 2^{s+1-k\varepsilon^2/2},
\end{aligned}$$

which implies that

$$s > \frac{k\varepsilon^2}{2} - \log(1/\varepsilon) - 2.$$

Thus, we can finally conclude that any one-way communication protocol that computes address on $\geq \frac{1}{2} + \varepsilon$ of its inputs must use $s > \frac{k\varepsilon^2}{2} - \log(1/\varepsilon) - 2 \geq \frac{n\varepsilon^2}{3} - \log(1/\varepsilon) - 2$ bits of communication. In other words, for any function $f : \{0,1\}^k \times [k] \to \{0,1\}$ computed by a one-way communication protocol over $s \leq \frac{n\varepsilon^2}{3} - \log(1/\varepsilon) - 2$ bits, it holds that

$$\Pr_{x,i}[f(x,i) = \mathsf{address}(x,i)] < \frac{1}{2} + \varepsilon. \tag{9.6}$$

By our discussion at the beginning of this section, it immediately follows that for any ROBP $f : \{0,1\}^n \to \{0,1\}$ of width $2^s \leq 2^{\frac{n\varepsilon^2}{3} - \log(1/\varepsilon) - 2}$, Equation (9.6) also holds. Furthermore, this must also hold for any $2^s \leq 2^{\frac{n\varepsilon^2}{9} - \log(1/\varepsilon)}$: this trivially holds when $n\varepsilon^2/9 < 1$ (any ROBP of width $2^s < 2$ must have width at most 1, and such ROBPs can only compute the address function with probability half), and when $n\varepsilon^2/9 \geq 1$ we have $2^{\frac{n\varepsilon^2}{9} - \log(1/\varepsilon)} \leq 2^{\frac{n\varepsilon^2}{3} - \log(1/\varepsilon) - 2}$. This completes the proof. □

9.4.2 Sampling lower bounds against input-output pairs

In the previous section, we gave an explicit function $b : \{0,1\}^n \to \{0,1\}$ that is hard to compute (even on average) for ROBPs of exponential width, but such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is *easy to sample* for ROBPs of just linear width. Next, we show how to find an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is *hard to sample* for ROBPs, thereby proving Theorem 9.2.

Theorem 9.12 (Sampling lower bounds against input-output pairs - Theorem 9.2, restated)**.** *There exists an explicit function $b : \{0,1\}^n \to \{0,1\}$ such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width at most $2^{n/16}$,*

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, b(\mathbf{U}_n))| \geq \frac{1}{2} - 128 \cdot 2^{-n/16}.$$

In order to prove this result, the main ingredient we use is an explicit two-source extractor (Part I). First, we show via the following lemma that $(\mathbf{U}_n, b(\mathbf{U}_n))$ is hard to sample for two-party communication protocols when b is a good enough two-source extractor. Then, we instantiate this lemma with the classical *inner product (mod* 2*)* two-source extractor [CG88], and use the connection between ROBPs and communication protocols to obtain Theorem 9.12.

Lemma 9.7. *Let $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a two-source extractor for min-entropy k with error ε. Then for any distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^{n+1}$ sampled by a two-party communication protocol with s bits of communication, it holds that*

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq \frac{1}{2} - \varepsilon - 2^{s-n+k+5}$$

Proof. The distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^{2n+1}$ sampled by Alice and Bob is of the form $(\mathbf{A}, \mathbf{B}, \mathbf{b}) \sim \{0,1\}^n \times \{0,1\}^n \times \{0,1\}$, where $\mathbf{A}$ are the bits output by Alice, $(\mathbf{B}, \mathbf{b})$ are the bits output by Bob, and these components are not necessarily independent. However, it is well-known [AST$^+$03] that if we fix the communication transcript between Alice and Bob, we can write $(\mathbf{X}, \mathbf{Y})$ as a convex combination of the form

$$(\mathbf{X}, \mathbf{Y}) = \sum_{i \in \mathcal{M}} p_i \cdot (\mathbf{A}^{(i)}, \mathbf{B}^{(i)}, \mathbf{b}^{(i)}),$$

where i runs over all possible message transcripts $\mathcal{M}$, and each $\mathbf{A}^{(i)} \sim \{0,1\}^n$ is independent from each $(\mathbf{B}^{(i)}, \mathbf{b}^{(i)}) \sim \{0,1\}^n \times \{0,1\}$. Consider now fixing each $\mathbf{b}^{(i)}$ to some $b \in \{0,1\}$. Then we can write

$$(\mathbf{X}, \mathbf{Y}) = \sum_{i \in \mathcal{M}, b \in \{0,1\}} p_{i,b} \cdot (\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b), \tag{9.7}$$

for some probabilities $p_{i,b}$. Notice also that $\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}$ remain independent.

Now, by definition of statistical distance, it holds that for any test $S \subseteq \{0,1\}^{2n+1}$,

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| \geq \Pr[(\mathbf{X}, \mathbf{Y}) \in S] - \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S].$$

We aim to construct a test S by keeping the abovementioned convex combination in mind. Towards this end, we start by letting t be a parameter that we will fix later. For each $i \in \mathcal{M}$ and $b \in \{0,1\}$, we define

$$\begin{aligned}
\mathsf{Bad}_A^{(i,b)} &:= \{x \in \{0,1\}^n : \Pr[\mathbf{A}^{(i,b)} = x] > 2^{-t}\},\\
\mathsf{Bad}_B^{(i,b)} &:= \{y \in \{0,1\}^n : \Pr[\mathbf{B}^{(i,b)} = y] > 2^{-t}\},\\
\mathsf{Bad}^{(i,b)} &:= \left\{(x, y, b) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\} : x \in \mathsf{Bad}_A^{(i,b)} \text{ or } y \in \mathsf{Bad}_B^{(i,b)}\right\}\\
\mathsf{Bad} &:= \bigcup_{i \in \mathcal{M}, b \in \{0,1\}} \mathsf{Bad}^{(i,b)}
\end{aligned}$$

Furthermore, we define the set

$$T := \{(x, y, b) \in \{0,1\}^n \times \{0,1\}^n \times \{0,1\} : \mathsf{Ext}(x, y) \neq b\}.$$

We are ready to define our test set $S \subseteq \{0,1\}^{2n+1}$ as:

$$S := T \cup \mathsf{Bad}.$$

The goal now is to lower bound $\Pr[(\mathbf{X}, \mathbf{Y}) \in S]$, and upper bound $\Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S]$. We start with the latter, as it is easier. Notice that it is impossible for $(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))$ to land in T, so we just have to worry about it landing in Bad. Notice also that each bad set of the form $\mathsf{Bad}_A^{(i,b)}$ or $\mathsf{Bad}_B^{(i,b)}$ has $< 2^t$ elements, or else it contradicts the definition of probability distribution. Thus for every $i \in \mathcal{M}, b \in \{0,1\}$,

$$\begin{aligned} |\mathsf{Bad}^{(i,b)}| &\leq |\mathsf{Bad}_A^{(i,b)} \times \{0,1\}^n \times \{0,1\}| + |\{0,1\}^n \times \mathsf{Bad}_B^{(i,b)} \times \{0,1\}| \\ &< 2^{t+n+1} + 2^{n+t+1} \\ &= 2^{t+n+2}. \end{aligned}$$

Since the set of messages $\mathcal{M} \subseteq \{0,1\} \times \{0,1\}^2 \times \cdots \times \{0,1\}^s$ has size $< 2^{s+1}$, we get

$$|\mathsf{Bad}| < 2^{s+1} \cdot 2 \cdot 2^{t+n+2} = 2^{s+t+n+4}.$$

Now notice that $(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))$ assigns each element in its support a probability of 2^{-2n}, so we have

$$\Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S] = \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in \mathsf{Bad}] \leq 2^{-2n} \cdot |\mathsf{Bad}| < 2^{-n+s+t+4}. \tag{9.8}$$

We now move towards lower bounding $\Pr[(\mathbf{X}, \mathbf{Y}) \in S]$. By Equation (9.7), it suffices to lower bound $\Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S]$ for every i, b. Now, let

$$p := \Pr[\mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)} \text{ and } \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}]$$

and note that we can rewrite $\Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S]$ as

$$\begin{aligned} &p \cdot \Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S \mid \mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)} \text{ and } \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}] \\ &+ (1-p) \cdot \Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S \mid \mathbf{A}^{(i,b)} \in \mathsf{Bad}_A^{(i,b)} \text{ or } \mathbf{B}^{(i,b)} \in \mathsf{Bad}_B^{(i,b)}] \end{aligned}$$

Notice that the probability attached to $(1-p)$ will always be 1, by our construction of S. On the other hand, in the term attached to p, we can replace S with T since $(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b)$ will never hit Bad in this conditioning. And the probability that $(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in T$ is, by definition of T, the probability that $\mathsf{Ext}(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}) \neq b$. Thus we can rewrite the above expression as

$$p \cdot \Pr[\mathsf{Ext}(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}) \neq b \mid \mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)} \text{ and } \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}] + (1-p).$$

Notice now that since $\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}$ were originally independent, the conditionings above keep them independent. In particular, we can define independent random variables $\widetilde{\mathbf{A}}^{(i,b)} := (\mathbf{A}^{(i,b)} \mid \mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)})$ and $\widetilde{\mathbf{B}}^{(i,b)} := (\mathbf{B}^{(i,b)} \mid \mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)})$ and the above expression becomes

$$p \cdot \Pr[\mathsf{Ext}(\widetilde{\mathbf{A}}^{(i,b)}, \widetilde{\mathbf{B}}^{(i,b)}) \neq b] + (1-p). \tag{9.9}$$

We would now like to get a lower bound on the entropy of each input to the extractor. Towards this end, we start by defining the probabilities

$$q_A := \Pr[\mathbf{A}^{(i,b)} \notin \mathsf{Bad}_A^{(i,b)}],$$
$$q_B := \Pr[\mathbf{B}^{(i,b)} \notin \mathsf{Bad}_B^{(i,b)}],$$

and we observe that $p = q_A \cdot q_B \leq \min\{q_A, q_B\}$. We now have two possible cases. In the first case, either q_A or q_B is at most $1/2$. In this case, $p \leq 1/2$ and $1 - p \geq 1/2$, which implies that Equation (9.9) is $\geq 1/2$.

In the second possible case, both q_A and q_B are $> 1/2$. In this case, it is straightforward to verify the following min entropy lower bounds:

$$\begin{aligned} H_\infty(\widetilde{\mathbf{A}}^{(i,b)}) &= \log\left(\frac{1}{\max_{x \notin \mathsf{Bad}_A^{(i,b)}} \Pr[\widetilde{\mathbf{A}}^{(i,b)} = x]}\right) \\ &= \log\left(\frac{q_A}{\max_{x \notin \mathsf{Bad}_A^{(i,b)}} \Pr[\mathbf{A}^{(i,b)} = x]}\right) \\ &> \log\left(\frac{\frac{1}{2}}{2^{-t}}\right) = t - 1, \end{aligned}$$

where the last inequality follows from the definition of bad sets. Of course, using the same reasoning,

$$H_\infty(\widetilde{\mathbf{B}}^{(i,b)}) > t - 1.$$

We are finally ready to pick t. We set it to $t := k + 1$, so that both of the above min-entropies become $> k$. Now, since Ext is a two-source extractor for min-entropy k with error ε, and since we are calling it on two independent sources of min-entropy k, the extractor property tells us

$$\begin{aligned} p \cdot \Pr[\mathsf{Ext}(\widetilde{\mathbf{A}}^{(i,b)}, \widetilde{\mathbf{B}}^{(i,b)}) \neq b] + (1-p) &\geq p \cdot (\frac{1}{2} - \varepsilon) + (1-p) \\ &= 1 - p \cdot (\frac{1}{2} + \varepsilon) \geq 1 - (\frac{1}{2} + \varepsilon) \\ &= \frac{1}{2} - \varepsilon. \end{aligned}$$

Thus, we finally see that in the case where both q_A and q_B are $> 1/2$, then Equation (9.9) is $\geq 1/2 - \varepsilon$ (given that we set $t := k + 1$). Thus in all cases, Equation (9.9) is $\geq 1/2 - \varepsilon$. And tracing back to the expression it originally represented, we get

$$\Pr[(\mathbf{A}^{(i,b)}, \mathbf{B}^{(i,b)}, b) \in S] \geq \frac{1}{2} - \varepsilon.$$

And we know that this holds for all i, b, since we made no assumption on their values. By Equation (9.7), this therefore implies

$$\Pr[(\mathbf{X}, \mathbf{Y}) \in S] \geq \frac{1}{2} - \varepsilon, \tag{9.10}$$

as long as we set $t = k+1$. Wrapping everything up, we combine Equation (9.8) and Equation (9.10) to get

$$\begin{aligned}|(\mathbf{X},\mathbf{Y}) - (\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n}))| &\geq \Pr[(\mathbf{X},\mathbf{Y}) \in S] - \Pr[(\mathbf{U}_{2n}, \mathsf{Ext}(\mathbf{U}_{2n})) \in S] \\ &> \frac{1}{2} - \varepsilon - 2^{-n+s+t+4} \\ &= \frac{1}{2} - \varepsilon - 2^{-n+s+k+5},\end{aligned}$$

as desired. □

Thus, we now know that two-party communication protocols have a hard time sampling $(\mathbf{U}_n, b(\mathbf{U}_n))$ when b is a good enough two-source extractor (Lemma 9.7). In order to actually instantiate this lemma and obtain our sampling lower bounds (Theorem 9.12), we will use the following classic two-source extractor.

Lemma 9.8 ([CG88]). *The function* $\mathsf{IP} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ *that computes the inner product of its inputs (mod 2) is a two-source extractor for min-entropy k with error* $\varepsilon = 2^{-(2k-n-1)/2}$.

Equipped with this extractor, we are ready to prove the main sampling lower bounds of this section.

Proof of Theorem 9.12. Let $n = 2\ell$ for some $\ell \in \mathbb{N}$, and let $\mathsf{Ext}\{0,1\}^\ell \times \{0,1\}^\ell \to \{0,1\}$ be the inner product extractor from Lemma 9.8 for min-entropy k and error $\varepsilon = 2^{-(2k-\ell-1)/2}$. By Lemma 9.7 we know that for any distribution $(\mathbf{X},\mathbf{Y}) \sim \{0,1\}^\ell \times \{0,1\}^{\ell+1}$ sampled by a two-party communication protocol with s bits of communication, it holds that

$$|(\mathbf{X},\mathbf{Y}) - (\mathbf{U}_n, \mathsf{Ext}(\mathbf{U}_n))| \geq \frac{1}{2} - \varepsilon - 2^{s-\ell+k+5} = \frac{1}{2} - 2^{-(2k-\ell-1)/2} - 2^{s-\ell+k+5}.$$

Plugging in $k = 3\ell/4$ yields

$$|(\mathbf{X},\mathbf{Y}) - (\mathbf{U}_n, \mathsf{Ext}(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{s-\ell/4+6} = \frac{1}{2} - 2^{s-n/8+6}$$

Finally, by our discussion at the beginning of this section on the connection between ROBPs and communication protocols, we know that for any $\ell \in \mathbb{N}$ and ROBP $F : \{0,1\}^\ell \to \{0,1\}^{n+1}$ of width $2^{s'}$,

$$|F(\mathbf{U}_n) - (\mathbf{U}_n, \mathsf{Ext}(\mathbf{U}_n))| \geq \frac{1}{2} - 2^{s'-n/8+7}.$$

Setting $s' = n/16$ completes the proof. □

We conclude this subsection by remarking that the above result gives an input-output distribution $(\mathbf{U}_n, b(\mathbf{U}_n))$ that cannot be sampled by exponential width ROBPs (even on average), but which can be sampled by AC^0 circuits. In particular, we can take b to be the inner product function, and combine the above result with the known result [IN96] that $(\mathbf{U}_n, \mathsf{IP}(\mathbf{U}_n))$ can be sampled in AC^0. Moreover, combining Theorem 9.12 with the fact that sampling lower bounds for ROBPs are stronger than correlation bounds (Theorem 9.8), we immediately recover strong correlation bounds against the function b.

9.4.3 Sampling lower bounds against general distributions

As a final application of the connection between communication protocols and ROBPs, we give a very simple distribution that is very hard to sample for ROBPs. In particular, we prove Theorem 9.3.

Theorem 9.13 (Sampling lower bounds against general distributions - Theorem 9.3, restated)**.** *There exists an explicit distribution* $\mathbf{Q} \sim \{0,1\}^n$ *such that for any ROBP* $F : \{0,1\}^\ell \to \{0,1\}^n$ *of width at most* $2^{n/12}$*,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 16 \cdot 2^{-n/12}.$$

Proof. Let $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ be a distribution sampled by a two-party communication protocol that communicates at most s bits. As we have seen, by fixing the message that Alice sends to Bob, we get that $(\mathbf{X}, \mathbf{Y})$ is a convex combination of $< 2^{s+1}$ distributions of the form $(\mathbf{X}', \mathbf{Y}') \sim \{0,1\}^n \times \{0,1\}^n$, where $\mathbf{X}', \mathbf{Y}'$ are independent. We consider any such $\mathbf{X}', \mathbf{Y}'$, and lower bound $|(\mathbf{X}', \mathbf{Y}') - (\mathbf{U}_n, \mathbf{U}_n)|$.

Towards this end, let t be a parameter we fix later, and let

$$\mathsf{BAD} := \left\{(s,s) \in \{0,1\}^n \times \{0,1\}^n : \Pr[\mathbf{X}' = s] \geq 2^{-t} \text{ or } \Pr[\mathbf{Y}' = s] \geq 2^{-t}\right\}.$$

Notice that $|\mathsf{BAD}| \leq 2 \cdot 2^t$, and furthermore for every $(s,s) \notin \mathsf{BAD}$ it holds that $\Pr[(\mathbf{X}', \mathbf{Y}') = (s,s)] < 2^{-2t}$. We let $S := \{(s,s) : s \in \{0,1\}^n\}$, and note that by definition of statistical distance we have

$$\begin{aligned}|(\mathbf{X}', \mathbf{Y}') - (\mathbf{U}_n, \mathbf{U}_n)| &\geq \Pr[(\mathbf{U}_n, \mathbf{U}_n) \in S - \mathsf{BAD}] - \Pr[(\mathbf{X}', \mathbf{Y}') \in S - \mathsf{BAD}] \\ &> 1 - 2^{-n} \cdot |\mathsf{BAD}| - 2^{-2t} \cdot |S - \mathsf{BAD}| \\ &\geq 1 - 2^{-n+t+1} - 2^{-2t+n}.\end{aligned}$$

We set $t = \frac{2n-1}{3}$ to equalize both exponents to $\frac{-n+2}{3}$, which yields

$$|(\mathbf{X}', \mathbf{Y}') - (\mathbf{U}_n, \mathbf{U}_n)| > 1 - 2^{\frac{-n+5}{3}}.$$

Now, recall that in the preliminaries of this thesis (Chapter 2), we proved that a convex combination of not-too-many distributions, each far from some target distribution $\mathbf{Q}$, is *itself* far from that target distribution $\mathbf{Q}$ (Fact 2.5). As a result, we immediately get that

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{U}_n, \mathbf{U}_n)| \geq 1 - 2^{s+1} \cdot 2^{\frac{-n+5}{3}} = 1 - 2^{\frac{-n+8}{3}+s} \geq 1 - 2^{-n/3+s+3}. \tag{9.11}$$

Finally, as discussed at the beginning of this section, for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{2n}$ of width $w = 2^{s'}$, the distribution $F(\mathbf{U}_\ell)$ can be sampled by a two-party communication protocol that uses $s'+1$ bits of communication. Combining this with Equation (9.11), we that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^{2n}$ of width s', it holds that

$$|F(\mathbf{U}_\ell) - (\mathbf{U}_n, \mathbf{U}_n)| \geq 1 - 2^{-n/3+s'+4},$$

which is at least $1 - 16 \cdot 2^{-n/6}$ for $s' \leq n/6$, as desired. □

Thus, we we have shown that ROBPs are remarkably bad at sampling the extremely simple distribution $(\mathbf{U}_n, \mathbf{U}_n)$. This provides a very strong separation between AC^0 circuits and ROBPs for the task of sampling, since AC^0 circuits can clearly sample this distribution. Moreover, combining Theorem 9.13 with the fact that sampling lower bounds for ROBPs are stronger than covariance bounds (Theorem 9.10), we immediately obtain strong covariance bounds against the equality function $\mathsf{EQ} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, which simply outputs 1 whenever its two inputs are the same. This completes the presentation, discussion, and proofs of our three basic results on the complexity of sampling (Theorems 9.1 to 9.3). With this foundation in place, we are finally ready to move towards proving the two main theorems of this chapter.

9.5 Sampling lower bounds against codes

In this section, we prove our first main theorem, giving tight sampling lower bounds against codes.

Theorem 9.14 (Sampling lower bounds against codes - Theorem 9.4, restated)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n,k,d) code of dimension k. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 12w \cdot 2^{-\frac{kd}{4n}}.$$

Theorem 9.15 (Tightness - Remark 9.1, formal)**.** *There is a universal constant $C > 0$ such that the following holds. For all $n,k,d \in \mathbb{N}$ such that there exists a linear $[n,k,d]$ code, there exists a distribution $\mathbf{Q} \sim \{0,1\}^n$ uniform over a linear $[n,k,d]$ code and an ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width $w \leq C \cdot 2^{C \cdot \frac{kd}{n} \log n}$ and length $\ell = n^2$ such that $F(\mathbf{U}_\ell) = \mathbf{Q}$.*

Theorem 9.14 asserts that for any distribution $\mathbf{Q} \sim \{0,1\}^n$ uniform over a good code, the output distribution of any ROBP of width $2^{\Omega(n)}$ has statistical distance $1-2^{-\Omega(n)}$ from $\mathbf{Q}$. Moreover, this statistical distance remains exponentially close to 1 (that is, of the form $1-2^{-n^{\Omega(1)}}$) for ROBPs of width $2^{n^{\Omega(1)}}$, even if the code parameters are of the form $kd \geq n^{1+\Omega(1)}$. In the context of AC^0 circuits, achieving such sampling lower bounds were left as an open question by Beck, Impagliazzo and Lovett [BIL12]. Finally, combining Theorem 9.14 with one of our sampling-computing equivalence theorems (Theorem 9.9), we immediately get that ROBPs of exponential width cannot test membership of a good code. In fact, using the average-case version of this equivalence (Theorem 9.10) yields the following stronger *covariance* bounds (Definition 9.9).

Corollary 9.3. *Let $b : \{0,1\}^n \to \{0,1\}$ be the indicator function of a good code $Q \subseteq \{0,1\}^n$. Then for any ROBP $F : \{0,1\}^n \to \{0,1\}$ of width $2^{\Omega(n)}$, it holds that $|\operatorname{cov}(F,b)| \leq 2^{-\Omega(n)}$.*

We now turn towards proving our tight sampling lower bounds.

9.5.1 Sampling lower bounds against list-decodable codes

In order to prove Theorem 9.14, we actually prove more general lower bounds against *list-decodable codes*.

Theorem 9.16 (Sampling lower bounds against list-decodable codes)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L)-list-decodable code of dimension k. Then for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w,*

$$|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq 1 - 8wL \cdot 2^{-\frac{\rho}{1+2\rho}k}.$$

As a reminder, list-decodable codes naturally generalize classical error-correcting codes (Definition 2.9). Indeed, a (ρ, L)-list-decodable code of dimension k is just a subset $Q \subseteq \{0,1\}^n$ of size 2^k such that any Hamming ball of radius ρn contains at most L points from Q. Thus, any classical (n,k,d) code Q is a $(\rho, 1)$-list-decodable code of dimension k for any $\rho < \frac{d}{2n}$ (Fact 2.10), and combining this with Theorem 9.16 immediately yields Theorem 9.14. Because of this, we will just focus on proving Theorem 9.16. In fact, we actually focus on proving such a result for *KRVZ samplers*, which are much easier to work with.

Theorem 9.17 (Sampling lower bounds against list-decodable codes)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over a (ρ, L)-list-decodable code of dimension k. Then for any KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ of width w,*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - 4wL \cdot 2^{-\frac{\rho}{1+2\rho}k}.$$

By combining the above result with our equivalence theorem for KRVZ and ROBP samplers (Theorem 9.6), we immediately obtain Theorem 9.16. Thus, in order to obtain our sampling lower bounds against codes for ROBPs, all we must do is prove Theorem 9.17. And indeed, this will be our main focus here. However, even though KRVZ samplers are easier to work with, that doesn't mean that Theorem 9.17 will be easy to prove. Thus, we start with a high-level overview of the plan, before giving the proof in full formality.

An overview of the proof

The proof of Theorem 9.17 uses two main ingredients. First, it uses a known result (Lemma 3.22) which says that any KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ can be written as a convex combination of a few product distributions. More formally, if the sampler has width w, then for any r, ℓ with $r\ell = n$, it can be written as a convex combination of w^r distributions of the form $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$, where each $\mathbf{Y}_i$ is independent. The second ingredient, which we will prove, is that product distributions, i.e., those of the above form, are statistically far from (distributions that are uniform over) good list-decodable codes.

At this point, it would be nice to conclude that the original KRVZ sampler $\mathbf{X}$ must also be far from a good list-decodable code $\mathbf{Q}$. In particular, we would like to argue that $\mathbf{X}$ is a convex combination of product distributions $\{\mathbf{Y}^{(j)}\}$, and each of these product distributions is far from $\mathbf{Q}$, so $\mathbf{X}$ must be far from $\mathbf{Q}$. Unfortunately, the bounds we are trying to lift are in the wrong direction: It is true that if each $\mathbf{Y}^{(j)}$ is close to $\mathbf{Q}$, then $\mathbf{X}$ is close to $\mathbf{Q}$, but it is not necessarily true that if each $\mathbf{Y}^{(j)}$ is far from $\mathbf{Q}$, then $\mathbf{X}$ is far from $\mathbf{Q}$. Indeed, each $\mathbf{Y}^{(j)}$ could be constant over a (different) codeword, which would make each $\mathbf{Y}^{(j)}$ extremely far from $\mathbf{Q}$, but still allow the overall convex combination over $\{\mathbf{Y}^{(j)}\}$ to exactly sample $\mathbf{Q}$.

Given the above counterexample, a new idea might be to try to argue that *as long as there aren't too many distributions* $\mathbf{Y}^{(j)}$ participating in the convex combination, then if each $\mathbf{Y}^{(j)}$ is far from $\mathbf{Q}$, then $\mathbf{X}$ is relatively far from $\mathbf{Q}$. It turns out this is true (Fact 2.5), but the corresponding lower bounds on $|\mathbf{X} - \mathbf{Q}|$ that it yields are still not as strong as we would like. To get the strongest possible bounds, we need a slightly more nuanced way to combine our two key ingredients. Below, we sketch a proof for our second ingredient, and show how to combine it with the first ingredient to yield our desired lower bound on $|\mathbf{X} - \mathbf{Q}|$.

Anti-concentration of product distributions in Hamming balls We now argue that product distributions are far from sampling good list-decodable codes. Let $\mathbf{Q} \sim \{0,1\}^n$ be a (ρ, L) list-decodable code, and let $\mathbf{Y} = (\mathbf{Y}_1, \dots, \mathbf{Y}_r) \sim (\{0,1\}^\ell)^r$ be such that each $\mathbf{Y}_i$ is independent and $r\ell = n$. Furthermore, we will need the product distribution to have a reasonable number of components r, or else it could clearly sample the code perfectly (if $r = 1$). Towards this end, we enforce the mild requirement $r \geq 1/\rho$, and thus $\ell \leq \rho n$. For a good list-decodable code, we can think of $\rho = \Omega(1)$, and thus we just require $r \geq O(1)$.

Now, the key intuition about product distributions is that for any point x in the space $\{0,1\}^n$ to which $\mathbf{Y}$ does not assign too much probability, the following must hold: if we draw a Hamming ball $\mathcal{B}(x)$ around x whose radius is not too small, then the vast majority of probability weight assigned to $\mathcal{B}(x)$ by $\mathbf{Y}$ does *not* land on x. In symbols, $\Pr[\mathbf{Y} \in \mathcal{B}(x)] \gg \Pr[\mathbf{Y} = x]$.

Let us formalize this intuition a little more. Fix any $x \in \{0,1\}^n$, and let $p := \Pr[\mathbf{Y} = x]$. Consider now the ball $\mathcal{B}_\ell(x)$ around x of radius ℓ. Now, parse x as $x = (x_1, \dots, x_r) \in (\{0,1\}^\ell)^r$. Since $\mathbf{Y}$ is a product distribution consisting of r components, there must be at least some $i \in [r]$ such that $\Pr[\mathbf{Y}_i = x_i] \leq p^{1/r}$. Consider now the set T of all strings of the form $(x_1, \dots, x_{i-1}, z, x_{i+1}, \dots, x_r) \in (\{0,1\}^\ell)^r$, where each x_j is fixed as before, but z can be taken as any element in $\{0,1\}^\ell$. Then $\mathbf{Y}$ assigns probability at least $\Pr[\mathbf{Y} = x]/p^{1/r}$ to the set T, and of course T is in the ball $\mathcal{B}_\ell(x)$. Thus, we get that $\Pr[\mathbf{Y} \in \mathcal{B}(x)] \geq \Pr[\mathbf{Y} = x]/p^{1/r}$, and therefore $\Pr[\mathbf{Y} \in \mathcal{B}(x)] \gg \Pr[\mathbf{Y} = x]$, as long as p is not too big.

Now, how can we use this to show $|\mathbf{Y} - \mathbf{Q}|$ is large? Well, by definition of statistical distance, it suffices to pick a set $S \subseteq \{0,1\}^n$ and show that $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y} \in S]$ is large. Our choice for S will be all codewords from $\mathbf{Q}$, with some "bad" codewords Bad removed. Then to lower bound $\Pr[\mathbf{Q} \in S]$, we just need to show that $\mathbf{Q}$ lands in Bad with not too high probability: in particular, we can just require that Bad is not too big. And to upper bound $\Pr[\mathbf{Y} \in S]$, we just need to show that the probability that $\mathbf{Y}$ lands on a "not-bad" codeword is small.

So what should we choose as the set Bad? You guessed it: a small set of codewords assigned the highest probability by $\mathbf{Y}$ (say, all codewords assigned probability $\geq p$ for some threshold probability p). As long as this set isn't too big (i.e., p isn't too small), we will have $\Pr[\mathbf{Q} \in S]$ be very close to 1. And as long as we removed the codewords assigned very high probability by $\mathbf{Y}$, we will have that $\Pr[\mathbf{Y} \in S]$ is very close to 0. We argue the latter, below.

To upper bound the probability that $\mathbf{Y}$ lands in S, we consider the sum $\sum_{q \in S} \Pr[\mathbf{Y} = q]$. By our anti-concentration observation above, this sum is at most $p^{1/r} \cdot \sum_{q \in S} \Pr[\mathbf{Y} \in \mathcal{B}_\ell(q)]$. Intuitively, we will now want to make sure that (i) r is not too big, because otherwise $p^{1/r}$ will be too big; and (ii) ℓ is not too big, because otherwise many of the balls $\{\mathcal{B}_\ell(q)\}$ in the sum will have big overlaps, causing probabilities to be multi-counted and the overall sum to be large.

It turns out that the best tradeoff occurs at setting $r = 1/\rho$ and $\ell = \rho n$. This is because a good list-decodable code will have $\rho = \Omega(1)$, which yields $p^{1/r} = p^{\Omega(1)}$, which will be quite small as long as we originally set our threshold probability p to be low enough. Similarly, by definition of list-decodability, we will have that any point x in the space $\{0,1\}^n$ will appear in at most L balls $\{\mathcal{B}_\ell(q)\}_{q \in S}$. This implies $\sum_{q \in S} \Pr[\mathbf{Y} \in \mathcal{B}_\ell(q)] \leq L$, since the probability $\mathbf{Y}$ assigns to any point $x \in \{0,1\}^n$ is counted at most L times. For a good list-decodable code, L is quite small, and we finally have that $\Pr[\mathbf{Y} \in S] \leq p^{1/r} \cdot L$ will be very close to 0.

Thus, as long as our original product distribution $\mathbf{Y} \sim (\{0,1\}^\ell)^r$ had $r \approx 1/\rho$ and $\ell \approx \rho n$, we have a set $S \subseteq \{0,1\}^n$ that makes $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y} \in S]$ very close to 1, implying the statistical distance $|\mathbf{Y} - \mathbf{Q}|$ is very close to 1.

From KRVZ samplers to product distributions Now, the question is: how do we use the fact that product distributions are far from good list-decodable codes in order to argue that KRVZ samplers are far from list-decodable codes? Well, let $\mathbf{X} \sim \{0,1\}^n$ be the KRVZ sampler, and $\mathbf{Q} \sim \{0,1\}^n$ be the list-decodable code. We need to show that $|\mathbf{X} - \mathbf{Q}|$ is large. So we write $\mathbf{X}$ as a convex combination of at most w^r product distributions $\{\mathbf{Y}^{(j)}\}_j$, each of the form $\mathbf{Y}^{(j)} = (\mathbf{Y}_1^{(j)}, \dots, \mathbf{Y}_r^{(j)}) \sim (\{0,1\}^\ell)^r$.

For any tester $S \subseteq \{0,1\}^n$, the statistical distance $|\mathbf{X} - \mathbf{Q}|$ is lower bounded by $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{X} \in S]$. Furthermore, it is easy to verify that this, in turn, is lower bounded by the worst $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y}^{(j)} \in S]$ (meaning the one that gives the smallest value). So what S should we pick?

From above, we know that as long as we set $r = 1/\rho$ and $\ell = \rho n$, then each $\mathbf{Y}^{(j)}$ has a test $S^{(j)}$ which makes $\Pr[\mathbf{Y}^{(j)} \in S^{(j)}]$ very close to 0. Furthermore $S^{(j)}$ is of the form $Q - \mathsf{Bad}^{(j)}$, where Q is the support of the code and $\mathsf{Bad}^{(j)}$ is some small bad set. Thus, since we want a *single* test $S \subseteq \{0,1\}^n$ guaranteed to make $\Pr[\mathbf{Q} \in S] - \Pr[\mathbf{Y}^{(j)} \in S]$ small *for every* j, we can simply take the test to be $S = Q - \cup_j \mathsf{Bad}^{(j)}$. Indeed, this guarantees that each $\Pr[\mathbf{Y}^{(j)} \in S]$ will be very close to 0, and as long as there are not too many elements in the convex combination, our total collection of bad elements won't be too big, and $\Pr[\mathbf{Q} \in S]$ will stay very close to 1. Thus we get that $|\mathbf{X} - \mathbf{Q}|$ is close to 1, as desired.

A formal presentation of the proof

Now that we have a high-level idea of how our proof will work, we are ready to present it in full formality.

Proof of Theorem 9.17. We start by writing our KRVZ sampler as a convex combination of random variables with nice structure. Let $\mathbf{W} = (\mathbf{W}_1, \dots, \mathbf{W}_n)$ be the vertices hit on the random walk that generates $\mathbf{X}$ (excluding the start vertex of the branching program). Let r, ℓ be positive integers that will be set later to ensure $r\ell = n$. Define $\mathbf{W}^* = (\mathbf{W}_\ell, \mathbf{W}_{2\ell}, \dots, \mathbf{W}_{r\ell})$, and recall the following standard observation (Lemma 3.22): for any $W \in \text{support}(\mathbf{W}^*)$, the random variable $(\mathbf{X} \mid \mathbf{W}^* = W)$ is of the form $\mathbf{X}^{(W)} := (\mathbf{X}_1^{(W)}, \mathbf{X}_2^{(W)}, \dots, \mathbf{X}_r^{(W)})$, where each $\mathbf{X}_i^{(W)} \sim \{0,1\}^\ell$ is independent. Thus the KRVZ sampler $\mathbf{X}$ is a convex combination of the form

$$\mathbf{X} = \sum_{W \in \text{support}(\mathbf{W}^*)} p_W \cdot \mathbf{X}^{(W)},$$

where each $p_W := \Pr[\mathbf{W}^* = W]$.

The goal now is to use the above decomposition to help us get a good lower bound on $|\mathbf{X} - \mathbf{Q}| = \max_S |\Pr[\mathbf{X} \in S] - \Pr[\mathbf{Q} \in S]|$. Towards this end, we note that for any S,

$$\begin{aligned} |\mathbf{X} - \mathbf{Q}| &\geq \Pr[\mathbf{Q} \in S] - \Pr[\mathbf{X} \in S] = \Pr[\mathbf{Q} \in S] - \sum_W p_W \cdot \Pr[\mathbf{X}^{(W)} \in S] \\ &\geq \Pr[\mathbf{Q} \in S] - \max_W \Pr[\mathbf{X}^{(W)} \in S]. \end{aligned}$$

Thus, we would like to pick a test S that maximizes the quantity $\Pr[\mathbf{Q} \in S]$ while minimizing the quantity $\max_W \Pr[\mathbf{X}^{(W)} \in S]$. A natural candidate for S is the entire codebook $Q = \text{support}(\mathbf{Q})$, minus some small set of "bad codewords" Bad, which are assigned too high of a probability by some $\mathbf{X}^{(W)}$. As such, we let $t > 0$ be a parameter to be set later, and we define $S = Q - \mathsf{Bad}$ where

$$\begin{aligned} \mathsf{Bad} &:= \bigcup_W \mathsf{Bad}^{(W)}, \\ \mathsf{Bad}^{(W)} &:= \{q \in Q : \Pr[\mathbf{X}^{(W)} = q] > 2^{-t}\}. \end{aligned}$$

Plugging in this definition of S, we get

$$\begin{aligned} |\mathbf{X} - \mathbf{Q}| &\geq \Pr[\mathbf{Q} \in Q - \mathsf{Bad}] - \max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}] \\ &\geq 1 - \Pr[\mathbf{Q} \in \mathsf{Bad}] - \max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}]. \end{aligned}$$

Thus, we would like to upper bound both quantities that are subtracted. To upper bound the first quantity, simply note that

$$\Pr[\mathbf{Q} \in \mathsf{Bad}] = 2^{-k} \cdot |\mathsf{Bad}| \leq 2^{-k} \sum_{W \in \text{support}(\mathbf{W}^*)} |\mathsf{Bad}^{(W)}| < 2^{-k+t+r\log(w)}$$

via the trivial upper bounds $|\text{support}(\mathbf{W}^*)| \leq w^r$ and $|\mathsf{Bad}^{(W)}| < 2^t$ for each W.

To upper bound the second quantity $\max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}]$, we start by making notation more convenient: let W^* be the maximizer of the above quantity, and define $\mathbf{Y} := \mathbf{X}^{(W^*)}$ and $\mathsf{Bad}^* := \mathsf{Bad}^{(W^*)}$. Of course we have $\max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}] = \Pr[\mathbf{Y} \in Q - \mathsf{Bad}^*]$, and we focus on upper bounding the latter.

Recall that $\mathbf{Y}$ is of the form $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_r)$ where each $\mathbf{Y}_i \sim \{0,1\}^\ell$ is independent, and $\mathsf{Bad}^* := \{q \in Q : \Pr[\mathbf{Y} = q] > 2^{-t}\}$ contains all codewords hit by $\mathbf{Y}$ with large probability. Thus each $q \in Q - \mathsf{Bad}^*$ must have $\Pr[\mathbf{Y} = q] \leq 2^{-t}$. So, if we parse each q as $(q_1, q_2, \ldots, q_r) \in (\{0,1\}^\ell)^r$, we have that $\Pr[\mathbf{Y} = q] = \Pr[\mathbf{Y}_1 = q_1] \cdot \Pr[\mathbf{Y}_2 = q_2] \cdots \Pr[\mathbf{Y}_r = q_r]$ by the independence of these random variables, and so there must be some $\pi(q) \in [r]$ such that $\Pr[\mathbf{Y}_{\pi(q)} = q_{\pi(q)}] \leq 2^{-t/r}$.

Now, for a string $x = (x_1, x_2, \ldots, x_r) \in (\{0,1\}^\ell)^r$, we let $x_{-i} := (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_r)$ denote x with its i^{th} chunk removed, and proceed as follows:

$$\begin{aligned}
\Pr[\mathbf{Y} \in Q - \mathsf{Bad}^*] &= \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y} = q] \\
&= \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y}_{\pi(q)} = q_{\pi(q)}] \cdot \Pr[\mathbf{Y}_{-\pi(q)} = q_{-\pi(q)}] \\
&\leq 2^{-t/r} \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y}_{-\pi(q)} = q_{-\pi(q)}] \\
&\leq 2^{-t/r} \sum_{q \in Q - \mathsf{Bad}^*} \Pr[\mathbf{Y} \in \mathsf{Ball}(q, \ell)] \\
&= 2^{-t/r} \sum_{v \in \{0,1\}^n} \Pr[\mathbf{Y} = v] \cdot \#\{q \in Q - \mathsf{Bad}^* : v \in \mathsf{Ball}(q, \ell)\} \\
&\leq 2^{-t/r} \sum_{v \in \{0,1\}^n} \Pr[\mathbf{Y} = v] \cdot \#\{q \in Q : \Delta(v, q) \leq \ell\} \\
&\leq 2^{-t/r} \sum_{v \in \{0,1\}^n} \Pr[\mathbf{Y} = v] \cdot |\mathsf{Ball}(v, \ell) \cap Q| \\
&\leq 2^{-t/r} \cdot \max_v |\mathsf{Ball}(v, \ell) \cap Q| \\
&\leq 2^{-t/r} \cdot L \textbf{ if } \ell \leq \rho n,
\end{aligned}$$

where the last line follows since Q is a (ρ, L)-list-decodable code (Definition 2.9). Thus, provided that we have selected $r, \ell \in \mathbb{N}$ such that $r\ell = n$ and $\ell \leq \rho n$, we can combine all of the above to get

$$\begin{aligned}
|\mathbf{X} - \mathbf{Q}| &\geq 1 - \Pr[\mathbf{Q} \in \mathsf{Bad}] - \max_W \Pr[\mathbf{X}^{(W)} \in Q - \mathsf{Bad}^{(W)}] \\
&= 1 - \Pr[\mathbf{Q} \in \mathsf{Bad}] - \Pr[\mathbf{Y} \in Q - \mathsf{Bad}^*] \\
&> 1 - 2^{-k+t+r\log w} - 2^{-t/r + \log L}.
\end{aligned}$$

Before picking r, ℓ, we set[16] $t = \frac{r}{r+1} \cdot (k - r\log w + \log L)$ as the value that equalizes the two exponents to $-\frac{1}{r+1} \cdot (k - r\log w + \log L) + \log L \leq -\frac{k}{r+1} + \log(wL)$ to obtain

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-t/r + \log L + 1} \geq 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1}.$$

Thus all that remains is to pick $r, \ell \in \mathbb{N}$ such that $r\ell = n$. If $1/\rho$ and ρn are integers, we simply set $r = 1/\rho$ and $\ell = \rho n$ to obtain

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1} = 1 - 2wL \cdot 2^{-\frac{\rho}{1+\rho}k}.$$

[16] Technically we originally asked for $t > 0$, but we may assume this without loss of generality, since if this setting of t is nonpositive, then the claimed result will become $|\mathbf{X} - \mathbf{Q}| \geq 0$, which is trivially true.

If $1/\rho$ and ρn are not integers, there is an easy way to slightly modify the proof so that everything works out (with a minor loss in parameters): first, recall that we originally defined $\mathbf{W}^* = (\mathbf{W}_\ell, \mathbf{W}_{2\ell}, \dots, \mathbf{W}_{r\ell})$ so that each $(\mathbf{X} \mid \mathbf{W}^* = W)$ is of the form $\mathbf{X}^{(W)} := (\mathbf{X}_1^{(W)}, \dots, \mathbf{X}_r^{(W)})$, where each $\mathbf{X}_i^{(W)}$ is independent and over ℓ bits. Observe that the argument actually does not require that each $\mathbf{X}_i^{(W)}$ has the same length; instead, it simply requires that each $\mathbf{X}_i^{(W)}$ has length at most ρn. Thus, we could have actually started with any $\mathbf{W}^*$ of the form

$$\mathbf{W}^* = (\mathbf{W}_{\alpha_1}, \mathbf{W}_{\alpha_2}, \dots, \mathbf{W}_{\alpha_r}),$$

where $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_r = n$, and each gap $\alpha_j - \alpha_{j-1}$ is bounded above by ρn. And the exact same argument as above yields

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1}. \tag{9.12}$$

Thus, we now have more flexibility in dealing with the case where $1/\rho$ and ρn are not integers: we can simply pick an integer r and define $0 < \alpha_1 < \alpha_2 < \cdots < \alpha_r = n$ such that each gap is at most ρn. In more detail, we can force the first $r - 1$ gaps to be exactly $\lfloor \rho n \rfloor$, while the last gap is at most $\lfloor \rho n \rfloor$, by picking $r := \left\lceil \frac{n}{\lfloor \rho n \rfloor} \right\rceil$ and setting $\alpha_j := \lfloor \rho n \rfloor \cdot j$ for all $j \in [r-1]$.

Before we plug the value of r into Equation (9.12), it is useful to get a clean lower bound on $\frac{1}{r+1}$:

$$\begin{aligned}
\frac{1}{r+1} &= \frac{1}{\left\lceil \frac{n}{\lfloor \rho n \rfloor} \right\rceil + 1} \geq \frac{1}{\frac{n}{\lfloor \rho n \rfloor} + 2} \geq \frac{1}{\frac{n}{\rho n - 1} + 2} \\
&= \frac{\rho n - 1}{n + 2(\rho n - 1)} = \frac{\rho n}{n + 2(\rho n - 1)} - \frac{1}{n + 2(\rho n - 1)} \\
&\geq \frac{\rho n}{n + 2\rho n} - \frac{1}{n} = \frac{\rho}{1 + 2\rho} - \frac{1}{n} \geq \frac{\rho}{1 + 2\rho} - \frac{1}{k},
\end{aligned}$$

where the last inequality follows since $k \leq n$, because a code's dimension cannot exceed the dimension in which it lives. At last, we can plug our lower bound for $1/(r+1)$ into Equation (9.12) to obtain

$$|\mathbf{X} - \mathbf{Q}| > 1 - 2^{-\frac{k}{r+1} + \log(wL) + 1} \geq 1 - 2^{-k \cdot (\frac{\rho}{1+2\rho} - \frac{1}{k}) + \log(wL) + 1} = 1 - 4wL \cdot 2^{-\frac{\rho}{1+2\rho} k},$$

which completes the proof. □

9.5.2 Sampling upper bounds against uniquely-decodable codes

In the previous section, we gave very general sampling lower bounds against list-decodable codes for KRVZ samplers, via Theorem 9.17. By combining this result with the list-decodability of (n, k, d) codes (Fact 2.10), we immediately obtain the following sampling lower bounds against (n, k, d) codes.

Theorem 9.18 (Sampling lower bounds against codes)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be uniform over an (n, k, d) code. Then for any KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ of width w,*

$$|\mathbf{X} - \mathbf{Q}| \geq 1 - 8w \cdot 2^{-\frac{kd}{4n}}.$$

In this section, we will prove that this result is almost tight, in the following sense. For almost all valid parameters n, k, d, there exists an (n, k, d) code $\mathbf{Q} \sim \{0,1\}^n$ that can be exactly sampled by a KRVZ sampler of width $w = 2^{\widetilde{O}(\frac{kd}{n})}$. In fact, we will actually prove a stronger result, and show that such a code can be generated by *simple samplers* (Definition 9.6) that are *granular* (Definition 9.7).

Theorem 9.19 (Tightness)**.** *There is a universal constant $C > 0$ such that the following holds. For all $n, k, d \in \mathbb{N}$ such that there exists a linear $[n, k, d]$ code, there exists a distribution $\mathbf{Q} \sim \{0,1\}^n$ uniform over a linear $[n, k, d]$ code that can be exactly generated by a 2^{-n}-granular simple sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w \leq C \cdot 2^{C \cdot \frac{kd}{n} \cdot \log n}$.*

Since we know that granular samplers can be perfectly simulated by ROBP samplers (Lemma 9.5), the above is all we need to show in order to immediately obtain our tightness result for ROBP samplers (Theorem 9.15). However, Theorem 9.19 will not be trivial to prove. Thus, let's begin with a proof overview.

An overview of the proof

Our proof of tightness is split into two cases. In the **first case**, we consider $k \leq 0.9n$. Here, the general idea is to define some n', k', d' such that there exists an (n', k', d') code Q', and then simply consider the repetition code $Q := Q' \times Q' \times \cdots \times Q'$, where n/n' copies of Q' participate in the Cartesian product. It is straightforward to verify that Q will be an $(n, k'n/n', d')$ code. Furthermore, it is not hard to see that a simple sampler of width w can sample any distribution with support size w (Fact 9.1), and that the product distribution of two distributions, each samplable by a simple sampler of width w, is also samplable by a simple sampler of width w (Fact 9.3). Thus, Q will be samplable by a simple sampler of width $2^{k'}$. Thus, if we can find a constant C and an (n', k', d') code with $n' = Cd, k' = Ckd/n, d' = d$, we are done with this case. Since $k \leq 0.9n$, the Gilbert-Varshamov bound (Theorem 2.1) guarantees this is always possible.

In the **second case**, we consider $k > 0.9n$. Here, the general idea is that $n - k$ will now be small. We consider two subcases: $k \geq n - 4d \log n$ and $k < n - 4d \log n$. We focus on the first subcase in this overview, as it is not too hard to extend the argument to work for the second subcase. In the first subcase, note that $n - k \leq 4d \log n$. Now, the main idea is that ROBPs can check membership of a k dimensional subspace $Q \subseteq \mathbb{F}_2^n$ using width 2^{n-k}, simply by keeping track of the $n - k$ parity checks that define Q. Furthermore, it is not too hard to show that for any ROBP of width w, the uniform distribution over its accepting strings can be generated by a simple sampler of width w (Theorem 9.9). Thus in this case, we can simply take any linear (n, k, d) code and uniformly sample from it using a simple sampler of width $2^{n-k} \leq 2^{4d \log n} \leq 2^{\frac{5kd}{n} \cdot \log n}$, where the last inequality follows from the current case $k > 0.9n$.

A formal presentation of the proof

With the above roadmap in mind, we now turn to our formal proof of tightness.

Proof of Theorem 9.19. As promised, we split the casework into $k \leq 0.9n$ and $k > 0.9n$.

Case 1 ($k \leq 0.9n$) We will show that there is a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$ such that the uniform distribution $\mathbf{Q}$ over Q can be sampled in width $w \leq C \cdot 2^{Ckd/n}$ for a sufficiently large constant C. To construct Q, the general idea will be to take an $[n', k', d']$ code Q' over a smaller space $\{0,1\}^{n'}$ and repeat it n/n' times. In more detail, let C be a sufficiently large constant to be chosen later, and set $n' = Cd$. We will aim to repeat our smaller code $t := \lfloor n/n' \rfloor$ times. We may assume $t \geq 20$: Otherwise, $20n'/n = 20Cd/n > 1$ and we can sample the linear $[n, k, d]$ code (guaranteed to exist by the hypothesis) by a simple sampler of width $w = 2^k < 2^{k \cdot 20Cd/n}$ (by Fact 9.1), and we are done. So we can henceforth assume $t \geq 20$.

Note that there must be some integers $n_1, \ldots, n_t$ such that each $n_i \geq n'$ and $\sum_i n_i = n$. Suppose now that there exist a collection of codes $\{Q_i\}_{i \in [t]}$ such that all of the following hold:

- Each Q_i is a linear $[n_i, k_i, d_i]$ code.

- Each $k_i = \lceil k/t \rceil$.
- Each $d_i = d$.

Then $\sum_i k_i \geq k$, and we may of course find a collection of linear codes $\{Q_i'\}_{i\in[t]}$ with the same properties as $\{Q_i\}_{i\in[t]}$, except that the property $k_i = \lceil k/t \rceil$ is traded for the properties $k_i \leq \lceil k/t \rceil$ and $\sum_i k_i = k$ (simply by reducing the dimension of each code by an appropriate amount). Notice that $Q_1' \times Q_2' \times \cdots \times Q_t'$ is then a linear $[n, k, d]$ code. Furthermore, by combining Fact 9.1 and Fact 9.3, this code can be exactly generated by a simple sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w \leq 2^{\lceil \frac{k}{t} \rceil} \leq 2 \cdot 2^{2C \cdot \frac{kd}{n}}$.

Thus all that remains for this case is to show the existence of a collection of codes $\{Q_i\}_{i\in[t]}$ with the above mentioned properties. For this, it suffices to show the existence of a linear $[n', k', d']$ code, where

$$\begin{aligned} n' &= Cd, \\ k' &= \left\lceil \frac{k}{t} \right\rceil = \left\lceil \frac{k}{\lfloor \frac{n}{Cd} \rfloor} \right\rceil, \\ d' &= d. \end{aligned}$$

By the Gilbert-Varshamov bound (Theorem 2.1), such a code exists as long as $2^{k'} \leq 2^{n'} / \binom{n'}{\leq d'-1}$. Plugging in the above values for n', k', d', a straightforward calculation (using the case condition that $k/n \leq 0.9$) shows that such a code must exist whenever $C \geq 250$. Thus we can always find a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$ such that the uniform distribution $\mathbf{Q}$ over Q can be sampled in width $w \leq 2 \cdot 2^{5000 \cdot kd/n}$.

Case 2 ($k > 0.9n$) We will show that there exists a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$ such that the uniform distribution $\mathbf{Q}$ over Q can be sampled in width $w \leq 2^{C \cdot \frac{kd}{n} \log n}$ for a sufficiently large constant C. To construct Q, the general idea will be to start with a code Q' of dimension $k' \gg k$, show that membership in Q' can be checked by a small width ROBP (by keeping track of parity checks), and then convert this into a simple sampler via Theorem 9.9. Then, it will not be too difficult to reduce the dimension of Q' to match the target dimension k, while barely affecting the width of the sampler. In more detail, let $k' := n - \lceil 4d \log n \rceil$. We consider the subcases $k \geq k'$ and $k < k'$.

Subcase 2.1 ($k \geq k'$) By the theorem hypothesis, we know that there is a linear $[n, k, d]$ code $Q \subseteq \{0,1\}^n$. Let $Q^\perp$ denote its dual, and recall that $Q^\perp$ must therefore have dimension $n - k$. In other words, we can find a basis $v^{(1)}, \dots, v^{(n-k)}$ of $Q^\perp$. Now, define a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ such that $f(x) = 1$ if and only if $\langle x, v^{(i)} \rangle = 0$ over $\mathbb{F}_2$, for all $i \in [n-k]$. That is, f accepts exactly the strings in $(Q^\perp)^\perp = Q$.

We can now design a low width ROBP that computes f. To do so, the ROBP keeps as its state a string $s \in \mathbb{F}_2^{n-k}$, which is originally initialized to the all zeroes vector. Upon reading a bit $x_i \in \{0,1\}$, the ROBP considers a vector $u^{(i)} \in \mathbb{F}_2^{n-k}$ which consists of the i^{th} bit of each of $v^{(1)}, \dots, v^{(n-k)}$ concatenated together. Then, the ROBP transitions to state $s + x_i \cdot u^{(i)}$. In its final layer, the ROBP treats the all zeroes state as the accept state, and every other state as a reject state.

It is straightforward to verify that the above ROBP has width 2^{n-k}, and that the state $s \in \mathbb{F}_2^{n-k}$ it reaches in the final layer is exactly $(\langle x, v^{(1)} \rangle, \dots, \langle x, y^{(n-k)} \rangle)$. Since the ROBP accepts if and only if this is the all zeroes vector, we see that the ROBP exactly computes f. And since $f^{-1}(1) = Q$, we can apply Theorem 9.9 to get a simple sampler of width $w = 2^{n-k}$ that samples the uniform distribution $\mathbf{Q}$ over Q. Since we have $k \geq k' = n - \lceil 4d \log n \rceil$, we know $n - k \leq \lceil 4d \log n \rceil \leq 5d \log n < 2 \cdot \frac{k}{n} \cdot 5d \log n$, where the last inequality follows from the case condition $k/n > 0.9$. Thus our simple sampler for our $[n, k, d]$ code $\mathbf{Q}$ has width $w < 2^{10 \cdot \frac{kd}{n} \log n}$, as desired.

Subcase 2.2 ($k < k'$) Let $t := k' - k \geq 1$. By the Gilbert-Varshamov bound (Theorem 2.1), there must exist a linear $[n, k', d]$ code $Q' \subseteq \mathbb{F}_2^n$. As we have seen above, Q' is easy to sample. But we would like to sample an $[n, k, d]$ code using approximately the same width. To do so, we will find a subcode of Q' that is easy to sample.

Since Q' is a linear $[n, k', d]$ code, it must have some vector q of Hamming weight d. Without loss of generality, we may assume the first d coordinates of q are 1, and the last $n - d$ coordinates are 0. Consider now the orthogonal complement S_q of $\{0, q\}$. Note that S_q has dimension $n - 1$. Furthermore, consider defining so-called "augmented elementary basis vectors" $\{\hat{e}^{(i)}\}_{i \in [n], i \neq d}$ as follows: for each $i < d$, let $\hat{e}^{(i)} := e^{(i)} + e^{(i+1)}$, and for each $i > d$, let $\hat{e}^{(i)} := e^{(i)}$, where each $e^{(i)} \in \mathbb{F}_2^n$ denotes a standard elementary basis vector. Then $\{\hat{e}^{(i)}\}$ is a basis for S_q.

Now let $v^{(1)}, \ldots, v^{(n-k')}$ be an arbitrary basis for the orthogonal complement $(Q')^{\perp}$. By straightforward linear algebra, there must be at least $k' - 1$ vectors in $\{\hat{e}^{(i)}\}_i$ that are mutually independent with $v^{(1)}, \ldots, v^{(n-k')}$. Without loss of generality, assume they are $\hat{e}^{(1)}, \ldots, \hat{e}^{(k'-1)}$. Notice now that $t \leq k' - 1$ (since we may assume $k \geq 1$), and consider the subspace $\widetilde{Q}$ spanned by basis vectors $v^{(1)}, \ldots, v^{(n-k')}, \hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$. Observe that $\widetilde{Q}$ has dimension $n - k' + t = n - k$.

Finally, let $Q^* := \widetilde{Q}^{\perp}$. Notice that Q^* has dimension $n - (n - k) = k$ and that Q^* is a subspace of $((Q')^{\perp})^{\perp} = Q'$. Thus Q^* has minimum distance $\geq d$. In fact, by our basis selection for $\widetilde{Q}$, it is straightforward to verify that the Hamming-weight d vector q defined earlier is also in Q^* (since q has inner product 0 with all the basis vectors of $\widetilde{Q}$). Thus, Q^* has minimum distance exactly d and it is therefore a linear $[n, k, d]$ code. Thus all that remains is to show that the uniform distribution over Q^* can be sampled by a low width simple sampler.

As in the first case of this proof, let $f : \mathbb{F}_2^n \to \mathbb{F}_2$ be defined such that $f(x) = 1$ if and only if $\langle x, v \rangle = 0$ for all $v \in \{v^{(1)}, \ldots, v^{(n-k')}, \hat{e}^{(1)}, \ldots, \hat{e}^{(t)}\}$. In other words, f tests membership in Q^*. Thus, if there is an ROBP of width w that computes f, then there is a simple sampler of width w that samples the uniform distribution $\mathbf{Q}$ over Q^*, by Theorem 9.9.

We now design a low width ROBP that computes f as follows. The state space of the ROBP will be $\mathbb{F}_2^{n-k'} \times \{0, 1\} \times \{0, 1\}$, and it will start at state $s = (\vec{0}, 0, 0)$. Informally, the first part of the state will keep track of the parity checks $v^{(1)}, \ldots, v^{(n-k')}$, while the remaining two parts will keep track of the parity checks $\hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$ through a more compressed representation (which is enabled by the fact that these basis vectors each have Hamming weight at most 2).

More formally, suppose the ROBP is reading the string $x \in \{0, 1\}^n$, and at time $i - 1$ it arrive at state $(z, b, c) \in \mathbb{F}_2^{n-k'} \times \{0, 1\} \times \{0, 1\}$. Upon reading the next bit x_i, the ROBP will transition to the state (z', b', c'), defined as follows. First, let $u^{(i)} \in \mathbb{F}_2^{n-k'}$ be the string that consists of the i^{th} bit of each of $v^{(1)}, \ldots, v^{(n-k')}$. Then, define $z' := z + x_i \cdot u^{(i)}$.

Next, define $b' := x_i$.

Finally, we define c' as follows. If $c = 1$ then keep c' as 1 (this indicates one of the parity checks $\hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$ has already been violated). If $i = 1$ then keep c' as 0. If $1 < i \leq d$ then set $c = 1$ if and only if $b \neq x_i$ (since this means $\langle x, \hat{e}^{(i-1)} \rangle = 1$). And if $d < i \leq n$, set $c = x_i$ (since $\langle x, \hat{e}^{(i)} \rangle = x_i$).

In the last layer of the ROBP, let all zeroes string $(\vec{0}, 0, 0)$ be the accept state, and every other string in $\mathbb{F}_2^{n-k'} \times \{0, 1\} \times \{0, 1\}$ be a reject state. It is straightforward to verify that the accept state is hit if and only if x has inner product 0 with each of $v^{(1)}, \ldots, v^{(n-k')}, \hat{e}^{(1)}, \ldots, \hat{e}^{(t)}$. Furthermore, this ROBP has width $w = |\mathbb{F}_2^{n-k'} \times \{0, 1\} \times \{0, 1\}| = 2^{n-k'+2} = 2^{\lceil 4d \log n \rceil + 2} < 2^{2\frac{k}{n} \cdot 4d \log n + 3}$, where the last inequality follows from the case condition $k/n > 0.9$. Thus by Theorem 9.9, there is a simple sampler of width $w < 8 \cdot 2^{8 \cdot \frac{kd}{n} \log n}$ that samples the uniform distribution $\mathbf{Q}$ over the linear $[n, k, d]$ code Q^*, as desired. □

This completes the presentation, discussion, and proof of our tight sampling lower bounds for ROBPs against codes. Next, let's move to pursue sampling lower bounds via a completely different technique.

9.6 Sampling lower bounds via direct product theorems

In this section, we present our direct product theorem, which gives a powerful and generic way to obtain sampling lower bounds. We prove the following strong direct product theorem for sampling with ROBPs.

Theorem 9.20 (Direct product theorem - Theorem 9.5, restated)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w, it holds that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$. Then for any $t \in \mathbb{N}$ and ROBP $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width w, it holds that*

$$|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}.$$

We view this result as the second main theorem and contribution of this chapter. At a high level, Theorem 9.20 says that if a distribution $\mathbf{Q} \sim \{0,1\}^n$ has statistical distance $\geq \delta$ from distributions sampled by ROBPs of width w, then $\mathbf{Q}^{\otimes t} \sim \{0,1\}^{nt}$ has statistical distance $\geq 1 - 2^{-\Omega(t\delta^2)}$ from distributions sampled by ROBPs of width w.[17] Notably, this blow-up in statistical distance is experienced without decreasing the power of the sampling model at all (e.g., by decreasing this width), and indeed, this is what we mean when we say that Theorem 9.20 is a *strong* direct product theorem. To start, let us sketch an overview of its proof.

An overview of the proof

To prove the above direct product theorem, we start by proving an analogous result for KRVZ samplers. Then, we use our equivalence theorem to obtain a direct product theorem for sampling with ROBPs. However, since our equivalence theorem (Theorem 9.6) has some loss in parameters (width), this will only yield a *weak* direct product theorem. Namely, the statistical distance still blows up from δ to $1 - 2^{-\Omega(t\delta^2)}$, *but only if* we also require the width to *decrease* from w to $w/14$. We would really like a *strong* direct product theorem, in the sense that the statistical distance blows up even if the ROBP is allowed to keep all of its width w. At the end of this subsection, we show how to build some extra machinery to make this happen.

A direct product theorem for KRVZ samplers We now proceed to sketch the proof of our direct product theorem for KRVZ samplers. Let $\mathbf{X} \sim \{0,1\}^{nt}$ be a distribution generated by a KRVZ sampler of width w, and parse it as $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_t)$, where each $\mathbf{X}_i \sim \{0,1\}^n$ need not be independent. Recall that $\mathbf{Q}^{\otimes t} \sim \{0,1\}^{nt}$ is of the form $\mathbf{Q}^{\otimes t} = (\mathbf{Q}_1, \dots, \mathbf{Q}_t)$, where each $\mathbf{Q}_i \sim \{0,1\}^n$ is an independent copy of $\mathbf{Q}$. We would like to argue

$$|(\mathbf{X}_1, \dots, \mathbf{X}_t) - (\mathbf{Q}_1, \dots, \mathbf{Q}_t)| \geq 1 - 2^{-\Omega(t\delta^2)}, \tag{9.13}$$

given that for any KRVZ sampler $\mathbf{X}' \sim \{0,1\}^n$ of width w it holds that $|\mathbf{X}' - \mathbf{Q}| \geq \delta$. The first observation is that any of the $\mathbf{X}_i \sim \{0,1\}^n$ can be generated by a KRVZ sampler of width w. Indeed, even though it represents a sequence of bits generated in the middle of the KRVZ sampler $\mathbf{X}$, it is easy to create a new KRVZ sampler $\mathcal{B}$ of the same width that only generates $\mathbf{X}_i$, via the following steps.

1. Copy the KRVZ sampler that creates $\mathbf{X}$.

[17] Here, recall that $\mathbf{Q}^{\otimes t}$ refers to a sequence of t independent copies of $\mathbf{Q}$.

2. Throw out all layers that do not produce bits corresponding to $\mathbf{X}_i$.

3. Add a new start vertex v_{start}.

4. Connect v_{start} to the first layer remaining in $\mathcal{B}$, using the appropriate probabilities.

5. Merge the first two layers of $\mathcal{B}$, to deal with the fact that the edges leaving v_{start} currently have no output labels.

Thus, we are guaranteed that for each $\mathbf{X}_i \sim \{0,1\}^n$ and $\mathbf{Q}_i \sim \{0,1\}^n$, it holds that $|\mathbf{X}_i - \mathbf{Q}_i| \geq \delta$ by the theorem hypothesis that every KRVZ sampler of width w is far from $\mathbf{Q}$. The question now is,

Is this enough to guarantee that the statistical distance blows up in Equation (9.13)?

Well, if each $\mathbf{X}_i$ were independent, it is not too hard to show that the answer is yes. However, this is of course not guaranteed to be the case, since the $\mathbf{X}_i$'s are consecutive slices of the same KRVZ sampler $\mathbf{X}$. Indeed, without further examination, it could potentially be the case that for any $x \in \{0,1\}^n$ and $i \in [n]$, the distributions $(\mathbf{X}_{-i} \mid \mathbf{X}_i = x)$ and $(\mathbf{Q}_{-i}^{\otimes t})$ are identical.[18] In this case, we cannot hope to lower bound Equation (9.13) by anything more than δ. In some sense, the above adversarial example represents a situation where each $\mathbf{X}_i$ is *not* contributing its "fair share" to the statistical distance in Equation (9.13). In order to force each $\mathbf{X}_i$ to be a contributing member, we would like a different guarantee than just $|\mathbf{X}_i - \mathbf{Q}_i| \geq \delta$. One natural way to encode the idea that each $\mathbf{X}_i$ is contributing its fair share is to require that for every $i \in [t]$ and $x \in (\{0,1\}^n)^{i-1}$,

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = x) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = x)| \geq \delta. \tag{9.14}$$

This leaves us with two questions:

(i) Given a guarantee like Equation (9.14), can we actually prove Equation (9.13)?

(ii) Is the guarantee given in Equation (9.14) even true?

If we can answer yes to both questions, we are done with our direct product theorem for KRVZ samplers.

It turns out that (i) is true, but it is a little cumbersome to do so using statistical distance. To avoid this, we use simple and well known facts to convert the statement into one about squared Hellinger distance, which can further be phrased in terms of the *Bhattacharyya coefficient*. Phrasing (i) in this way allows for a simple inductive proof, which we can then convert back to a result about statistical distance.

It also turns out that (ii) is true. To see why, note that $(\mathbf{Q}_i \mid \mathbf{Q}_{<i} = x)$ is just the same distribution as $\mathbf{Q} \sim \{0,1\}^n$, since each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$. Furthermore, it is straightforward to show that the distribution $(\mathbf{X}_i \mid \mathbf{X}_{<i} = x)$ can be generated by a width w KRVZ sampler, using a similar idea to the one we presented for why $\mathbf{X}_i$ has this property. Thus the hypothesis of the direct product theorem implies Equation (9.14), and our direct product theorem for KRVZ samplers is complete.

[18] Recall that for a random variable $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_t)$, the notation $\mathbf{X}_{-i}$ denotes $\mathbf{X}$ with $\mathbf{X}_i$ removed.

A direct product theorem for sampling with ROBPs It is now easy to obtain a direct product theorem for ROBP samplers, in a black-box manner, by combining the above direct product theorem with our equivalence theorem between KRVZ samplers and ROBP samplers (Theorem 9.6). However, as discussed at the beginning of this section, this will only yield a weak direct product theorem, since our equivalence theorem suffers a slight loss in parameters (i.e., width). If we want to obtain a *strong* direct product theorem for ROBP samplers, we must dig into the black box.

Looking back at the previous discussion, it is not too difficult to see that if one wants a strong direct product theorem for ROBP samplers (that suffers no loss in width), then it suffices to show the following: for any distribution $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$ sampled by an ROBP of width w, and any $i \in [t], x \in (\{0,1\}^n)^{i-1}$, the distribution $(\mathbf{X}_i \mid \mathbf{X}_{<i} = x)$ can be sampled by an ROBP of width w.

In order to show the above, our key ingredient is the following: for any w and probability distribution $p : [w] \to \mathbb{R}_{\geq 0}$, we construct an ROBP of width w such that a random walk over it hits the i^{th} vertex in the last layer with probability $p(i) + \gamma$, where $\gamma > 0$ can be arbitrarily small. A first attempt at constructing such an ROBP might use a *multi-thresholding function*, which sorts Boolean strings into buckets of various sizes - similar to the one from our equivalence theorems (Section 9.3.3). However, recall that our construction of such a function required width $2w$ (instead of the desired w), and we showed this was tight up to additive constants (Lemma 9.6).

To get the width down to w, we start by showing that for any biased coin $\mathbf{A} \sim \{0,1\}$, there is an ROBP of width 2 that samples a distribution arbitrarily close to it. At a high level, this argument works as follows: for any binary string $b \in \{0,1\}^\ell$, we show how to use its bits as "instructions" to construct a certain ROBP of length ℓ and width 2 in a layer-by-layer fashion. The constructed ROBP then guarantees that it accepts a random string with probability b, where b is interpreted as the binary representation of a number in $[0,1]$. Finally, it is not too hard to bootstrap such an object to create our desired ROBP of width w that hits the vertices in its last layer with probabilities close to $\{p(i)\}_{i \in [w]}$. As a result, we get our strong direct product theorem for sampling with ROBPs.

A formal presentation of the proof

Now that we have a high-level idea of how our proof will work, we are ready to get started with its formalization. Towards the end, we start by proving a simple new lemma on amplifying statistical distance in Section 9.6.1. Then, in Section 9.6.2, we show how this lemma can be combined with a basic fact about KRVZ samplers to obtain a strong direct product theorem for KRVZ samplers. Finally, in Section 9.6.3, we start by showing how this result can be combined with our KRVZ-ROBP equivalence theorem (Theorem 9.6) to obtain a weak direct product theorem for sampling with ROBPs. Then, we show how to turn this into a strong direct product theorem by employing some more involved tools, ultimately proving Theorem 9.20.

9.6.1 A simple lemma on amplifying statistical distance

As promised, let us begin with our simple new lemma on amplifying statistical distance, which is one of the key ingredients in the proofs of our direct product theorems. In fact, in some sense, this simple lemma can *itself* be viewed as a direct product theorem for sequences of somewhat-dependent random variables.

Lemma 9.9 (Lemma 9.1, restated)**.** *Let $\mathbf{X} \sim V^n$ and $\mathbf{Y} \sim V^n$ each be a sequence of n random variables over V, where elements in the sequence need not be independent. Suppose that for any $i \in [n]$ and $v \in V^{i-1}$,*

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)| \geq \delta.$$

Then

$$|\mathbf{X} - \mathbf{Y}| \geq 1 - e^{-n\delta^2/2}.$$

As it turns out, this lemma will be a little cumbersome to prove using statistical distance. Thus, we will convert the statement into one about more amenable notions of distance. The first such measure we consider is the *squared Hellinger distance*. Given two random variables $\mathbf{X}, \mathbf{Y}$ over some discrete space V, the squared Hellinger distance between $\mathbf{X}, \mathbf{Y}$ is denoted $H^2(\mathbf{X}, \mathbf{Y})$ and defined as follows:

$$H^2(\mathbf{X}, \mathbf{Y}) := \frac{1}{2} \sum_{v \in V} \left(\sqrt{\Pr[\mathbf{X} = v]} - \sqrt{\mathbf{Y} = v} \right)^2.$$

We would now like to express Lemma 9.9 using this more well-behaved notion of distance. For this, we can use the following fact, which is well-known and straightforward to show using Cauchy-Schwarz. It gives us estimates on statistical distance in terms of squared Hellinger distance.

Fact 9.4. *For any discrete random variables $\mathbf{X}, \mathbf{Y} \sim V$,*

$$H^2(\mathbf{X}, \mathbf{Y}) \leq |\mathbf{X} - \mathbf{Y}| \leq \sqrt{2} H(\mathbf{X}, \mathbf{Y}).$$

We now have the tools necessary to convert Lemma 9.9 into a statement about squared Hellinger distance. However, it turns out that there is another related measure of distance/similarity that will make Lemma 9.9 even easier to prove. It is known as the *Bhattacharyya coefficient*. Given two random variables $\mathbf{X}, \mathbf{Y}$ over some discrete space V, the Bhattacharyya coefficient between $\mathbf{X}$ and $\mathbf{Y}$ is denoted $\mathsf{BC}(\mathbf{X}, \mathbf{Y})$ and defined as follows:

$$\mathsf{BC}(\mathbf{X}, \mathbf{Y}) := \sum_{v \in V} \sqrt{\Pr[\mathbf{X} = v] \cdot \Pr[\mathbf{Y} = v]}.$$

It is easy to verify, via the definitions of squared Hellinger distance and Bhattacharyya coefficient, that $1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) = H^2(\mathbf{X}, \mathbf{Y})$. We can combine this observation with Fact 9.4 to obtain the following estimates on statistical distance in terms of the Bhattacharyya coefficient.

Fact 9.5. *For any discrete random variables $\mathbf{X}, \mathbf{Y} \sim V$,*

$$1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) \leq |\mathbf{X} - \mathbf{Y}| \leq \sqrt{2}\sqrt{1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y})}.$$

The goal now is to prove a version of Lemma 9.9 that uses the Bhattacharyya coefficient. We can then combine this result with Fact 9.5 to prove Lemma 9.9.

Lemma 9.10. *Let $\mathbf{X} \sim V^n$ and $\mathbf{Y} \sim V^n$ each be a sequence of n random variables over V, where elements in the sequence need not be independent. Suppose that for any $i \in [n]$ and $v \in V^{i-1}$,*

$$\mathsf{BC}((\mathbf{X}_i \mid \mathbf{X}_{<i} = v), (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)) \leq \delta.$$

Then

$$\mathsf{BC}(\mathbf{X}, \mathbf{Y}) \leq \delta^n.$$

Proof. We prove the slightly stronger statement that for any $i \in [n]$, it holds that $\mathsf{BC}(\mathbf{X}_{\leq i}, \mathbf{Y}_{\leq i}) \leq \delta^i$. We prove the result by induction on i. The base case $i = 1$ is immediate from the hypothesis. For the case $i \geq 2$

we have

$$\begin{aligned}
\mathsf{BC}(\mathbf{X}_{\le i}, \mathbf{Y}_{\le i}) &= \sum_{v \in V^i} \sqrt{\Pr[\mathbf{X}_{\le i} = v] \Pr[\mathbf{Y}_{\le i} = v]} \\
&= \sum_{v \in V^{i-1}} \sum_{b \in V} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{X}_i = b \mid \mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v] \Pr[\mathbf{Y}_i = b \mid \mathbf{Y}_{<i} = v]} \\
&= \sum_{v \in V^{i-1}} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v]} \sum_{b \in V} \sqrt{\Pr[\mathbf{X}_i = b \mid \mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_i = b \mid \mathbf{Y}_{<i} = v]} \\
&= \sum_{v \in V^{i-1}} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v]} \cdot \mathsf{BC}((\mathbf{X}_i \mid \mathbf{X}_{<i} = v), (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)) \\
&\le \delta \cdot \sum_{v \in V^{i-1}} \sqrt{\Pr[\mathbf{X}_{<i} = v] \Pr[\mathbf{Y}_{<i} = v]} \\
&= \delta \cdot \mathsf{BC}(\mathbf{X}_{<i}, \mathbf{Y}_{<i}) \\
&\le \delta \cdot \delta^{i-1} \\
&= \delta^i,
\end{aligned}$$

where the inequalities use the lemma hypothesis and the induction hypothesis. □

We now combine the above lemma with our estimates from Fact 9.5 to prove Lemma 9.9.

Proof of Lemma 9.9. By combining the lemma hypothesis with Fact 9.5, we know that for any i, v,

$$\begin{aligned}
\mathsf{BC}((\mathbf{X}_i \mid \mathbf{X}_{<i} = v), (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)) &\le 1 - \frac{|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Y}_i \mid \mathbf{Y}_{<i} = v)|^2}{2} \\
&\le 1 - \delta^2/2.
\end{aligned}$$

Thus by Lemma 9.10 we have

$$\mathsf{BC}(\mathbf{X}, \mathbf{Y}) \le (1 - \delta^2/2)^n \le e^{-n\delta^2/2},$$

where we use the standard inequality $1 + x \le e^x$ for all real x. Using Fact 9.5 once more we get

$$|\mathbf{X} - \mathbf{Y}| \ge 1 - \mathsf{BC}(\mathbf{X}, \mathbf{Y}) \ge 1 - e^{-n\delta^2/2},$$

as desired. □

9.6.2 A direct product theorem for KRVZ samplers

We now have everything we need in order to prove a strong direct product theorem for KRVZ samplers.

Theorem 9.21. *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ of width w, it holds that $|\mathbf{X} - \mathbf{Q}| \ge \delta$. Then for any KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^{nt}$ of width w,*

$$|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \ge 1 - e^{-t\delta^2/2}.$$

To prove this result, we will simply combine our lemma on amplifying statistical distance (Lemma 9.9) with the following claim, which is straightforward to verify using the definition of KRVZ sampler.

Claim 9.1 (KRVZ samplers are closed under slicing and preconditioning)**.** *Let $\mathbf{X} \sim \{0,1\}^n$ be a KRVZ sampler of width w. Then for any $1 \leq i \leq j \leq n$ and any $x \in \{0,1\}^{i-1}$, the distribution*

$$(\mathbf{X}_{i \to j} \mid \mathbf{X}_{<i} = x)$$

is also a KRVZ sampler of width w.

It is now not too difficult to formally prove our strong direct product theorem for KRVZ samplers.

Proof of Theorem 9.21. Parse $\mathbf{X}^*$ as $(\mathbf{X}_1, \ldots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$, and parse $\mathbf{Q}^{\otimes t}$ as $(\mathbf{Q}_1, \ldots, \mathbf{Q}_t) \sim (\{0,1\}^n)^t$. By definition, each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$. And by Claim 9.1, we know that for any fixing of $\mathbf{X}_{<i}$, the random variable $\mathbf{X}_i$ is a KRVZ sampler of width w. Thus for any $i \in [t]$ and $v \in (\{0,1\}^n)^{i-1}$, we have

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)| \geq \delta$$

by the hypothesis $|\mathbf{X} - \mathbf{Q}| \geq \delta$. Applying Lemma 9.9 completes the proof. □

9.6.3 A direct product theorem for ROBP samplers

Finally, we are ready to prove our main direct product theorem, which we restate below for convenience.

Theorem 9.22 (Theorem 9.20, restated)**.** *Let $\mathbf{Q} \sim \{0,1\}^n$ be a distribution such that for any ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w, it holds that $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$. Then for any $t \in \mathbb{N}$ and ROBP $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width w, it holds that*

$$|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}.$$

In order to prove this result, we would like to apply our KRVZ-ROBP equivalence theorem (Theorem 9.6) with our direct product theorem from KRVZ samplers (Theorem 9.21). Such a proof would look roughly as follows: first, if every ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w has $|F(\mathbf{U}_\ell) - \mathbf{Q}| \geq \delta$, then by the second bullet of Theorem 9.6, every KRVZ sampler $\mathbf{X} \sim \{0,1\}^n$ of width $w/7$ has roughly $|\mathbf{X} - \mathbf{Q}| \geq \delta$. Then, by Theorem 9.21, we know that every KRVZ sampler $\mathbf{X}^* \sim \{0,1\}^{nt}$ of width $w/7$ has roughly $|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}$. Finally, by the first bullet of Theorem 9.6, every ROBP $F : \{0,1\}^{\ell^*} \to \{0,1\}^{nt}$ of width $w/14$ must have $|F^*(\mathbf{U}_{\ell^*}) - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}$.

The above argument will give us statistical distance lower bounds of the form that we would like, but we lose a factor of 14 in the width. We would like to avoid this, and keep the direct product theorem *strong*, in the sense that the width need not decrease at all. Towards this end, the goal is to prove a version of Claim 9.1 for sampling using ROBPs. Just as in the proof of our direct product theorem for KRVZ samplers, we will then be able to combine this result with our lemma on amplifying statistical distance from Section 9.6.1 in order to prove Theorem 9.22. Thus, we proceed towards proving such a result, which will be significantly more challenging to prove than Claim 9.1 (which was easy to verify by the definition of KRVZ sampler).

Claim 9.2 (ROBP samplers are closed under slicing and preconditioning)**.** *Let $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^n$ be an ROBP of width w, and define $\mathbf{X} := F^*(\mathbf{U}_{\ell^*})$. For any $1 \leq i \leq j \leq n$ and any $x \in \{0,1\}^{i-1}$, and any $\varepsilon > 0$, the following holds. There exists an ROBP $F : \{0,1\}^\ell \to \{0,1\}^{j-i+1}$ of width w and length $\ell = \ell^* + 3w\log(w/\varepsilon)$ such that*

$$|F(\mathbf{U}_\ell) - (\mathbf{X}_{i \to j} \mid \mathbf{X}_{<i} = x)| \leq \varepsilon.$$

A natural approach to prove Claim 9.2 is to build the ROBP $F : \{0,1\}^\ell \to \{0,1\}^{j-i+1}$ by taking the appropriate slice of the ROBP F^* and simulating the conditioning $\mathbf{X}_{<i} = x$ by prepending another ROBP. The exact type of ROBP we would like to prepend should compute a function $g : \{0,1\}^{\ell'} \to [w]$ such that $g(\mathbf{U}_{\ell'}) = \alpha$ with roughly the same probability that F^* hits vertex α in layer i, conditioned on $\mathbf{X}_{<i} = x$.

To make things more formal, we will use a Σ-ROBP (Definition 9.2). In particular, we would like to construct a Σ-ROBP of width w that computes the g described above. Towards this end, one idea is to define g to compute a *multi-thresholding function*, which roughly splits the hypercube $\{0,1\}^{\ell'}$ into w consecutive buckets of an appropriate size, and on input x outputs the label of the bucket it falls into. Such a construction can indeed be used to compute the appropriate probabilities, and we use this in Section 9.3.3 to prove the equivalence between KRVZ and ROBP samplers. However, our Σ-ROBP from that section requires width $2w$, and we show that this is tight (Lemma 9.6). To compute g using width w, we need a new idea. Before presenting this new idea and proving Claim 9.2, let us show how it can be combined with our simple new lemma on amplifying statistical distance (Lemma 9.9) to prove Theorem 9.22.

Proof of Theorem 9.22. Let $\mathbf{X}^* = F^*(\mathbf{U}_{\ell^*})$. Parse $\mathbf{X}^*$ as $(\mathbf{X}_1, \dots, \mathbf{X}_t) \sim (\{0,1\}^n)^t$, and parse $\mathbf{Q}^{\otimes t}$ as $(\mathbf{Q}_1, \dots, \mathbf{Q}_t) \sim (\{0,1\}^n)^t$. Now, fix any $i \in [t]$ and $v \in (\{0,1\}^n)^{i-1}$. We want to get a lower bound on $|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)|$. Towards this end, set $\varepsilon := \delta \cdot (1 - 1/\sqrt{2})$. By Claim 9.2, we know that there exists an ROBP $F : \{0,1\}^\ell \to \{0,1\}^n$ of width w and length $\ell = \ell^* + 3w \log(\frac{\sqrt{2}w}{(\sqrt{2}-1)\delta})$ such that

$$|F(\mathbf{U}_\ell) - (\mathbf{X}_i \mid \mathbf{X}_{<i} = v)| \leq \delta \cdot (1 - 1/\sqrt{2})$$

Furthermore, by the theorem hypothesis, we know that $|F(\mathbf{U}_\ell) - \mathbf{Q}_i| \geq \delta$. Thus by the triangle inequality we have $|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - \mathbf{Q}_i| \geq \delta/\sqrt{2}$. And since each $\mathbf{Q}_i$ is an independent copy of $\mathbf{Q}$, we of course have $\mathbf{Q}_i = (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)$ and thus

$$|(\mathbf{X}_i \mid \mathbf{X}_{<i} = v) - (\mathbf{Q}_i \mid \mathbf{Q}_{<i} = v)| \geq \delta/\sqrt{2}.$$

By applying Lemma 9.9, we immediately get that $|\mathbf{X}^* - \mathbf{Q}^{\otimes t}| \geq 1 - e^{-t\delta^2/4}$, which completes the proof. Note that we can actually make the constant 4 arbitrarily close to 2 by picking a small enough ε, since the length $\ell = \ell^* + 3w \log(w/\varepsilon)$ of the ROBP F does not matter. □

Our goal now is to prove Claim 9.2. The new main ingredient we use is a Σ-ROBP that can achieve roughly the same sampling task that is achieved by our Σ-ROBP for the multi-thresholding function from Lemma 9.6, but using less width (while introducing a tiny amount of error). We prove the following.

Lemma 9.11 (Key ingredient for Theorem 9.22)**.** *For any distribution $\mathbf{X} \sim [w]$ and $\varepsilon > 0$, there exists a Σ-ROBP $f : \{0,1\}^\ell \to [w]$ of width w and length $\ell = 3w \log(w/\varepsilon)$ such that*

$$|f(\mathbf{U}_\ell) - \mathbf{X}| \leq \varepsilon.$$

This result says that any distribution over a support of size w can be sampled (with arbitrary precision) by a Σ-ROBP of width w. In other words, this can be viewed as an *ROBP version* of Fact 9.1, which claims the same result for simple samplers. Now, before we prove Lemma 9.11, let us see how it can be used to show that ROBP samplers are closed under slicing and preconditioning (Claim 9.2).

Proof of Claim 9.2. Let $F^* : \{0,1\}^{\ell^*} \to \{0,1\}^n$ be an ROBP of width w, and define $\mathbf{X} := F^*(\mathbf{U}_{\ell^*})$. Let $G = (V, E)$ be the graph underlying this ROBP with layers $V = V_0 \cup V_1 \cup \cdots \cup V_{\ell^*}$. Let γ_1 denote the number of output bits labeling each edge into V_1, let γ_2 denote the number of output bits labeling each edge into V_2, and so on. Note that $\sum_{i\in[\ell^*]}\gamma_i = n$.

Now, fix any $1 \leq i \leq j \leq n$ and $x \in \{0,1\}^{i-1}$ and $\varepsilon > 0$. The claim is easy to show if $i = 1$, so we henceforth assume $i > 1$. Recall that the goal is to construct an ROBP F such that $|F(\mathbf{U}_\ell) - (\mathbf{X}_{i\to j} \mid \mathbf{X}_{<i} = x)|$ is small. Towards this end, let $\alpha \leq \beta \in [\ell^*]$ be such that the i^{th} bit of $\mathbf{X}$ is outputted upon entering layer V_α, and the j^{th} bit of $\mathbf{X}$ is outputted upon entering layer V_β. That is, α is the smallest integer such that $\sum_{h\in[\alpha]}\gamma_h \geq i$ and β is the smallest integer such that $\sum_{h\in[\beta]}\gamma_h \geq j$.

We now construct a new branching program $G' = (V', E')$ that will eventually help us construct F. The layers of this new branching program are $V' = V'_{\alpha-1} \cup V'_\alpha \cup \cdots \cup V'_\beta$, where each V'_i is a copy of V_i from the original branching program G. The edge set E' of the new branching program will include all the edges from E that traverse between these layers (including their input and output labels), and nothing more.

We now perform a slight modification to the edges in E'. First, recall that by definition of each γ_h, we have $\sum_{h\in[\alpha-1]}\gamma_h \leq i - 1 < \sum_{h\in[\alpha]}\gamma_h$. Let $r := (i-1) - \sum_{h\in[\alpha-1]}\gamma_h \geq 0$ denote the number of bits on the edges into layer V'_α that will eventually by fixed by fixing $\mathbf{X}_{<i} = x$. Now, for each $v \in V'_{\alpha-1}$, do the following: consider its outgoing edges e_0 and e_1 with input labels 0 and 1, respectively. We will now examine whether each of these edges have the property that the first r bits in their output label match the last r bits of x. If neither of e_0, e_1 have this property, do nothing. If both of e_0, e_1 have this property, do nothing. If e_0 has this property but e_1 does not, then delete e_1 and replace it with a new copy of e_0 (this copy should connect the same vertices as e_0, it should have the same output label as e_0, but it should have the input label 1). If e_1 has this property but e_0 does not, then delete e_0 and replace it with a new copy of e_1 (this copy should connect the same vertices as e_1, it should have the same output label as e_1, but it should have the input label 0).

Our last modification to the edges of E' will be as follows. First, let $r' := \sum_{h\in[\beta]}\gamma_h - j$. We will erase the first r bits and last r' bits output by G'. In particular, for every edge entering V'_α, delete the first r bits of its output label. And for every edge entering V'_β, delete the last r' bits of its output label.

Now, label the vertices in $V_{\alpha-1}$ as $v_1, \ldots, v_w$, and let $v'_1, \ldots, v'_w$ denote the corresponding vertices in $V'_{\alpha-1}$. Consider again the original ROBP $G = (V, E)$. For each $s \in [w]$, let p_s denote the probability that a (uniform) random walk from the start vertex of G hits v_s, *conditioned on the event that the first* $i - 1$ *bits output by the random walk exactly match* x. Next, consider the new ROBP G'. For each $s \in [w]$, let $\mathbf{Y}_s \sim \{0,1\}^{j-i+1}$ denote the distribution generated by taking a (uniform) random walk over G', starting at vertex $v'_s \in V'_{\alpha-1}$ and outputting the output labels seen along the way. It is now straightforward to verify

$$\sum_{s\in[w]} p_s \cdot \mathbf{Y}_s = (\mathbf{X}_{i\to j} \mid \mathbf{X}_{<i} = x).$$

We would now like to construct a new ROBP that (almost) generates the distribution $\sum_{s\in[w]} p_s \cdot \mathbf{Y}_s$. First, let $\mathbf{A} \sim [w]$ denote the random variable corresponding to the distribution $\{p_s\}_{s\in[w]}$. By Lemma 9.11, there is a Σ-ROBP $f : \{0,1\}^{\ell'} \to [w]$ of width w and length $\ell' = 3w\log(w/\varepsilon)$ such that $|f(\mathbf{U}_{\ell'}) - \mathbf{A}| \leq \varepsilon$. For each $s \in [w]$, let $q_s := \Pr[f(\mathbf{U}_{\ell'}) = s]$. Now, let $G'' = (V'', E'')$ denote its underlying graph, which has layers $V'' = V''_0 \cup V''_1 \cup \cdots \cup V''_{\ell'}$. Label the vertices in $V''_{\ell'}$ as $u_1, \ldots, u_w$, so that a random walk on G'' hits u_s with probability q_s.

We are finally ready to construct the ROBP $F : \{0,1\}^\ell \to \{0,1\}^{j-i+1}$ advertised in the claim statement. Its underlying graph $G_F = (V_F, E_F)$ is formed as follows: first, set $V_F = V''_0 \cup \cdots \cup V''_{\ell'} \cup V'_{\alpha-1} \cup \cdots \cup V'_\beta$. Then, set $E_F = E'' \cup E'$. Finally, merge layers $V_{\ell''}$ and $V'_{\alpha-1}$ by identifying each $u_s \in V_{\ell''}$ with $v'_s \in V'_{\alpha-1}$. It is straightforward to verify that $F(\mathbf{U}_\ell) = \sum_{s\in[w]} q_s \cdot \mathbf{Y}_s$. Furthermore, notice that F has width w and length $\ell = \ell' + \beta - \alpha + 1 \leq \ell' + \ell^* = 3w\log(w/\varepsilon) + \ell^*$.

Thus, all that remains is to show $|\sum_{s\in[w]} q_s \cdot \mathbf{Y}_s - \sum_{s\in[w]} p_s \cdot \mathbf{Y}_s| \leq \varepsilon$. Using the definition of statistical distance, we have

$$\begin{aligned}
|\sum_s q_s\mathbf{Y}_s - \sum_s p_s\mathbf{Y}_s| &= \max_{T\subseteq\{0,1\}^{j-i+1}} (\Pr[\sum_s q_s\mathbf{Y}_s \in T] - \Pr[\sum_s p_s\mathbf{Y}_s \in T]) \\
&= \max_T \sum_s \Pr[\mathbf{Y}_s \in T] \cdot (q_s - p_s) \\
&\leq \max_T \sum_{s\,:\,q_s - p_s \geq 0} \Pr[\mathbf{Y}_s \in T] \cdot (q_s - p_s) \\
&\leq \sum_{s\,:\,q_s - p_s \geq 0} (q_s - p_s) \\
&= |f(\mathbf{U}_{\ell'}) - \mathbf{A}| \\
&\leq \varepsilon,
\end{aligned}$$

as desired. □

At last, all that remains is to prove our key ingredient, Lemma 9.11, which shows that any distribution over a support of size w can be sampled with arbitrary precision by a Σ-ROBP of width w. We do so, below.

Proof of Lemma 9.11. We would like to show that for any distribution $\mathbf{X} \sim [w]$ and $\varepsilon > 0$, there exists a Σ-ROBP $f : \{0,1\}^\ell \to [w]$ of width w and length $\ell = 3w\log(w/\varepsilon)$ such that $|f(\mathbf{U}_\ell) - \mathbf{X}| \leq \varepsilon$. Without loss of generality, we assume that $\Pr[\mathbf{X} = i] > 0$ for each $i \in [w]$. Furthermore, suppose for now that for any distribution $\mathbf{Y} \sim \{0,1\}$ and $\varepsilon' > 0$, there exists an ROBP $f' : \{0,1\}^{\ell'} \to \{0,1\}$ of width 2 and length $\ell' = \lceil\log(1/\varepsilon')\rceil$ such that $|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| \leq \varepsilon'$. We first show how this can be used to obtain the desired result, and then we show how to construct such an ROBP.

From width 2 to width w Let us return to the original distribution $\mathbf{X} \sim [w]$ that we would like to sample with error ε. For each $i \in [w-1]$, define a random variable $\mathbf{A}_i \sim \{0,1\}$ as follows. We set $\mathbf{A}_i = 0$ with probability $\Pr[\mathbf{X} = i \mid \mathbf{X} \geq i]$ and we set $\mathbf{A}_i = 1$ with probability $\Pr[\mathbf{X} > i \mid \mathbf{X} \geq i]$. Observe that

$$\Pr[\mathbf{X} = i] = \begin{cases} \Pr[\mathbf{A}_i = 0] \cdot \prod_{j<i} \Pr[\mathbf{A}_j = 1] & \text{if } i < w, \\ \Pr[\mathbf{A}_{i-1} = 1] \cdot \prod_{j<i-1} \Pr[\mathbf{A}_j = 1] & \text{if } i = w. \end{cases} \tag{9.15}$$

Now, for every $i \in [w-1]$, let $f_i' : \{0,1\}^{\ell'} \to \{0,1\}$ be an ROBP of width 2 and length $\ell' = \lceil\log(1/\varepsilon')\rceil$ such that $|f_i'(\mathbf{U}_{\ell'}) - \mathbf{A}_i| \leq \varepsilon'$. For convenience, we let $\mathbf{B}_i := f_i'(\mathbf{U}_{\ell'})$ so that $|\mathbf{A}_i - \mathbf{B}_i| \leq \varepsilon'$. Now, let $G^i = (V^i, E^i)$ denote the underlying graph of f_i' with layers $V^i = V_0^i \cup \cdots \cup V_{\ell'}^i$. Let start$^i \in V_0^i$ denote its start state, let accept$^i \in V_{\ell'}^i$ denote its accept state, and let reject$^i \in V_{\ell'}^i$ denote its reject state.

The goal now is to combine these ROBPs into one large ROBP that computes $f : \{0,1\}^\ell \to [w]$. To do so, we construct its underlying graph $G = (V, E)$ as follows. First, we concatenate all of the above ROBPs in a series configuration. That is, we set

$$V = (V_0^1 \cup \cdots \cup V_{\ell'}^1) \cup (V_0^2 \cup \cdots \cup V_{\ell'}^2) \cup \cdots \cup (V_0^{w-1} \cup \cdots \cup V_{\ell'}^{w-1}). \tag{9.16}$$

Next, we add all the edges $\bigcup_{i\in[w-1]} E^i$ to E. Then, for every $i \in [w-2]$, we draw two edges from accept^i to start^{i+1}, and give them input labels 0, 1, respectively. Next, for every $i \in [w-2]$, we do the following: for every layer W appearing (strictly) to the right of layer $V_{\ell'}^i$ in Equation (9.16), add a node called bucket^i_W. Then, draw two edges (labeled 0 and 1) from reject^i to bucket^i_W, where W is the layer immediately following $V^i_{\ell'}$. Then, for any consecutive layers W, W' that contain nodes $\mathsf{bucket}^i_W, \mathsf{bucket}^i_{W'}$, draw two edges (labeled 0 and 1) from bucket^i_W to $\mathsf{bucket}^i_{W'}$. Finally, for each $i \in [w-2]$, give the node $\mathsf{bucket}^i_{V^{w-1}_{\ell'}}$ the output label i; give the node reject^{w-1} the output label $w-1$; and give the node accept^{w-1} the output label w. This completes the construction of $G = (V, E)$ and $f : \{0,1\}^\ell \to [w]$.

Notice that the widest layer in G is $V^{w-1}_{\ell'}$, and it has width $2 + (w-2) = w$. Thus, G has width w and length $\ell = \ell' + (1+\ell')(w-2)$. In fact, notice we can contract the "trivial" edges between layers $V^i_{\ell'}, V^{i+1}_0$ for every $i \in [w-2]$, without changing the output distribution, thereby making the overall length $\ell = \ell' \cdot (w-1) \leq \ell' \cdot w$.

Now, set $\mathbf{X}' = f(\mathbf{U}_\ell)$ and observe via the above construction that

$$\Pr[\mathbf{X}' = i] = \begin{cases} \Pr[\mathbf{B}_i = 0] \cdot \prod_{j<i} \Pr[\mathbf{B}_j = 1] & \text{if } i < w, \\ \Pr[\mathbf{B}_{i-1} = 1] \cdot \prod_{j<i-1} \Pr[\mathbf{B}_j = 1] & \text{if } i = w. \end{cases} \tag{9.17}$$

Our goal now is to upper bound $|\mathbf{X}' - \mathbf{X}|$ using Equations (9.15) and (9.17). Towards this end, suppose we have a sequence of probabilities $p_1, \ldots, p_i$ and another sequence of probabilities $q_1, \ldots, q_i$ such that each q_j is at most $p_j + \varepsilon'$. It is then straightforward to use a hybrid-type argument to verify that $q_1 \cdot q_2 \cdots q_i \leq p_1 \cdot p_2 \cdots p_i + i \cdot \varepsilon'$. Thus, recalling that each $|\mathbf{A}_i - \mathbf{B}_i| \leq \varepsilon'$, we know by Equations (9.15) and (9.17) that for every $i \in [w]$,

$$\Pr[\mathbf{X}' = i] - \Pr[\mathbf{X} = i] \leq w \cdot \varepsilon'.$$

Thus, we have

$$\begin{aligned} |\mathbf{X}' - \mathbf{X}| &= \max_{T \subseteq [w]} \Pr[\mathbf{X}' \in T] - \Pr[\mathbf{X} \in T] \\ &= \max_{T \subseteq [w]} \sum_{i \in T} (\Pr[\mathbf{X}' = i] - \Pr[\mathbf{X} = i]) \\ &\leq \max_{T \subseteq [w]} \sum_{i \in T} w\varepsilon' \\ &\leq w^2 \cdot \varepsilon'. \end{aligned}$$

Finally, we see that if we set $\varepsilon' = \varepsilon / w^2$, then we have a Σ-ROBP $f : \{0,1\}^\ell \to [w]$ of width w and length $\ell \leq \ell' \cdot w = \lceil \log(1/\varepsilon') \rceil \cdot w = \lceil \log(w^2/\varepsilon) \rceil \cdot w \leq 3w \log(w/\varepsilon)$ such that $|f(\mathbf{U}_\ell) - \mathbf{X}| \leq \varepsilon$, as desired.[19]

Building a Σ-ROBP of width 2 All that remains is to show the claim we assumed at the beginning of this proof. Namely, that for any distribution $\mathbf{Y} \sim \{0,1\}$ and $\varepsilon' > 0$, there exists an ROBP $f' : \{0,1\}^{\ell'} \to \{0,1\}$ of width 2 and length $\ell' = \lceil \log(1/\varepsilon') \rceil$ such that $|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| \leq \varepsilon'$.

To prove the above, we specify the underlying graph $G' = (V', E')$ of f' as follows. The graph will consist of layers $V' = V'_0 \cup V'_1 \cup \cdots \cup V'_{\ell'}$, where each V'_i consists of vertices labeled u_i, v_i. We label u_0 as the start vertex, $u_{\ell'}$ as the reject state (outputs 0), and $v_{\ell'}$ as the accept state (outputs 1). Define $p := \Pr[\mathbf{Y} = 0]$ and assume without loss of generality that $0 < p < 1$. Next, we specify the edges E' of G'.

Let $b \in \{0,1\}^{\ell'}$ denote a parameter that we will specify later. Using b, we construct the edges entering each layer V'_i as follows. For every $i \in [\ell']$, do the following: if $b_i = 0$, draw two parallel edges of the form

[19] We remark that the w inside the log can be removed through a slightly more technical construction, where the ROBPs f'_i's are selected to take into account the errors made by the earlier f'_i's.

(v_{i-1}, v_i) and give them input labels $0, 1$; then, draw edges (u_{i-1}, u_i) with input label 0 and (u_{i-1}, v_i) with input label 1. On the other hand, if $b_i = 1$, then draw two parallel edges of the form (u_{i-1}, u_i) and give them input labels $0, 1$; then, draw edges (v_{i-1}, v_i) with input label 0 and (v_{i-1}, u_i) with input label 1.

Consider now a (uniform) random walk over G', starting at the start vertex u_0. For every $i \in [\ell']$, let q_i denote the probability that this random walk hits vertex u_i. Of course, the probability that the random walk hits v_i is then $1 - q_i$. Define $q_0 = 1$, and observe via our construction that the following holds for all $i \in [\ell']$:

$$q_i = \begin{cases} \frac{1}{2} \cdot q_{i-1} & \text{if } b_i = 0, \\ q_{i-1} + \frac{1}{2} \cdot (1 - q_{i-1}) = \frac{1}{2} \cdot (q_{i-1} + 1) & \text{if } b_i = 1, \end{cases}$$

which can be expressed more concisely as

$$q_i = \frac{1}{2} \cdot (q_{i-1} + b_i).$$

Recalling that $q_0 = 1$, it is then straightforward to show via induction that for any $i \in [\ell']$,

$$q_i = 2^{-i} + \sum_{h \in [i]} b_{i+1-h} \cdot 2^{-h}.$$

Since $u_{\ell'}$ is the reject state, we have

$$\Pr[f'(\mathbf{U}_{\ell'}) = 0] = q_{\ell'} = 2^{-\ell'} + \sum_{h \in [\ell']} b_{\ell'+1-h} \cdot 2^{-h}.$$

But observe that this quantity is simply a decimal written in binary using $\{b_1, b_2, \dots, b_{\ell'}\}$. In particular, if we consider any $0 < \tau < 1$ that can be written as $\tau = 2^{-\ell'} \cdot K$ for some $K \in \mathbb{N}$, then we can always find some $b \in \{0,1\}^{\ell'}$ such that the right hand side above evaluates to exactly τ. Now, recall that $p = \Pr[\mathbf{Y} = 0]$. Pick the smallest $K \in \mathbb{N}$ such that $2^{-\ell'} K \geq p > 0$, and pick $b \in \{0,1\}^{\ell'}$ such that $\Pr[f'(\mathbf{U}_{\ell'}) = 0] = 2^{-\ell'} K$. Then we must have

$$\Pr[\mathbf{Y} = 0] \leq \Pr[f'(\mathbf{U}_{\ell'}) = 0] < \Pr[\mathbf{Y} = 0] + 2^{-\ell'}.$$

In other words,

$$|f'(\mathbf{U}_{\ell'}) - \mathbf{Y}| < 2^{-\ell'}. \tag{9.18}$$

Finally, recall that the ROBP we constructed for f' had width 2, and thus we may set $\ell' = \lceil \log(1/\varepsilon') \rceil$ to upper bound Equation (9.18) by ε' and complete the proof. □

This fully completes our presentation, discussion, and proofs of not only our direct product theorems, but all the main theorems in this chapter (Theorems 9.1 to 9.5). Next, we briefly discuss two bonus applications.

9.7 Applications in data structures

As discussed, one of the main reasons to study sampling lower bounds is that they are strictly harder to obtain than classical lower bounds. Indeed, in Section 9.4.1 we demonstrated an explicit distribution that is easily samplable by ROBPs, but whose corresponding function is hard to compute. Furthermore, in Section 9.3.3, we showed that sampling lower bounds against ROBPs immediately yield computing lower bounds against ROBPs, which hold even *on average*. Thus, a natural application of our sampling lower bounds produces a rich host of *computing* lower bounds. However, in addition to this natural application of our sampling lower bounds, we also obtain results in a much more surprising domain: *data structure lower bounds*.

9.7.1 Data structure lower bounds for storing codewords

In the seminal paper of Viola [Vio12a] that launched a systematic study into the complexity of sampling, he observed that sampling lower bounds can in fact be used to obtain data structure lower bounds. Following his work, both of the papers that establish sampling lower bounds against codes for AC^0 circuits [LV12, BIL12] use Viola's observation to obtain data structure lower bounds against storing codewords (and retrieving them using AC^0 circuits). In this tradition, we combine our new sampling lower bounds (Theorem 9.16) with Viola's observation to get analogous data structure lower bounds in the context of ROBPs.

Corollary 9.4 (Data structure lower bounds)**.** *Let $Q \subseteq \{0,1\}^n$ be a (ρ, L) list decodable code of dimension k. Suppose that we can store the codewords of Q using only $k+r$ bits so that a codeword can be computed by an ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width w. Then*

$$r \geq \frac{\rho}{1+2\rho} \cdot k - \log(wL) - 3.$$

We repeat the proof of Viola [Vio12a] with our new lower bounds, and using ROBPs instead of circuits.

Proof of Corollary 9.4. Suppose the codewords of Q can be stored in $\{0,1\}^{k+r}$ bits of memory so that they can be retrieved by some ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width w. Let $\mathbf{Q}$ be uniform over Q, and let $\mathbf{U}_{k+r}$ be uniform over $\{0,1\}^{k+r}$, and observe that

$$|F(\mathbf{U}_{k+r}) - \mathbf{Q}| \leq 1 - 2^{-r}$$

by a simple calculation using the definition of statistical distance. But by Theorem 9.16, we know that

$$1 - 8wL \cdot 2^{-\frac{\rho}{1+2\rho}k} \leq |F(\mathbf{U}_{k+r}) - \mathbf{Q}|.$$

Combining the bounds yields the result. □

For (n, k, d) codes, we can use Fact 2.10 to specialize the above result to the following.

Corollary 9.5 (Corollary 9.1, restated)**.** *For any good code $Q \subseteq \{0,1\}^n$ of dimension k, there is a constant $c > 0$ such that if we can store codewords of Q using $k+r$ bits so that a codeword can be computed by an ROBP $F : \{0,1\}^{k+r} \to \{0,1\}^n$ of width at most 2^{cn}, then we must have redundancy $r \geq \lfloor cn \rfloor$.*

Thus for good codes, one must use $r = \Omega(n)$ bits of redundancy, even given an ROBP of width $2^{\Omega(n)}$. Furthermore, as discussed in Section 9.2, it is straightforward to show this is tight (up to constant factors).

9.8 Applications in communication complexity

In the previous section, we saw one unexpected application arising from our first main theorem (Theorem 9.4). In this section, we give an unexpected application arising from a key ingredient developed in the proof of our *second* main theorem (Theorem 9.5). In particular, using our simple new lemma on amplifying statistical distance (Section 9.6.1), we give a simple new proof of an interesting result of Göös and Watson [GW20] that establishes lower bounds on the communication complexity of sampling disjoint sets.

9.8.1 A simple new proof of known lower bounds for sampling disjoint sets

As we have seen, almost all of the prior work on the complexity of sampling has focused on sampling with AC^0 *circuits*. In this chapter, we have primarily focused on sampling with *read-once branching programs*. Now, we slightly switch gears and focus on sampling with *two-party communication protocols*, a subject first considered in the works [AST$^+$03, GW20]. To start, we give a formal description of how this is defined.

Sampling using two-party communication protocols We consider the following natural way to sample distributions with two-party communication protocols, as put forth in [AST$^+$03, GW20]. First, Alice and Bob receive private randomness $\mathbf{A}$ and $\mathbf{B}$, and no other input. Then, Alice and Bob take turns communicating bits to one another, until the protocol ends. We let $\Pi(\mathbf{A}, \mathbf{B})$ denote the sequence of bits communicated (the transcript). At the end, Alice uses $\mathbf{A}$ and $\Pi(\mathbf{A}, \mathbf{B})$ to output some string $\mathbf{X} \sim \{0,1\}^n$, while Bob uses $\mathbf{B}$ and $\Pi(\mathbf{A}, \mathbf{B})$ to output some string $\mathbf{Y} \sim \{0,1\}^n$. The distribution generated by the protocol is $(\mathbf{X}, \mathbf{Y})$.

A result of Göös and Watson One of the most classical problems in two-party communication complexity is the *disjointness* problem, where Alice and Bob receive inputs $x, y \in \{0,1\}^n$, and they must figure out if there is some i where $x_i = y_i = 1$. In a recent work [GW20], Göös and Watson study a natural *sampling* version of this historical problem. In particular, they seek out lower bounds on the complexity of generating disjoint strings using two-party communication protocols. More formally, they ask for the minimum number of bits communicated by any two-party protocol that generates the distribution $(\mathbf{S}, \mathbf{T}) \sim \{0,1\}^n \times \{0,1\}^n$ that is uniform over all strings $(s, t) \in \{0,1\}^n \times \{0,1\}^n$ with $s \wedge t = 0^n$. Their main result is the following.

Theorem 9.23 (Sampling lower bounds for disjoint sets [GW20])**.** *There is a universal constant $c > 0$ such that for any distribution $(\mathbf{X}, \mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ sampled by a two-party protocol using at most cn bits of communication,*

$$|(\mathbf{X}, \mathbf{Y}) - (\mathbf{S}, \mathbf{T})| \geq 1 - 2^{-cn}.$$

Below, we provide a simple new proof, by applying our new lemma on amplifying statistical distance (Lemma 9.9). We reminder the reader that this lemma has a simple, self-contained proof in Section 9.6.1.

An overview of the new proof

Our new proof of Theorem 9.23 will involve just five simple steps.

1. Observe that $(\mathbf{X}, \mathbf{Y})$ is a convex combination of 2^b product distributions $(\mathbf{X}', \mathbf{Y}')$ [AST$^+$03].
2. Apply the data processing inequality (Fact 2.3) to observe that
$$|(\mathbf{X}', \mathbf{Y}') - (\mathbf{S}, \mathbf{T})| \geq |(\mathbf{X}'_1, \mathbf{Y}'_1, \dots, \mathbf{X}'_n, \mathbf{Y}'_n) - (\mathbf{S}_1, \mathbf{T}_1, \dots, \mathbf{S}_n, \mathbf{T}_n)|.$$
3. Using a straightforward calculation, observe that $|(\mathbf{X}'_i, \mathbf{Y}'_i) - (\mathbf{S}_i, \mathbf{T}_i)| \geq \Omega(1)$ for each i, since $\mathbf{X}'_i, \mathbf{Y}'_i$ are independent and $(\mathbf{S}_i, \mathbf{T}_i)$ is uniform over $\{(0,0), (0,1), (1,0)\}$. Observe that this still holds even if you condition on any fixing of the random variables earlier on in the sequence, since this doesn't break the independence of $\mathbf{X}'_i, \mathbf{Y}'_i$, nor does it change the distribution of $(\mathbf{S}_i, \mathbf{T}_i)$.
4. Use our new Lemma 9.9 on amplifying statistical distance to obtain $|(\mathbf{X}', \mathbf{Y}') - (\mathbf{S}, \mathbf{T})| \geq 1 - 2^{-\Omega(n)}$.
5. Using the fact that the convex combination of a few far distributions is still far (Fact 2.5), conclude that $|(\mathbf{X}, \mathbf{Y}) - (\mathbf{S}, \mathbf{T})| \geq 1 - 2^{b-\Omega(n)}$, which is $1 - 2^{-\Omega(n)}$ for some $b = \Omega(n)$, as desired.

With this plan in mind, let us turn to the new, formal proof of Theorem 9.23.

A formal presentation of the new proof

New proof of Theorem 9.23. Suppose Alice and Bob have private randomness $\mathbf{A}$ and $\mathbf{B}$, respectively, and that they generate the distribution $(\mathbf{X},\mathbf{Y}) \sim \{0,1\}^n \times \{0,1\}^n$ using a protocol that exchanges b bits of communication. A simple observation (made in [AST+03]) is that if we condition on any fixing of the communication transcript $\Pi(\mathbf{A},\mathbf{B})$, then $\mathbf{X},\mathbf{Y}$ become independent. In other words, $(\mathbf{X},\mathbf{Y})$ is a convex combination of $\leq 2^b$ distributions of the form $(\mathbf{X}',\mathbf{Y}') \sim \{0,1\}^n \times \{0,1\}^n$, where $\mathbf{X}',\mathbf{Y}'$ are independent.

We now seek to lower bound each $|(\mathbf{X}',\mathbf{Y}') - (\mathbf{S},\mathbf{T})|$. Towards this end, define random variables $\mathbf{W} = (\mathbf{W}_1,\ldots,\mathbf{W}_n)$ and $\mathbf{R} = (\mathbf{R}_1,\ldots,\mathbf{R}_n)$, where each $\mathbf{W}_i := (\mathbf{X}'_i,\mathbf{Y}'_i)$ and each $\mathbf{R}_i := (\mathbf{S}_i,\mathbf{T}_i)$. Since $\mathbf{W}$ simply permutes the bits of $(\mathbf{X}',\mathbf{Y}')$, and $\mathbf{R}$ permutes the bits of $(\mathbf{S},\mathbf{T})$ in the same way, we have via the data processing inequality (Fact 2.3) that

$$|(\mathbf{X}',\mathbf{Y}') - (\mathbf{S},\mathbf{T})| \geq |\mathbf{W} - \mathbf{R}| = |(\mathbf{W}_1,\mathbf{W}_2,\ldots,\mathbf{W}_n) - (\mathbf{R}_1,\mathbf{R}_2,\ldots,\mathbf{R}_n)|.$$

Now, consider any $i \in [n]$, let v be in $(\{0,1\}^2)^{i-1}$, and consider the random variables $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ and $(\mathbf{R}_i \mid \mathbf{R}_{<i} = v)$. Notice that $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ is of the form $((\mathbf{X}'_i,\mathbf{Y}'_i) \mid (\mathbf{X}'_1,\mathbf{Y}'_1,\ldots,\mathbf{X}'_{i-1},\mathbf{Y}'_{i-1}) = v)$. Since $\mathbf{X}',\mathbf{Y}'$ are independent, $\mathbf{X}'_i,\mathbf{Y}'_i$ are independent, and fixing the first few bits of each random variable doesn't change this fact. Thus $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ is a random variable consisting of two independent bits. Let p denote the probability that its first bit is 1, and q denote the probability that its second bit is 1.

On the other hand, since $(\mathbf{S},\mathbf{T})$ is uniform over pairs $(x,y) \sim \{0,1\}^n \times \{0,1\}^n$ of disjoint strings, it is straightforward to verify that each $\mathbf{R}_i \sim \{0,1\}^2$ is an independent random variable that is uniform over the strings $\{(0,0),(0,1),(1,0)\}$. Thus, $(\mathbf{R}_i \mid \mathbf{R}_{<i} = v)$ is uniform over the same set of strings. Now that we have specified the distributions of $(\mathbf{W}_i \mid \mathbf{W}_{<i} = v)$ and $(\mathbf{R}_i \mid \mathbf{R}_{<i} = v)$, it is a straightforward calculation to verify that there is a universal constant $\delta > 0.1$ such that no matter the values of p, q,

$$|(\mathbf{W}_i \mid \mathbf{W}_{<i} = v) - (\mathbf{R}_i \mid \mathbf{R}_{<i} = v)| \geq \delta.$$

Thus, applying our lemma on amplifying statistical distance (Lemma 9.9), we get

$$|(\mathbf{X}',\mathbf{Y}') - (\mathbf{S},\mathbf{T})| \geq 1 - e^{-n\delta^2/2}.$$

To conclude, recall that $(\mathbf{X},\mathbf{Y})$ is a convex combination of $\leq 2^b$ distributions of the form $(\mathbf{X}',\mathbf{Y}')$, and thus applying Fact 2.5 we get

$$|(\mathbf{X},\mathbf{Y}) - (\mathbf{S},\mathbf{T})| \geq 1 - 2^b \cdot e^{-n\delta^2/2}.$$

Thus, as long as Alice and Bob communicated $b \leq \frac{n\delta^2}{4}\log_2 e = \Omega(n)$ bits, we get statistical distance at least $1 - 2^{-\frac{n\delta^2}{4}\log_2 e} = 1 - 2^{-\Omega(n)}$, as desired. □

9.9 Future directions

Recently, there have been a number of exciting works that study the complexity of sampling. In this new area of complexity, one seeks to understand the power of classical computational models for the task of sampling from distributions. In this chapter, we initiated a study of the complexity of sampling in limited *space*, and proved the first nontrivial sampling lower bounds against oblivious read-once branching programs. The overall study of the complexity of sampling is still a very new area, and many open questions remain. Below, we outline three such questions on the complexity of sampling with limited memory.

Sampling lower bounds using limited randomness In this chapter, we demonstrated distributions that cannot be sampled by ROBPs of limited width, given *unlimited* random bits as input. Can one demonstrate sampling lower bounds for ROBPs against a richer class of distributions if the ROBP is only provided with a limited number of random input bits (say, $\ell = \text{poly}(n)$)?

This question is especially interesting if one considers a more powerful model of multi-output ROBPs whose output bits need not be layered (as in Definition 9.3). It can be shown that such ROBPs can come arbitrarily close to sampling *any distribution* $\mathbf{X} \sim \{0,1\}^n$ using just width 3. However, the most natural way to construct such an ROBP requires $\ell \gg |\text{support}(\mathbf{X})|$ bits of randomness, which could be exponentially large in n. Thus, it would be interesting to understand the power of these ROBPs under the restriction $\ell \leq \text{poly}(n)$.

Sampling lower bounds for more general branching programs On the other hand, if we keep the output bits layered, and the randomness unlimited, it is natural to ask if there exist more general types of branching programs for which sampling lower bounds can be established.

One such generalization might be to unknown-order ROBPs [FK18], which are allowed to read their input bits and write their output bits in any unknown order. As it turns out, many of our results can be easily extended to this more general setting: For example, our results on sampling codes can be extended to the unknown-order setting using the fact that distance is preserved under permutating coordinates, and our sampling lower bounds against input-output pairs $(\mathbf{U}_n, b(\mathbf{U}_n))$ can be extended by replacing the two-source extractor with an extractor for interleaved sources (e.g., like the one from [RY11]).

Perhaps a more interesting generalization would be to read-k branching programs, where the branching program is allowed to read each input bit k times. It seems much more challenging to break the correlation between the output bits of a read-k branching program - thus, establishing sampling lower bounds for this more general setting would seem to require significantly new techniques.

A separation between sampling with ROBPs and AC^0 circuits As we've seen, it is not too hard to show that there exist simple distributions samplable by AC^0 circuits that cannot be sampled by ROBPs, even given exponential width. Can one find a distribution that cannot be sampled in AC^0, but can be sampled by ROBPs? This is possible if one can construct an extractor for AC^0 sources, which can be computed by small width ROBPs. In fact, even a disperser $\mathsf{Disp} : \{0,1\}^n \to \{0,1\}$ for min-entropy $n-1$ would suffice, as this would imply that both $\mathsf{Disp}^{-1}(1)$ and $\mathsf{Disp}^{-1}(0)$ can be generated in small space, yet one of these has min-entropy $n-1$ and thus cannot be generated in AC^0 (by definition of a disperser). Since all known extractors for AC^0 sources (Chapter 8) also work for distributions that can be generated in small space (Chapter 7), new extractors are needed.

Bibliography

[AA13] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Theory of Computing*, 9(4):143–252, 2013. Preliminary version in STOC 2011.

[Aar14] Scott Aaronson. The equivalence of sampling and searching. *Theory of Computing Systems*, 55(2):281–298, 2014. Preliminary version in CSR 2011.

[AAV13] Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: the quantum PCP conjecture. *ACM SIGACT News*, 44(2):47–79, 2013.

[AB09] Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

[ABGSD21] Divesh Aggarwal, Huck Bennett, Alexander Golovnev, and Noah Stephens-Davidowitz. Fine-grained hardness of CVP(P) – Everything that we can prove (and nothing else). In *ACM-SIAM Symposium on Discrete Algorithms (SODA 2021)*, pages 1816–1835. SIAM, 2021.

[ABN23] Anurag Anshu, Nikolas P Breuckmann, and Chinmay Nirkhe. NLTS hamiltonians from good quantum codes. In *55th Annual ACM Symposium on Theory of Computing (STOC 2023)*, pages 1090–1096, 2023.

[ACG$^+$22] Omar Alrabiah, Eshan Chattopadhyay, Jesse Goodman, Xin Li, and João Ribeiro. Low-degree polynomials extract from local sources. In *49th International Colloquium on Automata, Languages, and Programming (ICALP 2022)*, volume 229 of *LIPIcs*, pages 10:1–10:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[ACRT99] Alexander E Andreev, Andrea EF Clementi, José DP Rolim, and Luca Trevisan. Weak random sources, hitting sets, and bpp simulations. *SIAM Journal on Computing*, 28(6):2103–2116, 1999. Preliminary version in FOCS 1997.

[ADN$^+$19] Divesh Aggarwal, Ivan Damgård, Jesper Buus Nielsen, Maciej Obremski, Erick Purwanto, Joao Ribeiro, and Mark Simkin. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures. In *39th Annual International Cryptology Conference (CRYPTO 2019)*, pages 510–539. Springer, 2019.

[AGM03] Noga Alon, Oded Goldreich, and Yishay Mansour. Almost k-wise independence versus k-wise independence. *Information Processing Letters*, 88(3):107–110, 2003.

[Ajt83] Miklós Ajtai. σ_1^1-formulae on finite structures. *Annals of pure and applied logic*, 24(1):1–48, 1983.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004.

[All89] Eric Allender. A note on the power of threshold circuits. In *30th Annual Symposium on Foundations of Computer Science (FOCS 1989)*, pages 580–584. IEEE Computer Society, 1989.

[Alo86] Noga Alon. Eigenvalues, geometric expanders, sorting in rounds, and Ramsey theory. *Combinatorica*, 6(3):207–219, 1986.

[AMS99] Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and System Sciences*, 1(58):137–147, 1999. Preliminary version in STOC 1996.

[And87] A. E. Andreev. On a method for obtaining more than quadratic effective lower bounds for the complexity of π-schemes. *Moscow Univ. Math. Bull.*, 42(1):63–66, 1987.

[AOR$^+$20] Divesh Aggarwal, Maciej Obremski, Joao Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to extract useful randomness from unreliable sources. In *39th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2020)*, pages 343–372. Springer, 2020.

[AS16] Noga Alon and Joel H Spencer. *The probabilistic method*. John Wiley & Sons, 2016.

[AST$^+$03] Andris Ambainis, Leonard J Schulman, Amnon Ta-Shma, Umesh Vazirani, and Avi Wigderson. The quantum communication complexity of sampling. *SIAM Journal on Computing*, 32(6):1570–1585, 2003. Preliminary version in FOCS 1998.

[Bab87] László Babai. Random oracles separate pspace from the polynomial-time hierarchy. *Information Processing Letters*, 26(1):51–53, 1987.

[BBR88] Charles H Bennett, Gilles Brassard, and Jean-Marc Robert. Privacy amplification by public discussion. *SIAM Journal on Computing*, 17(2):210–229, 1988. Preliminary version in CRYPTO 1985.

[BC82] A Borodin and S Cook. A time-space tradeoff for sorting on a general sequential model of computation. *SIAM Journal on Computing*, 11(2):287–297, 1982. Preliminary version in STOC 1980.

[BCDT19] Avraham Ben-Aroya, Gil Cohen, Dean Doron, and Amnon Ta-Shma. Two-source condensers with low error and small entropy gap via entropy-resilient functions. In *23rd International Conference on Randomization and Computation (RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[BCS16] Itai Benjamini, Gil Cohen, and Igor Shinkar. Bi-lipschitz bijection between the boolean cube and the hamming ball. *Israel Journal of Mathematics*, 212(2):677–703, 2016. Preliminary version in FOCS 2014.

[BDF$^+$22] Andrej Bogdanov, Krishnamoorthy Dinesh, Yuval Filmus, Yuval Ishai, Avi Kaplan, and Akshayaram Srinivasan. Bounded indistinguishability for simple sources. In *13th Innovations in*

Theoretical Computer Science Conference (ITCS 2022). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[BDIR21] Fabrice Benhamouda, Akshay Degwekar, Yuval Ishai, and Tal Rabin. On the local leakage resilience of linear secret sharing schemes. *Journal of Cryptology*, 34:1–65, 2021. Preliminary version in CRYPTO 2018.

[BDL16] Jean Bourgain, Zeev Dvir, and Ethan Leeman. Affine extractors over large fields with exponential error. *computational complexity*, 25:921–931, 2016.

[BDSGM23] Marshall Ball, Dana Dachman-Soled, Eli Goldin, and Saachi Mutreja. Extracting randomness from samplable distributions, revisited. In *64th Annual Symposium on Foundations of Computer Science (FOCS 2023, to appear)*. IEEE, 2023.

[BDT19] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. An efficient reduction from two-source to nonmalleable extractors: Achieving near-logarithmic min-entropy. *SIAM Journal on Computing*, 51(2):STOC17–31, 2019. Preliminary version in STOC 2017.

[Bea91] Paul Beame. A general sequential time-space tradeoff for finding unique elements. *SIAM Journal on Computing*, 20(2):270–277, 1991. Preliminary version in STOC 1989.

[BEHL12] Ido Ben-Eliezer, Rani Hod, and Shachar Lovett. Random low-degree polynomials are hard to approximate. *computational complexity*, 21(1):63–81, 2012. Preliminary version in RANDOM 2009.

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In *Coding and Cryptology: Third International Workshop (IWCC 2011)*, pages 11–46. Springer, 2011.

[BFK$^+$81] Allan Borodin, Michael J Fischer, David G Kirkpatrick, Nancy A Lynch, and Martin Tompa. A time-space tradeoff for sorting on non-oblivious machines. *Journal of Computer and System Sciences*, 22(3):351–364, 1981. Preliminary version in FOCS 1979.

[BFM88] Manuel Blum, Paul Feldman, and Silvio Micali. Non-interactive zero-knowledge and its applications. In *20th Annual ACM Symposium on Theory of Computing (STOC 1988)*, pages 103–112, 1988.

[BFNV19] Adam Bouland, Bill Fefferman, Chinmay Nirkhe, and Umesh Vazirani. On the complexity and verification of quantum random circuit sampling. *Nature Physics*, 15(2):159–163, 2019.

[BG13] Andrej Bogdanov and Siyao Guo. Sparse extractor families for all the entropy. In *4th Conference on Innovations in Theoretical Computer Science (ITCS 2013)*, pages 553–560, 2013.

[BGK06] Jean Bourgain, Alexey A Glibichuk, and Sergei Vladimirovich Konyagin. Estimates for the number of sums and products and for exponential sums in fields of prime order. *Journal of the London Mathematical Society*, 73(2):380–398, 2006.

[BGLZ15] Abhishek Bhowmick, Ariel Gabizon, Thái Hoang Lê, and David Zuckerman. Deterministic extractors for additive sources. In *6th Conference on Innovations in Theoretical Computer Science (ITCS 2015)*, pages 277–286, 2015.

[BGM22] Marshall Ball, Oded Goldreich, and Tal Malkin. Randomness extraction from somewhat dependent sources. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[BIL12] Chris Beck, Russell Impagliazzo, and Shachar Lovett. Large deviation bounds for decision trees and sampling lower bounds for ac0-circuits. In *53rd Annual Symposium on Foundations of Computer Science (FOCS 2012))*, pages 101–110. IEEE, 2012.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM Journal on Computing*, 36(4):1095–1118, 2006. Preliminary version in FOCS 2004.

[BJS01] Paul Beame, Thathachar S Jayram, and Michael Saks. Time–space tradeoffs for branching programs. *Journal of Computer and System Sciences*, 63(4):542–572, 2001.

[BKS$^+$10] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. *Journal of the ACM (JACM)*, 57(4):1–52, 2010. Preliminary version in STOC 2005.

[BL73] Jacques Bernard and Gérard Letac. Construction d'evenements equiprobables et coefficients multinomiaux modulo p^n. *Illinois Journal of Mathematics*, 17(2):317–332, 1973.

[BL85] Michael Ben-Or and Nathan Linial. Collective coin flipping, robust voting schemes and minima of banzhaf values. In *26th Annual Symposium on Foundations of Computer Science (FOCS 1985)*, pages 408–416. IEEE, 1985.

[BL87] Ravi B Boppana and Jeffrey C Lagarias. One-way functions and circuit complexity. *Information and Computation*, 74(3):226–240, 1987. Preliminary version in CCC 1986.

[Bla79] George Robert Blakley. Safeguarding cryptographic keys. In *International Workshop on Managing Requirements Knowledge*, pages 313–313. IEEE, 1979.

[Blu86] Manuel Blum. Independent unbiased coin flips from a correlated biased source—a finite state markov chain. *Combinatorica*, 6:97–108, 1986. Preliminary version in FOCS 1884.

[BNS92] László Babai, Noam Nisant, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992. Preliminary version in STOC 1989.

[BO83] Michael Ben-Or. Another advantage of free choice: completely asynchronous agreement protocols. In *2nd Annual Symposium on Principles of Distributed Computing (PODC 1983)*, pages 27–30. ACM, 1983.

[BOPV06] Michael Ben-Or, Elan Pavlov, and Vinod Vaikuntanathan. Byzantine agreement in the full-information model in $O(\log n)$ rounds. In *38th Annual Symposium on Theory of Computing (STOC 2006)*, pages 179–186. ACM, 2006.

[Bou05] Jean Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1(01):1–32, 2005.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[Boy77] Phelim P Boyle. Options: A Monte Carlo approach. *Journal of financial economics*, 4(3):323–338, 1977.

[BPS07] Paul Beame, Toniann Pitassi, and Nathan Segerlind. Lower bounds for Lovász–Schrijver systems and beyond follow from multiparty communication complexity. *SIAM Journal on Computing*, 37(3):845–869, 2007. Preliminary version in ICALP 2005.

[Bra84] Gabriel Bracha. An asynchronous $\lfloor (n-1)/3 \rfloor$-resilient consensus protocol. In *3rd Annual Symposium on Principles of Distributed Computing (PODC 1984)*, pages 154–162. ACM, 1984.

[Bra08] Mark Braverman. Polylogarithmic independence fools ac 0 circuits. *Journal of the ACM (JACM)*, 57(5):1–10, 2008.

[BRC60] Raj Chandra Bose and Dwijendra K Ray-Chaudhuri. On a class of error correcting binary group codes. *Information and control*, 3(1):68–79, 1960.

[BRS93] Allan Borodin, Alexander Razborov, and Roman Smolensky. On lower bounds for read-k-times branching programs. *Computational Complexity*, 3:1–18, 1993.

[BRSW12] Boaz Barak, Anup Rao, Ronen Shaltiel, and Avi Wigderson. 2-source dispersers for $n^{o(1)}$ entropy, and Ramsey graphs beating the Frankl-Wilson construction. *Annals of Mathematics*, pages 1483–1543, 2012. Preliminary version in STOC 2006.

[BS00] Jürgen Bierbrauer and Holger Schellwat. Almost independent and weakly biased arrays: efficient constructions and cryptologic applications. In *20th Annual International Cryptology Conference (CRYPTO 2000)*, pages 533–543. Springer, 2000.

[BSG12] Eli Ben-Sasson and Ariel Gabizon. Extractors for polynomials sources over constant-size fields of small characteristic. In *16th International Conference on Randomization and Computation (RANDOM 2012)*, pages 399–410. Springer, 2012.

[BSHR$^+$01] Eli Ben-Sasson, Shlomo Hoory, Eyal Rozenman, Salil Vadhan, and Avi Wigderson. Extractors for affine sources. Unpublished manuscript, 2001.

[BSK12] Eli Ben-Sasson and Swastik Kopparty. Affine dispersers from subspace polynomials. *SIAM Journal on Computing*, 41(4):880–914, 2012. Preliminary version in STOC 2009.

[BSRZ15] Eli Ben-Sasson and Noga Ron-Zewi. From affine to two-source extractors via approximate duality. *SIAM Journal on Computing*, 44(6):1670–1697, 2015. Preliminary version in STOC 2011.

[BT94] Richard Beigel and Jun Tarui. On ACC. *Computational Complexity*, 4:350–366, 1994. Preliminary version in FOCS 1991.

[BT13] John C Baez and David Tweed. Monte Carlo methods in climate science. *Math Horizons*, 21(2):5–8, 2013.

[CC85] Benny Chor and Brian A Coan. A simple and efficient randomized Byzantine Agreement algorithm. *IEEE Transactions on Software Engineering*, SE-11(6):531–539, 1985.

[CDH$^+$00] Ran Canetti, Yevgeniy Dodis, Shai Halevi, Eyal Kushilevitz, and Amit Sahai. Exposure-resilient functions and all-or-nothing transforms. In *International Conference on the Theory and Application of Cryptographic Techniques (EUROCRYPT 2000)*, pages 453–469. Springer, 2000.

[CF01] Ran Canetti and Marc Fischlin. Universally composable commitments. In *21st Annual International Cryptology Conference (CRYPTO 2001)*, pages 19–40. Springer, 2001.

[CFG$^+$85] Benny Chor, Joel Friedman, Oded Goldreich, Johan Håstad, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *26th Annual Symposium on Foundations of Computer Science (FOCS 1985)*, pages 396–407. IEEE, 1985.

[CFL83] Ashok K Chandra, Merrick L Furst, and Richard J Lipton. Multi-party protocols. In *15th Annual ACM Symposium on Theory of Computing (STOC 1983)*, pages 94–99, 1983.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988. Preliminary version in FOCS 1985.

[CG17] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. *Journal of Cryptology*, 30:191–241, 2017.

[CG21] Eshan Chattopadhyay and Jesse Goodman. Improved extractors for small-space sources. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 610–621. IEEE, 2021.

[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 *61st Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 1226–1242. IEEE, 2020.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. In *52nd Annual Symposium on Theory of Computing (STOC 2020)*, pages 1184–1197. ACM, 2020.

[CGL20] Eshan Chattopadhyay, Vipul Goyal, and Xin Li. Nonmalleable extractors and codes, with their many tampered extensions. *SIAM Journal on Computing*, 49(5):999–1040, 2020. Preliminary version in STOC 2016.

[CGL21] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 622–633. IEEE, 2021.

[CGP$^+$15] J Carlson, Stefano Gandolfi, Francesco Pederiva, Steven C Pieper, Rocco Schiavilla, KE Schmidt, and Robert B Wiringa. Quantum Monte Carlo methods for nuclear physics. *Reviews of Modern Physics*, 87(3):1067, 2015.

[CGZ22] Eshan Chattopadhyay, Jesse Goodman, and David Zuckerman. The space complexity of sampling. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *LIPIcs*, pages 40:1–40:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[Cha20] Eshan Chattopadhyay. Guest Column: A recipe for constructing two-source extractors. *SIGACT News*, 51(2):38–57, June 2020.

[CL16a] Eshan Chattopadhyay and Xin Li. Explicit non-malleable extractors, multi-source extractors, and almost optimal privacy amplification protocols. In *57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pages 158–167. IEEE, 2016.

[CL16b] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *48th Annual ACM Symposium on Theory of Computing (STOC 2016)*, pages 299–311, 2016.

[CL18] Kuan Cheng and Xin Li. Randomness extraction in AC^0 and with small locality. *22nd International Conference on Randomization and Computation (RANDOM 2018)*, 2018.

[CL20] Eshan Chattopadhyay and Xin Li. Non-malleable codes, extractors and secret sharing for interleaved tampering and composition of tampering. In *Theory of Cryptography: 18th International Conference (TCC 2020)*, pages 584–613. Springer, 2020.

[CL22] Eshan Chattopadhyay and Jyun-Jie Liao. Extractors for sum of two sources. In *54th Annual ACM Symposium on Theory of Computing (STOC 2022)*, pages 1584–1597, 2022.

[CLP17] Ernie Croot, Vsevolod F Lev, and Péter Pál Pach. Progression-free sets in are exponentially small. *Annals of Mathematics*, pages 331–337, 2017.

[Coh16a] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. *SIAM Journal on Computing*, 45(4):1297–1338, 2016. Preliminary version in FOCS 2015.

[Coh16b] Gil Cohen. Making the most of advice: New correlation breakers and their applications. In *57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pages 188–196. IEEE, 2016.

[Coh17] Gil Cohen. Towards optimal two-source extractors and Ramsey graphs. In *49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, pages 1157–1170, 2017.

[Coh19] Gil Cohen. Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs. *SIAM Journal on Computing*, 50(3):STOC16–30, 2019. Preliminary version in STOC 2016.

[CRVW02] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. Randomness conductors and constant-degree lossless expanders. In *34th Annual ACM Symposium on Theory of Computing (STOC 2002)*, pages 659–668, 2002.

[CS16a] Gil Cohen and Leonard J Schulman. Extractors for near logarithmic min-entropy. In *57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pages 178–187. IEEE, 2016.

[CS16b] Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In *7th Conference on Innovations in Theoretical Computer Science (ITCS 2016)*, pages 47–58, 2016.

[CT15] Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In *19th International Conference on Randomization and Computation (RANDOM 2015)*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[CW89] Aviad Cohen and Avi Wigderson. Dispersers, deterministic amplification, and weak random sources. In *30th Annual Symposium on Foundations of Computer Science (FOCS 1989)*, pages 14–19. IEEE, 1989.

[CZ16] Eshan Chattopadhyay and David Zuckerman. New extractors for interleaved sources. In *31st Conference on Computational Complexity (CCC 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[CZ18] Xue Chen and David Zuckerman. Existence of simple extractors. In *Electron. Colloquium Comput. Complex.*, volume 25, page 116, 2018.

[CZ19] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653–705, 2019. Preliminary version in STOC 2016.

[DDV10] Francesco Davì, Stefan Dziembowski, and Daniele Venturi. Leakage-resilient storage. In *Security and Cryptography for Networks: 7th International Conference (SCN 2010)*, pages 121–137. Springer, 2010.

[DF92] Danny Dolev and Tomás Feder. Determinism vs. nondeterminism in multiparty communication complexity. *SIAM Journal on Computing*, 21(5):889–895, 1992. Preliminary version in FOCS 1989.

[DG10] Matt DeVos and Ariel Gabizon. Simple affine extractors using dimension expansion. In *IEEE 25th Annual Conference on Computational Complexity (CCC 2010)*, pages 50–57. IEEE, 2010.

[DGW09] Zeev Dvir, Ariel Gabizon, and Avi Wigderson. Extractors and rank extractors for polynomial sources. *Computational Complexity*, 18:1–58, 2009. Preliminary version in FOCS 2007.

[Din07] Irit Dinur. The PCP theorem by gap amplification. *Journal of the ACM (JACM)*, 54(3):12–es, 2007.

[DK11] Evgeny Demenkov and Alexander Kulikov. An elementary proof of a $3n - o(n)$ lower bound on the circuit complexity of affine dispersers. In *International Symposium on Mathematical Foundations of Computer Science*, pages 256–265. Springer, 2011.

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. *SIAM Journal on Computing*, 42(6):2305–2328, 2013. Preliminary version in FOCS 2009.

[DL12] Zeev Dvir and Shachar Lovett. Subspace evasive sets. In *44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 351–358, 2012.

[DMOZ22] Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. Nearly optimal pseudorandomness from hardness. *Journal of the ACM*, 69(6):1–55, 2022. Preliminary version in FOCS 2020.

[Dod00] Yevgeniy Dodis. *Exposure-Resilient Cryptography*. PhD thesis, Massachusetts Institute of Technology, 2000.

[Dod06] Yevgeniy Dodis. Fault-tolerant leader election and collective coin-flipping in the full information model, 2006.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im) possibility of cryptography with imperfect randomness. In *45th Annual Symposium on Foundations of Computer Science (FOCS 2004)*, pages 196–205. IEEE, 2004.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM Journal on Computing*, 38(1):97–139, 2008. Preliminary version in EUROCRYPT 2004.

[DP07] Stefan Dziembowski and Krzysztof Pietrzak. Intrusion-resilient secret sharing. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2007)*, pages 227–237. IEEE, 2007.

[Dvi12] Zeev Dvir. Extractors for varieties. *Computational complexity*, 21:515–572, 2012. Preliminary version in CCC 2009.

[DW11] Zeev Dvir and Avi Wigderson. Kakeya sets, new mergers, and old extractors. *SIAM Journal on Computing*, 40(3):778–792, 2011. Preliminary version in FOCS 2008.

[DW12] Anindya De and Thomas Watson. Extractors and lower bounds for locally samplable sources. *ACM Transactions on Computation Theory (TOCT)*, 4(1):1–21, 2012. Preliminary version in RANDOM 2011.

[Dwa72] Meyer Dwass. Unbiased coin tossing with discrete random variables. *The Annals of Mathematical Statistics*, 43(3):860–864, 1972.

[DY21] Yevgeniy Dodis and Kevin Yeo. Doubly-affine extractors, and their applications. In *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, 2021.

[EG17] Jordan S Ellenberg and Dion Gijswijt. On large subsets of with no three-term arithmetic progression. *Annals of Mathematics*, pages 339–343, 2017.

[Eli72] Peter Elias. The efficient construction of an unbiased random sequence. *The Annals of Mathematical Statistics*, 43(3):865–870, 1972.

[Ema08] Viola Emanuele. Norms, XOR lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4(1):137–168, 2008. Preliminary version in CCC 2007.

[Eus13] Alexander Eustis. *Hypergraph independence numbers*. PhD thesis, University of California, San Diego, 2013.

[EV13] Alex Eustis and Jacques Verstraëte. On the independence number of steiner systems. *Combinatorics, Probability and Computing*, 22(2):241–252, 2013.

[FG13] Jeff Ford and Anna Gál. Hadamard tensors and lower bounds on multiparty communication complexity. *computational complexity*, 22:595–622, 2013. Preliminary version in ICALP 2005.

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward Hirsch, and Alexander S Kulikov. A better-than-$3n$ lower bound for the circuit complexity of an explicit function. In *57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pages 89–98. IEEE, 2016.

[FK18] Michael A Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *59th Annual Symposium on Foundations of Computer Science (FOCS 2018)*, pages 946–955. IEEE, 2018.

[FLP85] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)*, 32(2):374–382, 1985. Preliminary version in PODS 1983.

[FLRS23] Yuval Filmus, Itai Leigh, Artur Riazanov, and Dmitry Sokolov. Sampling and certifying symmetric functions. *27th International Conference on Randomization and Computation (RANDOM 2023)*, 2023.

[FLW92] Alan M Ferrenberg, DP Landau, and Y Joanna Wong. Monte Carlo simulations: Hidden errors from “good” random number generators. *Physical Review Letters*, 69(23):3382, 1992.

[Fri92] Joel Friedman. On the bit extraction problem. In *33rd Annual Symposium on Foundations of Computer Science (FOCS 1992)*, pages 314–314. IEEE, 1992.

[FSS84] Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984. Preliminary version in FOCS 1981.

[GB10] Venkatesan Guruswami and Eric Blais. Notes 6: Reed–Solomon, BCH, Reed–Muller, and concatenated codes. *Introduction to Coding Theory CMU: Spring*, 2010.

[GGJS11] Sanjam Garg, Vipul Goyal, Abhishek Jain, and Amit Sahai. Bringing people of different beliefs together to do UC. In *Theory of Cryptography: 8th Theory of Cryptography Conference (TCC 2011)*, pages 311–328. Springer, 2011.

[GGN10] Oded Goldreich, Shafi Goldwasser, and Asaf Nussboim. On the implementation of huge random objects. *SIAM Journal on Computing*, 39(7):2761–2822, 2010. Preliminary version in FOCS 2003.

[Gil52] Edgar N Gilbert. A comparison of signalling alphabets. *The Bell system technical journal*, 31(3):504–522, 1952.

[GK08] Vipul Goyal and Jonathan Katz. Universally composable multi-party computation with an unreliable common reference string. In *Theory of Cryptography Conference*, pages 142–154. Springer, 2008.

[GK16] Alexander Golovnev and Alexander Kulikov. Weighted gate elimination: Boolean dispersers for quadratic varieties imply improved circuit lower bounds. In *7th Conference on Innovations in Theoretical Computer Science (ITCS 2016)*, pages 405–411, 2016.

[GK18a] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing. In *50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 685–698, 2018.

[GK18b] Vipul Goyal and Ashutosh Kumar. Non-malleable secret sharing for general access structures. In *38th Annual International Cryptology Conference (CRYPTO 2018)*, pages 501–530. Springer, 2018.

[GKW21] Alexander Golovnev, Alexander Kulikov, and Ryan Williams. Circuit depth reductions. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[GLW22] Venkatesan Guruswami, Xin Lyu, and Xiuhan Wang. Range avoidance for low-depth circuits and connections to pseudorandomness. In *26th International Conference on Randomization and Computation (RANDOM 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[GO14] Jens Groth and Rafail Ostrovsky. Cryptography in the multi-string model. *Journal of cryptology*, 27(3):506–543, 2014.

[Gol01] Oded Goldreich. *Foundations of Cryptography*, volume 1. Cambridge University Press, 2001.

[GPR95] David Grable, Kevin Phelps, and Vojtěch Rödl. The minimum independence number for designs. *Combinatorica*, 15:175–185, 1995.

[GR08] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008. Preliminary version in FOCS 2005.

[GRS06] Ariel Gabizon, Ran Raz, and Ronen Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. *SIAM Journal on Computing*, 36(4):1072–1094, 2006. Preliminary version in FOCS 2004.

[GRS22] Venkatesan Guruswami, Atri Rudra, and Madhu Sudan. Essential coding theory. Draft available at https://cse.buffalo.edu/faculty/atri/courses/coding-theory/book/, 2022.

[GRT18] Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In *50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 990–1002, 2018.

[GSV05] Shafi Goldwasser, Madhu Sudan, and Vinod Vaikuntanathan. Distributed computing with imperfect randomness. In *Distributed Computing: 19th International Conference (DISC 2005)*, pages 288–302. Springer, 2005.

[GSZ21] Vipul Goyal, Akshayaram Srinivasan, and Chenzhi Zhu. Multi-source non-malleable extractors and applications. In *40th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2021)*, pages 468–497. Springer, 2021.

[Gur04] Venkatesan Guruswami. Better extractors for better codes? In *36th annual ACM Symposium on Theory of Computing (STOC 2004)*, pages 436–444, 2004.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):1–34, 2009. Preliminary version in CCC 2007.

[GVJZ23] Zeyu Guo, Ben Lee Volk, Akhil Jalan, and David Zuckerman. Extractors for images of varieties. In *55th Annual ACM Symposium on Theory of Computing (STOC 2023)*, 2023.

[GVW15] Oded Goldreich, Emanuele Viola, and Avi Wigderson. On randomness extraction in AC^0. In *30th Conference on Computational Complexity (CCC 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[GW96] Ian Goldberg and David Wagner. Randomness and the Netscape browser. *Dr Dobb's Journal-Software Tools for the Professional Programmer*, 21(1):66–71, 1996.

[GW97] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: a quality-size trade-off for hashing. *Random Structures & Algorithms*, 1997. Preliminary version in STOC 1994.

[GW20] Mika Göös and Thomas Watson. A lower bound for sampling disjoint sets. *ACM Transactions on Computation Theory (TOCT)*, 12(3):1–13, 2020. Preliminary version in RANDOM 2019.

[Ham50] Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *18th Annual ACM Symposium on Theory of Computing (STOC 1986)*, pages 6–20, 1986.

[HG91] Johan Håstad and Mikael Goldmann. On the power of small-depth threshold circuits. *Computational Complexity*, 1:113–129, 1991. Preliminary version in FOCS 1990.

[HG17] Miguel Herrero-Collantes and Juan Carlos Garcia-Escartin. Quantum random number generators. *Reviews of Modern Physics*, 89(1):015004, 2017.

[HH64] J. M. Hammersley and D.C. Handscomb. *Monte Carlo methods*. Methuen, 1964.

[HHTT23] Pooya Hatami, William M Hoza, Avishay Tal, and Roei Tell. Depth-d threshold circuits vs. depth-$(d + 1)$ AND-OR trees. In *55th Annual ACM Symposium on Theory of Computing (STOC 2023)*, pages 895–904, 2023.

[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. Preliminary version in STOC 1989.

[HIV22] Xuangui Huang, Peter Ivanov, and Emanuele Viola. Affine extractors and AC^0-parity. In *26th International Conference on Randomization and Computation (RANDOM 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[HLW06] Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[Hoc59] Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffers*, 2:147–156, 1959.

[HR15] Pavel Hrubes and Anup Rao. Circuits with medium fan-in. In *30th Conference on Computational Complexity (CCC 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[HS70] Wassily Hoeffding and Gordon Simons. Unbiased coin tossing with a biased coin. *The Annals of Mathematical Statistics*, pages 341–352, 1970.

[IN96] Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. *Journal of cryptology*, 9(4):199–216, 1996. Preliminary version in FOCS 1989.

[IRW94] Russell Impagliazzo, Ran Raz, and Avi Wigderson. A direct product theorem. In *9th Annual Conference on Structure in Complexity Theory (CCC 1994)*, pages 88–96. IEEE, 1994.

[ISW00] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *32nd Annual ACM Symposium on Theory of Computing (STOC 2000)*, pages 1–10, 2000.

[IW97] Russell Impagliazzo and Avi Wigderson. $\mathsf{P} = \mathsf{BPP}$ if E requires exponential circuits: Derandomizing the XOR lemma. In *29th Annual ACM Symposium on Theory of Computing (STOC 1997)*, pages 220–229, 1997.

[Jai15] Rahul Jain. New strong direct product results in communication complexity. *Journal of the ACM (JACM)*, 62(3):1–27, 2015.

[JK99] Benjamin Jun and Paul Kocher. The Intel random number generator. *Cryptography Research Inc. White Paper*, 27:1–8, 1999.

[JSWZ13] Rahul Jain, Yaoyun Shi, Zhaohui Wei, and Shengyu Zhang. Efficient protocols for generating bipartite classical distributions and quantum states. *IEEE Transactions on Information Theory*, 59(8):5171–5178, 2013. Preliminary version in SODA 2013.

[JVV86] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical computer science*, 43:169–188, 1986. Preliminary version in ICALP 1985.

[KJS01] Kaoru Kurosawa, Thomas Johansson, and Douglas R Stinson. Almost k-wise independent sample spaces and their cryptologic applications. *Journal of Cryptology*, 14:231–253, 2001. Preliminary version in EUROCRYPT 1997.

[KKL88] Jeff Kahn, Gil Kalai, and Nathan Linial. The influence of variables on boolean functions. In *29th Annual Symposium on Foundations of Computer Science (FOCS 1988)*, pages 68–80. IEEE, 1988.

[KKMP21] Robert Kleinberg, Oliver Korten, Daniel Mitropolsky, and Christos Papadimitriou. Total functions in the polynomial hierarchy. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[KL20] Jonathan Katz and Yehuda Lindell. *Introduction to modern cryptography*. CRC press, 2020.

[Kla10] Hartmut Klauck. A strong direct product theorem for disjointness. In *42nd ACM Symposium on Theory of Computing (STOC 2010)*, pages 77–86, 2010.

[KLR09] Yael Tauman Kalai, Xin Li, and Anup Rao. 2-source extractors under computational assumptions and cryptography with defective randomness. In *50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 617–626. IEEE, 2009.

[KLRZ08] Yael Tauman Kalai, Xin Li, Anup Rao, and David Zuckerman. Network extractor protocols. In *49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 654–663. IEEE, 2008.

[KM04] Robert König and Ueli Maurer. Extracting randomness from generalized symbol-fixing and markov sources. In *International Symposium on Information Theory (ISIT 2004)*, page 232. IEEE, 2004.

[KM05] Robert König and Ueli Maurer. Generalized strong extractors and deterministic privacy amplification. In *Cryptography and Coding: 10th IMA International Conference*, pages 322–339. Springer, 2005.

[KMS19] Ashutosh Kumar, Raghu Meka, and Amit Sahai. Leakage-resilient secret sharing against colluding parties. In *60th Annual Symposium on Foundations of Computer Science (FOCS 2019)*, pages 636–660. IEEE, 2019.

[KMV14] Alexandr Kostochka, Dhruv Mubayi, and Jacques Verstraëte. On independent sets in hypergraphs. *Random Structures & Algorithms*, 44(2):224–239, 2014.

[KN96] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996.

[Kor21] Oliver Korten. The hardest explicit construction. In *62nd Annual Symposium on Foundations of Computer Science (FOCS 2021)*, pages 433–444. IEEE, 2021.

[KR19] Yael Tauman Kalai and Leonid Reyzin. *A Survey of Leakage-Resilient Cryptography*, page 727–794. Association for Computing Machinery, New York, NY, USA, 2019.

[KRVZ11] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. *Journal of Computer and System Sciences*, 77(1):191–220, 2011. Preliminary version in STOC 2006.

[KS21] Dakshita Khurana and Akshayaram Srinivasan. Improved computational extractors and their applications. In *41st Annual International Cryptology Conference (CRYPTO 2021)*, pages 566–594. Springer, 2021.

[KVMS12] Jeff Kinne, Dieter Van Melkebeek, and Ronen Shaltiel. Pseudorandom generators, typically-correct derandomization, and circuit lower bounds. *Computational complexity*, 21:3–61, 2012. Preliminary version in RANDOM 2009.

[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. Preliminary version in FOCS 2003.

[LCG$^+$20] Fuchun Lin, Mahdi Cheraghchi, Venkatesan Guruswami, Reihaneh Safavi-Naini, and Huaxiong Wang. Leakage-resilient secret sharing in non-compartmentalized models. In *1st Conference on Information-Theoretic Cryptography (ITC 2020)*, volume 163 of *LIPIcs*, pages 7:1–7:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[Lev86] Leonid A Levin. Average case complete problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

[Lew19] Mark Lewko. An explicit two-source extractor with min-entropy rate near $4/9$. *Mathematika*, 65(4):950–957, 2019.

[Li11a] Xin Li. Improved constructions of three source extractors. In *26th Annual Conference on Computational Complexity (CCC 2011)*, pages 126–136. IEEE, 2011.

[Li11b] Xin Li. A new approach to affine extractors and dispersers. In *26th Annual Conference on Computational Complexity (CCC 2011)*, pages 137–147. IEEE, 2011.

[Li12a] Xin Li. Design extractors, non-malleable condensers and privacy amplification. In *44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 837–854, 2012.

[Li12b] Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *53rd Annual Symposium on Foundations of Computer Science (FOCS 2012)*, pages 688–697. IEEE, 2012.

[Li13a] Xin Li. Extractors for a constant number of independent sources with polylogarithmic min-entropy. In *54th Annual Symposium on Foundations of Computer Science (FOCS 2013)*, pages 100–109. IEEE, 2013.

[Li13b] Xin Li. New independent source extractors with exponential improvement. In *45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 783–792, 2013.

[Li15] Xin Li. Three-source extractors for polylogarithmic min-entropy. In *56th Annual Symposium on Foundations of Computer Science (FOCS 2015)*, pages 863–882. IEEE, 2015.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pages 168–177. IEEE, 2016.

[Li17] Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *49th Annual ACM Symposium on Theory of Computing (STOC 2017)*, pages 1144–1156, 2017.

[Li19] Xin Li. Non-malleable extractors and non-malleable codes: partially optimal constructions. In *34th Computational Complexity Conference (CCC 2019)*, pages 1–49, 2019.

[Li23] Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. *arXiv preprint arXiv:2303.06802*, 2023.

[Liu23] Kuikui Liu. *Spectral Independence a New Tool to Analyze Markov Chains*. PhD thesis, University of Washington, 2023.

[LLS89] David Lichtenstein, Nathan Linial, and Michael Saks. Some extremal problems arising from discrete control processes. *Combinatorica*, 9(3):269–287, 1989. Preliminary version in STOC 1987.

[LLTT05] Chia-Jung Lee, Chi-Jen Lu, Shi-Chun Tsai, and Wen-Guey Tzeng. Extracting randomness from multiple independent sources. *IEEE Transactions on Information Theory*, 51(6):2224–2227, 2005.

[LM22] Xizhi Liu and Dhruv Mubayi. On explicit constructions of designs. *the electronic journal of combinatorics*, pages P1–53, 2022.

[LRVW03] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: Optimal up to constant factors. In *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 602–611, 2003.

[LS87] Zhenqin Li and Harold A Scheraga. Monte Carlo-minimization approach to the multiple-minima problem in protein folding. *Proceedings of the National Academy of Sciences (PNAS)*, 84(19):6611–6615, 1987.

[LS09] Troy Lee and Adi Shraibman. Lower bounds in communication complexity. *Foundations and Trends® in Theoretical Computer Science*, 3(4):263–399, 2009.

[LSŠ08] Troy Lee, Adi Shraibman, and Robert Špalek. A direct product theorem for discrepancy. In *2008 23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*, pages 71–80. IEEE, 2008.

[Lu04] Chi-Jen Lu. Encryption against storage-bounded adversaries from on-line strong extractors. *Journal of Cryptology*, 17(1), 2004. Preliminary version in CRYPTO 2002.

[LV12] Shachar Lovett and Emanuele Viola. Bounded-depth circuits cannot sample good codes. *computational complexity*, 21:245–266, 2012. Preliminary version in CCC 2011.

[LY22] Jiatu Li and Tianqi Yang. $3.1n - o(n)$ circuit lower bounds for explicit functions. In *54th Annual ACM Symposium on Theory of Computing (STOC 2022)*, pages 1180–1193, 2022.

[LZ19] Fu Li and David Zuckerman. Improved extractors for recognizable and algebraic sources. In *23rd International Conference on Randomization and Computation (RANDOM 2019)*, 2019.

[McE78] R. J. McEliece. A public-key cryptosystem based on algebraic coding theory. *Deep Space Network Progress Report*, 44:114–116, 1978.

[Mek17] Raghu Meka. Explicit resilient functions matching Ajtai-Linial. In *28th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2017)*, pages 1132–1148. SIAM, 2017.

[Mil76] Gary L Miller. Riemann's hypothesis and tests for primality. *Journal of Computer and System Sciences*, 13(3):300–317, 1976. Preliminary version in STOC 1975.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized algorithms*. Cambridge University Press, 1995.

[MU02] Elchanan Mossel and Christopher Umans. On the complexity of approximating the VC dimension. *Journal of Computer and System Sciences*, 65(4):660–671, 2002. Preliminary version in CCC 2001.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In *17th Annual International Cryptology Conference (CRYPTO 1997)*, pages 307–321. Springer, 1997.

[NS20] Jesper Buus Nielsen and Mark Simkin. Lower bounds for leakage-resilient secret sharing. In *39th Annual International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2020)*, pages 556–577. Springer, 2020.

[NT99] Noam Nisan and Amnon Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58(1):148–173, 1999.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994. Preliminary version in FOCS 1988.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996. Preliminary version in STOC 1993.

[PSL80] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *Journal of the ACM (JACM)*, 27(2):228–234, 1980.

[PTVF07] William H Press, Saul A Teukolsky, William T Vetterling, and Brian P Flannery. *Numerical recipes: The art of scientific computing (3rd edition)*. Cambridge University Press, 2007.

[Rab80] Michael O Rabin. Probabilistic algorithm for testing primality. *Journal of number theory*, 12(1):128–138, 1980.

[Rao07] Anup Rao. An exposition of Bourgain's 2-source extractor. *Electronic Colloquium on Computational Complexity (ECCC)*, TR07-034, 2007.

[Rao09a] Anup Rao. Extractors for a constant number of polynomially small min-entropy independent sources. *SIAM Journal on Computing*, 39(1):168–194, 2009. Preliminary version in STOC 2006.

[Rao09b] Anup Rao. Extractors for low-weight affine sources. In *24th Annual IEEE Conference on Computational Complexity (CCC 2009)*, pages 95–101. IEEE, 2009.

[Raz98] Ran Raz. A parallel repetition theorem. *SIAM Journal on Computing*, 27(3):763–803, 1998. Preliminary version in STOC 1995.

[Raz05] Ran Raz. Extractors with weak random seeds. In *37th Annual ACM Symposium on Theory of Computing (STOC 2005)*, pages 11–20, 2005.

[Rem16] Zachary Remscrim. The Hilbert function, algebraic extractors, and recursive fourier sampling. In *57th Annual Symposium on Foundations of Computer Science (FOCS 2016)*, pages 197–208. IEEE, 2016.

[RRV02] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *Journal of Computer and System Sciences*, 1(65):97–128, 2002. Preliminary version in STOC 1999.

[RŠ94] Vojtěch Rödl and Edita Šinajová. Note on independent sets in steiner systems. *Random Structures & Algorithms*, 5(1):183–190, 1994.

[RSW06] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. *SIAM Journal on Computing*, 35(5):1185–1209, 2006. Preliminary version in FOCS 2000.

[RT00] Jaikumar Radhakrishnan and Amnon Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000. Preliminary version in FOCS 1997.

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics*, 155:157–187, 2002. Preliminary version in FOCS 2000.

[RW93] Alexander Razborov and Avi Wigderson. $n^{\omega(\log n)}$ lower bounds on the size of depth-3 threshold circuits with AND gates at the bottom. *Information Processing Letters*, 45(6):303–307, 1993.

[RY11] Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *Journal of Computer and System Sciences*, 77(1):167–190, 2011. Preliminary version in FOCS 2008.

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020.

[RZ08] Anup Rao and David Zuckerman. Extractors for three uneven-length sources. In *12th International Conference on Randomization and Computation (RANDOM 2008)*, pages 557–570. Springer, 2008.

[Sam68] Paul A Samuelson. Constructing an unbiased random sequence. *Journal of the American Statistical Association*, 63(324):1526–1527, 1968.

[Sav98] John E Savage. *Models of computation*, volume 136. Addison-Wesley Reading, MA, 1998.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009.

[Sch80] Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980. Preliminary version in EUROSAM 1979.

[Sha48] Claude Elwood Shannon. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423, 1948.

[Sha49] Claude E Shannon. Communication theory of secrecy systems. *The Bell system technical journal*, 28(4):656–715, 1949.

[Sha79] Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.

[Sha04a] Ronen Shaltiel. Recent developments in explicit constructions of extractors. *Current Trends in Theoretical Computer Science: The Challenge of the New Century Vol 1: Algorithms and Complexity Vol 2: Formal Models and Semantics*, pages 189–228, 2004.

[Sha04b] Ronen Shaltiel. Towards proving strong direct product theorems. *Computational Complexity*, 12(1/2):1–22, 2004. Preliminary version in CCC 2001.

[Sha08] Ronen Shaltiel. How to get more mileage from randomness extractors. *Random Structures & Algorithms*, 33(2):157–186, 2008. Preliminary version in CCC 2006.

[Sha11a] Ronen Shaltiel. Dispersers for affine sources with sub-polynomial entropy. In *52nd Annual Symposium on Foundations of Computer Science (FOCS 2011)*, pages 247–256. IEEE, 2011.

[Sha11b] Ronen Shaltiel. An introduction to randomness extractors. In *International Colloquium on Automata, Languages, and Programming (ICALP 2011)*, pages 21–41. Springer, 2011.

[Sha11c] Ronen Shaltiel. Weak derandomization of weak algorithms: explicit versions of yao's lemma. *computational complexity*, 20:87–143, 2011. Preliminary version in CCC 2009.

[Sid95] Alexander Sidorenko. What we know and what we do not know about turán numbers. *Graphs and Combinatorics*, 11(2):179–199, 1995.

[Sid18] Alexander Sidorenko. Extremal problems on the hypercube and the codegree turán density of complete r-graphs. *SIAM Journal on Discrete Mathematics*, 32(4):2667–2674, 2018.

[Sid20] Alexander Sidorenko. On generalized erdős–ginzburg–ziv constants for $\mathbb{Z}_2^d$. *Journal of Combinatorial Theory, Series A*, 174:105254, 2020.

[SSZ98] Michael Saks, Aravind Srinivasan, and Shiyu Zhou. Explicit or-dispersers with polylogarithmic degree. *Journal of the ACM (JACM)*, 45(1):123–154, 1998. Preliminary version in STOC 1995.

[ST04] Daniel A Spielman and Shang-Hua Teng. Nearly-linear time algorithms for graph partitioning, graph sparsification, and solving linear systems. In *36th Annual Symposium on Theory of Computing (STOC 2004)*, pages 81–90. ACM, 2004.

[SU05] Ronen Shaltiel and Christopher Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM (JACM)*, 52(2):172–216, 2005. Preliminary version in FOCS 2001.

[SV86] Miklos Santha and Umesh V Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of computer and system sciences*, 33(1):75–87, 1986. Preliminary version in FOCS 1984.

[SV19] Akshayaram Srinivasan and Prashant Nalini Vasudevan. Leakage resilient secret sharing and applications. In *39th Annual International Cryptology Conference (CRYPTO 2019)*, pages 480–509. Springer, 2019.

[SW84] Quentin F Stout and Bette Warren. Tree algorithms for unbiased coin tossing with a biased coin. *The Annals of Probability*, 12(1):212–222, 1984.

[SZ99] Aravind Srinivasan and David Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28(4):1433–1459, 1999. Preliminary version in FOCS 1994.

[Ta-96] Amnon Ta-Shma. On extracting randomness from weak random sources. In *28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 276–285, 1996.

[Ta-02] Amnon Ta-Shma. Almost optimal dispersers. *Combinatorica*, 22(1):123–145, 2002. Preliminary version in STOC 1998.

[Tha98] Jayram S Thathachar. On separating the read-k-times branching program hierarchy. In *30th Annual ACM Symposium on Theory of Computing (STOC 1998)*, pages 653–662, 1998.

[TL18] Fang Tian and Zi-Long Liu. Bounding the independence number in some (n, k, ℓ, λ)-hypergraphs. *Graphs and Combinatorics*, 34:845–861, 2018.

[Tre01] Luca Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, 2001. Preliminary version in STOC 1999.

[TSG23] Karan Taneja, Richard Segal, and Richard Goodwin. Monte Carlo tree search for recipe generation using GPT-2. In *14th International Conference on Computational Creativity (ICCC 2023)*, 2023.

[TU12] Amnon Ta-Shma and Christopher Umans. Better condensers and new extractors from parvaresh-vardy codes. In *27th Conference on Computational Complexity (CCC 2012)*, pages 309–315. IEEE, 2012.

[TUZ07] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Lossless condensers, unbalanced expanders, and extractors. *Combinatorica*, 27:213–240, 2007. Preliminary version in STOC 2001.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *41st Annual Symposium on Foundations of Computer Science (FOCS 2000)*, pages 32–42. IEEE, 2000.

[TZ04] Amnon Ta-Shma and David Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004. Preliminary version in STOC 2001.

[TZS06] Amnon Ta-Shma, David Zuckerman, and Shmuel Safra. Extractors from Reed–Muller codes. *Journal of Computer and System Sciences*, 72(5):786–812, 2006. Preliminary version in FOCS 2001.

[Uma99] Christopher Umans. Hardness of approximating σ_2^p minimization problems. In *40th Annual Symposium on Foundations of Computer Science (FOCS 1999)*, pages 465–474. IEEE, 1999.

[Vad04] Salil P Vadhan. Constructing locally computable extractors and cryptosystems in the bounded-storage model. *Journal of Cryptology*, 17:43–77, 2004. Preliminary version in CRYPTO 2003.

[Vad12] Salil Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Var57] Rom Rubenovich Varshamov. Estimate of the number of signals in error correcting codes. *Docklady Akad. Nauk, SSSR*, 117:739–741, 1957.

[Vaz87a] Umesh Vazirani. Efficiency considerations in using semi-random sources. In *19th Annual ACM Symposium on Theory of Computing (STOC 1987)*, pages 160–168, 1987.

[Vaz87b] Umesh V Vazirani. Strong communication complexity or generating quasi-random sequences from two communicating semi-random sources. *Combinatorica*, 7(4):375–392, 1987. Preliminary version in STOC 1985.

[Vio05] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *computational complexity*, 13(3-4):147–188, 2005. Preliminary version in CCC 2003.

[Vio12a] Emanuele Viola. The complexity of distributions. *SIAM Journal on Computing*, 41(1):191–218, 2012. Preliminary version in FOCS 2010.

[Vio12b] Emanuele Viola. Extractors for turing-machine sources. In *16th International Conference on Randomization and Computation (RANDOM 2012)*, pages 663–671. Springer, 2012.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014. Preliminary version in FOCS 2011.

[Vio16] Emanuele Viola. Quadratic maps are hard to sample. *ACM Transactions on Computation Theory (TOCT)*, 8(4):1–4, 2016.

[Vio20] Emanuele Viola. Sampling lower bounds: boolean average-case and permutations. *SIAM Journal on Computing*, 49(1):119–137, 2020.

[Vio23] Emanuele Viola. New sampling lower bounds via the separator. In *38th Computational Complexity Conference (CCC 2023)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[vN51] John von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12(36-38):1, 1951.

[VV85] Umesh V Vazirani and Vijay V Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *26th Annual Symposium on Foundations of Computer Science (FOCS 1985)*, pages 417–428. IEEE, 1985.

[War36] Ewald Warning. Bemerkung zur vorstehenden arbeit von herrn chevalley. *Abh. Math. Sem. Univ. Hamburg*, 11:76–83, 1936.

[Wat13] Thomas Watson. Time hierarchies for sampling distributions. In *4th Conference on Innovations in Theoretical Computer Science (ITCS 2013)*, pages 429–440, 2013.

[Wat16] Thomas Watson. Nonnegative rank vs. binary rank. *Chicago Journal of Theoretical Computer Science*, 2:1–13, 2016.

[Wat20] Thomas Watson. Communication complexity with small advantage. *computational complexity*, 29(1):2, 2020. Preliminary version in CCC 2018.

[Wil16] Richard Ryan Williams. Strong ETH breaks with Merlin and Arthur: short non-interactive proofs of batch evaluation. In *31st Conference on Computational Complexity (CCC 2016)*, pages 1–17, 2016.

[WP23] Adam Bene Watts and Natalie Parham. Unconditional quantum advantage for sampling with shallow circuits. *arXiv preprint arXiv:2301.00995*, 2023.

[WZ99] Avi Wigderson and David Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. *Combinatorica*, 19:125–138, 1999. Preliminary version in STOC 1993.

[Yao79] Andrew Chi-Chih Yao. Some complexity questions related to distributive computing. In *11th Annual ACM Symposium on Theory of Computing (STOC 1979)*, pages 209–213, 1979.

[Yao82] Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (FOCS 1982)*, pages 80–91. IEEE, 1982.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles. In *26th Annual Symposium on Foundations of Computer Science (FOCS 1985)*, pages 1–10. IEEE, 1985.

[Yao90] AC-C Yao. On ACC and threshold circuits. In *31st Annual Symposium on Foundations of Computer Science (FOCS 1990)*, pages 619–627. IEEE, 1990.

[Yeh11] Amir Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245, 2011.

[YZ24] Huacheng Yu and Wei Zhan. Sampling, flowers and communication. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2024.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979.

[Zuc90] David Zuckerman. General weak random sources. In *31st Annual Symposium on Foundations of Computer Science (FOCS 1990)*, pages 534–543. IEEE, 1990.

[Zuc96a] David Zuckerman. On unapproximable versions of NP-complete problems. *SIAM Journal on Computing*, 25(6):1293–1304, 1996. Preliminary version in CCC 1993.

[Zuc96b] David Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16:367–391, 1996. Preliminary version in FOCS 1991.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997. Preliminary version in STOC 1996.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory of Computing*, 3:103–128, 2007. Preliminary version in STOC 2006.