Electronic Colloquium on Computational Complexity, Revision 5 of Report No. 106 (2020)

Improved Extractors for Small-Space Sources

Eshan Chattopadhyay*
Cornell University
eshanc@cornell.edu

Jesse Goodman*
Cornell University
jpmgoodman@cs.cornell.edu

August 24, 2021

Abstract

We study the problem of extracting random bits from weak sources that are sampled by algorithms with limited memory. This model of *small-space sources* was introduced by Kamp, Rao, Vadhan and Zuckerman (STOC'06), and falls into a line of research initiated by Trevisan and Vadhan (FOCS'00) on extracting randomness from weak sources that are sampled by computationally bounded algorithms. Our main results are the following.

1. We obtain near-optimal extractors for small-space sources in the polynomial error regime. For space s sources over n bits, our extractors require just $k \geq s \cdot \mathrm{polylog}(n)$ entropy. This is an exponential improvement over the previous best result, which required $k \geq s^{1.1} \cdot 2^{\log^{0.51} n}$ (Chattopadhyay and Li, STOC'16).
2. We obtain improved extractors for small-space sources in the negligible error regime. For space s sources over n bits, our extractors require entropy $k \geq n^{1/2+\delta} \cdot s^{1/2-\delta}$, whereas the previous best result required $k \geq n^{2/3+\delta} \cdot s^{1/3-\delta}$ (Chattopadhyay, Goodman, Goyal and Li, STOC'20).

To obtain our first result, the key ingredient is a new reduction from small-space sources to affine sources, allowing us to simply apply a good affine extractor.

To obtain our second result, we must develop some new machinery, since we do not have low-error affine extractors that work for low entropy. Our main tool is a significantly improved extractor for adversarial sources, which is built via a simple framework that makes novel use of a certain kind of leakage-resilient extractors (known as *cylinder intersection extractors*), by combining them with a general type of extremal designs. Our key ingredient is the first derandomization of these designs, which we obtain using new connections to coding theory and additive combinatorics.

*Supported by NSF CAREER Award 2045576.

ISSN 1433-8092

1 Introduction

Randomness is a powerful computational resource that has found beautiful applications in algorithm design, cryptography, and combinatorics (see [Vad12] for an excellent survey). Unfortunately, such applications require access to uniform bits, but randomness harvested from natural phenomena (e.g., radioactive decay, atmospheric noise) rarely looks so pure. Such motivates the study of *randomness extractors*, which are algorithms that convert these weak sources of randomness into distributions that are close to uniform:

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 an* extractor *for $\mathcal{X}$ with error ϵ if for every $\mathbf{X} \in \mathcal{X}$,*

$$|\mathsf{Ext}(\mathbf{X}) - \mathbf{U}_m| \leq \epsilon,$$

where $\mathbf{U}_m$ is the uniform distribution over $\{0,1\}^m$, and $|\cdot|$ denotes statistical distance.

Beyond purifying natural sources of randomness, extractors have found deep connections to complexity theory, cryptography, coding theory, and combinatorics (see, e.g., [Sha11, Vad12]). Constructing these objects has thus produced a fruitful line of research over the past 30 years, where various distribution families $\mathcal{X}$ and errors ϵ have been considered depending on the motivating application.

In order for extraction to be possible, each source $\mathbf{X} \in \mathcal{X}$ must have *some* randomness. In this field, it is standard to measure the randomness content of $\mathbf{X}$ as its *min-entropy*, defined as $H_\infty(\mathbf{X}) := \min_x \log(1/\Pr[\mathbf{X} = x])$. Unfortunately, it turns out that a min-entropy requirement alone is not enough to enable extraction. Indeed, an easy folklore argument shows that even if every source $\mathbf{X} \in \mathcal{X}$ has min-entropy $k \geq n - 1$, there cannot exist an extractor Ext for $\mathcal{X}$ that achieves nontrivial error $\epsilon < 1/2$.

To circumvent this impossibility result, researchers have considered two main directions. In the first direction, one assumes that each source $\mathbf{X} \in \mathcal{X}$ comes with a uniform seed $\mathbf{U}_d$, which can be used to extract uniform bits from the rest of the source, which has some min-entropy guarantee. Extractors in this setting are called *seeded extractors*, and near-optimal constructions of these objects are now known [LRVW03, GUV09, DKSS13]. In this paper, we focus on the second direction, where one assumes each source $\mathbf{X} \in \mathcal{X}$ has some additional structure beyond its min-entropy guarantee.

Samplable sources One natural way to equip each distribution $\mathbf{X} \in \mathcal{X}$ with some additional structure is to assume that it can be *sampled efficiently*, i.e., generated by an algorithm that has limited computational resources. Such sources were introduced by Trevisan and Vadhan [TV00], under the suggestion that they are a good model for distributions that would actually arise in nature. In [TV00], and the follow-up works of Viola [Vio14] and Li [Li16], the authors consider *circuit sources*: distributions that can be sampled by small circuits. Such sources can be thought of as distributions sampled by algorithms with limited *time*.

In this paper, we consider distributions that can be sampled by algorithms with limited *memory*. Known as *small-space sources*, this family of distributions was introduced by Kamp, Rao, Vadhan, and Zuckerman [KRVZ06], and further studied in recent work [CL16, CGGL20]. To define this class of sources formally, one uses *branching programs* to model the evolution of state in the small-space algorithm. 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 1.2 (Small-space source)**.** *A distribution* $\mathbf{X}$ *over* $\{0,1\}^n$ *is 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.*

Beyond their motivation in modeling distributions that one might actually find in nature, small-space sources are powerful enough to capture several other well-studied models. As noted in [KRVZ06], small-space sources can simulate: von Neumann's model of a coin with unknown bias [vN51]; the finite Markov chain model of Blum [Blu86]; the space-bounded models of Vazirani [Vaz87] and Koenig and Maurer [KM04, KM05]; and the popular models of oblivious bit-fixing and symbol-fixing sources [CGH+85, KZ06] and independent sources [CG88]. In fact, it is suggested in [KRVZ06] that the only model of sources that appears unrelated to small-space sources is the class of *affine sources* [GR08].

1.1 Summary of our results

In this paper, we explicitly construct two significantly improved extractors for small-space sources. Along the way, we prove a new structural result for small-space sources, and provide new explicit constructions of several related pseudorandom objects. Our extractors follow easily from these new key ingredients, which may be of independent interest. We formally state these results, below.

1.1.1 Small-space extractors for polylogarithmic entropy

In our first main theorem, we construct near-optimal extractors for small-space sources in the polynomial error regime.

Theorem 1. *There exists a universal constant* $C > 0$ *such that for all* $n, k, s \in \mathbb{N}$ *satisfying* $k \geq s \cdot \log^C(n)$*, there exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for space* s *sources with min-entropy* k*, which has output length* $m = (k/s)^{\Omega(1)}$ *and error* $\epsilon = n^{-\Omega(1)}$*.*

Thus, our extractor requires min-entropy $k \geq s \cdot \log^C(n)$, which is an exponential improvement over the previous best requirement [CL16] of $k \geq s^{1.1} \cdot 2^{\log^{0.51}(n)}$. In particular, in the natural setting of sources sampled by $s = \mathrm{polylog}(n)$ space algorithms, our extractor is the first construction that works for polylogarithmic entropy. Non-constructively, it is known that small-space extractors exist for min-entropy $k \geq O(s + \log n + \log(1/\epsilon))$, and thus our result is nearly optimal when the desired error is at most polynomially small.

The key ingredient we use to prove Theorem 1 is a new structural result, which establishes a connection between small-space sources and *affine sources*. An affine source $\mathbf{X}$ over n bits with min-entropy k is a distribution that is uniform over some (unknown) affine subspace of $\mathbb{F}_2^n$. A long line of work has considered the problem of constructing extractors for affine sources [GR08, DG10, Bou07, Yeh11, Li11, Rao09, Li16, CGL21], and in this work we show that such extractors can also extract from small-space sources. In particular, we prove the following.

Theorem 2. *Let* $\mathbf{X}$ *be a space* s *source over* $\{0,1\}^n$ *with min-entropy* k*. Then* $\mathbf{X}$ *is* $2^{-\Omega(k)}$*-close to a convex combination of affine sources with min-entropy* $\Omega(\frac{k}{s \log(n/k)})$*.*

By combining this structural result with the explicit affine extractor of Li [Li16], which works for polylog(n) min-entropy and has polynomially small error, we immediately obtain Theorem 1. Furthermore, if we are only interested in outputting one bit with constant error, we can use the recent affine extractor of Chattopadhyay, Goodman, and Liao [CGL21] to extract from small-space sources with min-entropy $k \geq s \cdot \log^{2+o(1)}(n)$.

1.1.2 Small-space extractors with exponentially small error

While polynomially small error suffices for many applications, it is sometimes important to achieve negligible error in applications such as cryptography [DOPS04]. However, since the best low-error affine extractors require entropy $k \geq \Omega(n/\sqrt{\log \log n})$ [Bou07, Yeh11, Li11], Theorem 2 does not yield any new result in the negligible error setting.

In our next main result, we develop some new machinery in order to obtain improved low-error extractors for small-space sources. Until recently, the best extractors for such sources [KRVZ06] required entropy $k \geq Cn^{1-\gamma}s^{\gamma}$, where $\gamma > 0$ is some tiny constant and C is a large one. In [CGGL20], the entropy requirement was improved to $k \geq Cn^{2/3+\delta}s^{1/3-\delta}$. We reduce this entropy requirement further, and prove the following.

Theorem 3. *For any fixed $\delta \in (0, 1/2]$ there is a constant $C > 0$ such that for all $n, k, s \in \mathbb{N}$ satisfying $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$, there exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources of min-entropy k, with output length $m = n^{\Omega(1)}$ and error $\epsilon = 2^{-n^{\Omega(1)}}$.*

Observe that the line of improvements described above (from [KRVZ06] to [CGGL20] to Theorem 3) is strict, since we always have $s < n$ (or else the bounds are trivial). In particular, note that for, say $s = n^{\delta}$ space, the entropy requirement has dropped from $k \geq O(n^{1-\gamma})$ to $k \geq O(n^{2/3+\delta})$ to $k \geq O(n^{1/2+\delta})$.

To prove Theorem 3, we start with the standard approach [KRVZ06] of reducing small-space sources to the class of *adversarial sources* [CGGL20]. Informally, an adversarial source $\mathbf{X}$ consists of many independent sources, where only a few of them are guaranteed to be "good" (i.e., contain some min-entropy). Formally, an (N, K, n, k)-adversarial source $\mathbf{X}$ consists of N independent sources $\mathbf{X}_1, \ldots, \mathbf{X}_N$, each over n bits, with the guarantee that at least K of them have min-entropy at least k. Such sources have applications in generating a (cryptographic) common random string in the presence of adversaries, and in harvesting randomness from unreliable sources.

To prove Theorem 3, we explicitly construct significantly improved extractors for adversarial sources:

Theorem 4. *There is a universal constant $C > 0$ such that for any fixed $\delta > 0$ and all sufficiently large $N, K, n, k \in \mathbb{N}$ satisfying $k \geq \log^C n$ and $K \geq N^{\delta}$, there exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources, with output length $m = k^{\Omega(1)}$ and error $\epsilon = 2^{-k^{\Omega(1)}}$.*

Previously, the best extractor for this setting [CGGL20] required $K \geq N^{0.5+o(1)}$ good sources, and our improvement to $K \geq N^{\delta}$ is crucial in obtaining better extractors for small-space sources. An added bonus is that our extractor construction is arguably much simpler compared to [CGGL20].

To prove Theorem 4, we develop a simple new framework for extracting from adversarial sources by combining (i) a general type of combinatorial design; and (ii) a specific kind of leakage-resilient extractor [KMS19, CGG+20]. While such leakage-resilient extractors were recently constructed explicitly in [CGG+20], the only known construction of such designs is probabilistic [RŠ94].

Thus, the key ingredient we use to prove Theorem 4, and subsequently Theorem 3, is the first explicit construction of such designs. In more detail, an (n, r, s)-design is an r-uniform hypergraph over n vertices with pairwise hyperedge intersections of size $< s$. To instantiate our framework, we need explicit (n, r, s)-designs with small independence number[1] α. Previously, Chattopadhyay, Goodman, Goyal and Li [CGGL20] constructed $(n, 3, 2)$-designs with independence number $\alpha \leq O(n^{0.923})$. To obtain our improved extractors in Theorems 3 and 4, we need designs with much smaller independence number. Our final main theorem constructs exactly such designs.

Theorem 5. *For all constants $r \geq s \in \mathbb{N}$ with r even, there exist explicit (n, r, s)-designs $(G_n)_{n\in\mathbb{N}}$ with independence number*

$$\alpha(G_n) \leq O(n^{\frac{2(r-s)}{r}}).$$

Theorem 5 gives the first derandomization of a result by Rödl and Šinajová [RŠ94], and our explicit designs are optimal up to a factor of 2 in the power. We show that it is easy to extend Theorem 5 to also work for odd r (up to a small loss in parameters), and we also show that our construction remains explicit for most *super-constant* r, s: we refer the reader to Section 4 for more detail.

Finally, we can combine our explicit designs with the leakage-resilient extractors from [CGG+20] to obtain our improved adversarial sources (Theorem 4), which immediately yields our improved extractors for small-space sources (Theorem 3). It is known that the technique of reducing small-space sources to adversarial sources has a barrier at min-entropy $\sqrt{n}$ (see Remark 6.7). Thus, the result in Theorem 3 has almost the best parameters one can hope to achieve using this technique.

2 Overview of Techniques

We use this section to sketch the explicit constructions of our small-space extractors. We start with our low-error small space extractors (Theorem 3) and the ingredients that go into it (Theorems 4 and 5). Then, we sketch the construction of our small-space extractor for polylogarithmic entropy (Theorem 1) and its key ingredient (Theorem 2).

2.1 Small-space extractors with exponentially small error

To construct our low-error small-space extractors, the first step is to use a standard reduction [KZ06] (which we slightly optimize) from small-space sources to adversarial sources. This reduction starts with the observation of [KRVZ06] that if we chop up the small space source $\mathbf{X}$ into t consecutive (equal-sized) chunks, and *condition on any fixing* of the vertices reached at the end of each chunk *in the random walk that generates* $\mathbf{X}$, then these t chunks become t independent sources. Furthermore, if $\mathbf{X}$ originally had k bits of entropy, then it follows from the entropy chain rule that $\mathbf{X}$ will still have roughly $k - st$ bits of entropy. A Markov argument then shows that at least a few of the t sources will have relatively high entropy. In other words, $\mathbf{X}$ now looks like an adversarial source, and we may now focus on constructing (low-error) extractors for adversarial sources.

[1] Recall that an *independent set* in a hypergraph is a subset of vertices that contain no hyperedge, and the *independence number* of a hypergraph is the size of its largest independent set.

Improved low-error extractors for adversarial sources To construct our low-error extractors for adversarial sources, we develop a new framework that combines a certain type of *leakage-resilient extractor* (LRE) with the (n, r, s)-designs discussed earlier. An LRE for r sources offers the guarantee that its output looks uniform *even conditioned on* the output of many *leakage* functions, each called on up to $r-2$ of the same inputs fed to the original LRE. Furthermore, recall that an (n, r, s)-design is an r-uniform hypergraph over n vertices with pairwise hyperedge intersections of size $< s$.

Now, given an (N, K, n, k)-adversarial source $\mathbf{X}$, we extract from it as follows, using an LRE and an $(N, r, r-1)$-design G with independence number $\alpha(G) < K$. First, we identify the vertices of our design with the N independent sources in $\mathbf{X}$. Then, for each hyperedge in our design, we call a leakage-resilient extractor on the r sources it contains, and finish by taking the bitwise XOR over the outputs of the LRE calls.

This construction successfully outputs uniform bits for the following reasons. Because $\alpha(G) < K$, we are guaranteed that *some* LRE call is given *only* good sources. By the *extractor* property of the LRE, this call will output uniform bits. Meanwhile, the *bounded intersection* property of the $(N, r, r-1)$-design, paired with the *leakage-resilience* property of the LRE, guarantees that these uniform bits still look uniform *even after taking their bitwise XOR with the outputs of all other LRE calls*. Using these ideas, we actually provide a slightly more general framework to combine (N, r, s)-designs with LREs of various strength. Our framework leverages the "*activation vs. fragile correlation*" paradigm introduced in [CGGL20], yet it is able to do so in a much more simple, general, and effective way, by combining two very general pseudorandom objects: LREs and designs.

To make our framework explicit, we will need explicit LREs and explicit designs with small independence number. Our explicit LREs will come from the work of Chattopadhyay et al. [CGG+20], where they gave the first explicit LREs that work for entropy $k = o(n)$, and in fact their LREs work for entropy $k \geq \text{polylog}(n)$. Thus all that remains is to provide an explicit construction of designs with small independence number.[2] We provide such a construction in this paper, and sketch it below.

Explicit designs with small independence number 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 then restrict it to the set $Q_r \subseteq Q$ of elements in Q that have Hamming weight r. Our design $G = (V, E)$ is constructed by identifying V with $[n]$, and by 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 guarantees that G is an (n, r, s)-design.

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 $S \subseteq \mathbb{F}_2^n$ that contains no vector in Q of weight r; in other words, since Q is a *linear* code, this means that the *subspace* $T^* := S \cap Q$ has no vector of Hamming weight r. If our linear code Q had very high dimension, then even if the subcube S was relatively small, we would have found a relatively large subspace T^* 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 it follows immediately from Sidorenko's recent bounds [Sid18, Sid20] on the size of sets in $\mathbb{F}_2^n$

[2] Explicit $(N, 3, 2)$-designs with independence number $\alpha < O(N^{0.923})$ were constructed in [CGGL20]. However, we need more general $(N, r, r-1)$-designs to push the independence number low enough to obtain our desired adversarial extractors, and (to the best of our knowledge) no such explicit designs were known prior to our work.

containing no r elements that sum to zero. Thus if Q has large enough dimension, S 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].

2.2 Small-space extractors for polylogarithmic entropy

Unfortunately, it is impossible to extract from small-space sources with entropy $k < \sqrt{n}$ using a reduction of the previous type (i.e., to adversarial sources), since setting $t \geq \sqrt{n}$ will leave $k - st \leq k - 1 \cdot \sqrt{n} < 0$ bits of entropy after the above fixing, while setting $t < \sqrt{n}$ will produce a chunk of size $n/t > \sqrt{n} > k$, which could hold all of the entropy and thus make extraction impossible. 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.

A reduction from small-space sources to affine sources 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 a nice observation of [CGGL20], which says that *any* source with entropy at least 1 is a convex combination of *affine sources* with entropy 1. 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 $k \geq \text{polylog}(n)$, 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 $k \geq \sqrt{n}$ entropy barrier arising from existing reductions to source types of fixed lengths (like *total-entropy sources* [KRVZ06] and *adversarial sources* [CGGL20]). Second, our reduction to a sequence of t' *independent sources with entropy* 1 (which we argue is an affine source with entropy t' using the observation of [CGGL20]) results in a *negligible* amount of lost entropy from each recursive step, whereas similar recursive reductions to *a constant number of sources with relatively high entropy* [CL16] are forced to lose much more entropy in each such step. As a result, we are able to bypass the $k \geq 2^{\sqrt{\log n}}$ entropy barrier of [CL16].

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 chain rule for min-entropy (Lemma 3.3) *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 [CL16].

Organization In Section 3 we provide several preliminaries. In the remainder of our paper, we follow a *bottom-up* strategy for presenting our main results. In Section 4, we provide an explicit construction of designs with small independence number, proving Theorem 5. In Section 5, we show how to combine these designs with leakage-resilient extractors to create a new, simple framework for extraction from adversarial sources. By instantiating our framework with our explicit designs and the explicit leakage-resilient extractors of [CGG+20], we obtain our improved extractors for adversarial sources, Theorem 4. In Section 6, we provide the standard reduction from small-space sources to adversarial sources for completeness, and we apply our adversarial extractors (Theorem 4) to obtain our small-space extractors with exponentially small error, Theorem 3. In Section 7, we provide our *new reduction* from small-space sources to *affine sources* (Theorem 2) and apply the affine extractor of Li [Li16] to obtain our small-space extractors for polylogarithmic entropy, Theorem 1. We conclude with some remarks and present some open problems in Section 8.

3 Preliminaries

General notation Given two strings $x, y \in \{0,1\}^m$, we let $x \oplus y$ denote their bitwise XOR. For a number $n \in \mathbb{N}$, $[n]$ denotes the interval $[1, n] \subseteq \mathbb{N}$. We let $\circ$ denote string concatenation, and for a collection $\{x_i : i \in I\}$ indexed by some finite set I, we let $(x_i)_{i \in I}$ denote the concatenation of all strings $x_i, i \in I$. If I is already equipped with some total order, this is used to determine the concatenation order; otherwise, I is arbitrarily identified with $[|I|]$ to induce a total ordering. Given a domain $\mathcal{D}$, and some string $x \in \mathcal{D}^N$, we let $x_i \in \mathcal{D}$ denote the value at the i^{th} coordinate of x. Given a subset $S \subseteq [N]$, we let $x_S := (x_i)_{i \in S}$. Even if $\mathcal{D} = \mathcal{R}^n$ for some other domain $\mathcal{R}$ and number $n \in \mathbb{N}$, the definition of $x_S \in \mathcal{D}^{|S|}$ does not change.

Basic coding theory and extractor definitions We let $\mathbb{F}_2$ denote the finite field of size two, and we let $\mathbb{F}_2^n$ denote a vector space over this field. The *Hamming weight* of a vector $x \in \mathbb{F}_2^n$ is defined as $\Delta(x) := \#\{i \in [n] : x_i = 1\}$, and the *Hamming distance* between two vectors $x, y \in \mathbb{F}_2^n$ is defined as $\Delta(x, y) := \Delta(x - y)$, where the subtraction is over $\mathbb{F}_2$. The *standard basis vectors* in $\mathbb{F}_2^n$ is the collection $\mathcal{E}^* := \{e_i\}_{i \in [n]}$, where $e_i \in \mathbb{F}_2^n$ holds a 1 at coordinate i and 0 everywhere else, and a *subcube* is a subspace spanned by some subset of $\mathcal{E}^*$. An (n, k, d)-code is a subset $Q \subseteq \mathbb{F}_2^n$ of size 2^k with the guarantee that any two distinct points $x, y \in Q$ have Hamming distance $\Delta(x, y) \geq d$. A linear $[n, k, d]$-code is simply an (n, k, d) code that is a subspace. Finally, we say that a source $\mathbf{X}$ over $\{0,1\}^n$ is an (n, k) source if it has min-entropy at least k, and we say that an extractor Ext an N-source extractor for entropy k if it is an extractor for a family of sources $\mathcal{X}$, where each $\mathbf{X} \in \mathcal{X}$ consists of N independent (n, k) sources.

Discrete probability In general, for a random variable $\mathbf{X} : \Omega \to V$, we are only concerned with the distribution over V induced by $\mathbf{X}$. We will therefore typically not define the outcome space Ω, and can assume it has any form we like (so long as the distribution induced by $\mathbf{X}$ does not change). Given random variables $\mathbf{X}, \mathbf{Y}$ and any $y \in \text{support}(\mathbf{Y})$, we let $(\mathbf{X} \mid \mathbf{Y} = y)$ denote a random variable that takes value x with probability $\Pr[\mathbf{X} = x \mid \mathbf{Y} = y]$. Given a random variable $\mathbf{X}$ and a family of random variables $\mathcal{Y}$, we say that $\mathbf{X}$ is a *convex combination* of random variables from $\mathcal{Y}$ if there exists a random variable $\mathbf{Z}$ such that for each $z \in \text{support}(\mathbf{Z})$, it holds that $(\mathbf{X} \mid \mathbf{Z} = z) \in \mathcal{Y}$.

We define the *statistical distance* between two random variables $\mathbf{X}, \mathbf{Y}$ over V as

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

and we say that $\mathbf{X}, \mathbf{Y}$ are ϵ-*close* if $|\mathbf{X} - \mathbf{Y}| \leq \epsilon$. Given these definitions, the following two standard facts are easy to show, and are extremely useful.

Fact 3.1. *For any random variable $\mathbf{X} \sim \{0,1\}^m$ and any constant $c \in \{0,1\}^m$, it holds that*

$$|\mathbf{X} - \mathbf{U}_m| = |(\mathbf{X} \oplus c) - \mathbf{U}_m|.$$

Fact 3.2. *For any random variables $\mathbf{X}, \mathbf{Y}$, where $\mathbf{X} \sim \{0,1\}^m$, it holds that*

$$|\mathbf{X} - \mathbf{U}_m| \leq \mathbb{E}_{y \sim \mathbf{Y}}[|(\mathbf{X} \mid \mathbf{Y} = y) - \mathbf{U}_m|].$$

Finally, we will need the following standard lemma about conditional min-entropy.

Lemma 3.3 ([MW97])**.** *Let $\mathbf{X}, \mathbf{Y}$ be random variables such that $\mathbf{Y}$ can take at most ℓ values. Then for any $\epsilon > 0$, it holds that*

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

4 Explicit extremal designs via slicing codes and zero-sum sets

In this section, we will construct our explicit designs and thereby prove Theorem 5. Before we state the formal theorem and proof, we begin with some background and discussion on (n, r, s)-designs.

4.1 Background and discussion

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. More formally, we say that an r-uniform hypergraph $G = (V, E)$ over n vertices is an (n, r, s)-*design*, or (n, r, s)-*partial Steiner system*, if $|e_1 \cap e_2| < s$ for all distinct $e_1, e_2 \in E$. Beyond the fact that they are pseudorandom objects themselves, it turns out that (n, r, s)-designs enjoy several interesting applications in pseudorandomness.

A notable application of designs is in the seminal work of Nisan and Wigderson [NW94], where they are used to construct *pseudorandom generators* (PRGs). In this application, the authors require (and provide) explicit designs that are *extremal* in the sense that they have a large number of hyperedges. More recently, explicit designs of a different extremal flavor have been used in the construction of extractors: in [CGGL20], Chattopadhyay, Goodman, Goyal, and Li show how to construct extractors for adversarial sources using explicit partial Steiner triple systems ($(n, 3, 2)$-designs) with *small independence number*.

Given these applications, it is natural to ask about the smallest possible independence number of more general (n, r, s)-designs. Rödl and Šinajová answered this question in 1994, proving the following:

Theorem 4.1 ([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.

In fact, they also showed this result is tight up to the term $C_{r,s}$ that depends only on r, s.

In order to prove Theorem 4.1, Rödl and Šinajová apply the Lovász Local Lemma to show that a *random* r-uniform hypergraph is such a design. Thus, while their result proves the existence of such designs, it does not provide an explicit way to construct them - and, unfortunately, an explicit construction is needed if one hopes to apply this result to construct other explicit objects (like extractors). Furthermore, all subsequent work appears to focus on improving the term $C_{r,s}$ [EV13, Eus13] or extending their result to more general types of designs [GPR95, KMV14, TL18], while still relying on probabilistic constructions.

In this section, we will provide explicit constructions of these extremal designs. Our designs give the first derandomization of Theorem 4.1, and differ from the optimal bound by just a square.

4.2 Main theorem about explicit designs

We are now ready to state our main theorem that describes our construction of explicit designs with small independence number.

Theorem 4.2 (Theorem 5, formal version)**.** *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 $\text{poly}\left(\binom{n}{r}\right)$ and outputs an (n, r, s)-design G with independence number*

$$\alpha(G) \leq C_{r,s} \cdot n^{\frac{2(r-s)}{r}}, \tag{1}$$

where $C_{r,s} = C \cdot r^4$ for some universal constant $C \geq 1$.

Remark 4.3. *It is easy to extend Theorem 4.2 to construct (n, r, s)-designs $(G_n)_{n\in\mathbb{N}}$ with odd r, at the expense of a small loss in parameters: simply construct an $(n, r+1, s)$-design G'_n using Theorem 4.2, and remove an arbitrary vertex from each hyperedge to create G_n. G_n will be an (n, r, s)-design, and will have independence number $\alpha(G_n) \leq C_{r+1,s} \cdot n^{\frac{2(r+1-s)}{r+1}}$.*

For all constants $r \geq s \in \mathbb{N}$ with r even, Theorem 4.2 constructs an explicit family of (n, r, s)-designs $(G_n)_{n\in\mathbb{N}}$ with small independence number. Like the non-explicit designs of Theorem 4.1 from [RŠ94], our derandomization focuses on the case where r, s are constant. However, it turns out that even for most *super-constant* r, s, our algorithm is still efficient. In particular, before proving Theorem 4.2, we make (and quickly prove) the following remark.

Remark 4.4. *Let $\mathcal{A}$ be the algorithm from Theorem 4.2, and let $m = m(n, r, s)$ be the number of hyperedges in the design produced by $\mathcal{A}$ on input (n, r, s). Then for any functions $r = r(n), s = s(n)$, Algorithm $\mathcal{A}$ is guaranteed to run in time $\text{poly}(n, m)$ over the collection $\mathcal{I} = \{(n, r(n), s(n))\}_{n\in\mathbb{N}}$ as long as at least one of the following holds:*

- *The functions r, s are constant: $r(n) = O(1)$ and $s(n) = O(1)$; or*
- *There is a constant $\epsilon > 0$ such that inequality (1) is bounded above by $O(n^{1-\epsilon}), \forall (n, r, s) \in \mathcal{I}$.*

The first bullet in Remark 4.4 reiterates the fact that the algorithm in Theorem 4.2 is efficient when r, s are constant. The second bullet gives a more general remark on the performance of Algorithm $\mathcal{A}$ on super-constant r, s: it says that as long as Theorem 4.2 gave a "non-trivial" 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 (and any constant r, s that achieve $\alpha(G) < n$ in Theorem 4.2 in fact achieve the second bullet).

To prove that Algorithm $\mathcal{A}$ is efficient given the condition in the second bullet, 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 $T(n, \beta, r) \geq \binom{n}{r}/\binom{\beta}{r}$. Thus, the second bullet implies the number of edges, $m = m(n, r, s)$, in the design is at least

$$T(n, Cn^{1-\epsilon} + 1, r) \geq T(n, n^{1-\epsilon/2}, r) \geq \binom{n}{r}/\binom{n^{1-\epsilon/2}}{r} \geq \binom{n}{r}/\binom{n}{r}^{1-\epsilon/4} \geq \binom{n}{r}^{\epsilon/4},$$

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 a simple application of Stirling's formula. Thus, Algorithm $\mathcal{A}$ runs in time $\binom{n}{r} = \text{poly}(n, m)$. 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 that the runtime of $\mathcal{A}$ is optimal up to constant powers. This completes our discussion on Remark 4.4.

4.3 Proof of Theorem 4.2

We now turn to proving Theorem 4.2. We start with the simple observation that hypergraphs over n vertices can be identified with subsets of $\mathbb{F}_2^n$. In particular, any subset $T \subseteq \mathbb{F}_2^n$ induces a hypergraph $G_T = (V, E)$ in the following way: identify V with $[n]$, and for each $x \in T$ add a hyperedge $e \subseteq [n]$ to E that contains exactly the coordinates that take the value 1 in x. 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.

To obtain our designs, we will need to explicitly 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. We can make sure this happens via the following two simple facts, which describe how these hypergraph properties can be identified with properties of subsets in $\mathbb{F}_2^n$.

Fact 4.5. *For any subset $T \subseteq \mathbb{F}_2^n$, the hypergraph 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 the first one guarantees that G_T will be r-uniform, and the second one guarantees that any two edges in G_T intersect at $< s$ points. They are both necessary because if the first does not hold, G_T will not be r-uniform, and if the first holds but the second does not, then two edges will end up sharing $\geq s$ points. □

Fact 4.6. *For any subset $T \subseteq \mathbb{F}_2^n$, the hypergraph G_T has independence number $\alpha(G_T) < \ell$ if and only if every subcube $A \subseteq \mathbb{F}_2^n$ of dimension at least ℓ has at least one point 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 $A := \textit{span}(\{e_i\}_{i \in S})$ 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. □

By Fact 4.5 and Fact 4.6, we see that the task of constructing an (n, r, s)-design G with small independence number is *equivalent* to the task of constructing a subset $T \subseteq \mathbb{F}_2^n$ with the following *three properties*:

1. T lies in the Hamming slice $\Delta_r := \{x \in \mathbb{F}_2^n : \Delta(x) = r\}$,
2. Points in T have pairwise Hamming distance $> 2(r - s)$, and
3. Any subcube of *relatively small* dimension intersects T.

In order to construct a set $T \subseteq \mathbb{F}_2^n$ with these three properties, we use connections to *coding theory* and *zero-sum problems*. In particular, recall that an (n, k, d)-code is a subset $Q \subseteq \mathbb{F}_2^n$ of size 2^k with the guarantee that any two distinct points $x, y \in Q$ 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 intersect it with the Hamming slice Δ_r, we obtain a set $T = Q \cap \Delta_r$ that enjoys properties (1) and (2). In order to endow it with property (3), we will need to start with some code Q such that for any relatively large subcube S, the set $S \cap T = S \cap (Q \cap \Delta_r) = (S \cap Q) \cap \Delta_r$ is non-empty.

The trick here is to start with a *linear* code Q. A *linear* $[n, k, d]$-code $Q \subseteq \mathbb{F}_2^n$ is simply an (n, k, d) code that is also a subspace. The condition $(S \cap Q) \cap \Delta_r \neq \emptyset$ required for property (3) now becomes more concrete: since Q is a subspace, $S \cap Q$ is also a subspace, and thus we can make sure it contains some vector of Hamming weight r as long as we can show that *every* large subspace contains such a vector. In particular, defining $\Lambda_r(n)$ to be the dimension of the largest subspace $R \subseteq \mathbb{F}_2^n$ containing no vector of Hamming weight r, we prove the following lemma.

Lemma 4.7. *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 $\alpha = \alpha(G_{Q \cap \Delta_r})$ that obeys the following inequality:*

$$\alpha - \Lambda_r(\alpha) \leq n - k$$

Proof. It follows immediately from Fact 4.5 that $G_{Q \cap \Delta_r}$ is an (n, r, s)-design. By Fact 4.6, 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 order to construct an (n, r, s)-design from Lemma 4.7 with the smallest possible independence number α, we will want an explicit $[n, k, d > 2(r-s)]$-linear code with the largest possible dimension k, along with a strong upper bound on $\Lambda_r(n)$. We start with the latter.

Getting a good upper bound on $\Lambda_r(n)$ is closely related to the theory of *zero-sum problems*. In this field, one parameter of great interest is the (generalized) *Erdős-Ginzburg-Ziv constant*(s) of a finite abelian group. Given $n \geq r \in \mathbb{N}$ where r is even, this parameter is defined for $\mathbb{F}_2^n$ as the smallest integer $s_r(n)$ such that any *sequence* of $s_r(n)$ values in $\mathbb{F}_2^n$ contains a subsequence of length r that sums to zero. For our application, it will be more convenient to use an almost identical parameter $\beta_r(n)$, defined as the size of the largest *subset* of $\mathbb{F}_2^n$ containing no r elements that sum to zero. Using slightly different terminology, the relationship between $\beta_r(n)$ and $\Lambda_r(n)$ was shown in [Sid20]. We include it here, in our language, for completeness.

Lemma 4.8 ([Sid20])**.** *For every $n \geq r \in \mathbb{N}$ where r is even,*

$$\beta_r(n - \Lambda_r(n)) \geq n.$$

Proof. Let $R \subseteq \mathbb{F}_2^n$ be a subspace of dimension $k := \Lambda_r(n)$ that contains no vector of Hamming weight r, and define $d := n - k$. Let $v_1, \ldots, v_d$ be a basis for the orthogonal complement of R, and define the matrix $M \in \mathbb{F}_2^{d \times n}$ so that its i^{th} row is v_i. Notice that R contains exactly the solutions to $Mx = 0$, and thus R has a vector of Hamming weight r if and only if there are r columns in M that sum to zero. By definition of R, we know R has no such vector, and thus $n \leq \beta_r(d) = \beta_r(n - \Lambda_r(n))$. □

To get a good upper bound on $\Lambda_r(n)$, we need a good upper bound on $\beta_r(n)$. In 2018, Sidorenko provided a very strong bound of this type:

Theorem 4.9 ([Sid18], Theorem 4.4)**.** *There is a universal constant $C > 0$ such that for every $n, r \in \mathbb{N}$ where r is even,*

$$\beta_r(n) \leq C \cdot r^3 \cdot 2^{2n/r}.$$

By plugging this bound into Lemma 4.8, we get the following corollary.

Corollary 4.10 ([Sid20])**.** *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*

$$\Lambda_r(n) \leq n - (r \log n - 3r \log r - r \log C)/2.$$

We are finally ready to prove our main design lemma, which reduces the problem of constructing (n, r, s)-designs with small independence number to constructing high-dimensional linear codes.

Lemma 4.11 (Main design lemma)**.** *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 4.10 into Lemma 4.7. □

To complete the proof of Theorem 4.2, we now just need to explicitly construct a linear code with very high dimension. In 1959-1960, Bose, Ray-Chaudhuri [BRC60], and Hocquenghem [Hoc59] explicitly constructed codes of exactly this type (see [GB10] for a great exposition of these codes, which are known as *BCH codes*). In particular, they proved the following theorem.

Theorem 4.12 ([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 $\text{poly}(n)$ time.*

By instantiating Lemma 4.11 with Theorem 4.12, we can finally prove Theorem 4.2.

Proof of Theorem 4.2. We start by assuming that $n = 2^m - 1$ for some $m \in \mathbb{N}$. Then, we let $t = r - s$, and use Theorem 4.12 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 4.11, 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 $\mathrm{poly}(\binom{n}{r})$ time if $Q \cap \Delta_r$ can be constructed in $\mathrm{poly}(\binom{n}{r})$ 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 4.12.

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. □

5 Extractors for adversarial sources via designs and LREs

Perhaps the most popular model of seedless extraction is to assume that each source $\mathbf{X}$ actually consists of several *independent sources* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N)$, each guaranteed to have some min-entropy. A long line of work has focused on constructing extractors for this setting [CG88, BIW06, Li15, Coh16, CZ19, Li19], and has culminated in extractors with a near-optimal entropy requirement [Li19]. Recently, the idea of generalizing this model to allow for *bad sources* with *no entropy guarantee* and/or *limited dependence* has received considerable attention [AOR+20, CGGL20, BGM19]. Motivated by applications in generating a (cryptographic) common random string in the presence of adversaries, and in harvesting randomness from unreliable sources, Chattopadhyay, Goodman, Goyal, and Li [CGGL20] introduced the class of *adversarial sources*:

Definition 5.1 (Adversarial sources)**.** *A source $\mathbf{X}$ over $(\{0,1\}^n)^N$ is 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 $\{0,1\}^n$, and* at least *K* of them are good*: i.e., there is some set $S \subseteq [N]$ of size K such that $H_\infty(\mathbf{X}_i) \geq k$, for all $i \in S$.*

In fact, the authors in [CGGL20] provide a more general definition that also allows for some limited dependence between the sources, but Definition 5.1 is already general enough to capture many of their motivating applications and generalize several well-studied settings: (N, N, n, k)-adversarial sources capture the *independent source* model [CG88], $(N, K, 1, 1)$-adversarial sources

capture *oblivious bit-fixing sources* [CGH+85], and (N, K, n, n)-adversarial sources capture so-called *symbol-fixing sources* [KZ06].

In this section, we will show how to combine our designs from Section 4 with a specific kind *leakage-resilient extractor* (LRE) known as *extractors for cylinder intersection* that was introduced in [KMS19] (see Definition 5.3), in order to obtain improved extractors for adversarial sources. The following is our main result of the section:

Theorem 5.2 (Theorem 4, restated)**.** *There is a universal constant $C > 0$ such that for any fixed $\delta > 0$ and all sufficiently large $N, K, n, k \in \mathbb{N}$ satisfying $k \geq \log^C n$ and $K \geq N^\delta$, there exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^n)^N \to \{0,1\}^m$ for (N, K, n, k)-adversarial sources, with output length $m = k^{\Omega(1)}$ and error $\epsilon = 2^{-k^{\Omega(1)}}$.*

Previously, the best explicit extractor for this setting was constructed by Chattopadhyay et al. [CGGL20], and required $K \geq N^{0.5+o(1)}$ good sources. On the other hand, it is easy to give a *non-explicit* extractor that requires just $K \geq 2$ good sources.[3] Thus, while our explicit constructions greatly improve the state-of-art (and most notably break the "$\sqrt{N}$ barrier"), there is still a lot of room for improvement. Further improvement, however, will require significantly new techniques.

In order to prove Theorem 5.2, we start by reviewing the *activation vs. fragile correlation* paradigm from [CGGL20] for extracting from adversarial sources in Section 5.1. We prove Theorem 5.2 in Section 5.2, where we will describe how to extend the activation vs. fragile correlation technique into a general framework for extracting from adversarial sources. We use this new framework by combining the recent explicit LREs from [CGG+20] with our new explicit designs from Section 4 to obtain our adversarial source extrators.

5.1 The *activation vs. fragile correlation* paradigm of [CGGL20]

Our construction leverages the "*activation vs. fragile correlation*" paradigm introduced in [CGGL20] for extracting from adversarial sources. This paradigm was first introduced in an attempt to construct a low-error extractor for (N, K, n, k)-adversarial sources, given just $k \geq \operatorname{polylog} n$ entropy and as a few good sources, K, as possible. Since there exists a three-source extractor Ext_0 for $k_0 \geq \operatorname{polylog} n$ entropy and exponentially small error [Li15], a natural idea is to somehow employ this object as a subroutine. Using this idea, [CGGL20] proposed an extractor for adversarial sources that works as follows. Given as input an adversarial source $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$, the extractor carefully selecting triples of sources, calls Ext_0 over each triple, and XORs the results. [CGGL20] argued that this procedure outputs uniform bits as long as the following two properties hold:

1. **Activation**: some Ext_0 call is *activated*, i.e., only given good sources as input.
2. **Fragile correlation**: fixing the (XOR of the) output of all other Ext_0 calls does not affect the output of the activated Ext_0 call (with high probability).

It is not hard to see why these conditions suffice: *activation* guarantees that some Ext_0 call outputs uniform bits, while *fragile correlation* guarantees that these uniform bits will be propagated through to the overall output of the extractor (by Fact 3.1 and Fact 3.2). Thus, the main

[3] This extractor calls an optimal two-source extractor over every pair of sources in the adversarial source, and takes the XOR of the results [CL16]. To see why this works, we refer the reader to a similar proof sketch for a slightly more involved construction, provided in the following paragraphs.

challenge considered in [CGGL20] is determining *how to select triples* such that activation and fragile correlation are guaranteed.

The key idea in [CGGL20] is to select triples using the hyperedges of a 3-uniform hypergraph, $G = (V, E)$. Then, we know that activation is guaranteed as long as the good sources *cover* some hyperedge $e \in E$, which is guaranteed to happen whenever $K > \alpha(G)$. In order to ensure fragile correlation, [CGGL20] observed that it suffices to require that $G = (V, E)$ is a *partial Steiner triple system*, also known as an $(N, 3, 2)$-design. Such a hypergraph guarantees that each Ext_0 call shares at most one source with the activated Ext_0 call. Thus, if we start by fixing all sources that are *not* inputs to the activated Ext_0 call, it is then easy to fix the outputs of all other Ext_0 calls without introducing correlation between the inputs to the activated call. Furthermore, by Lemma 3.3, we can show that this process barely decreases the entropy of the inputs to the activated call, and thus its output remains uniform.

This shows that the construction above provides a low-error extractor for (N, K, n, k)-adversarial sources, where $k \geq \operatorname{polylog} n$ and $K > \alpha(G)$. Thus, the goal becomes to explicitly construct an $(N, 3, 2)$-design $G = (V, E)$ with small independence number. Using cap set bounds, Chattopadhyay et al. [CGGL20] construct such an object with $\alpha(G) < N^{0.923}$, and thus gave an explicit extractor when there are $K \geq N^{0.923}$ good sources. In order to improve this requirement on K, it is natural to try to construct an $(N, 3, 2)$-design with smaller independence number. However, this seems difficult, and furthermore the tightness of Theorem 4.1 implies that this technique cannot possibly give an extractor that requires fewer than $K \geq N^{0.5+o(1)}$ good sources.

Chattopadhyay et al. [CGGL20] take a different approach. By using objects known as *strong two-source condensers* [BACDTS19] and *non-malleable extractors* [CGL20], the authors are able to create more robust versions of three-source extractors. These robust extractors have stronger conditioning properties, and allow the authors to use different hypergraphs (beyond $(N, 3, 2)$-designs) in their construction. As a result, they are able to reduce the requirement on good sources from $K \geq N^{0.923}$ to $K \geq N^{0.5+o(1)}$. Unfortunately, however, the conditioning properties of their robust subroutine extractors are extremely specific, and as a result they can only be combined with very specialized types of hypergraphs. These hypergraphs offer no clean generalization of $(N, 3, 2)$-designs, and furthermore they appear to be too specialized to offer any further improvement on K (and, in particular, break the "$\sqrt{N}$ barrier").

5.2 A new framework using leakage-resilient extractors and extremal designs

If one hopes to significantly improve K, it appears that one would need a multi-source extractor with *even stronger* conditioning properties to use as a subroutine. Recently, exactly such an object was constructed in [CGG+20], and is known as a *leakage-resilient extractor* (LRE). LREs are very general objects with extremely strong conditioning properties. The exact variant that will be useful here is actually a specialization known as *extractors for cylinder intersections*, first introduced in [KMS19]. Informally, we define an (r, s)-*leakage-resilient extractor* to be an r-source extractor LRE that outputs bits that look uniform, *even conditioned on* the output of several functions that each act on *fewer than* s of the inputs to LRE. Formally, it is defined as follows.

Definition 5.3 ([KMS19, CGG+20])**.** *A function* $\mathsf{LRE} : (\{0,1\}^n)^r \to \{0,1\}^m$ *is an* (r, s)-leakage-resilient extractor *for entropy* k *and error* ϵ *if the following holds. Let* $\mathbf{X} := (\mathbf{X}_1, \ldots, \mathbf{X}_r)$ *be any* r *independent* (n, k) *sources, let* $\mathcal{T} := \binom{[N]}{s-1}$*, and let* $\mathcal{L} := \{\mathsf{Leak}_T : (\{0,1\}^n)^{s-1} \to \{0,1\}^m\}_{T \in \mathcal{T}}$ *be*

any collection of functions. Then:

$$|\mathsf{LRE}(\mathbf{X}) \circ (\mathsf{Leak}_S(\mathbf{X}_S))_{S\in\mathcal{S}} - \mathbf{U}_m \circ (\mathsf{Leak}_S(\mathbf{X}_S))_{S\in\mathcal{S}}| \le \epsilon.$$

Given such a robust extractor, it is now easy to generalize the original extractor of [CGGL20] in a clean, natural way: instead of calling a three-source extractor over the hyperedges of an $(N, 3, 2)$-design and XORing the results, we call an (r, s)-leakage-resilient extractor over the hyperedges of an (N, r, s)-design and XOR the results. Once again, we can ensure *activation* as long as the number of good sources, K, exceeds the independence number of the design. On the other hand, instead of using Lemma 3.3 to ensure fragile correlation, we simply use the leakage-resilience of our leakage-resilient extractor: to see why this works, simply observe that an (N, r, s)-design guarantees that the intersection of two hyperedges has size $< s$, while a leakage-resilient extractor guarantees to output uniform bits *even conditioned on* several leaks that each act on $< s$ of its inputs.

Formally, we prove the following lemma, which provides a framework for combining leakage-resilient extractors with general designs in order to extract from adversarial sources.

Lemma 5.4. *Let $G = ([N], E)$ be an (N, r, s)-design with independence number α, and let $\mathsf{Ext}_0 : (\{0,1\}^n)^r \to \{0,1\}^m$ be an (r, s)-leakage resilient extractor for entropy k_0 with error ϵ_0. Then for any $K > \alpha$ and $k \ge k_0$, the function $\mathsf{Ext}_G : (\{0,1\}^n)^N \to \{0,1\}^m$ defined as*

$$\mathsf{Ext}_G(X) := \bigoplus_{e\in E(G)} \mathsf{Ext}_0(X_e)$$

is an extractor for (N, K, n, k) adversarial sources with error $\epsilon = \epsilon_0$.

Proof. Let $\mathbf{X}$ be an (N, K, n, k) adversarial source. We must show that $|\mathsf{Ext}_G(\mathbf{X}) - \mathbf{U}_m| \le \epsilon$. Because $K > \alpha$, there is some $e^* \in E$ containing only good sources, i.e., $\mathbf{X}_i$ has entropy at least k for each $i \in e^*$. Without loss of generality, we assume $e^* = [r]$. We now fix all other sources $\mathbf{Z}_1 = (\mathbf{X}_j)_{j\notin e^*}$, using Fact 3.2:

$$|\mathsf{Ext}_G(\mathbf{X}) - \mathbf{U}_m| \le \mathbb{E}_{z_1\sim\mathbf{Z}_1}[|(\mathsf{Ext}_G(\mathbf{X}) \mid \mathbf{Z}_1 = z_1) - \mathbf{U}_m|].$$

Consider any $z_1 = (x_j)_{j\notin e^*}$. For each $e \in E(G)$, we define the restriction $\mathsf{Ext}_0^e : (\{0,1\}^n)^{|e\cap e^*|} \to \{0,1\}^m$ as $\mathsf{Ext}_0^e(Y_1, \ldots, Y_{|e\cap e^*|}) = \mathsf{Ext}_0(Y_1, \ldots, Y_{|e\cap e^*|}, (x_j)_{j\in e\setminus e^*})$, so that we may write

$$(\mathsf{Ext}_G(\mathbf{X}) \mid \mathbf{Z} = z_1) = \bigoplus_{e\in E(G)} \mathsf{Ext}_0^e(\mathbf{X}_{e\cap e^*}) = \mathsf{Ext}_0(\mathbf{X}_{e^*}) \oplus \bigoplus_{e\in E(G)\setminus\{e^*\}} \mathsf{Ext}_0^e(\mathbf{X}_{e\cap e^*}).$$

Because G is an (N, r, s)-design, any two edges share at most $s - 1$ vertices. Thus, we may partition $E(G) \setminus \{e^*\}$ into $\binom{r}{s-1}$ sets, depending on the intersection behavior of each edge with e^*. In particular, for each $S \in \binom{e^*}{s-1}$, we define:

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

If any $e \in E$ ends up in more than one $\mathcal{W}_S$, we simply remove it from all but one of these sets. Now, for each $S \in \binom{e^*}{s-1}$, we define $\mathsf{Leak}_S : (\{0,1\}^n)^{s-1} \to \{0,1\}^m$ such that for any $X \in (\{0,1\}^n)^N$, $\mathsf{Leak}_S(X_S) = \bigoplus_{e\in\mathcal{W}_S} \mathsf{Ext}_0^e(X_{e\cap e^*})$, which is a valid definition because $e \cap e^*$ is always in S, by definition of $\mathcal{W}_S$. We may now write

$$(\mathsf{Ext}_G(\mathbf{X}) \mid \mathbf{Z}_1 = z_1) = \mathsf{Ext}_0(\mathbf{X}_{e^*}) \oplus \bigoplus_{S\in\binom{e^*}{s-1}} \mathsf{Leak}_S(\mathbf{X}_S). \tag{2}$$

To bound the distance of this random variable from uniform, we now define the second random variable we will fix, $\mathbf{Z}_2 := (\mathsf{Leak}_S(\mathbf{X}_S))_{S \in \binom{e^*}{s-1}}$. Fixing this random variable, we have:

$$\begin{aligned}|(\mathsf{Ext}_G(\mathbf{X}) \mid \mathbf{Z}_1 = z_1) - \mathbf{U}_m| &\leq \mathbb{E}_{z_2 \sim \mathbf{Z}_2}[|(\mathsf{Ext}_G(\mathbf{X}) \mid \mathbf{Z}_1 = z_1, \mathbf{Z}_2 = z_2) - \mathbf{U}_m|] \\ &= \mathbb{E}_{z_2 \sim \mathbf{Z}_2}[|(\mathsf{Ext}_0(\mathbf{X}_{e^*}) \mid \mathbf{Z}_2 = z_2) - \mathbf{U}_m|] \\ &= |\mathsf{Ext}_0(\mathbf{X}_{e^*}) \circ \mathbf{Z}_2 - \mathbf{U}_m \circ \mathbf{Z}_2|,\end{aligned}$$

where the first and last (in)equalities follow easily from the definition of statistical distance, and the second (in)equality follows from Equation (2) and the fact that adding a constant to a random variable does not change its distance from uniform. But notice that by definition of $\mathbf{Z}_2$ and the leakage-resilience of Ext_0, this quantity is bounded above by ϵ_0, which completes the proof. □

In order to highlight the generality of this framework, we observe that by Lemma 3.3, a standard three-source extractor is, in fact, a $(3, 2)$-leakage-resilient extractor (up to some negligible loss in parameters). Thus, by instantiating Lemma 5.4 with $r = 3, s = 2$, we recover the original extractor and analysis of [CGGL20]. Even better, since Theorem 4.1 tells us that the independence number α of an (N, r, s)-design decreases quickly as r, s grow large together, we see that Lemma 5.4 offers a concrete way to construct extractors for adversarial sources with much fewer good sources, K.

If we want to realize the above plan, we need two explicit objects. First, we need explicit (N, r, s)-designs with independence numbers that decrease quickly as r, s grow together. Theorem 5 of the current paper gives exactly this, and in fact the independence numbers of our designs decrease with r, s *almost as quickly as possible*, as shown by the tightness of Theorem 4.1.

Second, we need explicit leakage-resilient extractors for polylogarithmic entropy that have exponentially small error. Very recently, these exact objects were constructed:

Theorem 5.5 ([CGG+20])**.** *There is a universal constant $C > 0$ such that for any sufficiently large* constant $r \in \mathbb{N}$ *and all $n, k \in \mathbb{N}$ satisfying $k \geq \log^C n$, there exists an explicit $(r, r-1)$-leakage resilient extractor $\mathsf{Ext} : (\{0,1\}^n)^r \to \{0,1\}^m$ for min-entropy k with output length $m = k^{\Omega(1)}$ and error $\epsilon = 2^{-k^{\Omega(1)}}$.*

By combining these explicit LREs with our explicit designs, we can finally prove Theorem 5.2, which significantly improves the adversarial source extractors of [CGGL20].

Proof of Theorem 5.2. Let C be the same universal constant from Theorem 5.5, and let $r \in \mathbb{N}$ be a sufficiently large (even) constant such that $2/r < \delta$, and such that Theorem 5.5 guarantees the existence of an explicit $(r, r-1)$-leakage resilient extractor $\mathsf{Ext}_0 : (\{0,1\}^n)^r \to \{0,1\}^m$ for min-entropy $k \geq \log^C n$ with output length $m = k^{\Omega(1)}$ and error $\epsilon = 2^{-k^{\Omega(1)}}$. For sufficiently large $N \in \mathbb{N}$, Theorem 5 guarantees the existence of an $(N, r, r-1)$-design G with independence number $\alpha < N^\delta$ that is computable in $\mathrm{poly}(\binom{N}{r}) = \mathrm{poly}(N)$ time. The result now follows by instantiating Lemma 5.4 with Ext_0 and G. □

Next, we will show how to use our new and improved low-error extractors for adversarial sources (Theorem 5.2) to obtain improved improved low-error extractors for small-space sources.

6 A reduction from small-space sources to adversarial sources

In this section, we will show how to use our extractors from Section 5 to obtain better extractors for small-space sources (as defined by Definition 1.2). We will prove the following.

Theorem 6.1 (Theorem 3, restated)**.** *For any fixed $\delta \in (0, 1/2]$ there is a constant $C > 0$ such that for all $n, k, s \in \mathbb{N}$ satisfying $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$, there exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources of min-entropy k, with output length $m = n^{\Omega(1)}$ and error $\epsilon = 2^{-n^{\Omega(1)}}$.*

Until very recently, the best explicit extractor for this setting [KRVZ06] required entropy $k \geq Cn^{1-\gamma}s^{\gamma}$, where $\gamma > 0$ is a tiny constant and C is a large one. In [CGGL20], this requirement was significantly improved to $k \geq Cn^{2/3+\delta}s^{1/3-\delta}$, for an arbitrarily small constant $\delta > 0$, and the current paper (Theorem 6.1) further improves this to $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$. Note that this line of improvements is strict, since we always have $s < n$ (or else the bounds become trivial). In particular, for say $s = n^{\delta}$ space, the entropy requirement has dropped from $k \geq O(n^{1-\gamma})$ to $k \geq O(n^{2/3+\delta})$ to $k \geq O(n^{1/2+\delta})$.

Non-constructively, it is known [KRVZ06] that there exist extractors for space s sources that have error ϵ for min-entropy $k \geq O(s + \log n + \log(1/\epsilon))$. Thus, while Theorem 6.1 significantly improves the state-of-art in low-error extraction, there is still a lot of room for improvement. However, we note (in Remark 6.7) that any substantial improvements to our low-error extractors (i.e., beyond entropy requirement $k \geq \sqrt{n}$) will require a new type of reduction that bypasses the need for so-called *total-entropy sources*, which are used in [KRVZ06, CGGL20] and are used here. We will see exactly such a reduction in Section 7. (It will allow us to obtain near-optimal extractors with *polynomial error*. To obtain improved low-error extractors using this new reduction, one needs improved low-error affine extractors.)

We now proceed to prove Theorem 6.1. The techniques that follow, which will reduce the task of extracting from small space sources to the task of extracting from adversarial sources, are just slightly optimized versions of the exact arguments that appear in [KRVZ06, CGGL20]. However, we include them here for completeness. The first step is to reduce small-space sources to a class of sources known as *total entropy sources*, defined as follows.

Definition 6.2. *A random variable $\mathbf{X}$ over $(\{0,1\}^{\ell})^r$ is an (r, ℓ, k)*-total entropy source *if $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_r)$, where each $\mathbf{X}_i$ is an independent source over $\{0,1\}^{\ell}$, and $\sum_{i\in[r]} H_\infty(\mathbf{X}_i) \geq k$.*

In [KRVZ06], Kamp et al. showed that upon fixing a few positions in the random walk that generates a small space source $\mathbf{X}$, it is straightforward to use Lemma 3.3 to show that $\mathbf{X}$ becomes a total-entropy source, with high probability. We include the proof for completeness.

Lemma 6.3 ([KRVZ06])**.** *Let $\mathbf{X}$ be a space s source over $\{0,1\}^n$ with min-entropy k. Then for any $\alpha \in (0, 1/4]$ such that $r = \alpha k/s$ and $\ell = ns/(\alpha k)$ are positive integers, it holds that $\mathbf{X}$ is $2^{-k/4}$-close to a convex combination of $(r, \ell, k/2)$-total entropy sources.*

Proof. For each $i \in [n]$, let $\mathbf{W}_i \sim \{0,1\}^s$ be the random variable denoting the state reached in layer i 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 $\mathbf{W}^* := (\mathbf{W}_{i\ell+1})_{i\in[0,r-1]}$, 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. Furthermore, by Lemma 3.3, we know

$$\Pr_{w\sim\mathbf{W}^*}[H_\infty(\mathbf{X} \mid \mathbf{W}^* = w) \geq k - rs - k/4 = k - \alpha k - k/4 \geq k/2] \geq 1 - 2^{-k/4}. \tag{3}$$

Thus, 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 $w \sim \mathbf{W}^*$, which completes the proof. □

The next step is to show that a total-entropy source looks like an adversarial source, using a standard Markov-type argument:

Lemma 6.4. *Let $\mathbf{X}$ be an (r, ℓ, Γ)-total entropy source. Then for any $N, K, n, k \in \mathbb{N}$ with $Nn = r\ell$ and n a multiple of ℓ, $\mathbf{X}$ is also an (N, K, n, k)-adversarial source, as long as $Kn + Nk \leq \Gamma$.*

Proof. By definition of total-entropy source, $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_r)$, where each $\mathbf{X}_i$ is an independent source over $\{0,1\}^\ell$. By collecting the sources $\mathbf{X}_i$ into N buckets containing n/ℓ sources each, we see that $\mathbf{X}$ is also an (N, n, Γ)-total entropy source, and may be rewritten as $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_N)$, where each $\mathbf{X}_i$ is an independent source over $\{0,1\}^n$. If $\mathbf{X}$ were not an (N, K, n, k)-adversarial source, then the $K-1$ highest entropy sources in $\mathbf{X}$ each have entropy at most n, and the remaining each have entropy $< k$. This yields $H_\infty(\mathbf{X}) = \Gamma < (K-1)n + (N-(K-1))k < Kn + Nk$, contradicting the given lower bound on Γ. □

Given the above reduction, we can now use our improved adversarial source extractors (from Theorem 4) to give improved extractors for total-entropy sources.

Theorem 6.5. *For any fixed $\delta > 0$ and all sufficiently large $r, \ell, \Gamma \in \mathbb{N}$ with $\Gamma \geq \max\left\{(r\ell)^{1/2+\delta}, r^\delta \ell\right\}$, there exists an explicit extractor $\mathsf{Ext} : (\{0,1\}^\ell)^r \to \{0,1\}^m$ for (r, ℓ, Γ)-total entropy sources, with output length $m = (r\ell)^{\Omega(1)}$ and error $\epsilon = 2^{-(r\ell)^{\Omega(1)}}$.*

Proof. Fix any $N, n \in \mathbb{N}$ such that $Nn = r\ell$ and n is a multiple of ℓ. By Lemma 6.4, every (r, ℓ, Γ)-total entropy source is also an (N, K, n, k)-adversarial source, provided $Kn + Nk \leq \Gamma$. Thus, by Theorem 4, for any fixed $\delta_0 > 0$ there exists an explicit extractor $\mathsf{Ext}_0 : (\{0,1\}^r)^\ell \to \{0,1\}^m$ for (r, ℓ, Γ)-total entropy sources with output length $m = n^{\Omega(1)}$ and error $\epsilon = 2^{-n^{\Omega(1)}}$, provided $N^{\delta_0} n + N n^{\delta_0} \leq \Gamma$ and N, n are sufficiently large. To achieve the parameters claimed in the theorem statement, pick $\delta_0 = \delta/2$ and set N, n as follows: (i) if $r \geq \ell$, set $N = n = \sqrt{r\ell}$; (ii) if $r < \ell$, set $N = r$ and $n = \ell$. We conclude by remarking that this casework was motivated by trying to minimize the requirement on Γ by setting $N = n$. This is not possible in case (ii), but is possible in case (i) by assuming, without loss of generality, that $r = x^2 \ell$ for some $x \in \mathbb{N}$. □

Previously, the best low-error explicit extractors for total-entropy sources [CGGL20] required entropy $\Gamma \geq \max\{(r\ell)^{2/3+\delta}, r^{1/2+\delta}\ell\}$. Non-constructively, we know it is possible [KRVZ06] to achieve an entropy requirement of $\Gamma \geq O(\ell + \log r)$ and error of $2^{-\Omega(\Gamma)}$. Thus, while there is still a lot of room to give improved explicit extractors for total-entropy sources, we remark that our total-entropy extractor is almost optimal when the source consists of "a few long sources":

Remark 6.6. *The entropy requirement in Theorem 6.5 becomes $k \geq \ell^{1+\delta}$ when $\ell \geq r$, which is close to the optimal requirement of $k \geq O(\ell)$.*

Finally, we show how to combine our improved explicit extractors for total-entropy sources (Theorem 6.5) with the standard reduction from small-space sources to total-entropy sources (Lemma 6.3) to complete the proof of Theorem 6.1:

Proof of Theorem 6.1. Fix any $\delta \in (0, 1/2]$, and let $\alpha \in (0, 1/4]$ be a sufficiently small constant and $C > 0$ a sufficiently large constant. Given a space s source $\mathbf{X}$ over $\{0,1\}^n$ with entropy $k \geq Cn^{1/2+\delta}s^{1/2-\delta}$, we know by Lemma 6.3 that $\mathbf{X}$ is $\epsilon_0 = 2^{-k/4}$-close to a convex combination of $(r, \ell, k/2)$-total entropy sources, where $r = \alpha k/s$ and $\ell = ns/(\alpha k)$. (Here we assume $r, \ell \in \mathbb{N}$, but it is easy to extend the argument when this is not the case.) In particular, this means there is some random variable $\mathbf{Y}$ such that with probability at least $1 - \epsilon_0$ over $y \sim \mathbf{Y}$, the random variable $(\mathbf{X} \mid \mathbf{Y} = y)$ is an $(r, \ell, k/2)$-total entropy source.

Let $\mathsf{Ext}_0 : (\{0,1\}^\ell)^r \to \{0,1\}^m$ be the extractor from Theorem 6.5 for such total-entropy sources. We will argue that it also an extractor for the small-space source $\mathbf{X}$. Notice we have $|\mathsf{Ext}_0(\mathbf{X}) - \mathbf{U}_m| \leq \mathbb{E}_{y\sim\mathbf{Y}}[|\mathsf{Ext}_0(\mathbf{X} \mid \mathbf{Y} = y) - \mathbf{U}_m|] \leq \epsilon_0 + |\mathsf{Ext}_0(\mathbf{X}') - \mathbf{U}_m|$, where $\mathbf{X}'$ is some $(r, \ell, k/2)$-total entropy source. If we can argue that $r, \ell, k/2$ are sufficiently large and $k/2 \geq \max\{(r\ell)^{1/2+\delta}, r^\delta \ell\}$, then Theorem 6.5 tells us that $|\mathsf{Ext}_0(\mathbf{X}) - \mathbf{U}_m| \leq \epsilon_0 + |\mathsf{Ext}_0(\mathbf{X}') - \mathbf{U}_m| \leq 2^{-k/4} + 2^{-(r\ell)^{\Omega(1)}} = 2^{-n^{\Omega(1)}}$ and $m = (r\ell)^{\Omega(1)} = n^{\Omega(1)}$, which would prove the current theorem. We know that $r, \ell, k/2$ are sufficiently large because $r = \alpha k/s \geq \alpha C(n/s)^{1/2+\delta} \geq \alpha C$, and $\ell = ns/(\alpha k) \geq 1/\alpha$, and $k \geq C$, where α is sufficiently small and C is sufficiently large. Next, we know $k/2 \geq (r\ell)^{1/2+\delta} = n^{1/2+\delta}$ by the provided lower bound on k. Finally, to show $k/2 \geq r^\delta \ell = (\alpha k/s)^\delta ns/(\alpha k)$, rearrange the inequality to obtain $k^{2-\delta} \geq 2\alpha^{\delta-1}s^{1-\delta}n$, plug in the provided lower bound on k to obtain $(Cn^{1/2+\delta}s^{1/2-\delta})^{2-\delta} \geq 2\alpha^{\delta-1}s^{1-\delta}n$, and observe that it therefore suffices to show $(0.5C^{2-\delta}\alpha^{1-\delta}) \cdot n^{(1/2+\delta)(2-\delta)-1} \geq s^{1-\delta-(2-\delta)(1/2-\delta)}$, or rather

$$(0.5C^{2-\delta}\alpha^{1-\delta}) \cdot n^{2\delta-\delta/2-\delta^2} \geq s^{2\delta-\delta/2-\delta^2}.$$

This holds because $n \geq s$ (otherwise the provided lower bound on k gives $k > n$), because $2\delta - \delta/2 - \delta^2 \geq 0$ over $\delta \in (0, 1/2]$, and because C is sufficiently large. □

We conclude this section with a remark about the $\sqrt{n}$ "barrier" in this reduction technique.

Remark 6.7. *It is not possible to obtain an entropy requirement of $k < \sqrt{n}$ using the reduction from small-space sources to total-entropy sources from Lemma 6.3, no matter how r, ℓ are set. This is because $r\ell = n$ implies either (i) $\ell \geq \sqrt{n}$, or (ii) $r \geq \sqrt{n}$. In case (i), all of the entropy could be trapped in a single source of length $> k$, from which extraction is impossible. In case (ii), the application of Equation (3) in Lemma 6.3 leaves the source with 0 bits of entropy, from which extraction is impossible.*

In the following section, we will give a *new* reduction that allows us to bypass the $\sqrt{n}$ barrier (for polynomial error). We are able to do this because (like in [CL16]), we reduce to a type of independent sources whose lengths need not be determined ahead of time. Unlike total-entropy sources, this will allow us to recurse whenever we get stuck in a tricky situation like case (i) in Remark 6.7.

7 A reduction from small-space sources to affine sources

In this section, we construct extractors for small-space sources that can handle just polylogarithmic entropy in the polynomial error regime, proving Theorem 1.

Theorem 7.1 (Theorem 1, restated)**.** *There exists a universal constant $C > 0$ such that for all $n, k, s \in \mathbb{N}$ satisfying $k \geq s \cdot \log^C(n)$, there exists an explicit extractor $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ for space s sources with min-entropy k, which has output length $m = (k/s)^{\Omega(1)}$ and error $\epsilon = n^{-\Omega(1)}$.*

The main tool we use to prove this theorem is a new reduction from small-space sources to affine sources. As we have seen, an affine source is simply a uniform distribution over some affine subspace of $\mathbb{F}_2^n$. It will be useful, however, to have the following formal definition.

Definition 7.2 (Affine source)**.** *A distribution* $\mathbf{X}$ *over* $\mathbb{F}_2^n$ *is an* affine source *with min-entropy* k *if there exists some* shift vector $v_0 \in \mathbb{F}_2^n$ *and linearly independent basis vectors* $v_1, v_2, \ldots, v_k \in \mathbb{F}_2^n$ *such that* $\mathbf{X}$ *is generated by sampling* k *bits uniformly at random* $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k \sim \mathbb{F}_2$ *and computing* $v_0 + \sum_{i \in [k]} \mathbf{x}_i v_i$.

Given this definition, we are now ready to define the main lemma used in proving Theorem 7.1.

Lemma 7.3 (Theorem 2, restated)**.** *Let* $\mathbf{X}$ *be a space* s *source over* $\{0,1\}^n$ *with min-entropy* k*. Then* $\mathbf{X}$ *is* ϵ*-close to a convex combination of affine sources with min-entropy* Γ*, where*

$$\Gamma = \Omega\left(\frac{k}{s \log(n/k)}\right),$$

and $\epsilon = 2^{-\Omega(k)}$.

Before proving Lemma 7.3, we use it to prove Theorem 7.1. We recall the standard fact that if an extractor works for each source $\mathbf{X}$ in a family $\mathcal{X}$ of distributions, then it also works for any convex combination of sources from that family. In particular, this means that any extractor for affine sources is automatically an extractor for small-space sources, by Lemma 7.3. The following affine extractor of Li [Li16], which can handle polylogarithmic entropy, will be of particular interest.

Theorem 7.4 ([Li16])**.** *There exists a universal constant* $C > 0$ *such that for all* $n, k \in \mathbb{N}$ *satisfying* $k \geq \log^C n$*, there exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for affine sources with min-entropy* k*, which has output length* $m = k^{\Omega(1)}$ *and error* $\epsilon = n^{-\Omega(1)}$.

Resetting the universal constant C as necessary, Theorem 7.1 follows immediately by combining Lemma 7.3 and Theorem 7.4. Furthermore, since our reduction (Lemma 7.3) has extremely low error, we note that we can also combine it with a classical affine extractor of Bourgain [Bou07] to immediately get the following bonus result:

Theorem 7.5. *For any fixed constants* $C, \delta > 0$ *and all* $n, k, s \in \mathbb{N}$ *satisfying* $k \geq \delta n$ *and* $s \leq C$*, there exists an explicit extractor* $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ *for space* s *sources with min-entropy* k*, which has output length* $m = \Omega(n)$ *and error* $\epsilon = 2^{-\Omega(n)}$.

To the best of our knowledge, this is the only nontrivial small-space extractor that achieves super low error $\epsilon = 2^{-\Omega(n)}$, as all previous constructions [KRVZ06] have error at least $\epsilon = 2^{-\widetilde{\Omega}(n)}$. This improvement in error is extremely 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; our main application of it (Theorem 7.1), however, will use a polynomial-error affine extractor, whose error will subsume the very low error of the reduction.

At last, we are ready to prove Lemma 7.3, which will immediately yield Theorem 7.1 (and Theorem 7.5). We do so in the following subsection.

7.1 A reduction from small-space sources to simple bit-block sources

In this subsection, we actually show a stronger result than Lemma 7.3. In particular, we prove that the reduction holds even for a special case of affine sources called *bit-block sources*. Given a vector $v \in \mathbb{F}_2^n$, we define support$(v) \subseteq [n]$ to be the subset of all coordinates where v takes the value 1, and we define these sources as follows:

Definition 7.6 ([Vio14])**.** *A source* $\mathbf{X}$ *over* $\mathbb{F}_2^n$ *is a* bit-block source *with min-entropy* k *if it is an affine source with min-entropy* k *(as per Definition 7.2) with the additional guarantee that* $\mathit{support}(v_i) \cap \mathit{support}(v_j) = \emptyset$*, for all* $i \neq j \in [k]$*.*

In fact, we even show that the reduction holds for a special case of bit-block sources, called *simple bit-block sources*.

Definition 7.7. *A source* $\mathbf{X}$ *over* $\mathbb{F}_2^n$ *is a* simple bit-block source *with min-entropy* k *if it is a bit-block source with min-entropy* k *(as per Definition 7.6), with the additional guarantee that* $\max(\mathit{support}(v_i)) < \min(\mathit{support}(v_j))$ *for all* $i < j \in [k]$*.*

Given these definitions, we are now able to state the technical version of Lemma 7.3.

Lemma 7.8 (Lemma 7.3, technical version)**.** *Let* $\mathbf{X}$ *be a space* s *source over* $\{0,1\}^n$ *with min-entropy* k*. Then* $\mathbf{X}$ *is* ϵ*-close to a convex combination of simple bit-block sources with min-entropy* Γ*, where*

$$\Gamma = \Omega\left(\frac{k}{s \log(n/k)}\right),$$

and $\epsilon = 2^{-\Omega(k)}$*.*

Before we prove Lemma 7.8, we remark that the reduction also works in the *reverse direction*, implying that small-space sources and simple bit-block sources are *roughly equivalent*, up to a factor of about s.

Remark 7.9. *Using Definitions 1.2 and 7.7, it is relatively straightforward to show: a simple bit-block 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 7.8, we see that simple bit-block sources and space* s *sources are* roughly equivalent *(in the low-error convex combination sense), up to a factor of* $\widetilde{O}(s)$*.*

Now, in order to prove Lemma 7.8, 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 bit-block sources. We prove the latter first.

Definition 7.10. *A source* $\mathbf{X}$ *over* $\{0,1\}^n$ *is an* (n, r, k)-independent source sequence *if there exist some (unknown) lengths* $\ell_1, \ldots, \ell_r \in [n]$ *that sum to* n *such that* $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_r)$*, where each* $\mathbf{X}_i$ *is an independent* (ℓ_i, k)*-source.*

Lemma 7.11. *Let* $\mathbf{X}$ *be an* $(n, \Gamma, 1)$*-independent source sequence. Then* $\mathbf{X}$ *is a convex combination of simple bit-block sources with min-entropy* Γ*.*

Proof. By definition, $\mathbf{X} = (\mathbf{X}_1, \dots, \mathbf{X}_\Gamma)$, where each $\mathbf{X}_i$ is an independent $(\ell_i, 1)$-source, for some $\ell_i \in [n]$. We use the nice observation from [CGGL20] that any $(\ell, 1)$-source $\mathbf{Z}$ is a convex combination of affine sources with min-entropy exactly 1. Recall this observation goes as follows: since $\mathbf{Z}$ is an $(\ell, 1)$-source, it follows via a standard argument (see, e.g., [Vad12]) that it is a convex combination of *flat sources* with min-entropy exactly 1. But any such flat source $\mathbf{Z}'$ is, by definition, a uniform distribution over two distinct strings $x, y \in \{0,1\}^\ell$, which must differ at *some* coordinate $i^* \in [\ell]$. Thus $\mathbf{Z}'_{i^*}$ is a uniform bit, and it is easy to verify that every other bit $\mathbf{Z}'_j, j \neq i^*$ is either constantly 0, constantly 1, or equal to exactly $\mathbf{Z}'_{i^*}$ or $\mathbf{Z}'_{i^*} \oplus 1$. Using Definition 7.2, it is now straightforward to show this is an affine source with min-entropy 1.

Thus, we can write each $\mathbf{X}_i$ as a convex combination of affine sources over $\mathbb{F}_2^{\ell_i}$ with min-entropy 1. This means $\mathbf{X}$ is a convex combination of sources of the form $\mathbf{X}' = (\mathbf{X}'_1, \dots, \mathbf{X}'_\Gamma)$, where each $\mathbf{X}'_i$ is still independent and has the same length ℓ_i as before, but is now also guaranteed to be sampled by the process $v_0^{(i)} + \mathbf{b}_i v_1^{(i)}$, where $v_0^{(i)}, v_1^{(i)} \in \mathbb{F}_2^{\ell_i}$ are fixed vectors with $v_1^{(i)} \neq 0$, and $\mathbf{b}_i \sim \mathbb{F}_2$ is a uniform bit. Thus, we can show $\mathbf{X}'$ is a simple bit-block source with min-entropy Γ as follows. First, define $v_0 := (v_0^{(1)}, v_0^{(2)}, \dots, v_0^{(\Gamma)}) \in \mathbb{F}_2^n$. Next, for each $i \in [\Gamma]$, define $v_i := (\mathbb{1}[1 = i] \cdot v_1^{(1)}, \mathbb{1}[2 = i] \cdot v_1^{(2)}, \dots, \mathbb{1}[\Gamma = i] \cdot v_1^{(\Gamma)}) \in \mathbb{F}_2^n$, where $\mathbb{1}[\cdot]$ is the indicator function. Then it is straightforward to verify that $\mathbf{X}'$ is sampled by $v_0 + \sum_{i \in [k]} \mathbf{b}_i v_i$, and that the vectors $v_0, v_1, \dots, v_k$ satisfy Definition 7.7. Thus $\mathbf{X}'$ is a simple bit-block source with min-entropy k, and $\mathbf{X}$ is a convex combination of such sources. □

At last, we are ready to prove that small-space sources are close to a convex combination of independent source sequences. By combining the following lemma with Lemma 7.11, we immediately get Lemma 7.8.

Lemma 7.12. *Let $\mathbf{X}$ be a space s source over $\{0,1\}^n$ with min-entropy k. Then $\mathbf{X}$ is ϵ-close to a convex combination of $(n, \Gamma, 1)$-independent source sequences, where $\Gamma = \Omega\left(\frac{k}{s \log(n/k)}\right)$ and $\epsilon = 2^{-\Omega(k)}$.*

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 will pick layers to fix in the following manner. We will 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.[4] 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 $H_\infty(\mathbf{X}^{(a,b)} \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)}, (\mathbf{L}_i)_{i \in I^{(1)}} = \ell^{(1)}, \ldots, (\mathbf{L}_\tau)_{i \in I^{(\tau)}} = \ell^{(\tau)}) > H_\infty(\mathbf{X}^{(c,d)} \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)}, (\mathbf{L}_i)_{i \in I^{(1)}} = \ell^{(1)}, \ldots, (\mathbf{L}_\tau)_{i \in I^{(\tau)}} = \ell^{(\tau)})$; or if these min-entropy are identical and $a > c$. We now check the min-entropies of the slices in $A^{(\tau)}$. If $H_\infty(\mathbf{X}^a \mid (\mathbf{L}_i)_{i \in I^{(0)}} = \ell^{(0)}, (\mathbf{L}_i)_{i \in I^{(1)}} = \ell^{(1)}, \ldots, (\mathbf{L}_\tau)_{i \in I^{(\tau)}} = \ell^{(\tau)}) \geq 1$ 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 $(*)$.[5] This concludes our fixings.

[4] 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.

[5] 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.

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 $[2^s]^{2\Gamma-1} \cup [2^s]^{2\Gamma-1+\Gamma} \cup \dots \cup [2^s]^{2\Gamma-1+t\Gamma}$. 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 $1-\epsilon$ 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 3.3, which tells us that with probability at least $1-\epsilon$ over fixing $\mathbf{Q}$ to q, we have $H_\infty(\mathbf{X} \mid \mathbf{Q} = q) \geq k - \log(|\text{support}(\mathbf{Q})|) - \log(1/\epsilon) \geq k - (t+2)s\Gamma - \log(1/\epsilon)$. 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

$$(B^{(0)} \setminus A^{(0)}) \cup (B^{(1)} \setminus A^{(1)}) \cup \dots \cup (B^{(\tau^*)} \setminus A^{(\tau^*)}) \cup A^{\tau^*},$$

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 $(\beta, \beta') \in (B^{(0)} \setminus A^{(0)}) \cup (B^{(1)} \setminus A^{(1)}) \cup \dots \cup (B^{(\tau^*-1)} \setminus A^{(\tau^*-1)})$, 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 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 $(B^{(0)} \setminus A^{(0)}) \cup (B^{(1)} \setminus A^{(1)}) \cup \dots \cup (B^{(\tau^*-1)} \setminus A^{(\tau^*-1)}) \cup (B^{(\tau^*)} \setminus A^{(\tau^*)}) \cup A^{\tau^*}$. 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:

$$\begin{aligned} H_\infty(\mathbf{X} \mid \mathbf{Q} = q) &< (1 + \tau^* - 1) \cdot \Gamma \cdot 1 + (1+1) \cdot \Gamma \cdot \left(\frac{n}{2\Gamma} \cdot 2^{-\tau^*} \right) \\ &= \Gamma\tau^* + n \cdot 2^{-\tau^*} \\ &= \Gamma t + n 2^{-t}. \end{aligned}$$

Combining this with the assumption we made about q near the beginning of our analysis, we have:

$$k - (t+2)s\Gamma - \log(1/\epsilon) \leq H_\infty(\mathbf{X} \mid \mathbf{Q} = q) < \Gamma t + n2^{-t}. \tag{4}$$

We finally arrive at our last goal: setting parameters Γ, t, ϵ. We know that for any setting of these parameters that contradicts Equation (4), Case (2) simply cannot occur. Thus, for any such setting, we know that with probability $1 - \epsilon$ over fixing $\mathbf{Q} = q$, we have $H_\infty(\mathbf{X} \mid \mathbf{Q} = q) \geq k - (t+2)s\Gamma - \log(1/\epsilon)$, and this implies Case (1) must occur. In other words, with probability $1 - \epsilon$ 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 (4), 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 (4). In particular, these parameters just need to satisfy

$$\begin{gathered} k - (t+2)s\Gamma - \log(1/\epsilon) \geq \Gamma t + n2^{-t}, \text{ and} \\ \frac{n}{2\Gamma} \cdot 2^{-t} \geq 1. \end{gathered}$$

Combining these, we just require:

$$\Gamma \leq \min \left\{ \frac{k - n2^{-t} - \log(1/\epsilon)}{t \cdot (s-1) + 2s}, \frac{n}{2^{t+1}} \right\}.$$

We take $t := \log(4n/k)$ and $\epsilon := 2^{-k/2}$ and $\Gamma := \frac{k}{20s \cdot \log(n/k)}$ to complete the proof. □

8 Future directions

In this paper, we give new constructions of extractors for small-space sources based on (i) a new reduction from small-space sources to affine sources, and (ii) improved extractors for adversarial sources. The new key ingredient we use for our adversarial source extractors is (the first) derandomization of Rödl and Šinajová's probabilistic designs [RŠ94], which we combine with recent explicit constructions [KMS19, CGG+20] of a certain kind of leakage resilient extractors, known as extractors for cylinder intersections. These constructions demonstrate new applications of these two pseudorandom objects, and it would be interesting to explore whether these objects have further applications in pseudorandomness and complexity.

Beyond the above, the three most natural open problems are as follows.

Problem 1. Better low-error extractors for small-space sources: *Reduce the entropy requirement for low-error small-space extraction (Theorem 3) so that it is closer to the entropy requirement for polynomial-error small-space extraction (Theorem 1).*

Problem 2. Better extractors for adversarial sources: *Improve the requirement on good sources in Theorem 4 from $K \geq N^{\delta}$ to $K \geq \text{polylog}(N)$, or (less ambitiously) $K \geq N^{o(1)}$.*

Problem 3. Better explicit designs with small independence number: *Improve the constant in the power of n of Theorem 5 from 2 to 1.99.*

Given our new reduction from small-space extractors to affine sources, a concrete way to approach Problem 1 is to simply pursue the construction of better low-error affine extractors. In particular, solving the affine extraction problem would effectively also "finish off" the small-space extraction problem. Meanwhile, Problem 2 can be solved by constructing a leakage-resilient extractor against *number-on-forehead* leakage: that is, an extractor whose output looks uniform even conditioned on joint functions of all but one of its inputs. Finally, it would be interesting to see if Problem 3 could be answered using more elaborate properties of specific codes (i.e., beyond their distance and dimension).

References

[AOR+20] Divesh Aggarwal, Maciej Obremski, João Ribeiro, Luisa Siniscalchi, and Ivan Visconti. How to extract useful randomness from unreliable sources. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 343–372. Springer, 2020.

[BACDTS19] 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 *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

[BGM19] Marshall Ball, Oded Goldreich, and Tal Malkin. Randomness extraction from somewhat dependent sources. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 26, page 183, 2019.

[BIW06] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting randomness using few independent sources. *SIAM Journal on Computing*, 36(4):1095–1118, 2006.

[Blu86] Manuel Blum. Independent unbiased coin flips from a correlated biased sourcea finite state markov chain. *Combinatorica*, 6(2):97–108, 1986.

[Bou07] Jean Bourgain. On the construction of affine extractors. *GAFA Geometric And Functional Analysis*, 17(1):33–57, 2007.

[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.

[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.

[CGG+20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, Ashutosh Kumar, Xin Li, Raghu Meka, and David Zuckerman. Extractors and secret sharing against bounded collusion protocols. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1226–1242. IEEE, 2020.

[CGGL20] Eshan Chattopadhyay, Jesse Goodman, Vipul Goyal, and Xin Li. Extractors for adversarial sources via extremal hypergraphs. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 1184–1197, New York, NY, USA, 2020. Association for Computing Machinery.

[CGH+85] Benny Chor, Oded Goldreich, Johan Hasted, Joel Freidmann, Steven Rudich, and Roman Smolensky. The bit extraction problem or t-resilient functions. In *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*, pages 396–407. IEEE, 1985.

[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.

[CGL21] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. To Appear in the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS), 2021.

[CL16] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 299–311. ACM, 2016.

[Coh16] Gil Cohen. Local correlation breakers and applications to three-source extractors and mergers. *SIAM Journal on Computing*, 45(4):1297–1338, 2016.

[CZ19] Eshan Chattopadhyay and David Zuckerman. Explicit two-source extractors and resilient functions. *Annals of Mathematics*, 189(3):653–705, 2019.

[DG10] Matt DeVos and Ariel Gabizon. Simple affine extractors using dimension expansion. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 50–57. IEEE, 2010.

[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.

[DOPS04] Yevgeniy Dodis, Shien Jin Ong, Manoj Prabhakaran, and Amit Sahai. On the (im)possibility of cryptography with imperfect randomness. In *45th Annual IEEE Symposium on Foundations of Computer Science*, pages 196–205. IEEE, 2004.

[Eus13] Alexander Eustis. *Hypergraph independence numbers*. PhD thesis, UC San Diego, 2013.

[EV13] Alex Eustis and Jacques Verstraëte. On the independence number of Steiner systems. *Combinatorics, Probability & Computing*, 22(2):241–252, 2013.

[GB10] Venkatesan Guruswami and Eric Blais. Notes 6: Reed-Solomon, BCH, Reed-Muller and concatenated codes. *Introduction to Coding Theory CMU: Spring*, 2010.

[GPR95] David A. Grable, Kevin T. Phelps, and Vojtěch Rödl. The minimum independence number for designs. *Combinatorica*, 15(2):175–185, 1995.

[GR08] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, 28(4):415–440, 2008.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. Unbalanced expanders and randomness extractors from parvaresh–vardy codes. *Journal of the ACM (JACM)*, 56(4):20, 2009.

[Hoc59] Alexis Hocquenghem. Codes correcteurs d'erreurs. *Chiffres*, 2(2):147–56, 1959.

[KM04] Robert Koenig and Ueli Maurer. Extracting randomness from generalized symbol-fixing and markov sources. In *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.*, page 232. IEEE, 2004.

[KM05] Robert Koenig and Ueli Maurer. Generalized strong extractors and deterministic privacy amplification. In *IMA International Conference on Cryptography and Coding*, pages 322–339. Springer, 2005.

[KMS19] Ashutosh Kumar, Raghu Meka, and Amit Sahai. Leakage-resilient secret sharing against colluding parties. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 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.

[KRVZ06] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. Deterministic extractors for small-space sources. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 691–700. ACM, 2006.

[KZ06] Jesse Kamp and David Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. *SIAM Journal on Computing*, 36(5):1231–1247, 2006.

[Li11] Xin Li. A new approach to affine extractors and dispersers. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, pages 137–147, 2011.

[Li15] Xin Li. Three-source extractors for polylogarithmic min-entropy. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 863–882. IEEE, 2015.

[Li16] Xin Li. Improved two-source extractors, and affine extractors for polylogarithmic entropy. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 168–177. IEEE, 2016.

[Li19] Xin Li. Non-malleable extractors and non-malleable codes: Partially optimal constructions. In *34th Computational Complexity Conference, CCC 2019, July 18-20, 2019, New Brunswick, NJ, USA.*, pages 28:1–28:49, 2019.

[LRVW03] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 602–611. ACM, 2003.

[MW97] Ueli Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In *Annual International Cryptology Conference*, pages 307–321. Springer, 1997.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

[Rao09] Anup Rao. Extractors for low-weight affine sources. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 95–101. IEEE, 2009.

[RŠ94] Vojtěch Rödl and Edita Šinajová. Note on independent sets in Steiner systems. *Random Structures & Algorithms*, 5(1):183–190, 1994.

[Sha11] Ronen Shaltiel. An introduction to randomness extractors. In *International Colloquium on Automata, Languages, and Programming*, pages 21–41. Springer, 2011.

[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.

[TL18] Fang Tian and Zi-Long Liu. Bounding the independence number in some (n, k, ℓ, λ)-hypergraphs. *Graphs and Combinatorics*, 34(5):845–861, 2018.

[TV00] Luca Trevisan and Salil Vadhan. Extracting randomness from samplable distributions. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 32–42. IEEE, 2000.

[Vad12] Salil Vadhan. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science*, 7(1–3):1–336, 2012.

[Vaz87] Umesh Vazirani. Efficiency considerations in using semi-random sources. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 160–168, 1987.

[Vio14] Emanuele Viola. Extractors for circuit sources. *SIAM Journal on Computing*, 43(2):655–672, 2014.

[vN51] John von Neumann. Various techniques used in connection with random digits. *Appl. Math Ser*, 12(36-38):5, 1951.

[Yeh11] Amir Yehudayoff. Affine extractors over prime fields. *Combinatorica*, 31(2):245–256, 2011.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il