2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-6654-2055-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/FOCS52979.2021.00066

Improved Extractors for Small-Space Sources

Eshan Chattopadhyay
Cornell University
eshanc@cornell.edu

Jesse Goodman
Cornell University
jpmgoodman@cs.cornell.edu

***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 entropy $k \geq s^{1.1} \cdot 2^{\log^{0.51} n}$ (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.

***Index Terms*—adversarial sources; affine sources; designs; explicit constructions; extremal hypergraphs; randomness extractors; small-space sources**

I. Introduction

Randomness is a powerful computational resource that has found beautiful applications in algorithm design, cryptography, and combinatorics (see [1] 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 I.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., [1], [2]). 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 [3]–[5]. In this paper, we focus on the second direction, where one assumes each

2575-8454/21/$31.00 ©2021 IEEE
DOI 10.1109/FOCS52979.2021.00066

source $\mathbf{X} \in \mathcal{X}$ has some additional structure beyond its min-entropy guarantee.

a) 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 [6], under the suggestion that they are a good model for distributions that would actually arise in nature. In [6], and the follow-up works of Viola [7] and Li [8], 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 [9], and further studied in recent work [10], [11]. 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 I.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 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 [9], small-space sources can simulate: von Neumann's model of a coin with unknown bias [12]; the finite Markov chain model of Blum [13]; the space-bounded models of Vazirani [14] and Koenig and Maurer [15], [16]; and the popular models of oblivious bit-fixing and symbol-fixing sources [17], [18] and independent sources [19]. In fact, it is suggested in [9] that the only model of sources that appears unrelated to small-space sources is the class of *affine sources* [20].

A. 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) 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 [10] 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) k-dimensional affine subspace of $\mathbb{F}_2^n$. A long line of work has considered the problem of constructing extractors for affine sources [8], [20]–[26], 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 [8], which works for $\mathrm{polylog}(n)$ 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 [26] to extract from small-space sources with min-entropy $k \geq s \cdot \log^{2+o(1)}(n)$.

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 [27].

However, since the best low-error affine extractors require entropy $k \geq \Omega(n/\sqrt{\log\log n})$ [22]–[24], 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 [9] required entropy $k \geq Cn^{1-\gamma}s^{\gamma}$, where $\gamma > 0$ is some tiny constant and C is a large one. In [11], 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 [9] to [11] 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 [9] of reducing small-space sources to the class of *adversarial sources* [11]. 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 [11] 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 [11].

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 [28], [29]. While such leakage-resilient extractors were recently constructed explicitly in [29], the only known construction of such designs is probabilistic [30].

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 [11] 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 [30], 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 the full version of the paper for more details.

Finally, we can combine our explicit designs with the leakage-resilient extractors from [29] 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 $\sqrt{n}$. Thus, the result in Theorem 3 has almost the best parameters one can hope to achieve using this technique.

II. Overview of Techniques

We use this section to sketch the explicit constructions of our small-space extractors. We start with our low-error

[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.

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).

A. *Small-space extractors with exponentially small error*

To construct our low-error small-space extractors, the first step is to use a standard reduction [18] (which we slightly optimize) from small-space sources to adversarial sources. This reduction starts with the observation of [9] 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.

a) 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 [11], 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. [29], where they gave the first explicit LREs that work for entropy $k = o(n)$, and in fact their LREs work for entropy $k \geq \mathrm{polylog}(n)$. Thus all that remains is to provide an explicit construction of designs with small independence number. We provide such a construction in this paper, and sketch it below.

b) 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 [31], [32] on the size of sets in $\mathbb{F}_2^n$ 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 [33], [34].

B. *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) 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 an observation of [11], which says that *any* source with entropy at least 1 is a convex combination of *affine sources* with entropy 1. Given this, 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 can 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 $k \geq \mathrm{polylog}(n)$, 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* [9] and *adversarial sources* [11]). 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 [11]) 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* [10] 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 [10].

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 III.1) *once*, which keeps the error of our reduction very low: $2^{-\Omega(k)}$, compared to an error of $2^{-k^{\Omega(1)}}$ in the recursive reduction of [10].

b) An alternate construction of low-error extractors for small-space sources: It turns out that in the above reduction from small space sources to affine sources, we are actually reducing to a special type of affine sources known as *bit block sources*, which were introduced in [7]. While there are currently no explicit low-error extractors for affine sources with low entropy, we *do* have such objects for bit block sources [7], [25]. As a result, the above reduction actually provides an alternate construction of low error extractors for small space sources. However, the entropy requirement of this alternate construction is slightly worse than our construction that goes through explicit designs and adversarial extractors, and furthermore it does not provide these results (Theorems 4 and 5), which are of independent interest, along the way. For more detail, we refer the reader to the full version of the paper.

c) Organization: In Section III we provide several preliminaries. In the remainder of our paper, we follow a *bottom-up* strategy for presenting our main results. In Section IV, we provide an explicit construction of designs with small independence number, proving Theorem 5. In Section V, we combine these designs with the explicit leakage-resilient extractors of [29] to obtain our improved extractors for adversarial sources, Theorem 4. In Section VI, we observe how this immediately gives us our small-space extractors with exponentially small error, Theorem 3. In Section VII, we provide our *new reduction* from small-space sources to *affine sources* (Theorem 2) and apply the affine extractor of Li [8] to obtain our small-space extractors for polylogarithmic entropy, Theorem 1. We conclude with some remarks and present some open problems in Section VIII.

We refer the reader to [35] for the full version of this paper.

III. Preliminaries

a) 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.

b) 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.

c) 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

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

and we say that $\mathbf{X}, \mathbf{Y}$ are *ϵ-close* if $|\mathbf{X} - \mathbf{Y}| \leq \epsilon$. Finally, we will need the following standard lemma about conditional min-entropy.

Lemma III.1 ([36]). *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*

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

IV. Explicit extremal designs via slicing codes and zero-sum sets

In this section, we will construct our explicit designs and thereby prove Theorem 5. But before we do so, we begin with some background and discussion on (n, r, s)-designs.

A. 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 [37], 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 [11], 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 IV.1 ([30]). *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 IV.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).

In this section, we will provide explicit constructions of these extremal designs. Our designs give the first derandomization of Theorem IV.1, and differ from the optimal bound by just a square.

B. *Proof of Theorem 5*

We are now ready to explicitly construct our designs, and thereby prove Theorem 5. 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 IV.2. *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 IV.3. *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 $A := \mathit{span}(\{e_i\}_{i \in S})$ 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 IV.2 and Fact IV.3, 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 IV.4. *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 IV.2 that $G_{Q \cap \Delta_r}$ is an (n, r, s)-design. By Fact IV.3, there is a subcube $A = \mathit{span}(e_{i_1}, \ldots, e_{i_\alpha}) \subseteq \mathbb{F}_2^n$ 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 $(x_1, \dots, x_n) \mapsto (x_{i_1}, \dots, x_{i_\alpha})$, 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 $\alpha + k - n \leq dim(\pi(A')) \leq \Lambda_r(\alpha)$. □

To construct an (n, r, s)-design from Lemma IV.4 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 [32].

Lemma IV.5 ([32])**.** *For every $n \geq r \in \mathbb{N}$ where r is even,*

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

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 IV.6 ([31], 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 IV.5, we get the following corollary.

Corollary IV.7 ([32])**.** *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 IV.8 (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 IV.7 into Lemma IV.4. □

To complete the proof of Theorem 5, we now just need to explicitly construct a linear code with very high dimension. In 1959-1960, Bose, Ray-Chaudhuri [33], and Hocquenghem [34] explicitly constructed codes of exactly this type (see [38] for a great exposition of these codes, which are known as *BCH codes*). In particular, they proved the following theorem.

Theorem IV.9 ([33], [34])**.** *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 $\mathrm{poly}(n)$ time.*

By instantiating Lemma IV.8 with Theorem IV.9, we immediately obtain Theorem 5. We refer the reader to the full version for more details.

V. 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* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$, each guaranteed to have some min-entropy. A long line of work has focused on constructing extractors for this setting [19], [39]–[43], and has culminated in extractors with a near-optimal entropy requirement [43]. Recently, the idea of generalizing this model to allow for *bad sources* with *no entropy guarantee* and/or *limited dependence* has received considerable attention [11], [44], [45]. 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 [11] introduced the class of *adversarial sources*:

Definition V.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* $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$*, 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 this section, we will construct a significantly improved extractor for adversarial sources, thereby proving Theorem 4. Our proof of Theorem 4 builds upon and generalizes the so-called "*activation vs. fragile correlation*" paradigm introduced in [11] for extracting from adversarial sources. In particular, instead of combining various types of specialized extractors with various types of specialized extremal hypergraphs (as is done in [11]), we combine just one type of general robust extractor with one type of general extremal hypergraph. The general extremal hypergraphs we use are the designs constructed in Section IV, while the general robust extractor we use 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 [28]. 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 V.2 ([28], [29])**.** *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* $\mathbf{X} := (\mathbf{X}_1, \dots, \mathbf{X}_r)$ *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:*

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

By combining these robust extractors with our designs from Section IV, it is now easy to construct a new framework for extracting from adversarial sources. This framework, which generalizes the extractor constructions in [11], is formally captured in the following lemma (which is proven in the full version of this paper).

Lemma V.3. *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* $k \geq k_0$*, 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* $\epsilon = \epsilon_0$*.*

If we want to use the above framework to extract from adversarial sources with the fewest number of good sources possible, 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 IV.1.

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

Theorem V.4 ([29])**.** *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* $\epsilon = 2^{-k^{\Omega(1)}}$*.*

It is now not too difficult to instantiate our framework (Lemma V.3) with these LREs (Theorem V.4) and our explicit designs (Theorem 5) to obtain our significantly improved extractors for adversarial sources (Theorem 4). We refer the reader to the full version for more details.

VI. A Reduction from Small-Space Sources to Adversarial Sources

Now, we briefly discuss how our improved (low-error) extractors for adversarial sources immediately imply improved low-error extractors for small-space sources, thereby proving Theorem 3. We use an intermediate class of sources known as *total entropy sources*, defined as follows.

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

It is well known that small-space sources are close to a convex combination of total entropy sources [9], and that total entropy sources are, in fact, adversarial sources (for some appropriate parameters) [11]. By composing these two reductions, our improved extractors for adversarial sources immediately imply Theorem 3. Along the way, we also obtain the following improved extractor for total entropy sources.

Theorem VI.2. *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 $\epsilon = 2^{-(r\ell)^{\Omega(1)}}$.*

Previously, the best low-error explicit extractors for total-entropy sources [11] required entropy $\Gamma \geq \max\{(r\ell)^{2/3+\delta}, r^{1/2+\delta}\ell\}$. Non-constructively, we know it is possible [9] 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, our total-entropy extractor is almost optimal when the source consists of "a few long sources":

Remark VI.3. *The entropy requirement in Theorem VI.2 becomes $k \geq \ell^{1+\delta}$ when $\ell \geq r$, which is close to the optimal requirement of $k \geq O(\ell)$.*

VII. A REDUCTION FROM SMALL-SPACE SOURCES TO AFFINE SOURCES

Finally, we construct extractors for small-space sources that can handle just polylogarithmic entropy in the polynomial error regime, proving Theorem 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 VII.1 (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 $v_1, v_2, \dots, v_k \in \mathbb{F}_2^n$ such that $\mathbf{X}$ is generated by sampling k bits uniformly at random $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_k \sim \mathbb{F}_2$ 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 1.

Lemma VII.2 (Theorem 2, restated)**.** *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 Γ, where*

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

Before proving Lemma VII.2, we use it to prove Theorem 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 VII.2. The following affine extractor of Li [8], which can handle polylogarithmic entropy, will be of particular interest.

Theorem VII.3 ([8])**.** *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 1 follows immediately by combining Lemma VII.2 and Theorem VII.3. Furthermore, since our reduction (Lemma VII.2) has extremely low error, we note that we can also combine it with a classical affine extractor of Bourgain [22] to immediately get the following bonus result:

Theorem VII.4. *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 [9] have error at least $\epsilon = 2^{-\widetilde{\Omega}(n)}$.

At last, we are ready to prove Lemma VII.2, which will immediately yield Theorem 1 (and Theorem VII.4).

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

In this subsection, we actually show a stronger result than Lemma VII.2. 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 $\text{support}(v) \subseteq [n]$ to be the subset of all coordinates where v takes the value 1, and we define these sources as follows:

Definition VII.5 ([7])**.** *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 VII.1) with the additional guarantee that support$(v_i) \cap$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.

Definition VII.6. *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 VII.5), with the additional guarantee that* $\max(\textit{support}(v_i)) < \min(\textit{support}(v_j))$ *for all* $i < j \in [k]$.

Given these definitions, we are now able to state the technical version of Lemma VII.2.

Lemma VII.7 (Lemma VII.2, 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 VII.7, we briefly observe that 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 VII.7, 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 $\widetilde{O}(s)$.

We are now ready to prove Lemma VII.7. We will use an intermediate type of source, called an *independent source sequence*, which is a natural generalization of independent sources to allow for uneven (and unknown) length. We will show that small-space sources are (close to) a convex combination of independent source sequences, which are a convex combination of simple bit-block sources.

We start by defining independent source sequences.

Definition VII.8. *A source* $\mathbf{X}$ *over* $\{0,1\}^n$ *is an* (n,r,k)-independent source sequence *if there exist some (unknown) lengths* $\ell_1,\dots,\ell_r \in [n]$ *that sum to* n *such that* $\mathbf{X} = (\mathbf{X}_1,\dots,\mathbf{X}_r)$*, where each* $\mathbf{X}_i$ *is an independent* (ℓ_i,k)*-source.*

Next, we show that an independent source sequence is a convex combination of simple bit-block sources.

Lemma VII.9. *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.

The proof of this result (in the full version of the paper) uses the nice observation from [11] that any $(\ell,1)$-source $\mathbf{Z}$ is a convex combination of affine sources with min-entropy exactly 1. Finally, we show that small-space sources are close to a convex combination of independent source sequences. By combining the following lemma with Lemma VII.9, we immediately get Lemma VII.7.

Lemma VII.10. *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* $\epsilon = 2^{-\Omega(k)}$.

This is the key lemma in the proof of Lemma VII.7, and thus in the proof of Theorem 1. We refer the reader to Section II-B for a sketch of its proof, and to the full version of this paper for the complete proof.

VIII. Future Directions

In this paper, we demonstrated new applications of extremal designs and leakage-resilient extractors. It would be interesting to explore whether these objects have further applications in pseudorandomness and complexity. Beyond this, three 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* $K \geq \mathrm{polylog}(N)$*, 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.*

Acknowledgements

Eshan Chattopadhyay and Jesse Goodman are supported by NSF CAREER Award 2045576.

References

[1] S. Vadhan, "Pseudorandomness," *Foundations and Trends® in Theoretical Computer Science*, vol. 7, no. 1–3, pp. 1–336, 2012.

[2] R. Shaltiel, "An introduction to randomness extractors," in *International Colloquium on Automata, Languages, and Programming*. Springer, 2011, pp. 21–41.

[3] C.-J. Lu, O. Reingold, S. Vadhan, and A. Wigderson, "Extractors: Optimal up to constant factors," in *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM, 2003, pp. 602–611.

[4] V. Guruswami, C. Umans, and S. Vadhan, "Unbalanced expanders and randomness extractors from parvaresh–vardy codes," *Journal of the ACM (JACM)*, vol. 56, no. 4, p. 20, 2009.

[5] Z. Dvir, S. Kopparty, S. Saraf, and M. Sudan, "Extensions to the method of multiplicities, with applications to kakeya sets and mergers," *SIAM Journal on Computing*, vol. 42, no. 6, pp. 2305–2328, 2013.

[6] L. Trevisan and S. Vadhan, “Extracting randomness from samplable distributions,” in *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE, 2000, pp. 32–42.

[7] E. Viola, “Extractors for circuit sources,” *SIAM Journal on Computing*, vol. 43, no. 2, pp. 655–672, 2014.

[8] X. Li, “Improved two-source extractors, and affine extractors for polylogarithmic entropy,” in *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2016, pp. 168–177.

[9] J. Kamp, A. Rao, S. Vadhan, and D. Zuckerman, “Deterministic extractors for small-space sources,” in *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM, 2006, pp. 691–700.

[10] E. Chattopadhyay and X. Li, “Extractors for sumset sources,” in *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*. ACM, 2016, pp. 299–311.

[11] E. Chattopadhyay, J. Goodman, V. Goyal, and X. Li, “Extractors for adversarial sources via extremal hypergraphs,” in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, ser. STOC 2020. New York, NY, USA: Association for Computing Machinery, 2020, pp. 1184–1197. [Online]. Available: https://doi.org/10.1145/3357713.3384339

[12] J. von Neumann, “Various techniques used in connection with random digits,” *Appl. Math Ser*, vol. 12, no. 36-38, p. 5, 1951.

[13] M. Blum, “Independent unbiased coin flips from a correlated biased source—a finite state markov chain,” *Combinatorica*, vol. 6, no. 2, pp. 97–108, 1986.

[14] U. Vazirani, “Efficiency considerations in using semi-random sources,” in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, 1987, pp. 160–168.

[15] R. Koenig and U. Maurer, “Extracting randomness from generalized symbol-fixing and markov sources,” in *International Symposium onInformation Theory, 2004. ISIT 2004. Proceedings.* IEEE, 2004, p. 232.

[16] ——, “Generalized strong extractors and deterministic privacy amplification,” in *IMA International Conference on Cryptography and Coding*. Springer, 2005, pp. 322–339.

[17] B. Chor, O. Goldreich, J. Hasted, J. Freidmann, S. Rudich, and R. Smolensky, “The bit extraction problem or t-resilient functions,” in *26th Annual Symposium on Foundations of Computer Science (sfcs 1985)*. IEEE, 1985, pp. 396–407.

[18] J. Kamp and D. Zuckerman, “Deterministic extractors for bit-fixing sources and exposure-resilient cryptography,” *SIAM Journal on Computing*, vol. 36, no. 5, pp. 1231–1247, 2006.

[19] B. Chor and O. Goldreich, “Unbiased bits from sources of weak randomness and probabilistic communication complexity,” *SIAM Journal on Computing*, vol. 17, no. 2, pp. 230–261, 1988.

[20] A. Gabizon and R. Raz, “Deterministic extractors for affine sources over large fields,” *Combinatorica*, vol. 28, no. 4, pp. 415–440, 2008.

[21] M. DeVos and A. Gabizon, “Simple affine extractors using dimension expansion,” in *2010 IEEE 25th Annual Conference on Computational Complexity*. IEEE, 2010, pp. 50–57.

[22] J. Bourgain, “On the construction of affine extractors,” *GAFA Geometric And Functional Analysis*, vol. 17, no. 1, pp. 33–57, 2007.

[23] A. Yehudayoff, “Affine extractors over prime fields,” *Combinatorica*, vol. 31, no. 2, pp. 245–256, 2011.

[24] X. Li, “A new approach to affine extractors and dispersers,” in *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 2011, pp. 137–147.

[25] A. Rao, “Extractors for low-weight affine sources,” in *2009 24th Annual IEEE Conference on Computational Complexity*. IEEE, 2009, pp. 95–101.

[26] E. Chattopadhyay, J. Goodman, and J.-J. Liao, “Affine extractors for almost logarithmic entropy,” 2021, to Appear in the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS).

[27] Y. Dodis, S. J. Ong, M. Prabhakaran, and A. Sahai, “On the (im)possibility of cryptography with imperfect randomness,” in *45th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, 2004, pp. 196–205.

[28] A. Kumar, R. Meka, and A. Sahai, “Leakage-resilient secret sharing against colluding parties,” in *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2019, pp. 636–660.

[29] E. Chattopadhyay, J. Goodman, V. Goyal, A. Kumar, X. Li, R. Meka, and D. Zuckerman, “Extractors and secret sharing against bounded collusion protocols,” in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2020, pp. 1226–1242.

[30] V. Rödl and E. Šinajová, “Note on independent sets in Steiner systems,” *Random Structures & Algorithms*, vol. 5, no. 1, pp. 183–190, 1994.

[31] A. Sidorenko, “Extremal problems on the hypercube and the codegree Turán density of complete r-graphs,” *SIAM Journal on Discrete Mathematics*, vol. 32, no. 4, pp. 2667–2674, 2018.

[32] ——, “On generalized Erdős–Ginzburg–Ziv constants for $\mathbb{Z}_2^d$,” *Journal of Combinatorial Theory, Series A*, vol. 174, p. 105254, 2020.

[33] R. C. Bose and D. K. Ray-Chaudhuri, “On a class of error correcting binary group codes,” *Information and Control*, vol. 3, no. 1, pp. 68–79, 1960.

[34] A. Hocquenghem, “Codes correcteurs d’erreurs,” *Chiffres*, vol. 2, no. 2, pp. 147–56, 1959.

[35] E. Chattopadhyay and J. Goodman, “Improved extractors for small-space sources,” *Electron. Colloquium Comput. Complex.*, p. 106, 2020. [Online]. Available: https://eccc.weizmann.ac.il/report/2020/106

[36] U. Maurer and S. Wolf, “Privacy amplification secure against active adversaries,” in *Annual International Cryptology Conference*. Springer, 1997, pp. 307–321.

[37] N. Nisan and A. Wigderson, “Hardness vs randomness,” *Journal of Computer and System Sciences*, vol. 49, no. 2, pp. 149–167, 1994.

[38] V. Guruswami and E. Blais, “Notes 6: Reed-Solomon, BCH, Reed-Muller and concatenated codes,” *Introduction to Coding Theory CMU: Spring*, 2010.

[39] B. Barak, R. Impagliazzo, and A. Wigderson, “Extracting randomness using few independent sources,” *SIAM Journal on Computing*, vol. 36, no. 4, pp. 1095–1118, 2006.

[40] X. Li, “Three-source extractors for polylogarithmic min-entropy,” in *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. IEEE, 2015, pp. 863–882.

[41] G. Cohen, “Local correlation breakers and applications to three-source extractors and mergers,” *SIAM Journal on Computing*, vol. 45, no. 4, pp. 1297–1338, 2016.

[42] E. Chattopadhyay and D. Zuckerman, “Explicit two-source extractors and resilient functions,” *Annals of Mathematics*, vol. 189, no. 3, pp. 653–705, 2019. [Online]. Available: https://www.jstor.org/stable/10.4007/annals.2019.189.3.1

[43] X. 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.*, 2019, pp. 28:1–28:49.

[44] D. Aggarwal, M. Obremski, J. a. Ribeiro, L. Siniscalchi, and I. Visconti, “How to extract useful randomness from unreliable sources,” in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2020, pp. 343–372.

[45] M. Ball, O. Goldreich, and T. Malkin, “Randomness extraction from somewhat dependent sources,” in *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 26, 2019, p. 183.