Electronic Colloquium on Computational Complexity, Report No. 153 (2022)

Hardness against Linear Branching Programs and More

Eshan Chattopadhyay*
Cornell University
eshan@cs.cornell.edu

Jyun-Jie Liao*
Cornell University
jjliao@cs.cornell.edu

November 8, 2022

Abstract

In a recent work, Gryaznov, Pudlák and Talebanfard (CCC '22) introduced a linear variant of read-once branching programs, with motivations from circuit and proof complexity. Such a read-once linear branching program is a branching program where each node is allowed to make $\mathbb{F}_2$-linear queries, and are read-once in the sense that the queries on each path is linearly independent. As their main result, they constructed an explicit function with average-case complexity $2^{n/3-o(n)}$ against a slightly restricted model, which they call strongly read-once linear branching programs. The main tool in their lower bound result is a new type of extractor, called directional affine extractors, that they introduced.

Our main result is an explicit function with $2^{n-o(n)}$ average-case complexity against the strongly read-once linear branching program model, which is almost optimal. This result is based on a new connection from this problem to sumset extractors, which is a randomness extractor model introduced by Chattopadhyay and Li (STOC '16) as a generalization of many other well-studied models including two-source extractors, affine extractors and small-space extractors. With this new connection, our lower bound naturally follows from a recent construction of sumset extractors by Chattopadhyay and Liao (STOC '22). In addition, we show that directional affine extractors imply sumset extractors in a restricted setting. We observe that such restricted sumset sources are enough to derive lower bounds, and obtain an arguably more modular proof of the lower bound by Gryaznov, Pudlák and Talebanfard.

We also initiate a study of pseudorandomness against linear branching programs. Our main result here is a hitting set generator construction against regular linear branching programs with constant width. We derive this result based on a connection to Kakeya sets over finite fields.

1 Introduction

The central goal of complexity theory is to understand the power and limitation of different computation models. Motivated by this goal, it is natural to study the *lower bound problem*: given a computation model and a corresponding complexity measure, can we find an explicit function (e.g. computable in polynomial time) that has large complexity? Researchers have studied this problem on many interesting circuit models such as bounded-depth circuits (AC^0), DeMorgan formula and branching programs, and many interesting results have been found. For example, one of the most notable results in this field is that it requires exponential number of gates to compute parity in AC^0 [Yao85, Has89]. (See the excellent book by Jukna [Juk12] for more about circuit lower bound problems.)

Interestingly, circuit lower bound problems have found interesting connections with *randomness extraction*, another central problem in complexity theory. The theory of randomness extraction is concerned with the following problem: we are given an unknown distribution $\mathbf{X}$ which is guaranteed to have some amount of entropy, and our goal is to find an efficiently computable function Ext, which is called a *randomness extractor*, such that $\text{Ext}(\mathbf{X})$ is (close to) the uniform distribution. Unfortunately, it turns out to be impossible to design extractors in this generality, and a central line of inquiry has been to consider extracting random bits assuming some additional structure on $\mathbf{X}$. Randomness extractors have also found a variety of applications in other areas of theoretical computer science, including proving lower bounds for various

*Supported by NSF CAREER award 2045576

ISSN 1433-8092

computational models. For example, the state-of-the-art lower bound for Boolean circuits is based on affine extractors [LY22], which are extractors that work for weak sources that are uniform over affine subspaces. Affine extractors were also used to obtain almost optimal lower bounds for DNF of parities and parity decision trees [CS16]. As another example, extractors for sources sampled by small-space algorithms [KRVZ06] were shown to be average-case hard against read-once branching program [CGZ22].

The main idea behind this connection is as follows. Suppose one can show that for every function $f : \mathcal{X} \to \{0,1\}$ with small complexity measure, the uniform distribution over the larger pre-image (say, $f^{-1}(0)$) is a source $\mathbf{X}$ with some specific structure. If one can construct an extractor for weak sources with this structure, then $f(\mathbf{X})$ is a constant while $\mathrm{Ext}(\mathbf{X})$ is close to uniform, immediately implying that f and Ext cannot be the same function. In fact, f cannot even approximately compute Ext much better than a random guess, i.e., Ext exhibits *average-case hardness* against f. For instance, to derive average-case lower bounds for parity decision trees, for which it is not hard to see that the pre-image is a disjoint union of affine subspaces, one can choose Ext to be an affine extractor. However, the choice of the extractor is not always obvious. For example, the connection between general Boolean circuits and affine extractors [DK11, FGHK16, LY22] is more non-trivial.

In this paper, we study the lower bound problem for *read-once linear branching programs* [GPT22]. Our main contribution is a new connection between lower bound for read-once linear branching programs and *sumset extractors* [CL16], which we will discuss in later sections.

1.1 Linear branching programs

Read-once linear branching programs (ROLBPs) were first studied by Gryaznov, Pudlák and Talebanfard [GPT22], motivated by its connection to proof complexity. Roughly speaking, a ROLBP is a read-once branching program that can make linear queries. We leave the definition of "read-once" for later and define a linear branching program first.

Definition 1.1 (Linear branching program [GPT22])**.** *A linear branching program on $\mathbb{F}_2^n$ is a directed acyclic graph P with the following properties:*

- *There is only one source s in P.*
- *There are two sinks in P, labeled with 0 and 1 respectively.*
- *Every node v which is not a sink is labeled with a linear function $\ell_v : \mathbb{F}_2^n \to \mathbb{F}_2$. Moreover, there are exactly two outgoing edges from v, one is labeled with 1 and the other is labeled with 0.*

The size *of P is the number of non-sink nodes in P. We say P computes a boolean function $f : \mathbb{F}_2^n \to \{0,1\}$ in the following way. For every input $x \in \mathbb{F}_2^n$, we define the* computation path *of x as starting from s, and when on a node v which is not a sink, moving to the next node following the edge with label $\ell(x) \in \{0,1\}$. We repeat this process until the path ends at a sink. $f(x)$ is defined to be the label on this sink.*[1]

The most natural definition of "read-once" for a linear branching program is that the queries made on every path is linearly independent. In this paper, we focus on a more restricted model called *strongly read-once.*

Definition 1.2 (Strongly Read-Once [GPT22])**.** *For every node v in a branching program P, define Pre_v to be the span of all queries that appear on any path from the source to v, and Post_v to be the span of all queries that appear on any path from v to a sink. (For every non-sink node v, both Pre_v and Post_v include ℓ_v. For any sink w we define $\mathsf{Post}_w = \{0\}$.) We say P is* strongly read-once *if the following two properties hold.*

- *For every edge $e = (u \to v)$, $\mathsf{Pre}_u \cap \mathsf{Post}_v = \{0\}$.*
- *For every non-sink node v, $\mathsf{Pre}_v \cap \mathsf{Post}_v = \{0, \ell_v\}$.*

[1] In this paper, we sometimes abuse notation and also use P to denote the function computed by P.

As pointed out in [GPT22], although being more restricted than the natural definition of read-once, strongly read-once linear branching programs still generalize two important models: parity decision trees and read-once branching programs.[2] A parity decision tree is a decision tree which can make linear queries. This model was first defined by Kushilevitz and Mansour [KM93] for its connection with Fourier analysis, and has recently received attention because of its connections to special cases of the log-rank conjecture in communication complexity [TWXZ13, HHL18] and quantum query complexity [GTW21]. A read-once branching program is another generalization of decision tree such that different paths can share nodes, and can be used to model streaming algorithms and randomized small-space algorithms. Similar to how decision trees are generalized to PDT, it is natural to study ROBPs with linear queries.

The lower bound problem we are trying to answer is the following:

Question 1.3. *For a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, let $\mathsf{ROLBP}(f)$ denote the smallest possible size of a strongly read-once linear branching program that computes f. Can we find an explicit function f which is computable in polynomial time such that $\mathsf{ROLBP}(f)$ is as large as possible?*

Note that every function f has a trivial size upper bound $\mathsf{ROLBP}(f) \le 2^n$ (e.g. a trivial decision tree of depth n), so our goal is to find a function f such that $\mathsf{ROLBP}(f)$ is as close to 2^n as possible. We are also interested in answering the average-case lower bound problem:

Question 1.4. *For a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ and any $\varepsilon > 0$, let $\mathsf{ROLBP}_\varepsilon(f)$ denote the smallest size of strongly read-once linear BP P such that*

$$\Pr_{x \sim \mathbb{F}_2^n} [P(x) = f(x)] \ge \frac{1}{2} + \varepsilon.$$

Can we find a function f which is computable in polynomial time such that $\mathsf{ROLBP}_\varepsilon(f)$ is as large as possible?

1.2 Prior work

To obtain a lower bound for strongly read-once linear branching programs, [GPT22] introduced a new type of extractor called as *directional affine extractors*. (We refer the reader to Section 2 for standard notation in the context of extractors.)

Definition 1.5 (Directional Affine Extractor [GPT22])**.** *We say* $\mathrm{DAExt} : \mathbb{F}_2^n \to \mathbb{F}_2$ *is a (d, ε)-directional affine extractor if for any distribution $\mathbf{X} \in \mathbb{F}_2^n$ which is uniform over an affine subspace of dimension d, and any non-zero vector $a \in \mathbb{F}_2^n$, it holds that*

$$\mathrm{DAExt}(\mathbf{X} + a) + \mathrm{DAExt}(\mathbf{X}) \approx_\varepsilon \mathbf{U}_1.$$

[GPT22] proved that a directional affine extractor for small dimension has a large average-case lower bound for strongly read-once linear BP.

Theorem 1.6 ([GPT22, Theorem 16])**.** *Let* DAExt *be a (d, ε)-directional affine extractor. Then*

$$\mathsf{ROLBP}_{\sqrt{2\varepsilon}}(\mathrm{DAExt}) \ge \varepsilon 2^{n-d-1}.$$

In [GPT22], they constructed a directional affine extractor for dimension $(2/3 + o(1))n$, which implied a $2^{n/3 - o(n)}$ average-case lower bound for ROLBPs. A natural open question left in [GPT22] was to construct a directional affine extractor for dimension $d = o(n)$, which would directly imply a $2^{n-o(n)}$ average-case lower bound for ROLBPs. However, this seems like a challenging problem. Indeed, even constructing affine extractors for dimension $d = o(n)$ has been a difficult task that has been recently resolved [Li16, CGL21]; a directional affine extractor is an affine extractor with additional *non-malleable* properties (see Appendix B) and it is not clear how to use known techniques to construct such extractors for low dimension.

[2] This statement was not proved in [GPT22]. For completeness, we include a proof in Appendix A. Furthermore, our definition of strongly read-once is slightly more general than the original one in [GPT22]. We do not view it as a crucial difference, and we choose this definition merely for cleaner notation in the proof. We also discuss the difference in Appendix A.

1.3 Our results

In this work, we take a different approach and show that to get an average-case lower bound strongly read-once linear BP, it suffices to construct a *sumset extractor*. Informally, a sumset extractor is a function that can extract uniform randomness from sum of two independent weak sources (such sources are called sumset sources). The formal definition of sumset extractors is as follows.[3]

Definition 1.7 (Sumset Extractor [CL16]). *A function* $\mathrm{SumExt} : \mathbb{F}_2^n \to \{0,1\}$ *is a* (k_1, k_2, ε)*-sumset extractor if for any two independent distributions* $\mathbf{A}, \mathbf{B}$ *on* $\mathbb{F}_2^n$ *with* $\mathrm{H}_\infty(\mathbf{A}) \geq k_1$ *and* $\mathrm{H}_\infty(\mathbf{B}) \geq k_2$*,*

$$\mathrm{SumExt}(\mathbf{A} + \mathbf{B}) \approx_\varepsilon \mathbf{U}_1.$$

Our main theorem is as follows:

Theorem 1. *Let* SumExt *be a* (k_1, k_2, ε)*-sumset extractor. Then*

$$\mathsf{ROLBP}_{9\varepsilon}(\mathrm{SumExt}) > 2^{n-k_1-k_2-2}.$$

Sumset extractors were first introduced by Chattopadhyay and Li [CL16] as a "unified" extractor model for many other important extractor problems such as two-source extractors, affine extractors and small-space extractors. (We refer the reader to [CL22] for a more elaborate discussion on sumset extractors.) A recent work [CL22] gave an explicit construction of sumset extractors for polylogarithmic entropy.

Theorem 1.8 ([CL22]). *There is a* $(\mathrm{polylog}(n), \mathrm{polylog}(n), n^{-\Omega(1)})$*-sumset extractor that can be computed in polynomial time.*

Plugging this extractor into Theorem 1, we improve the best lower bound for strongly read-once linear BP from $2^{n/3-o(n)}$ to $2^{n-o(n)}$, which is almost optimal.

Theorem 2. *There is a function* SumExt *which can be computed in polynoimal time such that*

$$\mathsf{ROLBP}_{n^{-\Omega(1)}}(\mathrm{SumExt}) > 2^{n-\mathrm{polylog}(n)}.$$

1.4 On average-case lower bound with negligible error

One drawback of the average-case lower bound based on Theorem 1 is that we don't yet know any explicit construction of (k_1, k_2, ε)-sumset extractors such that $k_1 + k_2 \leq n$ and $\varepsilon = n^{-\omega(1)}$.[4] Thus we cannot directly use Theorem 1 to derive non-trivial average-case lower bound in the negligible error setting (for functions in P). Our main insight here is that the proof of Theorem 1 actually shows that it suffices to construct extractors for sumset sources $\mathbf{A} + \mathbf{B}$ with two additional properties, that we describe below.

Theorem 3. *Let* $\mathrm{SumExt}' : \mathbb{F}_2^n \to \{0,1\}$ *be a function such that* $\mathrm{SumExt}'(\mathbf{A}+\mathbf{B}) \approx_\varepsilon \mathbf{U}_1$ *for any independent distributions* $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$ *which satisfy* $\mathrm{H}_\infty(\mathbf{A}) \geq k_1, \mathrm{H}_\infty(\mathbf{B}) \geq k_2$ *and the following two additional properties.*

- $\mathbf{B}$ *is* almost affine*: the span of* $\mathrm{Supp}(\mathbf{B})$ *is of dimension* $\leq k_2 + 1$*.*
- $\mathbf{A}$ *and* $\mathbf{B}$ *have* non-intersecting span*:* $\mathrm{span}(\mathrm{Supp}(\mathbf{A})) \cap \mathrm{span}(\mathrm{Supp}(\mathbf{B})) = \{0\}$*.*

Then

$$\mathsf{ROLBP}_{9\varepsilon}(\mathrm{SumExt}') > 2^{n-k_1-k_2-2}.$$

We show two different extractor constructions that utilize the first and second property, respectively. Our first construction is exactly the directional affine extractors in [GPT22]. Our main observation is that directional affine extractors can extract from the restricted class of sumset sources with the almost affine property in Theorem 3, which gives an alternative proof of the average-case lower bound in [GPT22] (Theorem 1.6).

[3] For simplicity, we present the definition where the output length of the extractor is just 1 bit.

[4] A Paley graph extractor [CG88] with proper choice of parameters is actually a (k_1, k_2, ε)-sumset extractor for $k_1 + k_2 = (\frac{1}{2} + \gamma)n$ and negligible ε, for any constant $\gamma > 0$. (See [CZ16, Theorem 4.2].) However, it is not known how to compute such an extractor in polynomial time.

Furthermore, the proof of this statement is just a simple application of leftover hash lemma [ILL89]. We view this as a more modular proof of the lower bound result in [GPT22].

We note that a directional affine extractor is a stronger notion that sumset extractors with the almost affine property. Indeed given any sumset extractor, one can modify it to get a new sumset extractor so that the extractor ignores its first bit of input; however it is easy to see that the modified sumset extractor is not a directional affine extractor (see Remark 4.6). Thus, it could be an easier task to build sumset extractors with the almost affine property.

Our second construction is based on the interleaved-source extractor constructed in [CL20]. An interleaved source is a source over $\{0,1\}^{2n}$ of the form $(\mathbf{A} \circ \mathbf{B})_\pi$, where $\mathbf{A}, \mathbf{B}$ are independent sources over $\{0,1\}^n$, and π is a fixed but unknown permutation of the $2n$ bits. An interleaved source is a special case of sumset sources (see [CL16]). We observe that their extractor construction can be extended to work for sumset sources with the non-intersecting span property. In fact, we prove a slightly more general result.

Theorem 4. *For every constant $\delta > 0$, there is a function $\mathrm{ILExt} : \mathbb{F}_2^n \to \{0,1\}$ computable in polynomial time such that for every independent sources $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$ which satisfy $\mathrm{H}_\infty(\mathbf{A}) \geq (\frac{1}{3}+\delta)n$, $\mathrm{H}_\infty(\mathbf{B}) \geq (\frac{1}{3}+\delta)n$ and $\mathrm{H}_\infty(\mathbf{A}+\mathbf{B}) \geq (\frac{2}{3}+2\delta)n$,*

$$\mathrm{ILExt}(\mathbf{A}+\mathbf{B}) \approx_{2^{-\Omega(n)}} \mathbf{U}_1.$$

It's not hard to see that the additional entropy requirement on $\mathbf{A}+\mathbf{B}$ is implied by the non-intersecting span property, and hence we can apply Theorem 3 on the extractor in Theorem 4. Interestingly, while the two constructions are very different, they give the same average-case lower bound as in [GPT22].

Corollary 1.9. *For every constant $\delta > 0$, there exists a constant $\gamma > 0$ and a function $f : \mathbb{F}_2^n \to \{0,1\}$ computable in polynomial time such that*

$$\mathsf{ROLBP}_{2^{-\gamma n}}(f) > 2^{(1/3-\delta)n}.$$

1.5 Pseudorandomness against linear branching programs

Motivated by close connections between hardness and pseudorandomenss [NW94], we initiate the study of obtaining pseudorandomness results against linear branching programs. Generally speaking, in the pseudorandomness problem for a function class $\mathcal{F}$, our goal is to construct a *pseudorandom distribution* which can be generated with only $r \ll n$ random bits but is indistinguishable from the n-bit uniform distribution $\mathbf{U}_n$ by any function in $\mathcal{F}$. We now formally define a hitting set generator (HSG), which is the one-sided variant of a pseudorandom generator (PRG).

Definition 1.10 (Hitting Set Generators)**.** *We say a set $H \subseteq \{0,1\}^n$ is a* hitting *set with error ε for a class of functions $\mathcal{F}$ (on n-bit input), if for every function $f : \{0,1\}^n \to \{0,1\}$ in $\mathcal{F}$ such that $\Pr_{x \sim \mathbf{U}_n}[f(x) = 1] \geq \varepsilon$, there exists $h \in H$ such that $f(h) = 1$. Moreover, $G : \{0,1\}^d \to \{0,1\}^n$ is called a* hitting set generator *(HSG) with error ε for a class of functions $\mathcal{F}$ if $\{G(s)\}_{s \in \{0,1\}^d}$ is a hitting set for $\mathcal{F}$, and s is called the seed length of G.*

Constructing good HSGs for (standard) read-once branching programs is a central problem in complexity theory. If one can construct an explicit HSG with seed length $O(\log(n))$ and $O(1)$ error for ROBPs of size $\mathrm{poly}(n)$, this would imply $\mathbf{RL} = \mathbf{L}$, which is a major open problem in complexity theory. Interestingly, it was recently shown [CH20] that HSGs suffice to even derandomize **BPL**.

We note that for the above derandomization applications, it suffices to construct a HSG for oblivious ROBPs with ordered input. That is, given an n-bit input $x = (x_1, \ldots, x_n)$, the ROBPs read the bits $x_1, x_2, \ldots, x_n$ in order, regardless of what x is. For oblivious ROBPs with ordered input, the best known construction is due to Nisan [Nis92] that has seed length $O(\log^2(n))$ (which in fact is a pseudorandom generator). However, in spite of the improvement in several restricted sub-classes of ROBPs, Nisan's result remains the best known construction in the general setting after three decades of work.

A recent research direction has been to find approaches that are completely different from Nisan's construction. In this direction, researchers considered the task of constructing PRGs (and HSGs) for a natural generalization of ROBPs called as (oblivious) unordered ROBPs for which it is known that Nisan's construction fails to work [Tzu09]. An unordered ROBPs still read the bits of x in a fixed order that does not depend on x, but this order is unknown. By an impressive line of work culminating with a beautiful construction by Forbes and Kelly [FK18], we now have explicit PRGs with seed length $O(\log^3(n))$ for unordered ROBPs. The general approach used to construct PRGs for this model is based on analyzing the effects of random restrictions on ROBPs and leveraging bounds on the Fourier spectrum of branching programs [RSV13, CHRT18].

This gives us further motivation to study pseudorandomness against *oblivious ROLBPs*, which is a further generalization of unordered ROBPs.

Definition 1.11 (Oblivious ROLBPs). *We say a read-once linear branching programs P on input $\mathbb{F}_2^n$ is oblivious if the nodes can be divided into layers $L_0, \ldots, L_n$ such that*

- *L_0 only contains the source, and L_n consists of all the sinks.*
- *For every $0 \leq i < n$, every edge from nodes in L_i connects to a node in L_{i+1}.*
- *For every $0 \leq i < n$, every node on L_i is labeled with the same linear query ℓ_i.*
- *$(\ell_0, \ldots, \ell_{n-1})$ is a basis of $\mathbb{F}_2^n$.*

The width *of P is defined as* $\max_{i \in [n]}(|L_i|)$.

We note that unordered ROBPs correspond to the case of $(\ell_0, \ldots, \ell_{n-1})$ being a permutation of the standard basis. Thus, Nisan's PRG construction fails to work for oblivious ROLBPs. Further, it is not clear how to use the techniques of random restriction based constructions employed for unordered ROBPs when the layers can be arbitrary linear functions. Thus, it looks like we need new ideas to obtain pseudorandomness against oblivious ROLBs.

Our first observation is that the case of width $w = 2$ is easy since it is well known that a small-biased distribution [NN90, AGHP92, TS17] fools such programs.[5] This follows since small-biased distributions are invariant under full-rank linear transformations. Further, [BDVY13] proved that sum of small-biased distributions fools width-2 ROBPs that reads multiple bits. Thus, one can obtain a similar result for the linear analogue of these programs. It has been asked by Vadhan and Reingold (see [LV17]) whether sums of small-biased distributions can be employed to construct PRGs (or HSG) for general ROBPs. Indeed a positive answer to this question would immediately imply a PRGs (or HSG) for oblivious ROLBs. We are not able to resolve this conjectured approach, and take a different route that we describe below.

We take an initial step towards constructing HSGs against oblivious ROLBPs of width more than 2, and focus on the sub-class of regular oblivious ROLBPs. A regular linear branching program is a linear branching program in which every non-source node has in degree 2. We note that the sub-class of regular (standard and unordered) ROBPs have been well-studied [BRRY14, RSV13, BHPP22]. In fact, a recent result [BHPP22] proved that obtaining a HSG against regular ROBPs would imply a HSG with similar parameters against all ROBPs.

As our main result here, we construct a *hitting set generator* with $(1 - \Omega(1))n$ seed length for *regular* oblivious ROLBPs with *constant width*.

Theorem 5. *For every $w \in \mathbb{N}$, there is an explicit construction of HSG for regular oblivious ROLBPs of width w with seed length $(w-1) + \lceil (1 - 2^{-(w-1)})n \rceil$.*

Interestingly, our construction is based on a well-studied problem called rank-r Kakeya set [EOT10, KLSS11], which is a set that contains a r-dimensional affine subspace in every direction.

Definition 1.12. *A set $K \subseteq \mathbb{F}_2^n$ is called a* rank-r Kakeya set *(over $\mathbb{F}_2^n$) if for every r-dimension subspace $V \subseteq \mathbb{F}_2^n$, there exists $b \in \mathbb{F}_2^n$ such that $V + b \subseteq K$.*

We prove the following theorem.

Theorem 6. *A rank-r Kakeya set is a hitting set for oblivious read-once regular linear BP of width $(r+1)$.*

To get an efficiently computable HSG, we take the following simple construction of rank-r Kakeya set constructed by Kopparty, Lev, Saraf and Sudan [KLSS11].

[5] this result is due to Saks and Zuckerman, see [BDVY13] for sketch of a proof

Theorem 1.13 ([KLSS11]). *For every $r, n \in \mathbb{N}$ s.t. $r \le n$, there is an explicit construction of rank-r Kakeya set $K_{n,r} \subseteq \mathbb{F}_2^n$ with size at most $2^{\lceil (1-2^{-r})n \rceil + r}$, which is defined as follows. Let $I_1, \ldots, I_{2^r}$ be a parition of $[n]$, each having size at least $\lfloor 2^{-r} n \rfloor$. Then*

$$K_{n,r} = \bigcup_{t=1}^{2^r} \operatorname{span}\left(\{e_i\}_{i \notin I_i}\right).^{6}$$

In other words, $K_{n,r}$ is the union of 2^r boolean subcubes where the i-th subcube contains every point $x \in \mathbb{F}_2^n$ such that the x_{I_i} is 0.

To prove Theorem 5, observe that we can construct an efficient HSG with seed length $r + \lceil (1-2^{-r})n \rceil$ that uses the first r bits to select a set I_i and use the remaining $\lceil (1-2^{-r})n \rceil \ge n - |I_i|$ bits to choose a point in the subcube corresponding to I_i.

We note our approach based on Kakeya set does not seem to extend beyond regular ROLBPs. For non-regular oblivious ROLBPs, we observe that the construction in Theorem 1.13 is not a hitting set for width 3, because a read-once CNF $\bigwedge_{t=1}^{2^r} (\bigvee_{i \in I_t} x_i)$ always outputs 0 on $K_{n,r}$, and a read-once CNF can be computed by a width-3 ROBP.

Further, while one might hope to extend our result to larger width (for regular ROLBPs) with a better construction of Kakeya sets, we show that the construction in Theorem 1.13 is essentially optimal. This negative result also answers an open question in [KLSS11] (for the case of $\mathbb{F}_2^n$), where they asked whether there is a better construction of rank-r Kakeya sets than Theorem 1.13. This lower bounds may be of independent interest.

Theorem 7. *Every rank-r Kakeya set over $\mathbb{F}_2^n$ has size at least $2^{(1-2^{-r})(n+2)-r}$.*

Organization. We introduce preliminaries in Section 2. We prove Theorem 1 (and Theorem 3, which is a stronger version of Theorem 1) in Section 3. We discuss average-case lower bound results based on Theorem 3 in Section 4. We prove our results about HSGs and Kakeya sets (Theorem 6 and Theorem 7) in Section 5. Finally, we discuss some future directions in Section 6.

2 Preliminaries

2.1 Notation

Distributions and random variables. We sometimes abuse notation and treat distributions and random variables as the same. We always write a random variable/distribution in boldface font. Every log in this paper is of base 2 unless specified. We use $\mathrm{Supp}(\mathbf{X})$ to denote the support of a distribution. We use $\mathbf{U}_n$ to denote the uniform distribution on $\{0,1\}^n$. When $\mathbf{U}_n$ appears with other random variables in the same joint distribution, $\mathbf{U}_n$ is considered to be independent of other random variables. When there is a sequence of random variables $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_t$ in the context, for every set $S \subseteq [t]$ we use $\mathbf{X}_S$ to denote the sequence of random variables which use indices in S as subscript, i.e. $\mathbf{X}_S := \{\mathbf{X}_i\}_{i \in S}$.

Notation for $\mathbb{F}_2^n$. Throughout this paper, we treat $\mathbb{F}_2^n$ and $\{0,1\}^n$ as the same. We use $e_i \subseteq \mathbb{F}_2^n$ to denote the i-th standard basis vector, which as its i-th coordinate being 1 and other coordinates being 0. We sometimes use a vector $\ell \in \mathbb{F}_2^n$ to represents a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$ defined as $f(x) = \langle \ell, x \rangle$.

[6] Here e_i denote a the i-th standard basis vector in $\mathbb{F}_2^n$ which has its i-th coordinate being 1 and other coordinates being 0.

2.2 Statistical Distance

Definition 2.1. *Let* $\mathbf{D}_1, \mathbf{D}_2$ *be two distributions on the same universe* Ω. *The* statistical distance *between* $\mathbf{D}_1$ *and* $\mathbf{D}_2$ *is*

$$\begin{aligned}\Delta(\mathbf{D}_1; \mathbf{D}_2) &\coloneqq \max_{T \subseteq \Omega} \Big(\Pr[\mathbf{D}_1 \in T] - \Pr[\mathbf{D}_2 \in T] \Big) \\ &= \frac{1}{2} \sum_{s \in \Omega} |\mathbf{D}_1(s) - \mathbf{D}_2(s)|\,.\end{aligned}$$

We say $\mathbf{D}_1$ *is* ε*-close to* $\mathbf{D}_2$ *if* $\Delta(\mathbf{D}_1; \mathbf{D}_2) \leq \varepsilon$, *which is also denoted by* $\mathbf{D}_1 \approx_\varepsilon \mathbf{D}_2$. *When there are two joint distributions* $(\mathbf{X}, \mathbf{Z})$ *and* $(\mathbf{Y}, \mathbf{Z})$ *such that* $(\mathbf{X}, \mathbf{Z}) \approx_\varepsilon (\mathbf{Y}, \mathbf{Z})$, *we write* $(\mathbf{X} \approx_\varepsilon \mathbf{Y}) \mid \mathbf{Z}$ *for short.*

Throughout this paper, we frequently use the following standard properties without explicit referencing.

Lemma 2.2. *For every distribution* $\mathbf{D}_1, \mathbf{D}_2, \mathbf{D}_3$ *on the same universe, the following properties hold:*

- *For every function* f, $\Delta(f(\mathbf{D}_1); f(\mathbf{D}_2)) \leq \Delta(\mathbf{D}_1; \mathbf{D}_2)$.
- *(Triangle inequality)* $\Delta(\mathbf{D}_1; \mathbf{D}_3) \leq \Delta(\mathbf{D}_1; \mathbf{D}_2) + \Delta(\mathbf{D}_2; \mathbf{D}_3)$.
- *For any distribution* $\mathbf{Z}$,

$$\Delta((\mathbf{D}_1, \mathbf{Z}); (\mathbf{D}_2, \mathbf{Z})) = \mathop{\mathbb{E}}_{z \sim \mathbf{Z}} \left[\Delta(\mathbf{D}_1|_{\mathbf{Z}=z}; \mathbf{D}_2|_{\mathbf{Z}=z})\right].$$

- *(Markov argument) For any distribution* $\mathbf{Z}$, *if* $(\mathbf{D}_1 \approx_\varepsilon \mathbf{D}_2) \mid \mathbf{Z}$, *then*

$$\Pr_{z \sim \mathbf{Z}} \left[\mathbf{D}_1|_{\mathbf{Z}=z} \approx_{\sqrt{\varepsilon}} \mathbf{D}_2|_{\mathbf{Z}=z}\right] \geq 1 - \sqrt{\varepsilon}$$

2.3 Conditional Min-entropy

First we define (worst-case) conditional min-entropy.

Definition 2.3. *For a joint distribution* $(\mathbf{X}, \mathbf{Z})$, *the* conditional min-entropy *of* $\mathbf{X}$ *given* $\mathbf{Z}$ *is*

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) := \min_{z \in \mathrm{Supp}(\mathbf{Z})} \mathrm{H}_\infty(\mathbf{X}|_{\mathbf{Z}=z}).$$

A more fine-grained definition of conditional min-entropy called *average conditional min-entropy* was introduced in [DORS08].

Definition 2.4 ([DORS08])**.** *For a joint distribution* $(\mathbf{X}, \mathbf{Z})$, *the* average conditional min-entropy *of* $\mathbf{X}$ *given* $\mathbf{Z}$ *is*

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) := -\log\left(\mathop{\mathbb{E}}_{z \sim \mathbf{Z}} \left[\max_x (\Pr[\mathbf{X} = x \mid \mathbf{Z} = z])\right]\right).$$

For average conditional min-entropy we have the following nice property called *chain rule*:

Lemma 2.5 ([DORS08])**.** *Let* $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$ *be (correlated) random variables such that* $\mathrm{Supp}(\mathbf{Y}|_{\mathbf{Z}=z}) \leq 2^\lambda$ *for every* $z \in \mathrm{Supp}(\mathbf{Z})$. *Then*

$$\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid (\mathbf{Y}, \mathbf{Z})) \geq \widetilde{\mathrm{H}}_\infty((\mathbf{X}, \mathbf{Y}) \mid \mathbf{Z}) - \lambda \geq \widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) - \lambda.$$

The average conditional min-entropy can be converted into worst-case conditional min-entropy with the following lemma.

Lemma 2.6 ([DORS08, MW97])**.** *Let* $\mathbf{X}, \mathbf{Z}$ *be (correlated) random variables. For every* $\varepsilon > 0$,

$$\Pr_{z \sim \mathbf{Z}} \left[\mathrm{H}_\infty(\mathbf{X}|_{\mathbf{Z}=z}) \geq \widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{Z}) - \log(1/\varepsilon)\right] \geq 1 - \varepsilon.$$

2.4 Extractors

First we define a more general form of seeded extractors. (In the standard definition of seeded extractor, we consider $\mathbf{Y}$ to be the uniform distribution over $\mathcal{S}$.)

Definition 2.7. *Let $\mathcal{X}, \mathcal{S}$ be two finite sets. Let $\mathbf{Y}$ be a distribution over $\mathcal{S}$. We say $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}^m$ is a (k, ε)-extractor with seed $\mathbf{Y}$ if for every distribution $\mathbf{X} \in \mathcal{X}$ independent of $\mathbf{Y}$ such that $\mathrm{H}_\infty(\mathbf{X}) \geq k$,*

$$\mathrm{Ext}(\mathbf{X}, \mathbf{Y}) \approx_\varepsilon \mathbf{U}_m.$$

Furthermore, we say Ext is strong in $g(\mathbf{Y})$ for some deterministic function g if

$$(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}) \approx_\varepsilon \mathbf{U}_m) \mid g(\mathbf{Y}).$$

When Ext is strong in $\mathbf{Y}$ we simply say Ext is strong.

For strong seeded extractor we have the following standard lemma.

Lemma 2.8. *Suppose $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}^m$ is a (k, ε)-strong extractor with seed $\mathbf{Y}$, where $\mathbf{Y}$ is the uniform distribution over a set $S \subseteq \mathcal{S}$. Then for every $\mathbf{Y}'$ such that $\mathrm{Supp}(\mathbf{Y}') \subseteq S$ and $\mathrm{H}_\infty(\mathbf{Y}') \geq \mathrm{H}_\infty(\mathbf{Y}) - \Delta$, Ext is a $(k, 2^\Delta \varepsilon)$-strong extractor with seed $\mathbf{Y}'$.*

We need the following form of *leftover hash lemma*. This is more general than the original lemma in [ILL89], but is also standard in the literature. (See, e.g., [Vad12, Problem 6.3].)

Lemma 2.9 (Leftover Hash Lemma [ILL89])**.** *Consider any $h : \{0,1\}^n \times \mathcal{S} \to \{0,1\}^m$ and any distribution $\mathbf{Y} \in \mathcal{S}$ such that for every distinct $x_1, x_2 \in \{0,1\}^n$, $\Pr_{y \sim \mathbf{Y}}[h(x_1, y) = h(x_2, y)] \leq (1+\varepsilon)2^{-m}$. (We say h is ε-almost universal over randomness $\mathbf{Y}$ if h and $\mathbf{Y}$ satisfy the condition above.) Then h is a strong $(m + \log(1/\varepsilon), \sqrt{\varepsilon/2})$-extractor with seed $\mathbf{Y}$.*

We will also use the following lemma for seeded extractors on conditional min-entropy from [Vad12, Problem 6.8]. We need a more general form which works for the general seeded extractors defined above. For completeness we include a proof in Appendix C. (In the standard form of the following lemma, $\mathbf{Y}$ is a uniform over $\mathcal{S}$, and $\mathcal{X}_e = \mathcal{X}$ for every e.)

Lemma 2.10. *Let $(\mathbf{X}, \mathbf{Y}, \mathbf{E})$ be a joint distribution such that $\mathbf{X} \in \mathcal{X}$ and $\mathbf{Y} \in \mathcal{S}$ are independent conditioned on $\mathbf{E}$, and $\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{E}) \geq k$. Let $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}^m$ be a function which satisfies the following conditions for an error parameter $\varepsilon > 0$ and a deterministic function g: for every $e \in \mathrm{Supp}(\mathbf{E})$, there exists a set $\mathcal{X}_e \subseteq \mathcal{X}$ with size at least 2^{k+1} such that Ext when restricted to the domain $\mathcal{X}_e \times \mathcal{S}$ is a (k, ε)-extractor with seed $\mathbf{Y}|_{\mathbf{E}=e}$ and is strong in $g(e, \mathbf{Y})$. Then*

$$(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}) \approx_{3\varepsilon} \mathbf{U}_m) \mid (\mathbf{E}, g(\mathbf{E}, \mathbf{Y})).$$

3 Linear BP lower bounds based on sumset extractors

In this section, we prove Theorem 3 that we restate below. We note that Theorem 1 follows as a special case of this theorem.

Theorem 3 (restated)**.** *Let $\mathrm{SumExt}' : \mathbb{F}_2^n \to \{0,1\}$ be a function such that $\mathrm{SumExt}'(\mathbf{A} + \mathbf{B}) \approx_\varepsilon \mathbf{U}_1$ for any independent distributions $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$ which satisfy $\mathrm{H}_\infty(\mathbf{A}) \geq k_1, \mathrm{H}_\infty(\mathbf{B}) \geq k_2$ and the following two additional properties.*

- $\mathbf{B}$ *is* almost affine*: the span of $\mathrm{Supp}(\mathbf{B})$ is of dimension $\leq k_2 + 1$.*
- $\mathbf{A}$ *and* $\mathbf{B}$ *have* non-intersecting span*: $\mathrm{span}(\mathrm{Supp}(\mathbf{A})) \cap \mathrm{span}(\mathrm{Supp}(\mathbf{B})) = \{0\}$.*

Then

$$\mathsf{ROLBP}_{9\varepsilon}(\mathrm{SumExt}') > 2^{n-k_1-k_2-2}.$$

We first discuss the main ideas behind the proof before formally proving it. Given a read-once linear BP $P : \mathbb{F}_2^n \to \{0,1\}$ and any $b \in \{0,1\}$, the uniform distribution over the pre-image $P^{-1}(b)$ corresponds to the uniform distribution over all the computation path from the source s to the sink labeled b. For every edge e, whether a computation path pass goes through e and ends at a sink labeled b can be divided into two events: whether a path starting from s would reach e, and whether a path staring from e would end at a sink labeled b. The strongly read-once property guarantees that we can divide $\mathbb{F}_2^n$ into two complemented subspaces V_A, V_B such that the first event is determined by the projection of the input $x \in \mathbb{F}_2^n$ on V_A, and the second event is determined by the projection of x on V_B. Given a uniform input $\mathbf{X} \in \mathbb{F}_2^n$, the two projections are independent. Therefore, conditioned on the computation path passing through e and end at a sink labeled b, $\mathbf{X}$ can be written as the sum of two independent sources $\mathbf{A}+\mathbf{B}$, where $\mathrm{Supp}(\mathbf{A}) \subseteq V_A$ and $\mathrm{Supp}(\mathbf{B}) \subseteq V_B$. It remains to choose a cut E such that for every choice of $e \in E$, the two sources $\mathbf{A}, \mathbf{B}$ stated above both have enough entropy.

We formalize the ideas above as the following structural lemma:

Lemma 3.1. *Let $\mathbf{X}$ be a uniform random variable over $\mathbb{F}_2^n$. For every strongly read-once linear BP $P : \mathbb{F}_2^n \to \{0,1\}$ of size s and every $d \in [n]$, there exists a random variable $\mathbf{E}$, and random variables $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$, s.t.*

- *$\mathbf{E}$ has support size at most $2s$.*
- *$\mathbf{X} = \mathbf{A} + \mathbf{B}$*
- *For every $e \in \mathrm{Supp}(\mathbf{E})$, define $\mathbf{A}_e = \mathbf{A}|_{\mathbf{E}=e}$ and $\mathbf{B}_e = \mathbf{B}|_{\mathbf{E}=e}$. Then we have*
  - *$\mathbf{A}_e$ and $\mathbf{B}_e$ are independent.*
  - *$\mathbf{B}_e$ is uniform over an affine subspace V_e^B of dimension d*
  - *There exists a complemented subspace V_e^A of V_e^B such that $\mathbf{A}_e \in V_e^A$*
- *There is a deterministic function g s.t. $g(\mathbf{E}, \mathbf{B}) = f(\mathbf{X})$.*

Proof. We show that there exist some functions E, A, B s.t. $\mathbf{E} = E(\mathbf{X}), \mathbf{A} = A(\mathbf{X}), \mathbf{B} = B(\mathbf{X})$ satisfy the above claim. Fix any $x \in \mathbb{F}_2^n$. Consider the computation path of x, and let v be the first node on this path which satisfies that $\dim(\mathsf{Post}_v) \le d$. Note that v is well-defined because the last node w on this path satisfies $\dim(\mathsf{Post}_w) = 0 \le d$. Then we define $E(x) := (u \to v)$ to be the edge right before v in this path. (If v is the source, we define u to be a dummy node $\perp$, and define $\mathsf{Pre}_\perp = \{0\}$.) First we claim that $\dim(\mathsf{Pre}_u) \le n-d$. If $u = \perp$ then the claim is trivially true. Otherwise, observe that $\dim(\mathsf{Post}_u) \ge d+1$ by the definition of v, and by the strongly read-once property we have

$$\dim(\mathsf{Pre}_u) \le n + \dim(\mathsf{Pre}_u \cap \mathsf{Post}_u) - \dim(\mathsf{Post}_u) \le n + 1 - (d+1) = n - d.$$

Observe that $\mathsf{Pre}_u \cap \mathsf{Post}_v = \{0\}$ by the strongly read-once property. Now we choose an arbitrary basis $(b_1, \dots, b_n)$ of $\mathbb{F}_2^n$ such that $\mathrm{span}(\{b_i\}_{1 \le i \le \dim(\mathsf{Pre}_u)}) = \mathsf{Pre}_u$ and $\mathrm{span}(\{b_{n-i}\}_{0 \le i < \dim(\mathsf{Post}_v)}) = \mathsf{Post}_v$. Define $\mathsf{Pre}'_u = \mathrm{span}(\{b_i\}_{1 \le i \le n-d})$ and $\mathsf{Post}'_v = \mathrm{span}(\{b_i\}_{n-d < i \le n})$. Note that $\mathsf{Pre}_u \subseteq \mathsf{Pre}'_u$, $\mathsf{Post}_v \subseteq \mathsf{Post}'_v$ and Pre'_u and Post'_v are complemented subspaces. Then define $(A(x), B(x))$ to be the unique pair in $(\mathsf{Post}'_v)^\perp \times (\mathsf{Pre}'_u)^\perp$ s.t. $A(x) + B(x) = x$. It remains to prove that $\mathbf{E} = E(\mathbf{X}), \mathbf{A} = A(\mathbf{X}), \mathbf{B} = B(\mathbf{X})$ satisfy our claim.

First it's easy to see that the support size of $\mathbf{E}$ is upper bounded by $2s$: if the source s satisfies $\dim(\mathsf{Post}_s) \le d$, v is always the source s and $\mathbf{E}$ has support size 1; otherwise $\mathbf{E}$ is an edge in the branching program, and there are at most $2s$ choices. Moreover, $\mathbf{X} = \mathbf{A} + \mathbf{B}$ by definition of A and B. To prove the remaining two claims, consider any possible fixing $\mathbf{E} = e := (u \to v)$. Let $(A_e(x), B_e(x))$ denote the unique pair in $(\mathsf{Post}'_v)^\perp \times (\mathsf{Pre}'_u)^\perp$ s.t. $A_e(x) + B_e(x) = x$. We claim that there exists a set $S \subseteq (\mathsf{Post}'_v)^\perp$ so that $E(x) = e$ if and only if $A_e(x) \in S$. This implies that $(\mathbf{A}, \mathbf{B})|_{\mathbf{E}=e}$ is exactly the uniform distributions over $S \times (\mathsf{Pre}'_u)^\perp$, which satisfies the third claim by taking $V_e^B = (\mathsf{Pre}'_u)^\perp$ and $V_e^A = (\mathsf{Post}'_v)^\perp$. To prove this claim, observe that whether $E(x) = e$ can be decided by the following procedure. We follow the computation path of x, but stop and answer "NO" if we reach any node w such that either w cannot reach u (so that $E(x)$ can never be e regardless of the remaining queries) or $\dim(\mathsf{Post}_w) \le d$ (so that $E(x)$ would be the edge ending at w instead of e). Otherwise, if we reach the edge e we stop and answer "YES". Observe that every linear query ℓ we made in this procedure is in Pre_u. Moreover, for every such query, $\ell(x) = \ell(A_e(x)) + \ell(B_e(x)) = \ell(A_e(x))$ because $B_e(x) \in (\mathsf{Pre}'_u)^\perp \subseteq (\mathsf{Pre}_u)^\perp$. Therefore, the event $E(x) = e$ is completely determined by $A_e(x)$, which proves our claim. Finally, observe that conditioned on $E(x) = e$, the value of $f(x)$ is determined by queries in Post_v, and every such query ℓ satisfies that $\ell(x) = \ell(A_e(x)) + \ell(B_e(x)) = \ell(B_e(x)) = \ell(B(x))$ because $A_e(x) \in (\mathsf{Post}'_v)^\perp \subseteq (\mathsf{Post}_v)^\perp$. Therefore by choosing $g(e, \cdot)$ to be the subprogram of f starting at v, the last condition is also satisfied. □

Now we are ready to prove Theorem 3.

Proof of Theorem 3. Let SumExt' be a function which satisfies the conditions in Theorem 3 with parameters (k_1, k_2, ε), and let $f : \mathbb{F}_2^n \to \{0,1\}$ be any strongly read-once linear BP of size $s = 2^{n-k_1-k_2-2}$. Let $\mathbf{X}$ be a uniform random variable over $\mathbb{F}_2^n$. We want to show that

$$(\mathrm{SumExt}'(\mathbf{X}), f(\mathbf{X})) \approx_{9\varepsilon} (\mathbf{U}_1, f(\mathbf{X})), \tag{1}$$

which would imply $\Pr_{x\sim\mathbf{X}}\left[f(x) = \mathrm{SumExt}'(x)\right] \le \frac{1}{2} + 9\varepsilon$ for every f of size s, and hence $\mathsf{ROLBP}_{9\varepsilon}(\mathrm{SumExt}') > s$.

Let $\mathbf{E}, \mathbf{A}, \mathbf{B}$ be the random variables depending on $\mathbf{X}$ as in Lemma 3.1, by taking $d = k_2 + 1$. Recall that $\mathbf{E}, \mathbf{A}, \mathbf{B}$ have the following properties:

- $\mathbf{E}$ has support size at most $2s$.
- $\mathbf{X} = \mathbf{A} + \mathbf{B}$
- For every $e \in \mathrm{Supp}(\mathbf{E})$, define $\mathbf{A}_e = \mathbf{A}|_{\mathbf{E}=e}$ and $\mathbf{B}_e = \mathbf{B}|_{\mathbf{E}=e}$. Then we have
  - $\mathbf{A}_e$ and $\mathbf{B}_e$ are independent.
  - There exist complemented subspaces V_e^A, V_e^B of dimension $n-d$ and d such that $\mathbf{B}_e$ is uniform over V_e^B and $\mathbf{A}_e \in V_e^A$.
- There is a deterministic function g s.t. $g(\mathbf{E}, \mathbf{B}) = f(\mathbf{X})$.

Therefore we can rewrite Equation (1) as

$$(\mathrm{SumExt}'(\mathbf{A} + \mathbf{B}) \approx_{9\varepsilon} \mathbf{U}_1) \mid g(\mathbf{E}, \mathbf{B}). \tag{2}$$

Consider the function $\mathrm{Ext} : (\mathbb{F}_2^n)^2 \to \{0,1\}$ defined as $\mathrm{Ext}(a,b) = \mathrm{SumExt}'(a+b)$. We claim that for every $e \in \mathrm{Supp}(\mathbf{E})$, Ext restricted on the domain $V_e^A \times \mathbb{F}_2^n$ is a $(k_1, 3\varepsilon)$-extractor with seed $\mathbf{B}_e$ and is strong in $g(e, \mathbf{B}_e)$. This would imply Equation (2) because of the following. Observe that

$$\widetilde{\mathrm{H}}_\infty(\mathbf{A} \mid \mathbf{E}) = \widetilde{\mathrm{H}}_\infty(\mathbf{A} \mid (\mathbf{B}, \mathbf{E})) \ge \widetilde{\mathrm{H}}_\infty((\mathbf{A}, \mathbf{B}) \mid \mathbf{E}) - d \ge (n - \log(2s)) - d \ge k_1,$$

where the first equality is by the fact that $\mathbf{A}$ and $\mathbf{B}$ are independent conditioned on $\mathbf{E}$, and the first and second inequalities are by chain rule (Lemma 2.5). Furthermore, we can w.l.o.g. assume that $k_1 + k_2 \le n-2$ (since otherwise the bound is trivial), and this would imply $|V_e^A| = 2^{n-d} \ge 2^{k_1+1}$. Therefore we can apply Lemma 2.10 on Ext to get Equation (2).

Next we prove the claim. Let $\mathbf{A}' \in V_e^A$ be any distribution such that $\mathrm{H}_\infty(\mathbf{A}') \ge k_1$. By definition of SumExt', we have that for every random variable $\mathbf{B}' \in V_e^B$ such that $\mathrm{H}_\infty(\mathbf{B}') \ge \dim(V_e^B) - 1 = k_2$,

$$\mathrm{SumExt}'(\mathbf{A}' + \mathbf{B}') \approx_\varepsilon \mathbf{U}_1.$$

In other words, the function $\mathrm{Ext}' : V_e^B \times \mathbb{F}_2^n \to \{0,1\}$ defined as $\mathrm{Ext}'(b,a) = \mathrm{SumExt}'(a+b)$ is a (k_2, ε)-extractor with seed $\mathbf{A}'$. By chain rule, $\widetilde{\mathrm{H}}_\infty(\mathbf{B}_e \mid g(e, \mathbf{B}_e)) \ge \mathrm{H}_\infty(\mathbf{B}_e) - 1 = k_2$. Therefore, by Lemma 2.10 we can conclude that

$$(\mathrm{SumExt}'(\mathbf{A}' + \mathbf{B}_e) \approx_{3\varepsilon} \mathbf{U}_1) \mid g(e, \mathbf{B}_e),$$

and this is exactly what we claimed. □

Remark 3.2. *We can also get a shorter proof by applying Lemma 2.6 instead of Lemma 2.10 to deal with average conditional min-entropy. This would incur an extra* $\mathrm{poly}(\varepsilon)$ *factor on* $\mathsf{ROLBP}_{O(\varepsilon)}(\mathrm{SumExt}')$*, which only matters when* $\varepsilon = 2^{-\Omega(n)}$.

4 Average-case lower bound with negligible error

As we discussed in the introduction, Theorem 1 only implies average-case lower bound with polynomially small error because it is not known how to construct a (k_1, k_2, ε)-sumset extractor for entropy $k_1 + k_2 < n$ with negligible error ε. However, we proved a stronger theorem, Theorem 3, which says that we only need an extractor for sumset sources $\mathbf{A} + \mathbf{B}$ with two additional properties:

- $\mathbf{B}$ is almost affine: $\mathrm{Supp}(\mathbf{B})$ is contained in a linear subspace of dimension $\mathrm{H}_\infty(\mathbf{B}) + 1$, and
- $\mathbf{A}$ and $\mathbf{B}$ have non-intersecting span: $\mathrm{span}(\mathrm{Supp}(\mathbf{A})) \cap \mathrm{span}(\mathrm{Supp}(\mathbf{B})) = \{0\}$.

In this section we will see that we only need either of the two properties to prove a $2^{(1/3-\gamma)n}$ average-case lower bound with exponentially small error.

4.1 Sumset extractors for almost affine source

In this section, we show that a directional affine extractor can work for a sumset source $\mathbf{A} + \mathbf{B}$ as long as $\mathbf{B}$ is almost affine. The proof is simply an application of leftover hash lemma (Lemma 2.9).

Lemma 4.1. *Let* $\mathrm{DAExt} : \mathbb{F}_2^n \to \{0,1\}$ *be any* $(d, \varepsilon/2)$*-directional affine extractor. Then for any* $\mathbf{B} \in \mathbb{F}_2^n$ *which is uniform over an affine subspace of dimension* d*, and any* $\mathbf{A} \in \mathbb{F}_2^n$ *independent of* $\mathbf{B}$ *such that* $\mathrm{H}_\infty(\mathbf{A}) \geq \log(1/\varepsilon) + 1$*,*

$$(\mathrm{DAExt}(\mathbf{A} + \mathbf{B}) \approx_{\sqrt{\varepsilon/2}} \mathbf{U}_1) \mid \mathbf{B}.$$

Proof. Observe that for every distinct $a_1, a_2 \in \mathbb{F}_2^n$,

$$\Pr_{b \sim \mathbf{B}}[\mathrm{DAExt}(a_1 + b) = \mathrm{DAExt}(a_2 + b)] = \Pr_{b \sim \mathbf{B}}[(\mathrm{DAExt}(a_1 + b) + \mathrm{DAExt}(a_2 + b)) = 0] \leq \frac{1 + \varepsilon}{2},$$

by definition of $(d, \varepsilon/2)$-directional affine extractor. This means the function $h(a, b) = \mathrm{DAExt}(a + b)$ is ε-almost universal over randomness $\mathbf{B}$. By leftover hash lemma (Lemma 2.9), h is a $(\log(1/\varepsilon) + 1, \sqrt{\varepsilon/2})$-strong extractor with seed $\mathbf{B}$. In other words, for every distribution $\mathbf{A} \in \mathbb{F}_2^n$ independent of $\mathbf{B}$ such that $\mathrm{H}_\infty(\mathbf{A}) \geq \log(1/\varepsilon) + 1$,

$$(\mathrm{DAExt}(\mathbf{A} + \mathbf{B}) \approx_{\sqrt{\varepsilon/2}} \mathbf{U}_1) \mid \mathbf{B}.$$

□

Corollary 4.2. *Let* $\mathrm{DAExt} : \mathbb{F}_2^n \to \{0,1\}$ *be any* $(d, \varepsilon/2)$*-directional affine extractor. Then for any independent distributions* $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$ *such that* $\mathrm{H}_\infty(\mathbf{A}) \geq \log(1/\varepsilon) + 1$*,* $\mathrm{H}_\infty(\mathbf{B}) \geq d - 1$ *and* $\dim(\mathrm{span}(\mathrm{Supp}(\mathbf{B}))) \leq d$*,*

$$\mathrm{DAExt}(\mathbf{A} + \mathbf{B}) \approx_{3\sqrt{\varepsilon/2}} \mathbf{U}_1.$$

Proof. Let V be a linear subspace of dimension d such that $\mathrm{Supp}(\mathbf{B}) \subseteq V$, and let $\mathbf{B}'$ denote the uniform distribution over V. Define $\mathrm{Ext} : (\mathbb{F}_2^n)^2 \to \{0,1\}$ to bet $\mathrm{Ext}(a, b) = \mathrm{DAExt}(a + b)$. By Lemma 4.1, Ext is a strong $(\log(1/\varepsilon) + 1, \sqrt{\varepsilon/2})$-extractor with seed $\mathbf{B}'$. Since $\mathrm{H}_\infty(\mathbf{B}) \geq d - 1 = \mathrm{H}_\infty(\mathbf{B}') - 1$, by Lemma 2.8, Ext is a strong $(\log(1/\varepsilon) + 1, 3\sqrt{\varepsilon/2})$-extractor with seed $\mathbf{B}$, which is exactly what we want to prove. □

Apply Theorem 3 on Corollary 4.2 by taking $k_1 = \log(1/\varepsilon) + 1$ and $k_2 = d - 1$, we get an alternative proof of [GPT22, Theorem 16].

Theorem 4.3. *If* DAExt *is a* $(d, \varepsilon/2)$*-directional affine extractor, then*

$$\mathsf{ROLBP}_{27\sqrt{\varepsilon/2}}(\mathrm{DAExt}) \geq \varepsilon 2^{n-d-1}.$$

Remark 4.4. *The error in the above theorem is worse than [GPT22, Theorem 16] by a constant factor* 27*, but we note that our proof above is just a modular presentation of the proof in [GPT22, Theorem 16], and the factor* 27 *can be removed by a more careful analysis of this specific construction. That is, in the proof of Theorem 3 we actually need an affine source with* 1*-bit leakage instead of an almost affine source, so a factor* 9 *incurred by arguments related to average conditional min-entropy is unnecessary. Second, a seeded extractor based on leftover hash lemma can in fact work for average conditional min-entropy without any loss (see [DORS08]), so we can remove another factor* 3*.*

Recall that [GPT22] shows that there is an explicit $(d + \log(1/\varepsilon), \varepsilon/2)$-directional affine extractor. This implies the following corollary:

Corollary 4.5. *For every constant $\gamma > 0$, there exists an explicit function* $\mathrm{DAExt} : \mathbb{F}_2^n \to \{0,1\}$ *such that*

$$\mathsf{ROLBP}_{2^{-\gamma n}}(\mathrm{DAExt}) > 2^{(1/3-2\gamma)n-O(1)}.$$

Remark 4.6. *We note that while directional affine extractors imply sumset extractors with the additional "almost affine" restriction, the converse is not true. For example, if we take any sumset extractor* Ext *on n-bit input, and construct a new function* Ext' *on $(n+1)$-bit input which simply ignore the first bit and compute* Ext *on the last n bits, then* Ext' *is still a sumset extractor, but* Ext' *cannot be a directional affine extractor, because the shift $a = (1, 0, \dots, 0, 0) \in \mathbb{F}_2^{n+1}$ would make $\mathrm{Ext}'(\mathbf{X} + a) + \mathrm{Ext}'(\mathbf{X}) = 0$ for every source $\mathbf{X}$.*

4.2 Sumset extractors for non-intersecting span

To utilize the non-intersecting span property, we show that the interleaved-source extractor in [CL20] can be extended to work for the sum of two independent sources $\mathbf{A}, \mathbf{B}$ as long as both $\mathbf{A}, \mathbf{B}$ has entropy rate greater than $1/3$ and $\mathbf{A} + \mathbf{B}$ has entropy rate greater than $2/3$. Formally, we prove the following theorem which extends Theorem 8.1 in [CL20].[7]

Theorem 4 (restated)**.** *For every constant $\delta > 0$, there exists constants $\gamma, \tau > 0$ and an explicit function* $\mathrm{ILExt} : \mathbb{F}_2^n \to \{0,1\}^m$*, $m = \gamma n$, such that for any two independent sources $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$ which satisfies that*

- $\mathrm{H}_\infty(\mathbf{A}), \mathrm{H}_\infty(\mathbf{B}) \geq (\frac{1}{3} + \delta)n$
- $\mathrm{H}_\infty(\mathbf{A} + \mathbf{B}) \geq (\frac{2}{3} + 2\delta)n$

we have

$$\mathrm{ILExt}(\mathbf{A} + \mathbf{B}) \approx_{2^{-\tau n}} \mathbf{U}_m.$$

This theorem also implies a roughly $2^{n/3}$ average-case lower bound:

Corollary 4.7. *For every constant $\delta > 0$, there exists a constant $\tau > 0$ and an explicit function* $\mathrm{ILExt} : \mathbb{F}_2^n \to \{0,1\}$ *such that*

$$\mathsf{ROLBP}_{2^{-\tau n}}(\mathrm{ILExt}) > 2^{(1/3-2\delta)n}.$$

Proof. Let ILExt be the extractor in Theorem 4 with parameter $\delta > 0$, and let $\tau > 0$ be the corresponding constant in Theorem 4. (The output of ILExt is truncated to 1 bit.) Observe that given any two independent sources $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^n$, $\mathrm{span}(\mathrm{Supp}(\mathbf{A})) \cap \mathrm{span}(\mathrm{Supp}(\mathbf{B})) = \{0\}$ implies that for every $x \in \mathrm{Supp}(\mathbf{A}+\mathbf{B})$, there is a unique pair $(a, b) \in \mathrm{Supp}(\mathbf{A}) \times \mathrm{Supp}(\mathbf{B})$ such that $a + b = x$, where a is the projection of x on $\mathrm{span}(\mathrm{Supp}(\mathbf{A}))$ and b is the projection of x on $\mathrm{span}(\mathrm{Supp}(\mathbf{B}))$. This implies $\mathrm{H}_\infty(\mathbf{A} + \mathbf{B}) = \mathrm{H}_\infty(\mathbf{A}) + \mathrm{H}_\infty(\mathbf{B})$. Therefore, we can apply Theorem 3 on ILExt by taking $k_1 = k_2 = (1/3 + \delta)n$ and conclude that

$$\mathsf{ROLBP}_{2^{-\tau n}}(\mathrm{ILExt}) > 2^{(1/3-2\delta)n}.$$

∎

Before we formally prove Theorem 4, first we recall the construction of the interleaved-source extractor in [CL20]. The construction can be viewed as an affine variant of the three-source extractor in [Coh16], which is as follows. Suppose we have three independent sources $\mathbf{X}, \mathbf{Y}, \mathbf{Z} \in \{0,1\}^n$ with min-entropy δn. The first step is to apply a *somewhere random condenser* on $\mathbf{Z}$ to get $t = O(1)$ correlated sources $(\mathbf{S}_1, \dots, \mathbf{S}_t) \in (\{0,1\}^{d_1})^t$ such that there exists an unknown $i^* \in [t]$ for which $\mathbf{S}_{i^*}$ is guaranteed to have min-entropy $(1-\beta)d_1$, for some small enough constant $\beta > 0$. The second step is to compute $\mathbf{R}_i = \mathrm{Ext}(\mathbf{Y}, \mathbf{S}_i)$ for every $i \in [t]$ with some strong seeded extractor Ext. This makes sure that $\mathbf{R}_{i^*}$ is close to uniform, but we still don't know i^*, and $\mathbf{R}_{i^*}$ is correlated with other $\mathbf{R}_i$. To fix this problem, the final step is to apply a *correlation breaker*

[7] Note that we also improve the error from $2^{n^{-\Omega(1)}}$ in [CL20] to $2^{-\Omega(n)}$. This improvement comes from a better construction of affine correlation breakers in more recent works [CGL21, CL22].

to "break the correlation" between $(\mathbf{R}_1, \dots, \mathbf{R}_t)$ with the help of the remaining independent source $\mathbf{X}$, and merge them into a single uniform string by computing their parity.

In the interleaved source/sumset source setting, we are only given one source $\mathbf{A}+\mathbf{B}$. To apply the above three-source extractor construction, [CL20] takes a prefix of $\mathbf{A}+\mathbf{B}$ of length n_1, denoted by $\mathbf{A}_0+\mathbf{B}_0$, to play the role of $\mathbf{Z}$ in the above construction. Then $\mathbf{A}$ and $\mathbf{B}$ would play the roles of $\mathbf{X}$ and $\mathbf{Y}$ in the above construction respectively. In fact, since we do not have access to $\mathbf{A}$ and $\mathbf{B}$ separately, we would actually use $\mathbf{A}+\mathbf{B}$ to play the role of both $\mathbf{X}$ and $\mathbf{Y}$. We would take Ext to be a *strong linear seeded extractor*, and the correlation breaker to be an *affine correlation breaker*, so that $\mathbf{A}+\mathbf{B}$ can play the role of $\mathbf{B}$ and $\mathbf{A}$ respectively in the analysis. We will see the definitions of these primitives later.

To see why taking $\mathbf{Z}$ to be the prefix $\mathbf{A}_0+\mathbf{B}_0$ could possibly work, first observe that in the above construction, we only need a *block source* $(\mathbf{Z}, \mathbf{X})$ and another independent source $\mathbf{Y}$, instead of three independent sources. That is, we only need $(\mathbf{X}, \mathbf{Z})$ to be independent of $\mathbf{Y}$, and $\mathbf{X}$ to have enough entropy conditioned on $\mathbf{Z}$, because we would fix $\mathbf{Z}$ after the first step in the analysis. Therefore, as long as $\mathbf{A}_0$ has enough entropy, we can fix $\mathbf{B}_0$ in the first step, and $(\mathbf{A}, \mathbf{A}_0+\mathbf{B}_0)$ would become independent of $\mathbf{B}$. For the analysis to work, we need to make sure that after fixing both $\mathbf{A}_0$ and $\mathbf{B}_0$, both $\mathbf{A}$ and $\mathbf{B}$ still have enough entropy. Therefore, we need $\mathrm{H}_\infty(\mathbf{A}), \mathrm{H}_\infty(\mathbf{B})$ to be greater than n_1. At the same time, we also need n_1 to be large enough so that $\mathbf{A}_0$ contains enough entropy. (Note that $\mathbf{A}, \mathbf{B}$ are symmetric in the construction, so the analysis can also work if $\mathbf{B}_0$ contains enough entropy instead.) It turns out that it suffices to take $n_1 = n/3$ if $\mathbf{A}+\mathbf{B}$ is an interleaved source, and this is the only place where [CL20] needs $\mathbf{A}+\mathbf{B}$ to be an interleaved source. We observe that what we actually need in the analysis is that $\mathrm{H}_\infty(\mathbf{A}+\mathbf{B})$ is larger than $2n/3$.

Next we introduce the primitives that we mentioned in the above construction. First we define somewhere random sources and somewhere random condenser.

Definition 4.8. *We say* $(\mathbf{R}_1, \dots, \mathbf{R}_t) \in (\{0,1\}^n)^t$ *is an* elementary somewhere random k-source *if there exists* $i \in [t]$ *s.t.* $\mathrm{H}_\infty(\mathbf{R}_i) \geq k$. *A* somewhere random k-source *is a convex combination of elementary somewhere random* k*-sources.*

Definition 4.9. *We say* $\mathrm{SRCon} : \{0,1\}^n \to (\{0,1\}^m)^t$ *is a* $(\alpha_1 \to \alpha_2, \varepsilon)$-somewhere random condenser *if for every* $\mathbf{X} \in \{0,1\}^n$ *such that* $\mathrm{H}_\infty(\mathbf{X}) \geq \alpha_1 n$, $\mathrm{SRCon}(\mathbf{X})$ *is* ε*-close to a somewhere random* $(\alpha_2 m)$*-source.*

Lemma 4.10 ([BKS+10, Raz05, Zuc07])**.** *For every constants* $\delta, \beta > 0$*, there exist constants* $t \in \mathbb{N}$ *and* $\gamma_1, \gamma_2 > 0$ *such that the following holds. For every large enough* $n \in \mathbb{N}$*, there exists an explicit* $(\delta \to 0.99, \varepsilon)$*-somewhere random condenser* $\mathrm{SRCon} : \{0,1\}^n \to (\{0,1\}^{\gamma_1 n})^t$ *where* $\varepsilon = 2^{-\gamma_2 n}$*.*

The second primitive we need is a strong linear seeded extractors. We say a seeded extractor $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}^n$ is linear if for every $s \in \mathcal{S}$, $\mathrm{Ext}(\cdot, s)$ is a linear function. We need a linear seeded extractor with good dependence on the error, which can be constructed with a composition of GUV condenser [GUV09] and leftover hash lemma [ILL89]. (See, e.g., [CGL21] for a proof.)

Lemma 4.11. *For every* m *and* $\varepsilon > 0$*, and every* $d \geq 2m + 8\log(n/\varepsilon) + O(1)$*, there is an explicit* (k, ε)*-strong linear extractor* $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with seed* $\mathbf{U}_d$*, where* $k \geq m + 2\log(1/\varepsilon)$ *.*

Specifically, we want to choose ε small enough to get a seeded extractor that works for high-entropy seed.

Lemma 4.12. *For every* $d \geq 200\log(n)$*, there is an explicit function* $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{d/3}$*, such that for every distribution* $\mathbf{Y} \in \{0,1\}^d$ *which satisfies* $\mathrm{H}_\infty(\mathbf{Y}) \geq 0.99d$*,* LExt *is a* $(0.5d, 2^{-0.02d})$*-strong extractor with seed* $\mathbf{Y}$*.*

Proof. We claim that we can take LExt to be the (k, ε)-extractor in Lemma 4.11, where $\varepsilon = 2^{-0.03d}$ and $k = 0.5d$. Note that the restriction on k and d is satisfied by our choice of parameters. Since LExt is a strong-$(k, 2^{-0.03d})$ extractor with seed $\mathbf{U}_d$, Lemma 2.8 implies that for every distribution $\mathbf{Y} \in \{0,1\}^d$ with min-entropy $0.99d$, LExt is a $(k, 2^{-0.02d})$-strong extractor with seed $\mathbf{Y}$. $\square$

Finally we introduce (a special case of) affine correlation breakers. Roughly speaking, if we are given correlated random variables $(\mathbf{Y}_1, \dots, \mathbf{Y}_t)$ where $\mathbf{Y}_i$ is uniform, we can feed $(\mathbf{Y}_1, \dots, \mathbf{Y}_t)$ into a correlation breaker, and break the correlation of the i-th output from the other output, with the help of an extra independent source $\mathbf{X}$. We say a correlation breaker is an affine correlation breaker if we allow the extra source to be in the form $\mathbf{X} = \mathbf{A}+\mathbf{B}$ where $\mathbf{A}$ is an independent source but $\mathbf{B}$ can be correlated with $(\mathbf{Y}_1, \dots, \mathbf{Y}_t)$.

Definition 4.13 ([Li16, CL16]). *We say* $\mathrm{ACB} : \{0,1\}^n \times \{0,1\}^d \times [t] \to \{0,1\}^m$ *is a* (t,k,ε)*-affine correlation breaker if for every distribution* $\mathbf{A}, \mathbf{B} \in \{0,1\}^n$, $\mathbf{Y}_1, \dots, \mathbf{Y}_t \in \{0,1\}^d$ *and every* $i^* \in [t]$ *such that*

- $\mathrm{H}_\infty(\mathbf{A}) \geq k$,
- $\mathbf{A}$ *is independent of* $(\mathbf{B}, \mathbf{Y}_1, \dots, \mathbf{Y}_t)$,
- $\mathbf{Y}_{i^*} = \mathbf{U}_d$,

it holds that

$$(\mathrm{ACB}(\mathbf{A}+\mathbf{B}, \mathbf{Y}_{i^*}, i^*) \approx_\varepsilon \mathbf{U}_m) \mid \{\mathrm{ACB}(\mathbf{A}+\mathbf{B}, \mathbf{Y}_i, i)\}_{i \in [t] \setminus \{i^*\}}$$

We need the following construction of affine correlation breaker which can work for $\varepsilon = 2^{-\Omega(n)}$.

Lemma 4.14 ([CGL21, CL22]). *For every* $t = O(1)$*, there exists a universal constant* C *such that for* $\varepsilon > 0$ *and* $m \in \mathbb{N}$*, there exists an explicit* (t,k,ε)*-affine correlation breaker* $\mathrm{ACB} : \{0,1\}^n \times \{0,1\}^d \times [t] \to \{0,1\}^m$ *such that* $d = C\log(n/\varepsilon)$ *and* $k = C(m + \log(n/\varepsilon))$.

Now we are ready to prove Theorem 4.

Proof of Theorem 4. The construction of ILExt is as follows.

1. Take $\mathbf{X}_1$ to be a length-$(n/3)$ prefix of $\mathbf{X}$.
2. Compute $(\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_t) = \mathrm{SRCon}(\mathbf{X}_1)$, where $\mathrm{SRCon} : \{0,1\}^{n/3} \to (\{0,1\}^{\gamma_1 n})^t$ is the $(3\delta \to 0.99, 2^{-\gamma_2 n})$ somewhere random condenser from Lemma 4.10. ($t \in \mathbb{N}, \gamma_1 > 0, \gamma_2 > 0$ are constants depending on δ. Specifically, we can make $\gamma_1 < \delta$.)
3. Define $\gamma_3 = \min(\delta/3t, \gamma_1/3)$, and let $\mathrm{LExt} : \{0,1\}^n \times \{0,1\}^{\gamma_1 n} \to \{0,1\}^{\gamma_3 n}$ be the $(0.5\gamma_1 n, 2^{-0.01\gamma_1 n})$-strong linear extractor from Lemma 4.12 which can work for any seed with $0.99\gamma_1 n$ min-entropy. Note that for every constant $\gamma_1 > 0$ we can guarantee that $\gamma_1 n \geq 200\log(n)$ for large enough n.[8]
   For every $i \in [t]$, compute $\mathbf{R}_i = \mathrm{LExt}(\mathbf{X}, \mathbf{S}_i)$.
4. Output $\mathrm{ILExt}(\mathbf{X}) := \bigoplus_{i \in [t]} \mathrm{ACB}(\mathbf{X}, \mathbf{R}_i, i)$, where $\mathrm{ACB} : \{0,1\}^n \times \{0,1\}^{\gamma_3 n} \times [t] \to \{0,1\}^{\gamma n}$ is the $(t, (\delta/2)n, 2^{-\gamma_4 n})$-affine correlation breaker from Lemma 4.14, where $\gamma_4, \gamma > 0$ are small enough constants that satisfy the constraints $\gamma_3 n \geq C\log(n/\varepsilon)$ and $(\delta/2)n \geq C(\log(n/\varepsilon) + \gamma n)$ in Lemma 4.14. (C is a constant depending on t.) It suffices to choose $\gamma_4 = \min(\gamma_3/2C, \delta/4C)$ and $\gamma = \delta/8C$.

Next we prove the correctness of this construction. Let $\mathbf{A}_0$ be the prefixes of $\mathbf{A}$ of length $(1/3)n$ respectively, and $\mathbf{B}_0$ be the prefixed of $\mathbf{B}$ of length $(1/3)n$. First observe that either $\mathrm{H}_\infty(\mathbf{A}_0) \geq \delta n$ or $\mathrm{H}_\infty(\mathbf{B}_0) \geq \delta n$, since

$$\mathrm{H}_\infty(\mathbf{A}_0) + \mathrm{H}_\infty(\mathbf{B}_0) \geq \mathrm{H}_\infty(\mathbf{A}_0 + \mathbf{B}_0) \geq \mathrm{H}_\infty(\mathbf{A}+\mathbf{B}) - (2/3)n \geq 2\delta n.$$

Note that $\mathbf{A}$ and $\mathbf{B}$ are symmetric in this theorem, so without loss of generality we assume that $\mathrm{H}_\infty(\mathbf{A}_0) \geq \delta n$. By Lemma 2.5 and Lemma 2.6, we have $\mathrm{H}_\infty(\mathbf{B}|_{\mathbf{B}_0 = b_0}) \geq (\delta/2)n$ with probability $1 - 2^{-(\delta/2)n}$ over the fixing $\mathbf{B}_0 = b_0$. For the rest of the proof we fix $\mathbf{B}_0 = b_0$ and only consider b_0 which makes $\mathrm{H}_\infty(\mathbf{B}) \geq (\delta/2)n$, and add back the $2^{-(\delta/2)n} = 2^{-\Omega(n)}$ error in the end.

Observe that $\mathrm{H}_\infty(\mathbf{X}_0) = \mathrm{H}_\infty(\mathbf{A}_0 + b_0) \geq \delta n$. Therefore $\mathbf{S}_{[t]}$ is $2^{-\gamma_2 n}$-close to a somewhere random $0.99\gamma_1 n$-source. For every $i \in [t]$, define $\mathbf{R}_{A,i} = \mathrm{LExt}(\mathbf{A}, \mathbf{S}_i)$ and $\mathbf{R}_{B,i} = \mathrm{LExt}(\mathbf{B}, \mathbf{S}_i)$. Note that $\mathbf{R}_i = \mathbf{R}_{A,i} + \mathbf{R}_{B,i}$. Now assume that there exists $i \in [t]$ such that $\mathbf{S}_i$ has min-entropy $0.99\gamma_1 n$. Because $\mathbf{S}_i$ is independent of $\mathbf{B}$, and $\mathrm{H}_\infty(\mathbf{B}) \geq 0.5\delta n \geq 0.5\gamma_1 n$, we have then $\mathbf{R}_{B,i} \approx_{2^{-\Omega(n)}} \mathbf{U}_{\gamma_3 n}$ with probability $1 - 2^{-\Omega(n)}$ over the fixing of $\mathbf{S}_i$ by our choice of parameters of LExt and Markov argument. Moreover, after fixing $\mathbf{S}_i$, $\mathbf{R}_{B,i}$ is independent of $\mathbf{A}_0$. Therefore, with probability $1 - 2^{-\Omega(n)}$ over the fixing of $\mathbf{A}_0$ (which would also fix $\mathbf{S}_i$), $\mathbf{R}_{B,i} \approx_{2^{-\Omega(n)}} \mathbf{U}_{\gamma_3 n}$. Then observe that we can remove the assumption and use the fact that $\mathbf{S}_{[t]}$ is $2^{-\gamma_2 n}$-close to a somewhere random $0.99\gamma_1 n$-source to conclude that with probability $1 - 2^{-\Omega(n)}$ over the fixing of $\mathbf{A}_0$, there exists $i^* \in [t]$ such that $\mathbf{R}_{B,i^*} \approx_{2^{-\Omega(n)}} \mathbf{U}_{\gamma_3 n}$. Moreover, since $\mathbf{R}_{A,[t]}$ is independent of $\mathbf{R}_{B,i^*}$ after fixing $\mathbf{A}_0$, we have $\mathbf{R}_{i^*} \approx_{2^{-\Omega(n)}} \mathbf{U}_{\gamma_3 n}$ over any further fixing of $\mathbf{R}_{A,[t]}$.

[8]If $\gamma_3 < 1/3$ we can simply take the prefix of length $\gamma_3 n$ of the output. The output is still uniform, and LExt is still linear.

Next, observe that by Lemma 2.5,

$$\widetilde{\mathrm{H}}_\infty(\mathbf{A} \mid (\mathbf{A}_0, \mathbf{R}_{A,1}, \ldots, \mathbf{R}_{A,t})) \geq \mathrm{H}_\infty(\mathbf{A}) - (1/3)n - (t\gamma_3)n \geq (2/3)\delta n.$$

By Lemma 2.6 and union bound, we can conclude that with probability $1 - 2^{-\Omega(n)}$ over the fixing of $\mathbf{A}_0, \mathbf{R}_{A,1}, \ldots, \mathbf{R}_{A,t}$, we have $\mathbf{R}_{i^*} \approx_{2^{-\Omega(n)}} \mathbf{U}_{\gamma_3 n}$ and $\mathrm{H}_\infty(\mathbf{A}) \geq \delta/2n$. Moreover, observe that under any such fixing, $\mathbf{A}$ is independent of $(\mathbf{B}, \mathbf{R}_{[t]})$. Therefore, by Lemma 4.14 we can conclude that

$$(\mathrm{ACB}(\mathbf{A}+\mathbf{B}, \mathbf{R}_{i^*}, i^*) \approx_{2^{-\gamma_4 n}} \mathbf{U}_{\gamma n}) \mid \{\mathrm{ACB}(\mathbf{A}+\mathbf{B}, \mathbf{R}_i, i)\}_{i\in[t]\setminus\{i^*\}},$$

which implies

$$\mathrm{ILExt}(\mathbf{A}+\mathbf{B}) = \bigoplus_{i\in[t]} \mathrm{ACB}(\mathbf{A}+\mathbf{B}, \mathbf{R}_{i^*}, i^*) \approx_{2^{-\gamma_4 n}} \mathbf{U}_{\gamma n}.$$

Finally, after adding back all the $2^{-\Omega(n)}$ error that we mentioned above, the error is still $2^{-\Omega(n)}$. □

5 Kakeya sets and HSGs for regular ROLBPs

In this section, we prove Theorem 6, which says that rank-r Kakeya set is a hitting set for oblivious ROLBPs of width $(r+1)$, and Theorem 7, a size lower bound for rank-r Kakeya set over $\mathbb{F}_2^n$.

In [BRRY14], it was proved that a Hamming ball of radius $(w-1)$ is a hitting set for regular read-once branching program of width w.[9] Their proof relies on the fact that there are only $(w-1)$ "crucial layers" such that we can only make a "fatal decision" which goes from a "possibly accept" node to an "always reject" node in these layers. The formal statement is as follows.

Lemma 5.1 ([BRRY14])**.** *For a ROBP f on $\mathbb{F}_2^n$ with layers $L_0, L_1, \ldots, L_n$, we say a layer L_i is crucial if there exists $v \in L_i$ and an edge $(u \to v)$ such that u can reach an accepting state but v cannot.*[10] *Then for every $w \in \mathbb{N}$, a regular ROBP of width w has at most $(w-1)$ crucial layers.*

Based on this lemma, [BRRY14] observed that in order to find an input x of which the computation path reaches an accepting state, we only need to make sure that we do not make any fatal decision in the crucial layers, and the bits read in the other layer can simply be set to 0. Therefore, the Hamming ball of radius $(w-1)$ centered around 0 is a hitting set for regular ROBPs of width w, because the Hamming ball covers every possible decision in the crucial layers, no matter where the crucial layers are. This makes sure that we can find a string which does not make any fatal decision, and this string would reach the sink labeled with 1 in the end.

To generalize this argument to the setting of regular ROLBPs, we want to find a set H such that for every possible rotation R of $\mathbb{F}_2^n$, the rotation of H (denoted by $R(H)$) contains a string which does not make any fatal decision. A naive idea is to find a set which contains every possible rotation of Hamming balls centered at 0. However, this contains exactly the whole set $\mathbb{F}_2^n$. To deal with this issue, we observe that for the argument in [BRRY14] to work, we only need to make sure that for every possible choices of crucial layers $L_{i_1}, \ldots, L_{i_{w-1}}$, where $I = \{i_1, \ldots, i_{w-1}\} \subseteq [n]$, there exists a fixing of the bits outside the crucial layer, such that we enumerate over every possible choice of bits in the crucial layers. Note that the fixing does not need to be 0 and can depend on the choice of crucial layers I. That is, for every set $I \subseteq [n]$ of size at most $(w-1)$, we need to enumerate over a subcube with free bits in I and arbitrary fixing outside I. To ensure this for every possible rotation, what we need is exactly a *Kakeya set*. Next we give a formal proof of our argument.

Lemma 5.2. *Let $H \subseteq \mathbb{F}_2^n$ be a set which satisfies the following: for every $I \subseteq [n]$ of size $(w-1)$, there exists $b \in \mathbb{F}_2^n$ such that $b + \mathrm{span}(\{e_i\}_{i\in I}) \subseteq H$. Then H is a hitting set for regular branching programs of width w.*

Proof. Let f be a regular branching program of width w which accepts at least one string. By Lemma 5.1, there are at most $(w-1)$ crucial layers in f. Let I denote the set of indices of these crucial layers. By

[9] A Hamming ball of radius r centered around $c \in \{0,1\}^n$ is the set of all the strings which are different from c in at most r bits.

[10] Accepting states are the sinks with label 1.

assumption there exists $b \in \mathbb{F}_2^n$ such that $b + \mathrm{span}(\{e_i\}_{i\in I}) \subseteq H$. Now we define a string $b' \in \mathbb{F}_2^n$ inductively as follows. Let v_0 be the source of f, and for every non-sink node v and every $b \in \{0,1\}$ let $\mathsf{next}(v,b)$ denote the node which v connects to with an edge of label b. For i from 1 to n, we define b'_i (the i-th bit of b') as follows:

- If $i \notin I$, then set $b'_i = b_i$.
- If $i \in I$, then set $b'_i = 0$ if $\mathsf{next}(v_{i-1}, 0)$ can reach a accepting state. Otherwise set $b'_i = 1$.

Then we define $v_i = \mathsf{next}(v_{i-1}, b'_i)$. First observe that b' only differ from b on the bits with indices in I. Therefore $b' \in H$. It remains to prove that $f(b') = 1$. Next we prove by induction that every v_i can reach a accepting state. This means v_n is a accepting state, i.e. $f(b') = 1$. For the base case, note that v_0 is the source and hence can reach a accepting state by assumption. To prove that v_i can reach an accepting state assuming that v_{i-1} can reach a accepting state, consider two cases. If $i \notin I$, then the i-th layer is not crucial, which means v_i can reach a accepting state. If $i \in I$, observe that at least one node in $\{\mathsf{next}(v_{i-1}, 0), \mathsf{next}(v_{i-1}, 1)\}$ should be able to reach a accepting state, because they are the only nodes that v_{i-1} can connect to, and v_{i-1} can reach a accepting state. Therefore v_i can also reach a accepting state by definition of b'_i. □

Now we are ready to prove Theorem 6.

Proof of Theorem 6. Let K be a rank-r Kakeya set, and f be any oblivious ROLBP of width $(r+1)$ that accepts at least one string. Observe that there exists a full-rank matrix $R \in \mathbb{F}_2^{n\times n}$ and a read-once regular BP f' of width $(r+1)$ such that for every $x \in \mathbb{F}_2^n$ we have $f(x) = f'(Rx)$. We claim that f' accepts at least one string in $H = \{Rx : x \in K\}$, which implies that f accepts at least one string in K.

For every $I \subseteq [n]$ of size r, observe that there exists $b \in \mathbb{F}_2^n$ such that $b + \mathrm{span}(\{R^{-1}e_i\}_{i\in I}) \subseteq K$, by definition of Kakeya set. This implies that $Rb + \mathrm{span}(\{e_i\}_{i\in I}) \subseteq H$. By Lemma 5.2, H is a hitting set for regular branching programs of width $(r+1)$. Therefore f' accepts at least one string in H. □

Corollary 5.3. *For every $r, n \in \mathbb{N}$ s.t. $r \le n$, there is an explicit hitting set $K \subseteq \mathbb{F}_2^n$ for oblivious read-once regular linear BP of width $(r+1)$ such that $|K| \le 2^{\lceil (1-2^{-r})n\rceil + r}$.*

5.1 Limitation to our approach

Next we prove Theorem 7, which proves a lower bound on rank-r Kakeya sets and implies that the seed length of hitting set generator based on our approach cannot be improved by much.

Theorem 7 (restated). *Every rank-r Kakeya set over $\mathbb{F}_2^n$ has size at least $2^{(1-2^{-r})(n+2)-r}$.*

Proof. Let $s_{n,r}$ denote the minimum size of rank-r Kakeya set over $\mathbb{F}_2^n$. Clearly $S_{n,0} = 1$ for every $n \in \mathbb{N}$. We will show that for every n, r we have $S_{n,r}^2 \ge 2^{n+1} S_{n-1,r-1}$, and then the claimed bound easily follows by induction.

To prove this claim, consider any rank-r Kakeya set over $\mathbb{F}_2^n$, denoted by K, and for every non-zero $a \in \mathbb{F}_2^n$ define $K_a = \{v \in \mathbb{F}_2^n : v \in K \wedge v + a \in K\}$. We claim that for every a we have $|K_a| \ge 2S_{n-1,r-1}$. (Note that this also implies $|K| \ge 2S_{n-1,r-1}$ because every K_a is a subset of K.) To prove this, first we assume w.l.o.g. that the n-th bit of a is 1, and define $K'_a = \{v' \in \mathbb{F}_2^{n-1} : v' \circ 0 \in K_a\}$. Note that $|K'_a| = |K_a|/2$ because for every $v \in \mathbb{F}_2^n$ we have $v \in K_a$ if and only if $v + a \in K_a$, and exactly one of $\{v, v+a\}$ has the last bit being 0.

We claim that K'_a is a rank-$(r-1)$ Kakeya set over size $\mathbb{F}_2^{n-1}$, and hence has size at least $S_{n-1,r-1}$. To prove this, consider any subspace $V' \subseteq \mathbb{F}_2^{n-1}$ of dimension $(r-1)$, and let V denote the subspace of $\mathbb{F}_2^n$ which consists of vectors in V' padded with a 0 in the last bit. Since K is a rank-r Kakeya set, there exists $b \in \mathbb{F}_2^n$ such that $b + V + \{0^n, a\} \subseteq K$. W.l.o.g. we can assume that the last bit of b is 0, i.e. $b = b' \circ 0$ for some $b' \in \mathbb{F}_2^{n-1}$. Then observe that $V' + b' \subseteq K'_a$, because for every $v' \in V'$ we have that $(v' \circ 0) + (b' \circ 0) \in K$ and $(v' \circ 0) + (b' \circ 0) + a \in K$, which implies that $v' + b' \in K'_a$.

Since the same argument works for every subspace V' of dimension $(r-1)$, this means K'_a is a rank-r Kakeya set. Finally, consider the bijective function $f : \mathbb{F}_2^n \times \mathbb{F}_2^n \to \mathbb{F}_2^n \times \mathbb{F}_2^n$ defined as $f(v_1, v_2) = (v_1, v_2 - v_1)$. Observe that the image of f on $K \times K$ is exactly $(K \times \{0^n\}) \cup \bigcup_{a\in\mathbb{F}_2^n, a\ne 0^n} K_a \times \{a\}$. This implies $|K|^2 \ge 2^{n+1} S_{n-1,r-1}$, which is exactly the bound we want.

□

6 Future directions

There are several natural open questions that are raised by our work. We list a couple of interesting directions below.

- Obtain improved average-case hardness against ROLBPs in the small-error regime. Recall that we obtain optimal hardness results but can only achieve polynomially small error, with the bottleneck being lack of explicit constructions of low-error sumset extractors. While constructing low-error sumset sources would immediately resolve this problem, we saw in Section 4 that we just need restricted variants of sumset extractors, which we hope will be easier to construct.

- Construct improved hitting set generators (and more ambitiously pseudorandom generators) for oblivious ROLBPs. As discussed above, one way to make progress on this question would be to show that sum of small-biased distributions are pseudorandom against (standard) oblivious ROBPs. Another direction is to see if objects from linear algebraic pseudorandomness [FG15] can be leveraged for derandomization in this setting.

References

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

[BDVY13] Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff. Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9(1):283–293, 2013.

[BHPP22] Andrej Bogdanov, William M Hoza, Gautam Prakriya, and Edward Pyne. Hitting sets for regular branching programs. In *37th Computational Complexity Conference (CCC 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[BKS+10] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. *J. ACM*, 57(4):20:1–20:52, 2010.

[BRRY14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM Journal on Computing*, 43(3):973–986, 2014.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM J. Comput.*, 17(2):230–261, 1988.

[CG14] Mahdi Cheraghchi and Venkatesan Guruswami. Non-malleable coding against bit-wise and split-state tampering. In *Theory of Cryptography Conference*, pages 440–464. Springer, 2014.

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

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

[CH20] Kuan Cheng and William M Hoza. Hitting sets give two-sided derandomization of small space. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[CHRT18] Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 363–375, 2018.

[CL16] Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016*, pages 299–311. ACM, 2016.

[CL17] Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1171–1184, 2017.

[CL20] Eshan Chattopadhyay and Xin Li. Non-malleable codes, extractors and secret sharing for interleaved tampering and composition of tampering. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020*, volume 12552 of *Lecture Notes in Computer Science*, pages 584–613. Springer, 2020.

[CL22] Eshan Chattopadhyay and Jyun-Jie Liao. Extractors for sum of two sources. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1584–1597. ACM, 2022.

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

[CS16] Gil Cohen and Igor Shinkar. The complexity of DNF of parities. In Madhu Sudan, editor, *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science, Cambridge, MA, USA, January 14-16, 2016*, pages 47–58. ACM, 2016.

[CZ16] Eshan Chattopadhyay and David Zuckerman. New extractors for interleaved sources. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016*, volume 50 of *LIPIcs*, pages 7:1–7:28. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[DK11] Evgeny Demenkov and Alexander S. Kulikov. An elementary proof of a 3n - o(n) lower bound on the circuit complexity of affine dispersers. In Filip Murlak and Piotr Sankowski, editors, *Mathematical Foundations of Computer Science 2011 - 36th International Symposium, MFCS 2011, Warsaw, Poland, August 22-26, 2011. Proceedings*, volume 6907 of *Lecture Notes in Computer Science*, pages 256–265. Springer, 2011.

[DLWZ11] Yevgeniy Dodis, Xin Li, Trevor D. Wooley, and David Zuckerman. Privacy amplification and non-malleable extractors via character sums. In Rafail Ostrovsky, editor, *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 668–677. IEEE Computer Society, 2011.

[DORS08] Yevgeniy Dodis, Rafail Ostrovsky, Leonid Reyzin, and Adam D. Smith. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. *SIAM J. Comput.*, 38(1):97–139, 2008.

[DW09] Yevgeniy Dodis and Daniel Wichs. Non-malleable extractors and symmetric key cryptography from weak secrets. In *Proceedings of the 41st Annual ACM Symposium on Theory of Computing, STOC 2009*, pages 601–610. ACM, 2009.

[EOT10] Jordan S Ellenberg, Richard Oberlin, and Terence Tao. The kakeya set and maximal conjectures for algebraic varieties over finite fields. *Mathematika*, 56(1):1–25, 2010.

[FG15] Michael A. Forbes and Venkatesan Guruswami. Dimension expanders via rank condensers. In Naveen Garg, Klaus Jansen, Anup Rao, and José D. P. Rolim, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015, August 24-26, 2015, Princeton, NJ, USA*, volume 40 of *LIPIcs*, pages 800–814. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015.

[FGHK16] Magnus Gausdal Find, Alexander Golovnev, Edward A. Hirsch, and Alexander S. Kulikov. A better-than-3n lower bound for the circuit complexity of an explicit function. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11*

October 2016, Hyatt Regency, New Brunswick, New Jersey, USA, pages 89–98. IEEE Computer Society, 2016.

[FK18] Michael A. Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 946–955. IEEE Computer Society, 2018.

[GPT22] Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. Linear branching programs and directional affine extractors. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022.

[GTW21] Uma Girish, Avishay Tal, and Kewen Wu. Fourier growth of parity decision trees. *arXiv preprint arXiv:2103.11604*, 2021.

[GUV09] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes. *J. ACM*, 56(4):20:1–20:34, 2009.

[Has89] John Hastad. Almost optimal lower bounds for small depth circuits. *Adv. Comput. Res.*, 5:143–170, 1989.

[HHL18] Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for xor functions. *SIAM Journal on Computing*, 47(1):208–217, 2018.

[ILL89] Russell Impagliazzo, Leonid A. Levin, and Michael Luby. Pseudo-random generation from one-way functions (extended abstracts). In David S. Johnson, editor, *Proceedings of the 21st Annual ACM Symposium on Theory of Computing, May 14-17, 1989, Seattle, Washington, USA*, pages 12–24. ACM, 1989.

[Juk12] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012.

[KLSS11] Swastik Kopparty, Vsevolod F Lev, Shubhangi Saraf, and Madhu Sudan. Kakeya-type sets in finite vector spaces. *Journal of Algebraic Combinatorics*, 34(3):337–355, 2011.

[KM93] Eyal Kushilevitz and Yishay Mansour. Learning decision trees using the fourier spectrum. *SIAM J. Comput.*, 22(6):1331–1348, 1993.

[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, 2006.

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

[LV17] Chin Ho Lee and Emanuele Viola. Some limitations of the sum of small-bias distributions. *Theory of Computing*, 13(1):1–23, 2017.

[LY22] Jiatu Li and Tianqi Yang. $3.1n$ - $o(n)$ circuit lower bounds for explicit functions. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1180–1193. ACM, 2022.

[MW97] Ueli M. Maurer and Stefan Wolf. Privacy amplification secure against active adversaries. In Burton S. Kaliski Jr., editor, *Advances in Cryptology - CRYPTO '97, 17th Annual International Cryptology Conference*, volume 1294 of *Lecture Notes in Computer Science*, pages 307–321. Springer, 1997.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Comb.*, 12(4):449–461, 1992.

[NN90] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 213–223, 1990.

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

[Raz05] Ran Raz. Extractors with weak random seeds. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24, 2005*, pages 11–20. ACM, 2005.

[RSV13] Omer Reingold, Thomas Steinke, and Salil Vadhan. Pseudorandomness for regular branching programs via fourier analysis. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 655–670. Springer, 2013.

[TS17] Amnon Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 238–251, 2017.

[TWXZ13] Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 658–667. IEEE, 2013.

[Tzu09] Yoav Tzur. Notions of weak pseudorandomness and gf (2n)-polynomials. *Master's thesis, Weizmann Institute of Science*, 2009.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Found. Trends Theor. Comput. Sci.*, 7(1-3):1–336, 2012.

[Yao85] Andrew Chi-Chih Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *26th Annual Symposium on Foundations of Computer Science, Portland, Oregon, USA, 21-23 October 1985*, pages 1–10. IEEE Computer Society, 1985.

[Zuc07] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Comput.*, 3(1):103–128, 2007.

A Strongly read-once linear branching programs

A.1 Difference from the definition in [GPT22]

The difference between the definition of strongly read-once in [GPT22] and our definition (Definition 1.2) is as follows. First, let Pre'_u denote the span of all the linear queries on a path to u, *excluding the query* ℓ_u *on* u. The definition of strongly read-once in [GPT22] is that $\mathsf{Pre}'_u \cap \mathsf{Post}_u = \{0\}$ for every node u. First we show that the definition in [GPT22] implies Definition 1.2.

Claim 1. *In a linear branching program, if for every node v it holds that $\mathsf{Pre}'_v \cap \mathsf{Post}_v = \{0\}$, then*

- *For every edge (u, v), $\mathsf{Pre}_u \cap \mathsf{Post}_v = \{0\}$.*
- *For every node v, $\mathsf{Pre}_v \cap \mathsf{Post}_v = \{0, \ell_v\}$.*

Proof. Let $P(v)$ denote the set of all nodes u such that there is an edge $(u \to v)$. Observe that $\mathsf{Pre}'_v = \mathrm{span}(\bigcup_{u \in P(v)} \mathsf{Pre}_u)$. Therefore, if $\mathsf{Pre}'_v \cap \mathsf{Post}_v = \{0\}$, then $(\mathsf{Pre}_u \cap \mathsf{Post}_v) \subseteq (\mathsf{Pre}'_v \cap \mathsf{Post}_v)$ for every $u \in P(v)$, which implies $\mathsf{Pre}_u \cap \mathsf{Post}_v = \{0\}$ for every $u \in P(v)$. To prove the second property, note that $\mathsf{Pre}_v = \mathsf{Pre}'_v \cup (\mathsf{Pre}'_v + \ell_v)$. Because Post_v is a subspace that contains ℓ_v, we have $(\mathsf{Pre}'_v + \ell_v) \cap \mathsf{Post}_v = (\mathsf{Pre}'_v + \ell_v) \cap (\mathsf{Post}_v + \ell_v) = (\mathsf{Pre}'_v \cap \mathsf{Post}_v) + \ell_v = \{\ell_v\}$, which implies $\mathsf{Pre}_v \cap \mathsf{Post}_v = \{0, \ell_v\}$. $\square$

Next we show that our definition is strictly more general.

Claim 2. *There exists a linear branching program which satisfies the strongly read-once definition in Definition 1.2, but contains some node w such that $\mathsf{Pre}'_w \cap \mathsf{Post}_w \neq \{0\}$.*

Proof. To see why this is the case, consider a linear branching programs with four non-sink nodes, s, v_1, v_2, w, and the two edges of w connect to two sink labeled with 0 and 1 respectively. Furthermore, we choose the queries on these nodes to be $\ell_s = e_3$, $\ell_{v_1} = e_1$, $\ell_{v_2} = e_2$ and $\ell_w = e_1 + e_2$. Then observe that both Pre'_w and Post_w contain $e_1 + e_2$, but this linear branching program satisfies the definition in Definition 1.2. $\square$

In fact, from the example above, one can see that our definition of strongly read-once is technically incomparable with the "weakly read-once" model defined in [GPT22], which requires that $\ell_v \notin \mathsf{Pre}'_v$ for every v. However, our definition is closer to strongly read-once in [GPT22] because we still need the fact that the queries before v and the queries after v do not affect each other.

Finally let us elaborate what the second property in our definition means, because it might seem less intuitive. Given the first definition, we see that every edge $e = (u \to v)$ decompose $\mathbb{F}_2^n$ into two complemented subspace. The second property is to make sure that the dimension of both subspaces in this decomposition change by at most 1 when we move one step from an edge $(u \to v)$ to another edge $(v \to w)$. This is to make sure that for every path we can find an edge e such that the dimension of Pre_u and Post_v is exactly what we want. Without this property, the size lower bound in Theorem 3 would become roughly $2^{n-k_1-2k_2}$ because the dimension of Post_v can drop down to half after one step, and the directional affine extractor in [GPT22] would no longer work.

A.2 ROBPs and PDTs are strongly read-once

Next we show that (standard) read-once branching programs and parity decision trees are indeed strongly read-once, as claimed in [GPT22]. We prove this claim with the definition of strongly reda-once in [GPT22] which would imply our definition as discussed above. The fact that ROBPs are strongly read-once is easy to see: if there is some v such that $\mathsf{Pre}'_v \cap \mathsf{Post}_v \neq \{0\}$, then there is a bit x_i, a node u which queries x_i and is on a path to v and another node w which also queries x_i and is on a path from v. Then x_i is queried twice on the path from u to v then to w, which contradicts to the definition of ROBPs.

The fact that PDTs are strongly read-once is less obvious. For example, it might be possible that there is a node u with children v_1, v_2 such that $\ell_u = \ell_{v_1} + \ell_{v_2}$. However, for this case we can query ℓ_{v_1} on node v_2 instead and simulate the original query ℓ_{v_2} with $\ell_u + \ell_{v_1}$. In general, observe that the following operation can convert a PDT T into another PDT T' with the same size.

Fact 1. *Let u, v be two distinct internal nodes in the PDT T such that u is an ancestor of v. Moreover, suppose v is in the subtree of u which corresponds to $\ell_u(x) = b$, for some $b \in \mathbb{F}_2$. Then we can replace the query on v with $\ell_u + \ell_v$, and add b to the labels on the edges from v.*

We claim that we can find a sequence of operation that turns the above tree into a strongly read-once branching program.

Claim 3. *For every PDT T, there exists another PDT T' such that the size of T' is not larger than the size of T.*

Proof. Without loss of generality, assume that T is weakly read-once, i.e. $\ell_u \notin \mathsf{Pre}'_u$ for every internal node u. (Otherwise the value of $\ell_u(x)$ is fixed whenever we reach u, and hence u and one of the branch from u can be completely removed.) Next we perform a sequence of operations in Fact 1 and maintain a set of internal nodes S with the following invariant. For every node v, let ℓ_v denote the linear query on node u. (After we perform the operation in Fact 1, ℓ_v is changed correspondingly.) The invariant is defined as

- There does not exist $u \in S$ and $W \subseteq S \backslash \{u\}$ such that u is an ancestor of every node in W, and $\sum_{w \in W} \ell_w = \ell_u$.

Next, fix a topological order. We will add all the internal nodes into S in this order, and prove that the invariant holds by induction. Suppose S consists of all the nodes before v in the topological order, and the invariant is true for S. We say v violates the invariant with $u \in S$, if there exists $W \subseteq S$ such that u is an ancestor of v and every node in W, and $\ell_v + \sum_{w \in W} \ell_w = \ell_u$. (Note that we only need to consider the case

$u \in S$ but not the case $u = v$, because v cannot be the ancestor of any node in S, since we add the nodes in a topological order.) If v does not violate the invariant with any u, then we simply add v into S, and the invariant remains true. Otherwise, pick $u \in S$ with the *smallest depth* such that v violates the invariant with u. Then we replace ℓ_v with $\ell_u + \ell_v$ by performing the operation in Fact 1. We claim that after this operation, if v violates the invariant with any u', then the depth of u' is strictly greater than the depth of u. Therefore, by repeating this operation for finite number of times, eventually v would not violate the invariant with any u, which means we can add v into S without violating the invariant.

To prove the claim, let u' be any node such that v violates the invariant with u' after the operation. Because v violates the invariant with u before the operation, there exists $W \subseteq S \backslash \{u\}$ such that u is an ancestor of v and every node in W, and $\ell_v + \sum_{w \in W} \ell_w = \ell_u$. Furthermore, because v violates the invariant with u' after the operation, there exists $W' \subseteq S \backslash \{u'\}$ such that u' is an ancestor of v and every node in W', and $\ell_v + \ell_u + \sum_{w \in W'} \ell_w = \ell_{u'}$. Let T denote the symmetric difference between W and W'. Note that $T \subseteq S$. Observe that $\ell'_u = \sum_{w \in T} \ell_w$. If u' has depth smaller than or equal to u, then $u' \notin T$ and u' is an ancestor of every node in W and W', which violates the invariant.

□

B Directional affine extractors are non-malleable

Recently, stronger variants of seeded and seedless extractors, called non-malleable extractors have been studied, with motivations from cryptography and pseudorandomness [DW09, CG14]. In this section we show that directional affine extractors are equivalent to affine extractors that are non-malleable against tampering functions that are constant shift.

We refer the reader to [CG14] for the general definition of seedless non-malleable extractors, and present the definition specialized to our setting below.

Definition B.1. *We say* $\mathrm{Ext} : \mathbb{F}_2^n \to \{0,1\}$ *is a* (d, ε)-non-malleable affine extractor against shifts *if for every source* $\mathbf{X} \in \mathbb{F}_2^n$ *which is uniform over an affine subspace of dimension* d*, and every non-zero shift* $a \in \mathbb{F}_2^n$*,*

$$(\mathrm{Ext}(\mathbf{X}) \approx_\varepsilon \mathbf{U}_1) \mid \mathrm{Ext}(\mathbf{X} + a).$$

It's easy to see that a (d, ε)-non-malleable affine extractor against shifts is also a (d, ε)-directional affine extractor. We prove the converse below.

Theorem B.2. *For every* $d \in \mathbb{N}, \varepsilon > 0$ *such that* $d \geq \log(1/\varepsilon)$*, a* (d, ε)*-directional affine extractor is also a* $(d, O(\sqrt{\varepsilon}))$*-non-malleable affine extractor.*

Proof. To prove this theorem, we need an extension of Vazirani's XOR lemma, which can be found in [DLWZ11, Lemma 3.8]. We only state the special case we need here.

Lemma B.3. *Let* $(\mathbf{W}, \mathbf{W}')$ *be a random variable over* $(\mathbb{F}_2)^2$*. If* $\mathbf{W} \approx_\varepsilon \mathbf{U}_1$ *and* $(\mathbf{W} + \mathbf{W}') \approx_\varepsilon \mathbf{U}_1$*, then*

$$(\mathbf{W} \approx_{4\varepsilon} \mathbf{U}_1) \mid \mathbf{W}'.$$

With this lemma, it suffices to prove that for every (d, ε)-directional affine extractor $\mathrm{DAExt} : \mathbb{F}_2^n \to \{0,1\}$ the following holds. for every source $\mathbf{X} \in \mathbb{F}_2^n$ which is uniform over an affine subspace of dimension d, and every non-zero shift $a \in \mathbb{F}_2^n$,

- $\mathrm{DAExt}(\mathbf{X}) \approx_{\sqrt{\varepsilon}} \mathbf{U}_1$, and
- $\mathrm{DAExt}(\mathbf{X}) + \mathrm{DAExt}(\mathbf{X} + a) \approx_{\sqrt{\varepsilon}} \mathbf{U}_1$.

The second condition is directly implied by the definition of DAExt. It remains to prove the first condition. Let V be the linear subspace which is a shift of the affine subspace $\mathrm{Supp}(\mathbf{X})$, and let $\mathbf{V}$ denote the uniform distribution over V which is independent of $\mathbf{X}$. Observe that $\mathbf{V} + \mathbf{X}$ is the same distribution as $\mathbf{X}$, and $\mathrm{H}_\infty(\mathbf{V}) \geq d \geq \log(1/\varepsilon)$. Then, by Lemma 4.1 we have $\mathrm{DAExt}(\mathbf{X} + \mathbf{V}) \approx_{O(\sqrt{\varepsilon})} \mathbf{U}_1$. □

We note that Chattopadhyay and Li [CL17] considered the problem of constructing non-malleable extractors against the more general class of all linear functions, but their results requires to the affine source to have dimension $0.99n$. However, it appears difficult to extend their techniques to handle smaller min-entropy, even against the weaker class of shifts.

C Extractors for average conditional min-entropy, generalized

In this section we prove the following lemma.

Lemma 2.10 (restated)**.** *Let* $(\mathbf{X}, \mathbf{Y}, \mathbf{E})$ *be a joint distribution such that* $\mathbf{X} \in \mathcal{X}$ *and* $\mathbf{Y} \in \mathcal{S}$ *are independent conditioned on* $\mathbf{E}$*, and* $\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{E}) \geq k$*. Let* $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}^m$ *be a function which satisfies the following conditions for an error parameter* $\varepsilon > 0$ *and a deterministic function* g*: for every* $e \in \mathrm{Supp}(\mathbf{E})$*, there exists a set* $\mathcal{X}_e \subseteq \mathcal{X}$ *with size at least* 2^{k+1} *such that* Ext *when restricted to the domain* $\mathcal{X}_e \times \mathcal{S}$ *is a* (k, ε)*-extractor with seed* $\mathbf{Y}|_{\mathbf{E}=e}$ *and is strong in* $g(e, \mathbf{Y})$*. Then*

$$(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}) \approx_{3\varepsilon} \mathbf{U}_m) \mid (\mathbf{E}, g(\mathbf{E}, \mathbf{Y})).$$

The proof follows the outline in [Vad12, Problem 6.8], but each step in the proof needs to be extended to our more general definition of seeded extractors. First we need the following lemma.

Lemma C.1. *Let* $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}^m$ *be a* (k, ε)*-extractor with seed* $\mathbf{Y}$*, where* $k \leq \log(|\mathcal{X}|) - 1$*, and is strong in* $g(\mathbf{Y})$ *for some deterministic function* g*. Then for every* $0 < t \leq k$*,* $\mathrm{Ext} : \mathcal{X} \times \mathcal{S} \to \{0,1\}$ *is also a* $(k - t, 2^{t+1}\varepsilon)$*-extractor with seed* $\mathbf{Y}$ *that is strong in* $g(\mathbf{Y})$*.*

Proof. Let $\mathcal{G} = \mathrm{Supp}(g(\mathbf{Y}))$. It suffices to prove that for every $T \subseteq \{0,1\}^m \times \mathcal{G}$ and every $\mathbf{X}$ such that $\mathrm{H}_\infty(\mathbf{X}) \geq k - t$, it holds that

$$\Pr\left[(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}), g(\mathbf{Y})) \in T\right] - \Pr\left[(\mathbf{U}_m, g(\mathbf{Y})) \in T\right] \leq (2^{t+1} - 1)\varepsilon$$

For every $x \in \mathcal{X}$, define $\delta(x) = \Pr\left[(\mathrm{Ext}(x, \mathbf{Y}), g(\mathbf{Y})) \in T\right] - \Pr[(\mathbf{U}_m, g(\mathbf{Y})) \in T]$. Let $N = |\mathcal{X}|$, and consider an ordering of the elements in $\mathcal{X}$, $x_1, \ldots, x_N$ such that $\delta(x_1) \geq \delta(x_2) \geq \ldots \geq \delta(x_N)$. Define a step function $f : (0, N] \to \mathbb{R}$ to be $f(r) = \delta(x_{\lceil r \rceil})$. Note that f is decreasing. Since Ext is a (k, ε) extractor, observe that for every $0 \leq m \leq N - 2^k$ it holds that $-\varepsilon \leq 2^{-k} \int_m^{m+2^k} f(t)\, dt \leq \varepsilon$, because $2^{-k} \int_m^{m+2^k} f(t)\, dt$ corresponds to $\Pr\left[(\mathrm{Ext}(\mathbf{X}', \mathbf{Y}), g(\mathbf{Y})) \in T\right] - \Pr\left[(\mathbf{U}_m, g(\mathbf{Y})) \in T\right]$ for some $\mathbf{X}'$ of min-entropy k. Then observe that

$$\begin{aligned}
&\Pr\left[(\mathrm{Ext}(\mathbf{X}, \mathbf{Y}), g(\mathbf{Y})) \in T\right] - \Pr\left[(\mathbf{U}_m, g(\mathbf{Y})) \in T\right] \\
&\leq 2^{t-k} \int_0^{2^{k-t}} f(t)\, dt \\
&= 2^{t-k} \left(\int_0^{2^k} f(t)\, dt - \int_{2^{k-t}}^{2^k} f(t)\, dt \right) \\
&\leq 2^{t-k} \left(\int_0^{2^k} f(t)\, dt - \frac{2^k - 2^{k-t}}{2^k} \int_{N-2^k}^{N} f(t)\, dt \right) \text{ (since } 2^k \leq N - 2^k \text{ and } f \text{ is decreasing)} \\
&\leq (2^{t+1} - 1)\varepsilon \\
&\leq 2^{t+1}\varepsilon.
\end{aligned}$$

□

Next we prove Lemma 2.10.

Proof of Lemma 2.10. For every $e \in \mathrm{Supp}(\mathbf{E})$, write $\mathbf{X}_e = \mathbf{X}|_{\mathbf{E}=e}$ and $\mathbf{Y}_e = \mathbf{Y}|_{\mathbf{E}=e}$ for short. Note that

$(\mathbf{X}, \mathbf{Y}) \mid (\mathbf{E} = e)$ is equivalent to independent distributions $(\mathbf{X}_e, \mathbf{Y}_e)$. Therefore,

$$
\begin{aligned}
&\Delta\left((\mathrm{Ext}(\mathbf{X}, \mathbf{Y}), \mathbf{E}, g(\mathbf{E}, \mathbf{Y})); (\mathrm{Ext}(\mathbf{X}, \mathbf{Y}), \mathbf{E}, g(\mathbf{E}, \mathbf{Y}))\right) \\
&= \mathop{\mathbb{E}}_{e \sim \mathbf{E}}\left[\Delta\left((\mathrm{Ext}(\mathbf{X}_e, \mathbf{Y}_e), g(e, \mathbf{Y}_e)); (\mathrm{Ext}(\mathbf{X}_e, \mathbf{Y}_e), g(e, \mathbf{Y}_e))\right)\right] \\
&= \sum_{e : \mathrm{H}_\infty(\mathbf{X}_e) \geq k} \Pr[\mathbf{E} = e] \cdot \Delta\left((\mathrm{Ext}(\mathbf{X}_e, \mathbf{Y}_e), g(e, \mathbf{Y}_e)); (\mathrm{Ext}(\mathbf{X}_e, \mathbf{Y}_e), g(e, \mathbf{Y}_e))\right) \\
&+ \sum_{e : \mathrm{H}_\infty(\mathbf{X}_e) < k} \Pr[\mathbf{E} = e] \cdot \Delta\left((\mathrm{Ext}(\mathbf{X}_e, \mathbf{Y}_e), g(e, \mathbf{Y}_e)); (\mathrm{Ext}(\mathbf{X}_e, \mathbf{Y}_e), g(e, \mathbf{Y}_e))\right) \\
&= \sum_{e : \mathrm{H}_\infty(\mathbf{X}_e) \geq k} \Pr[\mathbf{E} = e] \cdot \varepsilon \\
&+ \sum_{e : \mathrm{H}_\infty(\mathbf{X}_e) \geq k} \Pr[\mathbf{E} = e] \cdot 2^{k+1-\mathrm{H}_\infty(\mathbf{X}_e)} \varepsilon \text{ (by Lemma C.1)} \\
&\leq \sum_e \Pr[\mathbf{E} = e] \cdot \varepsilon + \sum_e \Pr[\mathbf{E} = e] \cdot 2^{k+1-\mathrm{H}_\infty(\mathbf{X}_e)} \varepsilon \\
&= \varepsilon + 2^{-k} \cdot 2^{1+\widetilde{\mathrm{H}}_\infty(\mathbf{X} \mid \mathbf{E})} \cdot \varepsilon \\
&\leq 3\varepsilon.
\end{aligned}
$$

□

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il