# Explicit Directional Affine Extractors and Improved Hardness for Linear Branching Programs

**Xin Li**
Johns Hopkins University, Baltimore, MD, USA

**Yan Zhong**
Johns Hopkins University, Baltimore, MD, USA

## Abstract

Affine extractors give some of the best-known lower bounds for various computational models, such as $\mathsf{AC}^0$ circuits, parity decision trees, and general Boolean circuits. However, they are not known to give strong lower bounds for read-once branching programs (ROBPs). In a recent work, Gryaznov, Pudlák, and Talebanfard (CCC' 22) introduced a stronger version of affine extractors known as directional affine extractors, together with a generalization of ROBPs where each node can make linear queries, and showed that the former implies strong lower bound for a certain type of the latter known as strongly read-once linear branching programs (SROLBPs). Their main result gives explicit constructions of directional affine extractors for entropy $k > 2n/3$, which implies average-case complexity $2^{n/3-o(n)}$ against SROLBPs with exponentially small correlation. A follow-up work by Chattopadhyay and Liao (CCC' 23) improves the hardness to $2^{n-o(n)}$ at the price of increasing the correlation to polynomially large, via a new connection to sumset extractors introduced by Chattopadhyay and Li (STOC' 16) and explicit constructions of such extractors by Chattopadhyay and Liao (STOC' 22). Both works left open the questions of better constructions of directional affine extractors and improved average-case complexity against SROLBPs in the regime of small correlation.

This paper provides a much more in-depth study of directional affine extractors, SROLBPs, and ROBPs. Our main results include:

- An explicit construction of directional affine extractors with $k = o(n)$ and exponentially small error, which gives average-case complexity $2^{n-o(n)}$ against SROLBPs with exponentially small correlation, thus answering the two open questions raised in previous works.
- An explicit function in $\mathsf{AC}^0$ that gives average-case complexity $2^{(1-\delta)n}$ against ROBPs with negligible correlation, for any constant $\delta > 0$. Previously, no such average-case hardness is known, and the best size lower bound for any function in $\mathsf{AC}^0$ against ROBPs is $2^{\Omega(n)}$.

One of the key ingredients in our constructions is a new linear somewhere condenser for affine sources, which is based on dimension expanders. The condenser also leads to an unconditional improvement of the entropy requirement of explicit affine extractors with negligible error. We further show that the condenser also works for general weak random sources, under the Polynomial Freiman-Ruzsa Theorem in $\mathsf{F}_2^n$, recently proved by Gowers, Green, Manners, and Tao (arXiv' 23).

**2012 ACM Subject Classification** Theory of computation → Expander graphs and randomness extractors; Theory of computation → Circuit complexity; Theory of computation → Pseudorandomness and derandomization

**Keywords and phrases** Randomness Extractors, Affine, Read-once Linear Branching Programs, Low-degree polynomials, $\mathsf{AC}^0$ circuits

**Digital Object Identifier** 10.4230/LIPIcs.CCC.2024.10

**Related Version** *Full Version*: https://eccc.weizmann.ac.il/report/2023/058/

**Funding** *Xin Li*: Supported by NSF CAREER Award CCF-1845349 and NSF Award CCF-2127575. *Yan Zhong*: Supported by NSF CAREER Award CCF-1845349.

**Acknowledgements** We thank anonymous reviewers for their helpful comments and a reviewer for pointing us to [20].







## 1 Introduction

Randomness extractors are functions that extract almost uniform random bits from weak random sources that have poor quality. Although the original motivation of randomness extractors comes from bridging the gap between the quality of randomness required in typical applications and that available in practice, as pseudorandom objects, they turn out to have broad applications in computer science. For example, the kind of extractors known as *affine extractors* are shown to be closely connected to complexity theory. Indeed, they give strong size lower bounds for $\mathsf{AC}^0$ circuits (constant depth circuits with NOT gates and unbounded fan-in AND, OR gates) by the standard switching lemma [23], and are shown to give exponential size lower bounds for DNF circuits with a bottom layer of parity gates, together with strong average-case hardness for parity decision trees [14]. Via sophisticated gate elimination techniques, they also give the best-known size lower bounds for general Boolean circuits [16, 18, 28]. We define affine extractors below.

▶ **Definition 1** (Affine extractor). *An $(n, k)$ affine source is the uniform distribution over some affine subspace with dimension $k$, of the vector space $\mathsf{F}_2^n$.*[1] *A function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ is an affine extractor for entropy $k$ with error $\varepsilon$ if for every $(n, k)$ affine source $X$, we have*

$$\mathsf{Ext}(X) \approx_\varepsilon U_m,$$

*where $U_m$ stands for the uniform distribution over $\{0,1\}^m$, and $\approx_\varepsilon$ means $\varepsilon$ close in statistical distance. We say $\mathsf{Ext}$ is explicit if it is computable by a polynomial-time algorithm.*

However, affine extractors are not known to imply strong lower bounds for computational models that measure space complexity. For example, a natural model in this context is a branching program, which is a directed acyclic graph with one source and two sinks, and each non-sink node has out-degree 2. To define the computation of the branching program, one marks each non-sink node with the index of an input bit, and labels the two outgoing edges by 0 and 1, respectively. Furthermore, one sink is labeled by 1 and the other is labeled by 0. The program now computes any input by following the natural path from the source to one sink, while reading the corresponding input bits and going through the corresponding edges. The program accepts the input if and only if the path ends in the sink with label 1, and the size of the branching program is defined as the number of its nodes, which roughly corresponds to $2^{O(s)}$ where $s$ is the space complexity of the computation.

Proving non-trivial lower bounds of an explicit function for general branching programs turns out to be a challenging problem. The best known bound is $\Omega(\frac{n^2}{\log^2 n})$ [33] after decades of effort, which is not enough to separate $\mathsf{P}$ from $\mathsf{LOGSPACE}$. Thus, most research on lower bounds for branching programs has focused on restricted models, and the most well-studied is the model of *read-once branching program*, where on any computational path, any input bit is read at most once. Exponential lower bounds are known in this model [41, 43, 17, 24, 27, 39, 36, 19, 5, 1, 26], however, it is not clear if affine extractors imply strong lower bounds here. For example, the inner product is a good affine extractor for any entropy $k > n/2$, but it can be computed by a read-once branching program of size $O(n)$.

In a recent work [22], Gryaznov, Pudlák, and Talebanfard introduced a generalization of affine extractors called *directional affine extractors* and a generalization of standard read-once branching programs called *read-once linear branching programs*, and show that explicit constructions of the former imply strong lower bounds for certain cases of the latter. We define the two generalizations below.

[1] More generally, affine sources and affine extractors can be defined over any finite field, but in this paper we focus on the binary field $\mathsf{F}_2$.

▶ **Definition 2** (Directional affine extractor). *A function* $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ *is a directional affine extractor for entropy* $k$ *with error* $\varepsilon$ *if for every* $(n,k)$ *affine source* $X$ *and every non-zero vector* $a \in \mathsf{F}_2^n$*, we have*

$$(\mathsf{DAExt}(X), \mathsf{DAExt}(X+a)) \approx_\varepsilon (U_m, \mathsf{DAExt}(X+a)).$$

*We say the function is a (zero-error) directional affine disperser if there exists some* $b \in \{0,1\}^m$ *such that*

$$\left|\mathsf{Supp}\left(\mathsf{DAExt}(X) \mid \mathsf{DAExt}(X+a) = b\right)\right| = 2^m.$$

▶ Remark 3. Our definition is slightly more general than the definition in [22], since we allow the extractor to output more than one bits. In the special case of $m = 1$, our definition implies that in [22], the reverse is also true up to a small loss in parameters as shown in [12].

▶ **Definition 4** (Linear branching program [22]). *A linear branching program on* $\mathsf{F}_2^n$ *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 non-sink node* $v$ *is labeled with a linear function* $\ell_v : \mathsf{F}_2^n \to \mathsf{F}_2$*. 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* $P$*.* $P$ *computes a Boolean function* $f : \{0,1\}^n \to \{0,1\}$ *in the following way. For every input* $x \in \mathsf{F}_2^n$*,* $P$ *follows the computation path by starting from* $s$*, and when on a non-sink node* $v$*, moves to the next node following the edge with label* $\ell_v(x) \in \{0,1\}$*. The computation ends when the path ends at a sink, and* $f(x)$ *is defined to be the label on this sink.*

[22] defines two kinds of read-once linear branching programs (ROLBP for short). Specifically, given any linear branching program $P$ and any node $v$ in $P$, let $\mathsf{Pre}_v$ denote the span of all linear queries that appear on any path from the source to $v$, excluding the query $\ell_v$. Let $\mathsf{Post}_v$ denote the span of all linear queries in the subprogram starting at $v$.

▶ **Definition 5** (Weakly read-once linear branching program). *A linear branching program* $P$ *is weakly read-once if for every inner node* $v$ *of* $P$*, it holds that* $\ell_v \notin \mathsf{Pre}_v$*.*

▶ **Definition 6** (Strongly read-once linear branching program). *A linear branching program* $P$ *is strongly read-once if for every inner node* $v$ *of* $P$*, it holds that* $\mathsf{Pre}_v \cap \mathsf{Post}_v = \{0\}$*.*

In this paper, we will focus on strongly read-once linear branching programs, and use SROLBP as a shorthand. As observed in [22] and [12], even the more restricted SROLBPs generalize several important and well-studied computational models, for example, decision trees, parity decision trees, and standard read-once branching programs. These models have applications in diverse areas, such as learning theory, streaming algorithms, communication complexity and query complexity. Thus, just as the natural generalizations from $\mathsf{AC}^0$ circuits to $\mathsf{AC}^0[\oplus]$ circuits ($\mathsf{AC}^0$ with parity gates), and from decision trees to parity decision trees, studying the generalization from ROBPs to ROLBPs is also a natural direction. In addition, as observed in [22], parity decision trees are the only case in $\mathsf{AC}^0[\oplus]$ for which we have strong average-case lower bounds, and they are closely related to tree-like resolution refutation proof systems. Thus studying ROLBPs as a generalization of parity decision trees is of particular interest (in fact, this is the original motivation in [22]). We now define two complexity measures of SROLBPs below.

▶ **Definition 7.** *For a Boolean function $f : \{0,1\}^n \to \{0,1\}$, let $\mathsf{SROLBP}(f)$ denote the smallest possible size of a strongly read-once linear branching program that computes $f$, and $\mathsf{SROLBP}_\varepsilon(f)$ denote the smallest possible size of a strongly read-once linear branching program $P$ such that*

$$\mathbf{Pr}_{x \leftarrow_U \mathsf{F}_2^n}[P(x) = f(X)] \geq \frac{1}{2} + \varepsilon.$$

*The definition can be adapted to* ROBP*s naturally.*

The main contribution of [22] is to show that directional affine extractors give strong average-case hardness for SROLBPs. Specifically, they show that for any directional affine extractor DAExt for entropy $k$ with error $\varepsilon$, we have $\mathsf{SROLBP}_{\sqrt{\varepsilon/2}}(\mathsf{DAExt}) \geq \varepsilon 2^{n-k-1}$. In addition, they give an explicit construction of directional affine extractor for $k \geq \frac{2n}{3} + c$ with $\varepsilon \leq 2^{-c}$, which also implies exponential average-case hardness for SROLBPs of size up to $2^{\frac{n}{3}-o(n)}$. Thus, directional affine extractors are indeed stronger than standard affine extractors and give strong lower bounds in more computational models. [22] left open the question of explicit constructions of directional affine extractors for $k = o(n)$.

In a follow-up work, Chattopadhyay and Liao [12] showed that another kind of extractors, known as *sumset extractors*, also give strong average-case hardness for SROLBPs. These extractors were introduced by Chattopadhyay and Li [9], which are extractors that work for the sum of two (or more) independent weak random sources. By using existing constructions of such extractors in [11], they give an explicit function Ext such that $\mathsf{SROLBP}_{n^{-\Omega(1)}}(\mathsf{Ext}) \geq 2^{n-\log^{O(1)} n}$, i.e., the branching program size lower bound becomes close to optimal, but the correlation increases from exponentially small to polynomially large. Similarly, [12] left open the question of obtaining improved average-case hardness against SROLBPs in the small correlation regime.

We remark that directional affine extractors are a special case of *affine non-malleable extractors*, which are defined by Chattopadhyay and Li [10]. Roughly, an affine non-malleable extractor is an affine extractor such that the output is still close to uniform, even conditioned on the output of the extractor where the input affine source is modified by any affine function with no fixed points.

In this context, directional affine extractors just correspond to the case where the tampering function adds a non-zero affine shift to the source. Previously, the best affine non-malleable extractor due to Li [32] works for entropy $k \geq (1-\gamma)n$ for some small constant $\gamma < 1/3$ with error $2^{-\Omega(n)}$. Thus this does not give a better construction of directional affine extractors. However, [32] does give an improved sumset extractor, which yields an explicit function Ext such that $\mathsf{SROLBP}_\varepsilon(\mathsf{Ext}) \geq 2^{n-O(\log n)}$ for any constant $\varepsilon > 0$, i.e., the branching program size lower bound becomes optimal up to the constant in $O(.)$, but the correlation increases to any constant.

## 1.1 Our Results

In this paper, we present a much more in-depth study of directional affine extractors, affine non-malleable extractors, SROLBPs, and standard ROBPs. To begin with, we observe that it is not a prior clear that SROLBPs are more powerful than standard ROBPs. Indeed, it is easy to see that $\mathsf{AC}^0[\oplus]$ and parity decision trees are exponentially more powerful than $\mathsf{AC}^0$ circuits and standard decision trees, respectively, since parity requires exponential size $\mathsf{AC}^0$ circuits and decision trees. However, any parity function can be computed by an ROBP of size $O(n)$. Nevertheless, there are previous works [34, 25, 20] which showed that computing

explicit characteristic functions of certain affine subspaces require ROBPs of size $2^{\Omega(n)}$ (e.g., the satisfiable Tseitin formulas in [20]). Since such functions are easily computable by an SROLBP of size $O(n)$, this provides a separation between SROLBP and ROBP and shows that indeed SROLBPs are exponentially more powerful than ROBPs.

In turn, this further demonstrates that directional affine extractors have stronger properties than standard affine extractors, as they imply strong lower bounds for SROLBPs. Next, we give explicit constructions of directional affine extractors with much better parameters than that in [22]. Our construction works for any linear entropy with exponentially small error.

▶ **Theorem 8.** *For any constant $0 < \delta \leq 1$, there exists a family of explicit directional affine extractors $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ for entropy $k \geq \delta n$ with error $\varepsilon = 2^{-\Omega(n)}$ and output length $m = \Omega(n)$.*

In fact, our construction can work for slightly sub-linear entropy.

▶ **Theorem 9.** *There exists a constant $c > 1$ and an explicit family of directional affine extractors $\mathsf{DAExt} : \{0,1\}^n \to \{0,1\}^m$ for entropy $k \geq cn(\log\log\log n)^2/\log\log n$ with error $\varepsilon = 2^{-n^{\Omega(1)}}$ and output length $m = n^{\Omega(1)}$, as well as an explicit family of directional affine dispersers for entropy $k \geq cn(\log\log n)^2/\log n$ with $m = n^{\Omega(1)}$.*

This theorem immediately gives much improved average-case hardness for SROLBPs.

▶ **Theorem 10.** *There is an explicit function DAExt such that $\mathsf{SROLBP}_{2^{-n^{\Omega(1)}}}(\mathsf{DAExt}) \geq 2^{n-\widetilde{O}(\frac{n}{\log\log n})}$, where $\tilde{O}(.)$ hides $(\log\log\log n)^2$ factors.*

In particular, we can achieve exponentially small correlation while obtaining a $2^{n-o(n)}$ size lower bound for SROLBPs, which is almost optimal. This significantly improves the $2^{n/3-o(n)}$ size lower bound in [22] and the polynomially large correlation in [12]. Thus, Theorem 9 and 10 provide positive answers to the two open questions in [22] and [12] mentioned before.

We remark that under our new definition, a directional affine extractor is strictly stronger than a standard affine extractor. Thus Theorem 9 also improves the entropy requirement of negligible error affine extractors, from the previously best-known result of $\frac{n}{\sqrt{\log\log n}}$ [42, 29] to $\frac{cn(\log\log\log n)^2}{\log\log n}$.

We also revisit the hardness results for standard ROBPs. As mentioned before, exponential and even close to optimal size lower bounds are known for explicit functions in this model, where the current best result is an explicit function that requires ROBPs (in fact, SROLBPs) of size $2^{n-O(\log n)}$ [32]. However, there has also been a lot of interest in finding functions in lower complexity classes that give strong lower bounds for ROBPs. It is clear that the class $\mathsf{NC}^0$ is not sufficient. Thus the next possible class is $\mathsf{AC}^0$. Indeed there are previous works giving explicit $\mathsf{AC}^0$ functions that require ROBPs of size $2^{\Omega(\sqrt{n})}$[24, 27, 19, 5] and even $2^{\Omega(n)}$ [20], yet there is no average-case hardness as far as we know. Here, we improve both the size lower bound and the average-case hardness by giving an explicit $\mathsf{AC}^0$ function that has negligible correlation with ROBPs of size $2^{(1-\delta)n}$ for any constant $\delta > 0$.

▶ **Theorem 11.** *For any constant $\delta > 0$ there is an explicit function $\mathsf{AC}^0$-Ext in $\mathsf{AC}^0$ such that $\mathsf{ROBP}_{2^{-\mathrm{poly}\log n}}(\mathsf{AC}^0\text{-Ext}) \geq 2^{(1-\delta)n}$.*

One of the key ingredients in our constructions is a new linear somewhere condenser for affine sources. Specifically, we have

▶ **Definition 12.** *For any $0 < \delta < \gamma < 1$, a function $\mathsf{SCond} : \mathsf{F}_2^n \to (\mathsf{F}_2^m)^\ell$ is a $(\delta, \gamma)$ affine somewhere condenser, if it satisfies the following property: for any affine source $X$ over $\mathsf{F}_2^n$ with entropy $\delta n$, let $(Y_1, \cdots, Y_\ell) = \mathsf{SCond}(X) \in (\mathsf{F}_2^m)^\ell$, then there exists at least one $i \in [\ell]$ such that $Y_i$ is an affine source over $\mathsf{F}_2^m$ with entropy at least $\gamma m$.*

▶ **Theorem 13.** *There exists a constant $\beta > 0$ such that for any $0 < \delta \leq 1/2$, there is an explicit $(\delta, 1/2 + \beta)$ affine somewhere condenser $\mathsf{SCond} : \mathsf{F}_2^n \to (\mathsf{F}_2^m)^t$, where $t = \mathsf{poly}(1/\delta)$ and $m = n/\mathsf{poly}(1/\delta)$. Moreover, $\mathsf{SCond}$ is a linear function.*

We further show that (a slight modification of) this condenser works for general weak random sources, under the well-known Polynomial Freiman-Ruzsa Theorem in $\mathsf{F}_2^n$, once one of the most important conjectures in additive combinatorics and very recently proved by Gowers, Green, Manners, and Tao [21].

Previously, all condensers of this kind are based on sum-product theorems, and the function is a polynomial with degree $\mathsf{poly}(1/\delta)$ [3, 38, 44]. In contrast, there exist constructions of linear *seeded* extractors, where if one lists the outputs of the extractor for all possible seeds, then we get a somewhere random source such that at least one output is close to uniform, and the function is a linear function. However, in many applications such as ours, one needs to use a somewhere condenser instead of simply listing all outputs of an extractor, since the former only gives a small number (e.g., a constant) of outputs as opposed to $\mathsf{poly}(n)$ outputs from the extractor. Hence, our linear somewhere condenser complements the existing sum-product theorem based somewhere condensers. Moreover, our construction of the condenser is based on *dimension expanders*, which are algebraic pseudorandom objects previously studied based on their own interests, with no clear applications in computer science as far as we know. Thus, our construction can be viewed as one of the first applications of dimension expanders in computer science.

Finally, we study the question of whether directional affine extractors can give strong lower bounds for the class of $\mathsf{AC}^0[\oplus]$ in a black box way. Cohen and Tal [15] showed via probablistic methods that standard affine extractors do not suffice since depth-3 $\mathsf{AC}^0[\oplus]$ circuits can compute optimal affine extractors. Using a slightly modified argument as that in [15], we show that even the stronger version of directional affine extractors does not suffice. Specifically, depth-3 $\mathsf{AC}^0[\oplus]$ circuits can also compute optimal directional affine extractors. This in turn provides a strong separation of $\mathsf{AC}^0[\oplus]$ from $\mathsf{SROLBP}$.

▶ **Theorem 14.** *There exists a function $f : \{0,1\}^n \to \{0,1\}$ which is a directional affine extractor for entropy $k$ with error $\varepsilon$, where $k = \log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + O(1)$ such that the following properties hold.*

1. *$f$ is a polynomial of degree $\log \frac{n}{\varepsilon^2} + \log\log \frac{n}{\varepsilon^2} + O(1)$.*
2. *$f$ can be realized by a $\mathsf{XOR\text{-}AND\text{-}XOR}$ circuit of size $O((n/\varepsilon)^2 \cdot \log^3(n/\varepsilon))$.*
3. *$f$ can be realized by a De Morgan formula of size $O((n^5/\varepsilon^2) \cdot \log^3(n/\varepsilon))$.*

## 2 Overview of the Techniques

Here we give a sketch of the main ideas used in this paper. For clarity, we shall be informal at places and ignore some technical details.

### 2.1 Directional affine extractors

Our starting point is the construction of affine extractors by Li [29], which works for sub-linear entropy with exponentially small error. We first briefly recall the construction there. Divide an affine source $X$ of entropy rate $\delta$ into $O(1/\delta)$ blocks. By choosing the size of the blocks appropriately, one can show that there exists a "good" block $X_g$ of entropy rate $\Omega(\delta)$, and the source $X$ still has a lot of entropy conditioned on $X_g$ (i.e., we get an affine *block source*). If we know the position of $X_g$, randomness extraction is easy: we apply a somewhere condenser (e.g., those in [3, 38, 44]) to condense $X_g$ into a matrix with a constant number of rows,

such that at least one row has entropy rate $1 - \delta/2$. At this point, we can apply a linear two-source extractor (e.g., the inner product function) to each row of the matrix and the source $X$ to get an affine *somewhere random* source, conditioned on the fixing of $X_g$. This is another matrix with a constant number of rows, such that at least one row is uniform, and one can apply existing techniques to deterministically extract random bits from this source [37].

However, when $\delta$ is small, we don't know which block $X_g$ is good. Thus in [29], the construction tries all blocks, and then combines them together. To make this process work, the construction crucially maintains the following property: (*) for each block $X_i$, the output bits produced from this block are constant degree polynomials of the input bits, and the degrees decrease geometrically from the first block to the last block. With this property, the analysis goes by focusing on the first good block $X_g$. Notice that we can fix all the outputs produced from blocks before $X_g$, while all outputs produced from blocks after $X_g$ have degrees less than those from $X_g$. Thus if we take the XOR of all these outputs, an XOR lemma of polynomials [40, 4] guarantees the final output is still close to uniform. We note that the XOR lemma of polynomials only works for degree up to $\log n$. Hence it is important to keep the degree $c$ of the outputs from each block to be as small as possible. Roughly, we will need $c^{O(1/\delta)} < \log n$.

Our strategy now is to adapt this construction to directional affine extractors. Towards this, we use techniques from constructions of non-malleable extractors since, as we remark before, directional affine extractors are a special case of affine non-malleable extractors. Recent constructions of non-malleable extractors usually consist of two steps: first, generate a small advice that is different from the tampered version with high probability, and then use the advice together with other tools (e.g., correlation breakers) to achieve non-malleability. Thus, our goal is to adapt these two steps to directional affine extractors while, at the same time, still maintaining property (*), which is crucial to achieving any linear entropy or slightly sub-linear entropy. We now explain both steps.

As before, for each block $X_i$ we will get an output $U_i$, which is close to uniform if $X_i$ is a good block. Divide $U_i$ into two parts $U_i = U_{i1} \circ U_{i2}$. We will use $U_{i1}$ to generate the advice and $U_{i2}$ for the rest of the construction. Notice that from the tampered input $X' = X + a$ we also have a tampered version $U'_i = U'_{i1} \circ U'_{i2}$. In the following, we will always use letters with prime to denote the corresponding random variables produced from the tampered input. If $U_{i1} \neq U'_{i1}$ then we are done, otherwise we use $U_{i1} = U'_{i1}$ to sample some $\Omega(\delta^2 n)$ bits $H_i$ from an encoding of $X$, using an asymptotically good binary linear code. Since $X' = X + a$, we have that $H_i + H'_i$ basically corresponds to the sampled bits from the encoding of $a$. Thus $H_i \neq H'_i$ with high probability by the distance of the linear code. However, we cannot just do sampling naively since we need to keep the degree to be a constant. Therefore, we also divide both $U_{i1}$ and the encoding of $X$ into $\Omega(\delta^2 n)$ blocks where each block contains a constant number of bits, and use each block of $U_{i1}$ to sample one bit from the corresponding block of the encoding of $X$. By the distance property of the code, there are $\Omega(\delta^2 n)$ blocks of the encoding of $X$ and $X'$ that are different. Thus we still have $H_i \neq H'_i$ with high probability, and now each bit of $H_i$ is a constant degree polynomial of the bits of $U_{i1}$ and $X$. The advice string is now $U_{i1} \circ H_i$.

Once we have the advice, we can append it to another string extracted from $X$ by using a linear seeded extractor and $U_{i2}$ as the seed. Now notice that the string produced from $X$ is different from the string produced from $X'$ with high probability, and they are linearly correlated conditioned on the fixing of $(U_i, U'_i)$. Thus we can apply, for example, a known affine non-malleable extractor (the state-of-the-art affine non-malleable extractor

with negligible error only works for high entropy). However, the known construction of affine non-malleable extractor in [10] has super constant degree. Indeed, even one application of this extractor results in a polynomial of degree larger than $\log n$, which already defeats our purpose to get a directional affine extractor (we can still get a directional affine disperser, though).

To solve this problem, we develop new ideas that make use of the special structure of $X' = X + a$. Recall that in our construction, for every block $X_i$ we get a $U_{i2}$, which is close to uniform if $X_i$ is good, and $X$ still has enough entropy conditioned on $X_i$. Our idea now is to use a *seeded non-malleable extractor* $\mathsf{snmExt}$ instead, which is an extractor with a uniform random seed, such that if an adversary tampers with the seed but not the source, then the output of the extractor on the original inputs is close to uniform given the output on the tampered inputs. By appending the advice string to $U_{i2}$ and getting $\tilde{U}_i = U_i \circ H_i$, we have $\tilde{U}_i \neq \tilde{U}'_i$ with high probability, and the seed $\tilde{U}_i$ has high entropy if $H_i$ has small size, which suffices for the seeded non-malleable extractor as long as the extractor is strong. Now, if the seeded non-malleable extractor is also *linear* conditioned on any fixing of the seed, then we have $\mathsf{snmExt}(X', \tilde{U}'_i) = \mathsf{snmExt}(X, \tilde{U}'_i) + \mathsf{snmExt}(a, \tilde{U}'_i)$. Since $\mathsf{snmExt}(X, \tilde{U}_i)$ is close to uniform given $\mathsf{snmExt}(X, \tilde{U}'_i)$ (because it is a non-malleable extractor), and the extractor is strong (we can fix the seeds $(\tilde{U}_i, \tilde{U}'_i)$), this implies that $\mathsf{snmExt}(X, \tilde{U}_i)$ is close to uniform given $\mathsf{snmExt}(X', \tilde{U}'_i)$. [2]

Luckily, there are previous constructions of linear seeded non-malleable extractors due to Li [30], which are based on the inner product function. Moreover, this extractor also has the property that each output bit is a constant degree polynomial of the input bits. Thus everything seems to work out, except for one problem: the non-malleable extractor in [30] only works when the source has entropy rate $> 1/2$, but here our goal is to work for any linear (or slightly sub-linear) entropy. A natural idea would be to use the somewhere condenser (e.g., in [3, 38, 44]) to boost the entropy rate of $X$. However, all known condensers of this kind are based on sum-product theorems, which are non-linear functions, and applying them changes the structure of $X' = X + a$, which is important for our construction. Another idea is to apply a linear seeded extractor to $X$ and try all possible seeds. This indeed keeps the structure of $X' = X + a$, but will result in a $\mathsf{poly}(n)$ number of outputs, and combining them together will result in a polynomial of large, super constant degree.

This motivates another key ingredient in our construction, a new linear somewhere condenser for affine sources. In short, we construct a linear function which, given any affine source on $n$ bits with entropy rate $0 < \delta \leq 1/2$, outputs $\mathsf{poly}(1/\delta)$ rows such that each row has $n/\mathsf{poly}(1/\delta)$ bits, and at least one row has entropy rate $1/2 + \beta$ for some absolute constant $\beta > 0$. This complements the sum-product based somewhere condensers, and can be viewed as a separate contribution of our work. We will explain the construction of this condenser later, but finish the description of our directional affine extractor here, assuming that we have the linear somewhere condenser.

The rest of the construction roughly goes as follows. We apply the linear somewhere condenser to the source $X$ to get a constant number of rows, then apply $\mathsf{snmExt}$ to each row using $\tilde{U}_i$ as the seed. Thus we get a constant number of outputs such that at least one of them is close to uniform conditioned on the corresponding tampered output. Now we apply an *affine correlation breaker* such as those in [31, 8, 11] to further break the correlations between different outputs, and combine these outputs together by taking the XOR. The

[2] The actual analysis involves more details since here $X$ is not independent of $(\tilde{U}_i, \tilde{U}'_i)$, but the property still holds due to the affine structure. We omit the details here.

correlation breaker guarantees that the final output is close to uniform conditioned on the tampered output. To keep the degree small, we need to replace all seeded extractors used in the correlation breaker with a constant degree linear seeded extractor in [29]. This keeps the output bits to be constant degree polynomials of the input bits, and the remaining construction is essentially the same as that in [29].

## 2.2 Linear somewhere condenser

We now describe our construction of the linear somewhere condenser. This is based on another pseudorandom object known as *dimension expander*. Informally, a dimension expander is a set of linear mappings from a vector space $\mathsf{F}^n$ to itself, such that for any linear subspace $V \subset F^n$ with small dimension $k \leq n/2$, the span of the union of all the images of $V$ under the set of linear mappings has dimension at least $(1+\alpha)k$ for some absolute constant $\alpha > 0$. Readers familiar with expander graphs can see that this is a linear algebraic analog of expander graphs. Thus, it is desirable to give explicit constructions of the set of linear mappings which has as few number of mappings as possible, where this number $d$ is called the degree. Dimension expanders were first introduced by Barak, Impagliazzo, Shpilka, and Wigderson [2], who also showed the existence of such objects. Later, Bourgain and Yehudayoff [6, 7] gave explicit constructions of dimension expanders with degree $d = O(1)$ over any field. Interestingly, as far as we know, there are no previous applications of dimension expanders in computer science, and they are mainly studied based on their own interests and connections to other algebraic pseudorandom objects. Thus our construction can be viewed as one of the first applications of dimension expanders in computer science.

Given an explicit dimension expander $\{T_i\}_{i\in[d]}$ where each $T_i$ is a linear mapping, and any affine source $X$ with entropy rate $\delta \leq 1/2$, we first construct a basic somewhere condenser as follows. Divide $X$ equally into $X = X_1 \circ X_2$, and our condenser produces $2d+2$ outputs: $(X_1, X_2, \{X_1 + T_i(X_2)\}_{i\in[d]}, \{T_i(X_1) + X_2\}_{i\in[d]})$. We show that at least one output has entropy rate $(1+\gamma)\delta$ for some constant $\gamma > 0$, and we give some intuition below. By the structure of affine sources, one can show that there exists another affine source $X_3$ independent of $X_1$ such that $X_2 = X_3 + L(X_1)$ for some linear function $L$. Let $H(X_1) = s$, $H(X_3) = r$ and $H(L(X_1)) = t$, then we have $s + r = \delta n$. If either $s$ or $r$ is small, e.g., $s \ll \delta n/2$, then we must have $r \gg \delta n/2$ and thus $H(X_2) = r + t \geq (1+\gamma)\delta n/2$. Therefore the entropy rate of $X_2$ is at least $(1+\gamma)\delta$. The case of $r \ll \delta n/2$ is similar. Hence, we only need to consider the case where $s \approx \delta n/2$ and $r \approx \delta n/2$, and notice that we must have either $s \leq \delta n/2$ or $r \leq \delta n/2$. Furthermore, in this case, $t$ must be small, since otherwise, we would again have $H(X_2) = r + t \geq (1+\gamma)\delta n/2$.

For simplicity, assume that $s = r = \delta n/2$, and $t = 0$. Hence both $X_1$ and $X_2$ have entropy rate $\delta \leq 1/2$, and they are independent. Without loss of generality, assume the supports of both $X_1$ and $X_2$ are linear subspaces. By the property of the dimension expander, $\mathsf{Span}(\cup_{i\in[d]} T_i(X_1))$ has dimension at least $(1+\alpha)\delta n/2$. We now argue that there exists an $i \in [d]$ such that the support of $T_i(X_1) + X_2$ has dimension at least $(1+\alpha/d)\delta n/2$, which implies that $T_i(X_1) + X_2$ has entropy rate at least $(1+\alpha/d)\delta$. To see this, assume otherwise, then for any $i \in [d]$, any vector in the support of $T_i(X_1) + X_2$ can be expressed as a linear combination of the $r = \delta n/2$ basis vectors in the support of $X_2$ and $< (\alpha/d)\delta n/2$ other vectors. This implies that $\mathsf{Span}(\cup_{i\in[d]} T_i(X_1))$ has dimension $< \delta n/2 + d \cdot (\alpha/d)\delta n/2 = (1+\alpha)\delta n/2$, since any vector in $\mathsf{Span}(\cup_{i\in[d]} T_i(X_1))$ can be expressed as a linear combination of the $r = \delta n/2$ basis vectors in the support of $X_2$ and $< d \cdot (\alpha/d)\delta n/2$ other vectors. This contradicts the property of the dimension expander.

Thus, in all cases, we get the desired entropy rate boost. Our final somewhere condenser involves repeated uses of the basic condenser, as in previous works. It is easy to see that the entropy rate of at least one output will increase to $1/2 + \beta$ for some absolute constant $\beta > 0$ after $O(\log(1/\delta))$ uses of the basic condenser. The number of outputs is, therefore, $\mathsf{poly}(1/\delta)$ and each output has $n/\mathsf{poly}(1/\delta)$ bits. Finally, it is clear that the condenser is a linear function.

Once we have this linear condenser, we can even replace the somewhere condensers used in [29] by the new condenser. This further reduces the degree of the polynomials of the output bits (since previous somewhere condensers are polynomials instead of linear functions). Therefore we can push the entropy requirement of our directional affine extractor to be even better than that in [29], from $\frac{n}{\sqrt{\log \log n}}$ to $\frac{cn(\log \log \log n)^2}{\log \log n}$.

We show that a slight modification of our linear condenser also works for general weak random sources, under the Polynomial Freiman-Ruzsa Theorem. Roughly, the idea is to use a careful analysis of subsources and collision probability. Specifically, it is known that if the collision probability of a distribution is small, then the distribution is close to having high min-entropy. On the other hand, if the collision probability is large, then (without loss of generality) assuming the distribution is the uniform distribution over some unknown subset, existing results in additive combinatorics imply that there is a large subset $A$ in the support of the distribution such that the size of $A + A$ is not much larger than $A$. The Polynomial Freiman-Ruzsa Theorem then implies that there is another large subset $A' \subset A$ which is "close" to an affine subspace, which roughly reduces the analysis to the case of affine sources.

## 2.3 $\mathsf{AC}^0$ average-case hardness for ROBPs

To show $\mathsf{AC}^0$ average-case hardness for ROBPs, we use a standard observation that if one conditions on an inner node, then the input bits priori to this node and the input bits after this node are still independent. We then construct an appropriate extractor in $\mathsf{AC}^0$, which we call $\mathsf{AC}^0$-Ext, for sources with such a structure. Specifically, given any ROBP of size $s$ and any constant $\delta > 0$, we can find a cut or anti-chain (a maximal subset of vertices such that none of which is an ancestor of any other vertex) of size $O(s)$ at roughly depth $\delta n$ above the sinks, so that conditioned on the fixing of any vertex in the cut, the input uniform random string $X$ now becomes two independent weak sources $A$ and $B$, where $A$ corresponds to the first part of the program and $B$ corresponds to the second part. Since we don't know the order of bits queried by the ROBP, the bits of the two sources are interleaved, and we view $X = A + B$. Using a standard averaging argument, one can show that with high probability, the following properties are satisfied: (1) $A$ and $B$ are supported on disjoint subsets of input bits; (2) $A$ has min-entropy roughly $(1 - \delta)n - \log s$ and $B$ has min-entropy $\delta n$; and (3) $B$ is an oblivious bit-fixing source, which is obtained by fixing some unknown bits in a uniform random string. If $s \leq 2^{(1-2\delta)n}$ then both $A$ and $B$ have entropy rate roughly $\delta$. Now, our goal is to construct an extractor in $\mathsf{AC}^0$ for sources with this structure, that is also *strong* in $B$. This means that even if we condition on the fixing of the vertex in the cut and $B$, the output of the extractor is still close to uniform. On the other hand, the output of the ROBP is completely determined by the vertex and $B$. Thus our extractor is average-case hard for ROBPs of size up to $2^{(1-2\delta)n}$.

As usual, the function $\mathsf{AC}^0$-Ext will be compositions of different, more basic extractors as building blocks. Thus we need all these building blocks to be computable in $\mathsf{AC}^0$. Here, we leverage the constructions from two previous works on extractors in $\mathsf{AC}^0$: (1) the $\mathsf{AC}^0$-computable extractors $\mathsf{AC}^0$-BFExt for bit-fixing source by Cheng and Li [13], and (2) the $\mathsf{AC}^0$-computable strong linear seeded extractors $\mathsf{AC}^0$-LExt by Papakonstantinou, Woodruff, and Yang [35].

Now we can describe our main idea of construction. Divide $X$ into $t = O(1/\delta)$ blocks, and by an averaging argument, there exists a block $B_g$ of $B$ with entropy rate $\Omega(\delta)$. Now for the block $X_g = A_g + B_g$, we can fix $A_g$ so that $X_g$ is an oblivious bit-fixing source of entropy rate $\Omega(\delta)$ and is a deterministic function of $B$. We next fix the bits from $B$ outside of the $g$-th block so that the source $X$ outside of $X_g$ is a deterministic function of $A$ and thus independent of $X_g$. Moreover, $A$ and $X$ still have enough entropy left.

Applying the above-mentioned extractor $\mathsf{AC}^0$-$\mathsf{BFExt}$ for bit-fixing sources to each block $X_i$, we convert $X$ into a *somewhere random source* $Y = Y_1 \circ \cdots \circ Y_t$ where the row $Y_g$ is a deterministic function of $B_g$ and close to uniform, while all the other rows are deterministic functions of $A$. At this point, we can simply take the XOR of the $Y_i$'s to obtain a close-to-uniform output. However, as mentioned before, we need the extractor to be strong in $B$ and this simple approach is not sufficient. Instead, we fix all the outputs produced by $\mathsf{AC}^0$-$\mathsf{BFExt}$ for $X_i$ where $i \neq g$. Note that these are all deterministic functions of $A$. Thus conditioned on this fixing, $Y$ becomes a deterministic function of $B$, which is independent of $A$. Moreover, as long as the output size of $\mathsf{AC}^0$-$\mathsf{BFExt}$ is not too large, $A$ still has enough entropy left. Since $X = A + B$, we can now apply a *strong* $t$-affine correlation breaker as in [31, 11] with each $Y_i$ as the seed to extract from $X$ a random string, and take the XOR of them. The property of the correlation breaker guarantees that the string produced from $Y_g$ and $X$ is close to uniform conditioned on all the other outputs and $Y$. Hence the XOR is also close to uniform conditioned on $B$. To ensure the correlation breaker is computable in $\mathsf{AC}^0$, we replace all the strong (linear) seeded extractors in the known constructions of $t$-affine correlation breakers with the above-mentioned $\mathsf{AC}^0$-$\mathsf{LExt}$. Since $t = O(1/\delta)$ is a constant, the correlation breaker involves a constant number of compositions of $\mathsf{AC}^0$-$\mathsf{LExt}$, which is still in $\mathsf{AC}^0$.

## 3 Open Problems

Our work leaves several natural open problems. The most obvious is to further improve the constructions of directional affine extractors and the average-case hardness for $\mathsf{SROLBP}$s. It would also be quite interesting to show any hardness of explicit functions for $\mathsf{WROLBP}$s, which appears to require new ideas. Finally, it is an interesting question to see if there exist functions in $\mathsf{AC}^0$ that achieve optimal hardness for $\mathsf{ROBP}$s, or strong hardness for $\mathsf{SROLBP}$s.

## References


1. Alexander E. Andreev, Juri L. Baskakov, Andrea E. F. Clementi, and José D. P. Rolim. Small pseudo-random sets yield hard functions: New tight explict lower bounds for branching programs. In Jirí Wiedermann, Peter van Emde Boas, and Mogens Nielsen, editors, *Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings*, volume 1644 of *Lecture Notes in Computer Science*, pages 179–189. Springer, 1999. doi:10.1007/3-540-48523-6_15.
2. Boaz Barak, Russel Impagliazzo, Amir Shpilka, and Avi Wigderson. Definition and existence of dimension expanders. Discussion (no written record), 2004.
3. Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2005.
4. Arnab Bhattacharyya, Swastik Kopparty, Grant Schoenebeck, Madhu Sudan, and David Zuckerman. Optimal testing of reed-muller codes. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 488–497. IEEE Computer Society, 2010. doi:10.1109/FOCS.2010.54.

5 Beate Bollig and Ingo Wegener. A very simple function that requires exponential size read-once branching programs. *Inf. Process. Lett.*, 66(2):53–57, 1998. doi:10.1016/S0020-0190(98)00042-8.

6 Jean Bourgain. Expanders and dimensional expansion. *Comptes Rendus Mathematique*, 347(7):357–362, 2009.

7 Jean Bourgain and Amir Yehudayoff. Expansion in $SL_2(\mathbb{R})$ and monotone expanders. *Geometric and Functional Analysis*, 23(1):1–41, 2013.

8 Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. Affine extractors for almost logarithmic entropy. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 622–633, 2022. doi:10.1109/FOCS52979.2021.00067.

9 Eshan Chattopadhyay and Xin Li. Extractors for sumset sources. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC, Cambridge, MA, USA, June 18-21, 2016*, pages 299–311. ACM, 2016. doi:10.1145/2897518.2897643.

10 Eshan Chattopadhyay and Xin Li. Non-malleable codes and extractors for small-depth circuits, and affine functions. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 1171–1184. ACM, 2017. doi:10.1145/3055399.3055483.

11 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. doi:10.1145/3519935.3519963.

12 Eshan Chattopadhyay and Jyun-Jie Liao. Hardness against linear branching programs and more. In *Proceedings of the Conference on Proceedings of the 38th Computational Complexity Conference*, CCC '23, Dagstuhl, DEU, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

13 Kuan Cheng and Xin Li. Randomness extraction in AC0 and with small locality. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 37:1–37:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.APPROX-RANDOM.2018.37.

14 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. doi:10.1145/2840728.2840734.

15 Gil Cohen and Avishay Tal. Two structural results for low degree polynomials and applications. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2015*, 2015.

16 Evgeny Demenkov and Alexander Kulikov. An elementary proof of 3n-o(n) lower bound on the circuit complexity of affine dispersers. In *Proceedings of the 36th international conference on Mathematical foundations of computer science*, pages 256–265, 2011.

17 Paul E. Dunne. Lower bounds on the complexity of 1-time only branching programs. In Lothar Budach, editor, *Fundamentals of Computation Theory, FCT '85, Cottbus, GDR, September 9-13, 1985*, volume 199 of *Lecture Notes in Computer Science*, pages 90–99. Springer, 1985. doi:10.1007/BFb0028795.

18 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 *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 89–98, 2016. doi:10.1109/FOCS.2016.19.

19 Anna Gál. A simple function that requires exponential size read-once branching programs. *Inf. Process. Lett.*, 62(1):13–16, 1997. doi:10.1016/S0020-0190(97)00041-0.

**20** Ludmila Glinskih and Dmitry Itsykson. Satisfiable Tseitin Formulas Are Hard for Nondeterministic Read-Once Branching Programs. In Kim G. Larsen, Hans L. Bodlaender, and Jean-Francois Raskin, editors, *42nd International Symposium on Mathematical Foundations of Computer Science (MFCS 2017)*, volume 83 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.MFCS.2017.26.

**21** W. T. Gowers, Ben Green, Freddie Manners, and Terence Tao. On a conjecture of marton, 2023. arXiv:2311.05762.

**22** Svyatoslav Gryaznov, Pavel Pudlák, and Navid Talebanfard. Linear Branching Programs and Directional Affine Extractors. In *37th Computational Complexity Conference (CCC 2022)*, volume 234, pages 4:1–4:16, 2022.

**23** Johan Håstad. Almost optimal lower bounds for small depth circuits. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 6–20. ACM, 1986. doi:10.1145/12130.12132.

**24** Stasys Jukna. Entropy of contact circuits and lower bounds on their complexity. *Theor. Comput. Sci.*, 57:113–129, 1988. doi:10.1016/0304-3975(88)90166-1.

**25** Stasys Jukna. A note on read-k times branching programs. *RAIRO - Theoretical Informatics and Applications*, 28:75–83, January 1995.

**26** Valentine Kabanets. Almost k-wise independence and hard boolean functions. *Theor. Comput. Sci.*, 297(1-3):281–295, 2003. doi:10.1016/S0304-3975(02)00643-6.

**27** Matthias Krause, Christoph Meinel, and Stephan Waack. Separating the eraser turing machine classes $L_e$, $NL_e$, co-$NL_e$ and $P_e$. *Theor. Comput. Sci.*, 86(2):267–275, 1991. doi:10.1016/0304-3975(91)90021-S.

**28** Jiatu Li and Tianqi Yang. 3.1n - o(n) circuit lower bounds for explicit functions. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2022, pages 1180–1193, New York, NY, USA, 2022. Association for Computing Machinery. doi:10.1145/3519935.3519976.

**29** Xin Li. A new approach to affine extractors and dispersers. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC*, 2011.

**30** Xin Li. Non-malleable extractors, two-source extractors and privacy amplification. In *2012 IEEE 53rd Annual Symposium on Foundations of Computer Science*, 2012.

**31** Xin Li. Improved non-malleable extractors, non-malleable codes and independent source extractors. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, STOC 2017, pages 1144–1156, New York, NY, USA, 2017. Association for Computing Machinery.

**32** Xin Li. Two source extractors for asymptotically optimal entropy, and (many) more. Technical report, Arxiv, 2023. arXiv:2303.06802.

**33** E. I. Nechiporuk. On a boolean function. *Doklady of the Academy of Sciences of the USSR*, 164(4):765–766, 1966.

**34** EA Okolnishnikova. On lower bounds for branching programs. *Siberian Advances in Mathematics*, 3:152–156, January 1993.

**35** Periklis A Papakonstantinou, David P Woodruff, and Guang Yang. True randomness from big data. *Scientific reports*, 6:33740, 2016.

**36** Stephen Ponzio. A lower bound for integer multiplication with read-once branching programs. *SIAM Journal on Computing*, 28(3):798–815, 1998. doi:10.1137/S0097539795290349.

**37** Anup Rao. Extractors for low-weight affine sources. In *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*, CCC '09, pages 95–101. IEEE Computer Society, 2009.

**38** Ran Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

**39** Janos Simon and Mario Szegedy. A new lower bound theorem for read-only-once branching programs and its applications. In *Advances In Computational Complexity Theory*, 1992.

40 Emanuele Viola and Avi Wigderson. Norms, xor lemmas, and lower bounds for polynomials and protocols. *Theory of Computing*, 4(7):137–168, 2008. doi:10.4086/toc.2008.v004a007.

41 Ingo Wegener. On the complexity of branching programs and decision trees for clique functions. *J. ACM*, 35(2):461–471, 1988. doi:10.1145/42282.46161.

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

43 Stanislav Zák. An exponential lower bound for one-time-only branching programs. In Michal Chytil and Václav Koubek, editors, *Mathematical Foundations of Computer Science 1984, Praha, Czechoslovakia, September 3-7, 1984, Proceedings*, volume 176 of *Lecture Notes in Computer Science*, pages 562–566. Springer, 1984. doi:10.1007/BFb0030340.

44 David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Theory of Computing*, 2007.