2016 IEEE 57th Annual Symposium on Foundations of Computer Science

# Improved Two-Source Extractors, and Affine Extractors for Polylogarithmic Entropy

Xin Li
*Department of Computer Science*
*Johns Hopkins University*
*lixints@cs.jhu.edu*

***Abstract*—In a recent breakthrough [1], Chattopadhyay and Zuckerman gave an explicit two-source extractor for min-entropy $k \geq \log^C n$ for some large enough constant $C$, where $n$ is the length of the source. However, their extractor only outputs one bit. In this paper, we improve the output of the two-source extractor to $k^{\Omega(1)}$, while the error remains $n^{-\Omega(1)}$ and the extractor remains strong in the second source. In the non-strong case, the output can be increased to $k$. Our improvement is obtained by giving a better extractor for $(q, t, \gamma)$ non-oblivious bit-fixing sources, which can output $t^{\Omega(1)}$ bits instead of one bit as in [1].**

**We also give the first explicit construction of deterministic extractors for affine sources over $\mathbb{F}_2$, with entropy $k \geq \log^C n$ for some large enough constant $C$, where $n$ is the length of the source. Previously the best known results are by Bourgain [2], Yehudayoff [3] and Li [4], which require the affine source to have entropy at least $\Omega(n/\sqrt{\log \log n})$. Our extractor outputs $k^{\Omega(1)}$ bits with error $n^{-\Omega(1)}$. This is done by reducing an affine source to a non-oblivious bit-fixing source, where we adapt the alternating extraction based approach in previous work on independent source extractors [5] to the affine setting. Our affine extractors also imply improved extractors for circuit sources studied in [6].**

**We further extend our results to the case of zero-error dispersers, and give two applications in data structures that rely crucially on the fact that our two-source or affine extractors have large output size.**

***Keywords*-extractor; affine; two-source;**

## I. Introduction

Randomness extraction is a broad area that studies the problem of converting biased random sources into nearly uniform random bits. The natural motivation comes from the wide application of randomness in computation, such as in algorithms, distributed computing and cryptography, and the requirement that the random bits used should be uniformly distributed. In reality, however, natural random sources almost always have serious biases, and can leak information to an adversary because of side channel attacks. These defective random sources are known as weak random sources. Therefore, intuitively, a randomness extractor takes as input one or more weak random sources, and outputs a distribution that is statistically close to uniform.

Formally, a weak random source is modeled as a probability distribution over $n$ bit strings with some entropy $k$. In the context of randomness extraction, the standard measure of entropy is the so called *min-entropy*, which is defined as follows.

**Definition I.1.** The *min-entropy* of a random variable $X$ is

$$H_\infty(X) = \min_{x \in \mathsf{supp}(X)} \log_2(1/\Pr[X = x]).$$

For $X \in \{0,1\}^n$, we call $X$ an $(n, H_\infty(X))$-source, and we say $X$ has *entropy rate* $H_\infty(X)/n$.

However, one can easily show that it is impossible to construct deterministic randomness extractors for one $(n, k)$ source, even if $k$ is as large as $n - 1$. Thus, the study of randomness extractors has been pursued in two different directions. The first one is to allow the extractor itself to be randomized. In this case one ends up with the notion of *seeded extractors* [7], where the extractor is given a short independent uniform random seed (typically of length say $O(\log n)$). It is now possible to construct such extractors for all weak random sources. Typically, one also requires the output of the extractor to be close to uniform even given the seed. Such extractors are known as *strong seeded extractors*. Seeded extractors have a lot of applications in theoretical computer science and have been studied extensively, resulting in almost optimal constructions [8], [9], [10].

Another direction is to impose some special structure on the weak source, and thereby allows the construction of deterministic randomness extractors. This is the focus of this paper. Formally, we have the following definition.

**Definition I.2.** (Deterministic extractors for structured sources) Let $\mathcal{C}$ be a class of distributions on a finite set

Partially supported by NSF Grant CCF-1617713.

0272-5428/16 $31.00 © 2016 IEEE
DOI 10.1109/FOCS.2016.26

$\Omega$. A function $E : \Omega \to \{0,1\}^m$ is an extractor for $\mathcal{C}$ with entropy threshold $k$ and error $\epsilon$, if for any weak source $X \in \mathcal{C}$ with entropy at least $k$, we have

$$|E(X) - U_m| \leq \epsilon.$$

Here $U_m$ is the uniform distribution over $\{0,1\}^m$ and $|\cdot|$ stands for the statistical distance.

In this paper we will study the following classes of weak sources.

*Independent Sources:* Here, the extractor is given as input more than one general weak random sources, and the sources are independent of each other. Using the probabilistic method, one can show that there exists a deterministic extractor for just two independent sources with each having logarithmic min-entropy, which is optimal since extractors for one weak source do not exist. In fact, the probabilistic method shows that with high probability a random function is such a two-source extractor. However, the most interesting and important part is to give explicit constructions of such functions, which turns out to be highly challenging.

The first explicit construction of a two-source extractor appeared in [11], where Chor and Goldreich showed that the well known Lindsey's lemma gives an extractor for two independent $(n,k)$ sources with $k > n/2$. Since then there has been essentially no progress on two-source extractors until in 2005 Bourgain [12] gave a construction that breaks the entropy rate $1/2$ barrier, and works for two independent $(n, 0.49n)$ sources. In a different work, Raz [13] gave an incomparable result of two source extractors which requires one source to have min-entropy larger than $n/2$, while the other source can have min-entropy $O(\log n)$.

Given the difficulty of constructing explicit two-source extractors, much research has been focusing on a slightly more general model, where the extractor is allowed to have more then two independent sources as the input. Starting from [14], there has been a long line of fruitful results [14], [13], [12], [15], [16], [17], [18], [5], [19], [20], which introduced many new techniques and culminated in the three source extractor of exponentially small error by the author [19]. However, in the two source case the situation has not been improved.

Recently, Chattopadhyay and Zuckerman [1] made a breakthrough to this problem by giving the first explicit two-source extractors for $(n,k)$ sources with $k \geq \log^C n$ for some large enough constant $C$. This dramatically improves the situation of two-source extractors and is actually near optimal. However, their construction only outputs one bit and thus a natural question is whether one can achieve a significantly larger output length.

*Affine Sources:* An affine source is the uniform distribution over some unknown subspace of a vector space, and affine extractors are deterministic extractors for such sources.

**Definition I.3.** (affine source) Let $\mathbb{F}_q$ be the finite field with $q$ elements. Denote by $\mathbb{F}_q^n$ the $n$-dimensional vector space over $\mathbb{F}_q$. A distribution $X$ over $\mathbb{F}_q^n$ is an $(n,k)_q$ affine source if there exist linearly independent vectors $a_1, \cdots, a_k \in \mathbb{F}_q^n$ and another vector $b \in \mathbb{F}_q^n$ s.t. $X$ is sampled by choosing $x_1, \cdots, x_k \in \mathbb{F}$ uniformly and independently and computing

$$X = \sum_{i=1}^{k} x_i a_i + b.$$

In this paper we focus on the case where $q = 2$. Using the probabilistic method, it is not hard to show that there exists a deterministic affine extractor, as long as $k > 2\log n$ and the output length $m < k - O(1)$. The problem is to given an explicit construction of such a function.

There has also been a lot of work studying affine extractors and dispersers. For example, Gabizon and Raz [21] constructed explicit extractors for affine sources even with entropy 1. However, their constructions require the field size to be much larger than $n$, i.e., $q > n^{\Omega(1)}$, in order to use Weil's theorem. DeVos and Gabizon [22] constructed explicit extractors for $(n,k)_q$ affine sources when $q = \Omega((n/k)^2)$ and the characteristic of the field $\mathbb{F}_q$ is $\Omega(n/k)$. As the field size gets smaller, constructing explicit affine extractors becomes significantly harder.

The extreme and hardest case where the field is $\mathbb{F} = \mathrm{GF}(2)$, is the focus of the rest of the paper. Note that in this case the min-entropy $H_{\infty(X)}$ is the same as the standard Shannon entropy $H(X)$. Here, it is well known how to construct extractors for affine sources with entropy rate greater than $1/2$. However the problem becomes much harder as the entropy rate drops to $1/2$ and below $1/2$. Bourgain [2] used sophisticated character sum estimates to give an extractor for affine sources with entropy $k = \delta n$ for any constant $\delta > 0$. This was later slightly improved to $k = \Omega(n/\sqrt{\log\log n})$ by Yehudayoff [3] and the author [4], which have remained the best known results. Rao [23] constructed extractors for affine sources with entropy as small as $\mathrm{polylog}(n)$, as long as the subspace of $X$ has a basis of low-weight vectors. In the case where one only wishes to output one bit with support $\{0,1\}$ (i.e., a *disperser*),

Ben-Sasson and Kopparty [24] gave constructions for entropy $\Omega(n^{4/5})$, and Shaltiel [25] gave a construction for entropy $2^{\log^{0.9} n}$.

*Circuit Sources:* Trevisan and Vadhan [26] considered the question of extracting random bits from *samplable sources*, which are $n$-bit distributions generated by some small circuit from $\ell$ uniform bits. They showed that such extractors imply circuit lower bounds for related circuits. They also constructed explicit extractors for such sources with min-entropy $k = \Omega(n)$ under some necessary computational assumptions (such as the existence of a function computable in time $2^{O(n)}$ which requires $2^{\Omega(n)}$ size $\Sigma_5$ circuits). Subsequently, Viola [6] constructed unconditional extractors for sources generated by local circuits and circuits of small depth (e.g., $\mathsf{NC}^0$ and $\mathsf{AC}^0$ circuits). However, for both these sources, the extractors in [6] require min-entropy at least $n^{2/3+\Omega(1)}$.

*Non-oblivious bit-fixing Sources:* As in [1], an intermediate class of sources we use in our construction is a special kind of non-oblivious bit-fixing source. Non-oblivious bit-fixing sources are sources on $n$ bits which are uniform except some unknown $q$ bits. However the $q$ bits can depend arbitrarily on the $n - q$ uniform bits. Extractors for non-oblivious bit-fixing sources are equivalent to resilient functions, and were studied in [27], [28], [29], [6]. What we need here is a special kind of non-oblivious bit-fixing sources which only requires bounded independence in the "good" bits. Such sources were first defined by Viola in [6], where he constructed an extractor that extracts one bit from a non-oblivious bit-fixing source with $q \approx \sqrt{n}$ bad bits, and the "good" bits are $\mathrm{polylog}(n)$-wise independent. Subsequently, [1] gave an improved one-bit extractor that can handle $q = n^{1-\delta}$ for any constant $\delta > 0$. We now formally define such sources.

**Definition I.4.** A distribution $\mathcal{D}$ on n bits is t-wise independent if the restriction of $\mathcal{D}$ to any $t$ bits is uniform. Further $\mathcal{D}$ is a $(t, \epsilon)$-wise independent distribution if the distribution obtained by restricting $\mathcal{D}$ to any $t$ coordinates is $\epsilon$-close to uniform.

**Definition I.5.** A source $X$ on $\{0,1\}^n$ is called a $(q, t)$-non-oblivious bit-fixing source if there exists a subset of coordinates $Q \subseteq [n]$ of size at most $q$ such that the joint distribution of the bits indexed by $\overline{Q} = [n] \setminus Q$ is $t$-wise independent. The bits in the coordinates indexed by $Q$ are allowed to arbitrarily depend on the bits in the coordinates indexed by $\overline{Q}$.

If the joint distribution of the bits indexed by $\overline{Q}$ is $(t, \gamma)$-wise independent then $X$ is said to be a $(q, t, \gamma)$-non-oblivious bit-fixing source.

### A. Our Results

In this paper, we improve the output length of the two-source extractor in [1] to $k^{\Omega(1)}$.

**Theorem I.6.** *There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ with $k \geq \log^C n$, there exists a polynomial time computable function $\mathsf{2Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ with $m = k^{\Omega(1)}$ and $\epsilon = n^{-\Omega(1)}$ satisfying the following: if $X, Y$ are two independent $(n, k)$ sources, then*

$$|(\mathsf{2Ext}(X, Y), Y) - (U_m, Y)| \leq \epsilon.$$

Since the extractor is strong in $Y$, if we don't need a strong two-source extractor, then we can use the output of 2Ext to extract from $Y$ and output almost all the min-entropy. For example, by using a strong seeded extractor from [30] that uses $O(\log^2 n \log k)$ bits to extract all the min-entropy (and requiring that say $m \geq \log^3 n$), we have the following theorem.

**Theorem I.7.** *There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ with $k \geq \log^C n$, there exists a polynomial time computable function $\mathsf{2Ext} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ with $m = k$ and $\epsilon = n^{-\Omega(1)}$ satisfying the following: if $X, Y$ are two independent $(n, k)$ sources, then*

$$|\mathsf{2Ext}(X, Y) - U_m| \leq \epsilon.$$

Next we give an affine extractor over $\mathbb{F}_2$ that works for entropy $k \geq \mathrm{polylog}(n)$, thus significantly improving all previous results in terms of the entropy requirement (even in the disperser case). Our extractor outputs $k^{\Omega(1)}$ bit and has error $n^{-\Omega(1)}$.

**Theorem I.8.** *There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ with $k \geq \log^C n$, there exists a polynomial time computable function $\mathsf{AExt} : \{0,1\}^n \to \{0,1\}^m$ with $m = k^{\Omega(1)}$ and $\epsilon = n^{-\Omega(1)}$ satisfying the following: for any $(n, k)$ affine source $X$, we have that*

$$|\mathsf{AExt}(X) - U_m| \leq \epsilon.$$

Table I summarizes our result compared to previous constructions of extractors and dispersers for affine sources over $\mathbb{F}_2$.

By using a reduction from $\mathsf{NC}^0$ and $\mathsf{AC}^0$ sources to affine sources in [6], we also obtain improved extractors for $\mathsf{NC}^0$ and $\mathsf{AC}^0$ sources, which only require min-entropy $n^{1/2+\Omega(1)}$.

**Theorem I.9.** *For any constant $\alpha > 0, d = O(1)$ and any $n, k \in \mathbb{N}$ with $k \geq n^{1/2+\alpha}$, there is an explicit extractor $\mathsf{acExt} : \{0,1\}^n \to \{0,1\}^m$ with $m = k^{\Omega(1)}$*

| Construction | Entropy | Output | Error |
|---|---|---|---|
| [2] | $k \geq \delta n$, any constant $\delta$ | $\Theta(n)$ | $2^{-\Omega(n)}$ |
| [3], [4] | $k = \Omega(n/\sqrt{\log\log n})$ | $n^{\Omega(1)}$ | $2^{-n^{\Omega(1)}}$ |
| [24] | $k = \Omega(n^{4/5})$ | 1 | Disperser |
| [25] | $k \geq 2^{\log^{0.9} n}$ | 1 | Disperser |
| This work | $k \geq \log^C n$, some $C > 1$. | $k^{\Omega(1)}$ | $n^{-\Omega(1)}$ |

Table I
SUMMARY OF AFFINE EXTRACTORS AND DISPERSERS OVER $\mathbb{F}_2$.

*and $\epsilon = n^{-\Omega(1)}$ such that if $X$ is an $(n,k)$ source generated by a depth-$d$ $\mathsf{AC}^0$ circuit of size $n^d$, then*

$$|\mathsf{acExt}(X) - U_m| \leq \epsilon.$$

All the above extractors are based on an extractor for $(q,t,\gamma)$-non-oblivious bit-fixing sources. In particular, we have the following theorem which improves the output length of such an extractor in [1] from one bit to $t^{\Omega(1)}$.

**Theorem I.10.** *There exists a constant $c > 0$ such that for any constant $\delta > 0$ and all $n, q, t \in \mathbb{N}$ with $q \leq n^{1-\delta}$, $t \geq c\log^{21} n$ and any $\gamma \leq 1/n^{t+1}$, there exists an explicit extractor $\mathsf{BFExt} : \{0,1\}^n \to \{0,1\}^m$ with $m = t^{\Omega(1)}$ and $\epsilon = n^{-\Omega(1)}$ such that for any $(q,t,\gamma)$ non-oblivious bit-fixing source $X$ on $n$ bits, we have*

$$|\mathsf{BFExt}(X) - U_m| \leq \epsilon.$$

*Subsequent work.:* In a subsequent work, Meka [31] constructed explicit resilient functions matching the randomized construction by Ajtai and Linial [32]. His construction also gives improved two-source extractors which require min-entropy $\log^C n$ for a smaller constant $C$. Similarly, his construction can also be used to obtain affine extractors for entropy $\log^C n$ with a smaller constant $C$. In another subsequent work by Chattopadhyay and the author [33], the techniques of this paper were used to construct explicit extractors for a new model of weak sources called *sumset sources*, which also give improved extractors for sources generated by small space computation.

### *B. Zero-error dispersers and applications*

We now extend our results to the case of zero-error dispersers. We first have the following definition.

**Definition I.11.** (Zero-error dispersers and strongly hitting dispersers) [34] Let $\mathcal{C}$ be a class of distributions on a finite set $\Omega$. A function $D : \Omega \to \{0,1\}^m$ is a disperser for $\mathcal{C}$ with entropy threshold $k$ and error $\epsilon$, if for any weak source $X \in \mathcal{C}$ with entropy at least $k$,

$$|\mathsf{Supp}(D(X))| \geq (1-\epsilon)2^m.$$

The function $D$ is called a zero-error disperser if $\epsilon = 0$. It is called a $\mu$-strongly hitting disperser if in addition for every $z \in \{0,1\}^m$, $\Pr[D(X) = z] \leq \mu$.

Naturally, we consider both independent sources and affine sources. Zero-error independent source dispersers are intimately connected to explicit constructions of Ramsey graphs [16], [1], [35], and zero-error affine dispersers for sub-linear entropy give explicit Boolean functions with the best known general circuit lower bounds of $3.01n$ [36]. Here we will be interested in zero-error dispersers with large output length (say $\Omega(k)$). Previously, such dispersers for independent sources are only known when $k \geq \delta n$ (for two sources) or $k \geq n^\delta$ (for $O(1/\delta)$ sources) [34]. The dispersers of [1], [35], which work for $k \geq \log^C n$, only output one bit. For affine sources, such dispersers are only known when $k = \Omega(n/\sqrt{\log\log n})$. In this case Gabizon and Shaltiel [37] achieved output length $k - \beta n/\sqrt{\log\log n}$ for some constant $0 < \beta < 1$. All other known zero-error dispersers, such as the one by Ben-Sasson and Kopparty [24] which works for entropy $\Omega(n^{4/5})$, and the one by Shaltiel [25] which works for entropy $2^{\log^{0.9} n}$, only output one bit.

Note that if the error of an extractor is small enough compared to its output length, then the extractor is automatically a strongly hitting disperser and zero error disperser. For example, we have

**Fact I.12.** *Let $f : \Omega \to \{0,1\}^m$ be an extractor for a class $\mathcal{C}$ of sources with entropy $k$ and error $\epsilon \leq 2^{-(m+1)}$. Then $f$ is a $2^{-(m+1)}$-strongly hitting disperser for the same class $\mathcal{C}$ and entropy $k$.*

The error of our two-source extractors and affine extractors can be made $n^{-C}$ for any constant $C > 0$, at the price of slightly increasing the running time of the extractors (but still polynomial in $n$). Thus we have the following direct corollary.

**Corollary I.13.** *There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ with $k \geq \log^C n$ and any constant $C' > 0$, there exist explicit constructions of strongly $2^{-(m+1)}$-hitting dispersers for two independent $(n,k)$ sources or affine sources on $n$ bits with entropy $k$, with output length $m = C' \log n$.*

Using a generic technique to increase the output length in [34], we obtain the following:

**Theorem I.14.** *There exist constants $C > 0, \eta > 0$ such that for all $n, k \in \mathbb{N}$ with $k \geq \log^C n$ and $m =$*

*$\eta k$, there exists an explicit construction of a $2^{-(m+3)}$-strongly hitting disperser $D' : (\{0,1\}^n)^2 \to \{0,1\}^m$ for two independent $(n,k)$ sources.*

Gabizon and Shaltiel [34] further showed that such strongly hitting dispersers can be applied to the problem of *implicit probe search*, which, informally speaking, is to search for an element in a table with few probes, while using no additional information beyond the stored elements themselves.

**Definition I.15.** (Implicit probe search scheme) [34]. For integer parameters $n, k, q$, the implicit probe search problem is as follows: Store a subset $S \subseteq \{0,1\}^n$ of size $2^k$ in a table $T$ of size $2^k$, (where every table entry holds only a single element of $S$), such that given any $x \in \{0,1\}^n$ we can determine whether $x \in S$ using $q$ queries to $T$. A solution to this problem is called an implicit $q$-probe scheme with table size $2^k$ and domain size $2^n$.

The main goal for this problem is to give a scheme that uses as few number of queries as possible (e.g., $O(1)$ queries) and has table size as small as possible. Fiat and Naor [38] studied implicit $O(1)$-probe schemes, where the number of queries is a constant that does not depend on $n$ or $k$. They showed that if $n$ is large enough compared to $k$, then no such schemes can exist. This improves a previous bound by Yao [39]. They also proved that non-explicitly, such schemes exist as long as $n \leq 2^{\mathrm{poly}(k)}$. In addition, they gave an efficient implicit $O(1)$-probe scheme in the case where $k = \delta n$ for any constant $\delta > 0$. Gabizon and Shaltiel [34] showed that zero-error dispersers for a constant number of independent sources can be used to construct implicit $O(1)$-probe schemes, and they achieved $O(1/\delta)$-probe schemes in the case where $k = n^\delta$ for any constant $\delta > 0$. By using our improved zero-error disperser, we obtain the following implicit $O(1)$-probe scheme, which almost matches the non-explicit construction in [38].

**Theorem I.16.** *There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ with $k = \log^C n$ there is an efficiently computable implicit $O(1)$-probe scheme with table size $2^k$ and domain size $2^n$.*

Note that the construction crucially relies on a strongly hitting disperser with output length $\Omega(k)$, which is made possible only because our two-source extractor has output length at least $c \log n$ for some constant $c > 1$ and error $n^{-c}$.

In the case of affine sources, Gabizon and Shaltiel [37] also gave a generic method to transform a zero-error affine disperser with $c \log n$ output bits into another zero-error affine disperser which outputs almost all entropy. Combining their transformation with our zero-error disperser, we obtain the following theorem, which improves the entropy requirement and output length of previous zero-error affine dispersers.

**Theorem I.17.** *There exists a constant $C > 0$ such that for all $n, k \in \mathbb{N}$ with $k \geq \log^C n$, there is an explicit zero-error affine disperser $D : \{0,1\}^n \to \{0,1\}^m$ for entropy $k$, where $m = k - \log^C n$.*

Such zero-error affine dispersers can be applied to the following problem of constructing schemes for defective memory with stuck-at and adversarial errors, as shown in [37]. The problem was first studied by Kuznetsov and Tsybakov [40], where we have a memory of $n$ cells each storing a symbol from some finite alphabet (we focus on the Boolean alphabet here). However, some subset of at most $s$ cells are stuck (e.g., fixed to some particular string $u$) and cannot be modified. The goal then is to store a string $z \in \{0,1\}^m$ in the memory so that later we can read the memory and retrieve $z$, even without any information about which of the cells are stuck. Tsybakov [41] extended this to a more general model where besides the stuck-at errors, an adversary can choose to corrupt few cells after the string is stored. Formally, we have the following definition.

**Definition I.18.** [37]. An $(n, s, e)$-stuck-at noisy memory scheme consists of

- a (possibly randomized) encoding function $E$ such that given any $S \subset [n]$ with $|S| \leq s$, $u \in \{0,1\}^{|S|}$ and $z \in \{0,1\}^m$, $E$ returns $x \in \{0,1\}^n$ such that $x|_S = u$, and
- a decoding function $D : \{0,1\}^n \to \{0,1\}^m$ such that for any $x \in \{0,1\}^n$ produced by $E$ with input $z$ (and any inputs $S$ and $u$ as above), and any 'noise vector' $\xi \in \{0,1\}^n$ of hamming weight at most $e$, $D(x + \xi) = z$.

The rate of the scheme is $m/n$, and the natural goal here is to tolerate as many errors as possible while making the rate (or equivalently, $m$) as large as possible. While one can use standard error-correcting codes for this problem, Gabizon and Shaltiel [37] showed that by using "invertible" zero-error dispersers, one can do much better. Specifically, they constructed an explicit stuck-at noisy memory scheme that can tolerate $s(n) = pn$ stuck-at errors for $p < 1/2$ and $e(n) = o(\sqrt{n})$ adversarial errors, with $m = n - s(n) - e(n) \log n - O(n/\sqrt{\log \log n})$ (which achieved rate $1 - p - o(1)$ for the first time) and they left open the problem of improving $m$ to $n - s(n) - e(n) \log n - \log^{O(1)} n$.

As shown in [37], a longer output of the zero-error affine disperser directly translates into a larger $m$ in the scheme. Using our improved disperser, we obtain the following improved scheme.

**Theorem I.19.** *There exists a constant $C > 0$ such that for any $p < 1/2$, $e(n) = o(\sqrt{n})$ and $s(n) \leq pn$, there is an explicit $(n, s(n), e(n))$-stuck-at noisy memory scheme with $m = n - s(n) - e(n)\log n - \log^C n$.*

## II. Overview of the constructions

### A. Improved two-source extractor

Here we give a brief overview of our improved two-source extractor. Since it follows easily from the extractor for non-oblivious bit-fixing sources, we first describe our new extractor for the $(q, t, \gamma)$ non-oblivious bit-fixing source on $n$ bits with $q \leq n^{1-\delta}$ for any constant $\delta > 0$, and $\gamma \leq 1/n^{t+1}$. Our starting point is the one-bit deterministic extractor for such sources in [1], which we will call $\mathsf{BitExt}$. We note that from the construction of [1], (by setting the parameters appropriately) this function has the following properties. First, it is a depth-4 $\mathsf{AC}^0$ circuit with size $n^{O(1)}$. Second, since it's an extractor, for any $(q, t, \gamma)$ non-oblivious bit-fixing source $X$, we have $\mathsf{BitExt}(X)$ is $n^{-\Omega(1)}$-close to uniform. Third, it's a resilient function, in the sense that any coalition of $q$ bits has influence[1] at most $q/n^{1-\frac{\delta}{2}}$, even when the rest of the bits are only $(t, \gamma)$-wise independent.

We now describe how to extract more than one bit. One natural idea is to divide the source $X$ into many blocks and then apply $\mathsf{BitExt}$ to each block. Indeed this is our first step. In the source $X$, we denote the "bad bits" by $Q$, and the "good bits" by $\overline{Q}$. To ensure that no block consists of only bad bits, we will divide $X$ into $n^\alpha$ blocks for some constant $\alpha < \delta$ (it suffices to take $\alpha = \delta/4$). Thus we get $\ell = n^\alpha$ blocks $\{X_i, i \in [\ell]\}$ with each block containing $n' = n^{1-\alpha}$ bits. We now apply $\mathsf{BitExt}$ to each block to obtain a bit $Y_i$. Of course, we will set up the parameters such that $\mathsf{BitExt}$ is an extractor for $(q, t, \gamma)$ non-oblivious bit-fixing source on $n^{1-\alpha}$ bits.

Now consider any block. Our observation is that since each block can contain at most $q \leq n^{1-\delta}$ bits from $Q$, the coalition of the bad bits in this block still has small influence. In particular, a simple calculation shows that $q < n'^{1-\frac{3\delta}{4}}$ and thus for each block the influence of the bad bits is bounded by $q/n'^{1-\frac{3\delta}{8}} < n^{-\frac{3\delta}{8}}$. This means that with probability at least $1 - n^{-\frac{3\delta}{8}}$ over the fixing of $X_i \cap \overline{Q}$, we have that $Y_i$ is fixed. Thus, by a simple union bound, with probability at least $1 - n^\alpha n^{-\frac{3\delta}{8}} = 1 - n^{-\frac{\delta}{8}}$ over the fixing of $\overline{Q}$, we have that all $\{Y_i, i \in [\ell]\}$ are fixed.

Now consider another distribution $X'$, which has the same distribution as $X$ for the bits in $\overline{Q}$, while the bits in $Q$ are fixed to 0 independent of the bits in $\overline{Q}$. We let $Y'_i, i \in [\ell]$ be the corresponding $Y_i$'s obtained from $X'$ instead of $X$. By the above argument, with probability at least $1 - n^{-\frac{\delta}{8}}$ over the fixing of $\overline{Q}$, $\{Y_i\}$ and $\{Y'_i\}$ are the same. Thus the joint distribution of $\{Y_i\}$ and $\{Y'_i\}$ are within statistical distance $n^{-\frac{\delta}{8}}$. Moreover, the bits in $\overline{Q}$ are $(t, \gamma)$-wise independent and thus they are $n^t\gamma \leq 1/n$-close to a truly $t$-wise independent distribution. From now on we will treat $\overline{Q}$ as being truly $t$-wise independent, since this only adds $1/n$ to the final error.

We will now choose a parameter $m = t^{\Omega(1)}$ for the output length. In addition, we take the generating matrix $G$ of an asymptotically good linear binary code with message length $m$, codeword length $r = O(m)$ and distance $d = \Omega(m)$. It is well known how to construct such codes (and thus the generating matrix) explicitly. Note that $G$ is an $m \times r$ matrix and any codeword can be generated by $w = vG$ for some vector $v \in \{0,1\}^m$, where all operations are in $\mathbb{F}_2$. We choose $m$ so that $r = O(m) \leq \ell$ and now we let $Y = (Y_1, \cdots, Y_r)$ be the random vector in $\mathbb{F}_2^r$ obtained from $\{Y_i, i \in [\ell]\}$. Similarly, we have $Y' = (Y'_1, \cdots, Y'_r)$. The output of our extractor will now be $Z = (Z_1, \cdots, Z_m) = GY$, where all operations are in $\mathbb{F}_2$.

For the analysis let us consider $Z' = (Z'_1, \cdots, Z'_m) = GY'$. We will show that $Z'$ is close to uniform and then it follows that $Z$ is also close to uniform since they are within statistical distance $n^{-\Omega(1)}$ (as they are deterministic functions of $Y$ and $Y'$ respectively). To show this, we will use the XOR lemma. Consider any non-empty subset $S \subseteq [m]$ and $V'_S = \bigoplus_{i \in S} Z'_i$. Note that this is just $(\sum_{i \in S} G_i)Y'$ where $G_i$ stands for the $i$'th row of $G$. Note that $\sum_{i \in S} G_i$ is a codeword and thus has at least $d = \Omega(m)$ 1's. On the other hand, it can have at most $r = O(m)$ 1's.

Note that the parity of up to $r$ bits can be computed by a depth-2 $\mathsf{AC}^0$ circuit of size $2^{O(r)} = 2^{O(m)}$. Recall that each input bit $Y_i$ can be computed by a depth-4 $\mathsf{AC}^0$ circuit of size $n^{O(1)}$. Thus we see that each $V'_S$ can be computed by a depth-6 $\mathsf{AC}^0$ circuit of size at most $2^{O(m)} n^{O(1)} = 2^{O(m)}$ if we choose $m > \log n$.[2]

[1] Informally, the influence of a set of bits is the probability over the rest of the bits such that the function is not fixed.

[2] We can get rid of the intermediate negation gates with only a constant factor of blow-up in the circuit size, by standard tricks.

Note that all bits in $Q$ are fixed to 0. Thus the inputs of the circuits are only from $\overline{Q}$.

Now our goal is to ensure that $V'_S$ can be fooled by $t$-wise independent distributions with error $\epsilon = 2^{-m}$. By the results of Braverman [42] and Tal [43], it suffices to take $t = O(\log(2^{O(m)}/\epsilon)^{21}) = O(m^{21})$. Thus we can take $m = \Omega(t^{\frac{1}{21}})$ and it follows that $V'_S$ cannot distinguish between $t$-wise independent distributions and uniform distribution. On the other hand, if $\overline{Q}$ is the uniform distribution, then $V'_S$ is the XOR of at least $d = \Omega(m)$ independent random variables, with each being $n^{-\Omega(1)}$-close to uniform. Thus in this case $V'_S$ is $(n^{-\Omega(1)})^d = 2^{-\Omega(m \log n)}$-close to uniform. Together this means that $V'_S$ is $2^{-\Omega(m \log n)} + 2^{-m} < 2^{1-m}$ close to uniform. Since this is true for any non-empty subset $S$, by a standard XOR lemma it now follows that $Z'$ is $2^{-\Omega(m)}$-close to uniform. Adding back the errors we see that $Z$ is $n^{-\Omega(1)}$-close to uniform.

We can now apply the reduction from two independent sources to a non-oblivious bit-fixing source, which was implicit in [19] and explicit in [1]. This reduction reduces two independent $(n, k)$ sources to a $(q, t, \gamma)$-non-oblivious bit-fixing source with $t = k^{\Omega(1)}$,[3] thus by applying the above extractor we also get an improved two-source extractor with output length $k^{\Omega(1)}$.

### *B. Affine extractor*

On the high level, our construction of affine extractors follows the framework of the recent two-source extractor construction by Chattopadhyay and Zuckerman [1]. Specifically, we will first reduce an affine source to a $(q, t, \gamma)$-non-oblivious bit-fixing source, and then apply our improved deterministic extractor for such sources.

We now describe our reduction. We will mainly adapt techniques from previous work on extractors for independent sources. Specifically, by using ideas from alternating extraction (Figure **??**) [44], [45], [18], [5], one of the author's previous work [19] obtained a somewhere random source with $N = \text{poly}(n)$ rows from two independent $(n, k)$ sources with $k \geq \text{polylog}(n)$. The somewhere random source is a random matrix, with the additional property that except for a small fraction of "bad" rows, the rest of the rows are almost $t$-wise independent for $t = k^{\Omega(1)}$ in the sense that any $t$ of these rows are $\gamma = 2^{-k^{\Omega(1)}}$-close to uniform. Thus, these rows (or, say, taking one bit each row) form exactly a $(q, t, \gamma)$-non-oblivious bit-fixing source.

Now we need to adapt that construction to affine sources. Of course we now only have one affine source instead of two independent sources. However, due to the special structure of affine sources we can still apply similar ideas as in [5], [19]. Specifically, we will use a special kind of strong seeded extractors called *linear seeded extractors*. These extractors have the property that for any fixed seed, the output is a linear function of the source. We take such a seeded extractor with seed length $O(\log n)$ and error $\epsilon$, and use every possible seed to extract from the affine source $X$. This gives us a matrix (or somewhere random source) of $N = \text{poly}(n)$ rows, where each row corresponds to the output of the extractor on a particular seed. A standard argument shows that if $X$ is affine, then at least $1 - 2\epsilon$ fraction of the rows are truly uniform, although they may depend on each other in arbitrary ways. Note that this is even better for our purpose than in the case of two independent general weak random sources, since there we have to use some other ideas to reduce the error, while here the error in the somewhere random source is essentially zero. We further restrict the size of each row, so that the length is much smaller than the entropy of $X$.

We can now use these rows and the source $X$ itself to do the same alternating extraction protocol as in [5], [19] to make the "good" rows almost $t$-wise independent for $t = k^{\Omega(1)}$, with error $\gamma = 2^{-k^{\Omega(1)}}$. To see why alternating extraction works in this case, consider one particular uniform row $Y$. Note that $Y$ is a linear function of $X$, so $Y$ is also an affine source. Recall that the length of $Y$ is much smaller than the entropy of $X$. A standard argument shows that $X$ can be decomposed into $X = A + B$ where both $A, B$ are affine sources, $A = L(Y)$ for some linear bijection $L$, and $B$ is independent of $Y$. Thus, to do the alternating extraction, we can first take a small slice of $Y$ to be $S_1$, and use a linear seeded extractor $\mathsf{Ext}$ to compute $R_1 = \mathsf{Ext}(X, S_1)$. Note that $R_1 = \mathsf{Ext}(X, S_1) = \mathsf{Ext}(A, S_1) + \mathsf{Ext}(B, S_1)$. By the property of a strong extractor we know that with high probability over the fixing of $S_1$, $\mathsf{Ext}(B, S_1)$ is close to uniform (since $S_1$ is independent of $B$). Note that $S_1$ is a deterministic function of $A$ and $A$ is independent of $B$, thus $R_1 = \mathsf{Ext}(A, S_1) + \mathsf{Ext}(B, S_1)$ is also uniform conditioned on the fixing of $S_1$.

Next, suppose the length of $R_1$ is much smaller than the length of $Y$ ,we can then use $R_1$ and apply $\mathsf{Ext}$ back to $Y$ to extract $S_2 = \mathsf{Ext}(Y, R_1)$. The reason is that we can first fix $\mathsf{Ext}(A, S_1)$. Note that we have already fixed $S_1$ so this is a deterministic function of $A$ (or $Y$). Therefore after fixing it, $\mathsf{Ext}(B, S_1)$ is still uniform and independent of $Y$ (since now it is a deterministic function of $B$), and now $R_1 = \mathsf{Ext}(A, S_1) + \mathsf{Ext}(B, S_1)$

[3]The original reduction in [19] only gives $q = \Omega(n)$, but a simple modification can also give $q \leq n^{1-\delta}$ for any constant $\delta > 0$.

is independent of $Y$. Since the length of $R_1$ is small, conditioned on this fixing $Y$ still has a lot of entropy left. Therefore we can now extract $S_2 = \mathsf{Ext}(Y, R_1)$. After this we can further fix $\mathsf{Ext}(B, S_1)$ and thus also $R_1$. We know that with high probability over this fixing, $S_2$ is still close to uniform. Moreover conditioned on this fixing, $B$ still has a lot of entropy left, and is still independent of $Y$. Now $S_2$ is a deterministic function of $Y$. Continue doing this, we can see that alternating extraction works as long as we always use a strong linear seeded extractor and keep the size of each $R_i, S_i$ to be small. Intuitively, it's like alternating extraction between the two independent affine sources $Y$ and $B$. Now we can use similar arguments as in [5] to make the somewhere random source almost $t$-wise independent.[4]

However, there are a few subtle technical problems we need to deal with. First, when we generalize the above alternating extraction to run for $t$ rows $Y^1, Y^2, \cdots, Y^t$ simultaneously, we will need to consider the concatenation $Y = Y^1 \circ Y^2 \circ \cdots \circ Y^t$ and decompose $X$ into $X = A + B = L(Y) + B$. This ensures that we can condition on the fixing of all the intermediate random variables obtained from $Y^1, Y^2, \cdots, Y^t$ without affecting $B$. We can do this if we choose the parameters appropriately so that both $t = k^{\Omega(1)}$ and the size of $Y^i$ are small compared to the entropy of $X$. Thus in the decomposition $B$ still has sufficient entropy. Another subtlety arises in the analysis as follows. The alternating extraction will take some $b < \log n$ rounds, with each round consisting of some $k^{\Omega(1)}$ steps. In each round $j$, we start the alternating extraction using a random variable $Y^{ij}$ obtained from $Y^i$, and at the end we obtain a random variable $R^{ij}$ from $X$. Our goal is to show that these $\{R^{ij}\}$ will gradually become independent of each other, until at the end they become all independent, thus achieving $t$-wise independent. Towards this, at the end of round $j$, for each $Y^i$ we need to use $R^{ij}$ to extract $Y^{i(j+1)}$ from $Y^i$ to start the next round. Here we would like to argue that for those $\{R^{\ell j}\}$ that have already become independent of $R^{ij}$, we can first fix all $\{Y^{\ell(j+1)}\}$ and all the $R$ variables produced in round $j+1$, and $Y^{i(j+1)}$ is still uniform. This ensures that whatever is already independent will remain independent. While this is true in the case of two independent sources, it is no longer true in the case of an affine source. The reason is that, as explained above, when we fix $R = \mathsf{Ext}(A, S) + \mathsf{Ext}(B, S)$, the part of $\mathsf{Ext}(A, S)$ is a function of $A$ (and $Y$). Thus this fixing may cause $Y^{i(j+1)}$ to lose entropy (note that fixing $\mathsf{Ext}(B, S)$ will not since $B$ is independent of $Y$). Fortunately, we can get around this by restricting the length of the $R$ variables to be much smaller than the length of $Y^{i(j+1)}$. We note that if we take a seeded extractor with error $\epsilon$, and use a seed that loses $\ell$ bits of entropy, then the extractor still works with error increased to $2^\ell \epsilon$. Thus by appropriately choosing the parameters (making $\ell$ small enough compared to the seed length of the seeded extractor), we can still use $Y^{i(j+1)}$ to start the next round of alternating extraction, and the whole construction goes through.

[4]We remark that we can also use the flip-flop alternating extraction developed in [20], which may result in an improvement in the constants. However in this paper we do not try to optimize the constant $C$ in our final result where $k \geq \log^C n$.

One final point is that the extractor for non-oblivious bit-fixing source in [1], as well as our improved extractor can only handle the case where $q \leq N^{1-\delta}$ for any constant $\delta > 0$. This means that to convert the affine source $X$ into a somewhere random source in the first step, we need to take a strong linear seeded extractor with seed length $O(\log n)$ and error $\epsilon = 1/\mathrm{poly}(n)$, i.e., an extractor with optimal seed length. Previously, such a linear seeded extractor was not known. In this paper we construct such a strong linear seeded extractor by combining the lossless condenser in [9] and another strong linear seeded extractor in [46]. We note that the condenser in [9] itself may not be linear, but can be made linear with the same parameters by a careful instantiation, following a result in [47]. Thus in this step we can use $O(\log n)$ bits to condense the source into a $(n' = O(k), k)$ source with error $1/\mathrm{poly}(n)$. We then use the linear seeded extractor in [46], which has seed length $d = O\left(\log n' + \frac{\log n'}{\log k} \log\left(\frac{1}{\epsilon}\right)\right)$. Note that $n' = O(k)$. Thus if we take $\epsilon = 1/\mathrm{poly}(n)$ we get $d = O(\log n)$. Altogether we get a strong seeded extractor with seed length $O(\log n)$ and error $\epsilon = 1/\mathrm{poly}(n)$. Since both the condenser and the extractor are linear, the combined extractor is also linear.

## III. Conclusions and Open Problems

Constructing explicit two-source extractors and affine extractors are two related challenging problems. Through a long line of research, the recent breakthrough result of Chattopadhyay and Zuckerman [1] has finally brought us close to the optimal two-source extractor. In this paper we managed to improve the output length of the two-source extractors in [1] from 1 to $k^{\Omega(1)}$ in the strong case, and to $k$ in the non-strong case. We also construct the first explicit affine extractor for poly-logarithmic entropy, thus bringing affine extractors close to optimal. However, in both cases the error remains

$n^{-\Omega(1)}$. The most obvious open problem is to improve the error (say to exponentially small).

This seems challenging and requiring new ideas. Specifically, the current approach is to first reduce the sources to a $(q, t, \gamma)$ non-oblivious bit-fixing source, and then apply a deterministic extractor for such sources. The analysis is done by bounding the influence of a coalition of variables. If the non-oblivious bit-fixing source has length $n^{O(1)}$ (to ensure polynomial time computability), then even one bit can have influence $\Omega(\log n/n^{O(1)})$ by the result of [28]. Thus we cannot hope to get error $n^{-\omega(1)}$ through this approach alone.

Another related open problem is to increase the output length of our affine extractor, which is currently $k^{\Omega(1)}$. We note that we can use a general technique by Shaltiel [48] to try to improve the output length. However for that purpose we need to use a linear seeded extractor with seed length $O(\log n)$, since the error will be increased by a factor of $2^d$ where $d$ is the seed length. A linear seeded extractor with such seed length can possibly achieve output length $k^{0.9}$ [46], but we are not aware of any construction with output length $\Omega(k)$. On the other hand, if we can make the error smaller, then we can afford a larger seed length (such as $O(\log^2 n)$), which is enough to output almost all the entropy.

In the case of $\mathsf{NC}^0$ and $\mathsf{AC}^0$ sources, there is still much room for improvement. Currently it is not known how to extract from sources with min-entropy smaller than $n^{1/2}$, even if the source is generated by a circuit where each output bit depends on at most 2 input bits.

Finally, an interesting observation of our work is that actually the bias of the one bit extractor in [1] is not very important (in [1] it has bias $n^{-\Omega(1)}$). Indeed, even if it only has constant bias, after the step of using the generating matrix $G$, we can see that the XOR of $\Omega(m)$ copies will have bias $2^{-\Omega(m)}$. However, at this moment this observation doesn't seem to help improve the parameters.

## REFERENCES


[1] E. Chattopadhyay and D. Zuckerman, "Explicit two-source extractors and resilient functions," in *STOC*, 2016.

[2] J. Bourgain, "On the construction of affine-source extractors," *Geometric and Functional Analysis*, vol. 1, pp. 33–57, 2007.

[3] A. Yehudayoff, "Affine extractors over prime fields," *Combinatorica*, 2011.

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

[5] ——, "Extractors for a constant number of independent sources with polylogarithmic min-entropy," in *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science*, 2013, pp. 100–109.

[6] E. Viola, "Extractors for circuit sources," in *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011.

[7] N. Nisan and D. Zuckerman, "Randomness is linear in space," *Journal of Computer and System Sciences*, vol. 52, no. 1, pp. 43–52, 1996.

[8] C. J. Lu, O. Reingold, S. Vadhan, and A. Wigderson, "Extractors: Optimal up to constant factors," in *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003, pp. 602–611.

[9] V. Guruswami, C. Umans, and S. Vadhan, "Unbalanced expanders and randomness extractors from Parvaresh-Vardy codes," *Journal of the ACM*, vol. 56, pp. 1–34, 2009.

[10] Z. Dvir, S. Kopparty, S. Saraf, and M. Sudan, "Extensions to the method of multiplicities, with applications to Kakeya sets and mergers," in *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science*, 2009, pp. 181–190.

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

[12] J. Bourgain, "More on the sum-product phenomenon in prime fields and its applications," *International Journal of Number Theory*, vol. 1, pp. 1–32, 2005.

[13] R. Raz, "Extractors with weak random seeds," in *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, 2005, pp. 11–20.

[14] B. Barak, R. Impagliazzo, and A. Wigderson, "Extracting randomness using few independent sources," in *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004, pp. 384–393.

[15] A. Rao, "Extractors for a constant number of polynomially small min-entropy independent sources," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[16] B. Barak, A. Rao, R. Shaltiel, and A. Wigderson, "2 source dispersers for $n^{o(1)}$ entropy and Ramsey graphs beating the Frankl-Wilson construction," in *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[17] X. Li, "Improved constructions of three source extractors," in *Proceedings of the 26th Annual IEEE Conference on Computational Complexity*, 2011, pp. 126–136.

[18] ——, "New independent source extractors with exponential improvement," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing*, 2013, pp. 783–792.

[19] ——, "Three source extractors for polylogarithmic min-entropy," in *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

[20] G. Cohen, "Local correlation breakers and applications to three-source extractors and mergers," in *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science*, 2015.

[21] A. Gabizon and R. Raz, "Deterministic extractors for affine sources over large fields," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, 2005.

[22] M. DeVos and A. Gabizon, "Simple affine extractors using dimension expansion," in *Proc. of the 25th CCC*, 2010.

[23] A. Rao, "Extractors for low-weight affine sources," in *Proc. of the 24th CCC*, 2009.

[24] E. Ben-Sasson and S. Kopparty, "Affine dispersers from subspace polynomials," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 2009.

[25] R. Shaltiel, "Dispersers for affine sources with sub-polynomial entropy," in *Proceedings of the 52nd Annual IEEE Symposium on Foundations of Computer Science*, 2011.

[26] L. Trevisan and S. Vadhan, "Extracting randomness from samplable distributions," in *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[27] M. Ben-Or and N. Linial, "Collective coin flipping," *Randomness and Computation*, 1978.

[28] J. Kahn, G. Kalai, and N. Linial, "The influence of variables on boolean functions (extended abstract)," in *Proceedings of the 29th Annual IEEE Symposium on Foundations of Computer Science*, 1988.

[29] J. Kamp and D. Zuckerman, "Deterministic extractors for bit-fixing sources and exposure-resilient cryptography," *SIAM Journal on Computing*, vol. 36, no. 5, pp. 1231–1247, 2007.

[30] R. Raz, O. Reingold, and S. Vadhan, "Extracting all the randomness and reducing the error in trevisan's extractors," *JCSS*, vol. 65, no. 1, pp. 97–128, 2002.

[31] R. Meka, "Explicit resilient functions matching Ajtai-Linial," *CoRR*, vol. abs/1509.00092, 2015.

[32] M. Ajtai and N. Linial, "The influence of large coalitions," *Combinatorica*, vol. 13, no. 2, pp. 129–145, 1993.

[33] E. Chattopadhyay and X. Li, "Extractors for sumset sources," in *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, 2016.

[34] A. Gabizon and R. Shaltiel, "Increasing the output length of zero-error dispersers," in *Random 2008*, 2008.

[35] G. Cohen, "Two-source dispersers for polylogarithmic entropy and improved Ramsey graphs," in *STOC*, 2016.

[36] M. G. Find, A. Golovnev, E. Hirsch, and A. Kulikov, "A better-than-$3n$ lower bound for the circuit complexity of an explicit function," ECCC, Tech. Rep. TR15-166, 2015.

[37] A. Gabizon and R. Shaltiel, "Invertible zero-error dispersers and defective memory with stuck-at errors," in *Randomization, Approximation, and Combinatorial Optimization. Proceedings of RANDOM-APPROX '12*, 2012.

[38] A. Fiat and M. Naor, "Implicit $O(1)$ probe search," *SIAM Journal on Computing*, vol. 22, pp. 1–10, 1993.

[39] A. C.-C. Yao, "Should tables be sorted?" *Journal of the ACM*, vol. 28, pp. 615–628, 1981.

[40] A. V. Kuznetsov and B. S. Tsybakov, "Coding in a memory with defective cells." *Probl. Peredachi Inf.*, vol. 10, 1974.

[41] B. S. Tsybakov, "Defect and error correction." *Probl. Peredachi Inf.*, vol. 11, 1975.

[42] M. Braverman, "Polylogarithmic independence fools ac0 circuits." *Journal of the ACM*, vol. 57, no. 5, 2010.

[43] A. Tal, "Tight bounds on the fourier spectrum of ac0." ECCC: Electronic Colloquium on Computational Complexity, Tech. Rep. TR14-174, 2014.

[44] S. Dziembowski and K. Pietrzak, "Intrusion-resilient secret sharing," in *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, ser. FOCS '07, 2007, pp. 227–237.

[45] Y. Dodis and D. Wichs, "Non-malleable extractors and symmetric key cryptography from weak secrets," in *Proceedings of the 41st Annual ACM Symposium on Theory of Computing*, 2009, pp. 601–610.

[46] R. Shaltiel and C. Umans, "Simple extractors for all min-entropies and a new pseudorandom generator," *Journal of the ACM*, vol. 52, pp. 172–216, 2005.

[47] M. Cheraghchi and P. Indyk, "Nearly optimal deterministic algorithm for sparse walsh-hadamard transforms," Electronic Colloquium on Computational Complexity, Tech. Rep. TR15-076, 2015.

[48] R. Shaltiel, "How to get more mileage from randomness extractors," in *Proceedings of the 21th Annual IEEE Conference on Computational Complexity*, 2006, pp. 49–60.