On Randomness Extraction in AC0

Oded Goldreich[*] Emanuele Viola[†] Avi Wigderson[‡]

January 1, 2015

Abstract

We consider randomness extraction by $\mathcal{AC}^0$ circuits. The main parameter, n, is the length of the source, and all other parameters are functions of it. The additional extraction parameters are the min-entropy bound $k = k(n)$, the seed length $r = r(n)$, the output length $m = m(n)$, and the (output) deviation bound $\epsilon = \epsilon(n)$.

For $k \leq n/\log^{\omega(1)} n$, we show that $\mathcal{AC}^0$-extraction is possible if and only if $\frac{m}{r} \leq 1 + \text{poly}(\log n) \cdot \frac{k}{n}$; that is, the extraction rate m/r exceeds the trivial rate (of one) by an additive amount that is proportional to the min-entropy rate k/n. In particular, non-trivial $\mathcal{AC}^0$-extraction (i.e., $m \geq r+1$) is possible if and only if $k \cdot r > n/\text{poly}(\log n)$. For $k \geq n/\log^{O(1)} n$, we show that $\mathcal{AC}^0$-extraction of $r + \Omega(r)$ bits is possible when $r = O(\log n)$, but leave open the question of whether more bits can be extracted in this case.

The impossibility result is for constant ϵ, and the possibility result supports $\epsilon = 1/\text{poly}(n)$. The impossibility result is for (possibly) non-uniform $\mathcal{AC}^0$, whereas the possibility result hold for uniform $\mathcal{AC}^0$. All our impossibility results hold even for the model of bit-fixing sources, where k coincides with the number of non-fixed (i.e., random) bits.

We also consider deterministic $\mathcal{AC}^0$ extraction from various classes of restricted sources. In particular, for any constant $\delta > 0$, we give explicit $\mathcal{AC}^0$ extractors for poly$(1/\delta)$ independent sources that are each of min-entropy rate δ; and four sources suffice for $\delta = 0.99$. Also, we give non-explicit $\mathcal{AC}^0$ extractors for bit-fixing sources of entropy rate $1/\text{poly}(\log n)$ (i.e., having $n/\text{poly}(\log n)$ unfixed bits). This shows that the known analysis of the "restriction method" (for making a circuit constant by fixing as few variables as possible) is tight for $\mathcal{AC}^0$ even if the restriction is picked deterministically depending on the circuit.

Keywords: AC0, randomness extractors, general min-entropy sources, block sources, bit-fixing sources, multiple-source extraction.

[*] Weizmann Institute of Science, Rehovot, Israel. Partially supported by the Minerva Foundation with funds from the Federal German Ministry for Education and Research.

[†] Northeastern and Harvard University. Supported by NSF grant CCF-1319206. Work partially done while a visiting scholar at Harvard University.

[‡] Institute for Advanced Study, Princeton, NJ 08540, USA. Partially supported by NSF grant CCF-1412958.

Contents

1 Introduction

Randomness extractors, hereafter referred to as extractors, are procedures that transform sources of "weak randomness" into sources of almost perfect randomness. The feasibility of such a transformation depends on the specific notion of "weak randomness", and in most cases the transformation must be provided with a short (perfectly) random seed. Indeed, this fundamental problem has several versions, and each of them comes with a set of parameters. (See Shaltiel's survey [52] for a wide perspective as well as for a snapshot of the state of the art a decade ago.)

The most popular and general notion of weak randomness is parameterized by a probability bound, denoted 2^{-k}, such that no outcome may appear with probability that exceeds it. In such a case, k is called the min-entropy of the source. Additional parameters of the extraction problem include the length of the source, denoted n, the length of the seed, denoted r, the length of the (extracted) output, denoted m, and an upper bound on its deviation (from perfect randomness), denoted ϵ. In fact, n is viewed as the main parameter, and all other parameters are stated as functions of n. A function $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ is called a (k,ϵ)-extractor if for every X of min-entropy k it holds that $E(X,U_r)$ is ϵ-close to U_m, where U_ℓ denotes the uniform distribution over $\{0,1\}^\ell$. It is called a strong (k,ϵ)-extractor if for such X's it holds that $E(X,U_r)\circ U_r$ is ϵ-close to U_{m+r}. (Note that if E is a strong (k,ϵ)-extractor, then $E'(x,u) = E(x,u)\circ u$ is an (k,ϵ)-extractor.)

When ignoring computational issues, the exact trade-off between the various extraction parameters is known, but much research has been devoted to obtaining explicit constructions that approach the optimal bounds. Traditionally, an extractor is called explicit if it can be computed efficiently (i.e., in polynomial-time) or alternatively if Boolean circuits computing it can be constructed in poly(n)-time. It is known that some constructions are even more explicit than that; for example, the popular constructions of universal hashing functions [14] (known as the "mother of all extractors") are computable by highly uniform $\mathcal{AC}^0[2]$ circuits (i.e., constant-depth circuits of polynomial-size with parity gates). The same holds for Trevisan's celebrated extractor [53]. *Can one get any lower* (indeed to $\mathcal{AC}^0$)*?* This is the question we study here.

1.1 The most relevant prior work

Our starting point is the following negative result by Viola [56].

Theorem 1.1 (severe limitations on extraction in $\mathcal{AC}^0$ [56, Thm. 6.4]): *If a* $(k, 0.999)$*-extractor* $E : \{0,1\}^n \times \{0,1\}^{0.999m} \to \{0,1\}^m$ *is computable by a circuit* C (with negations and unbounded fan-in and and or gates)*, then* $\text{size}(C) \geq \exp(\Omega(n/k)^{1/(\text{depth}(C)-1)})$*. In particular, if* E *can be computed by a family of* $\mathcal{AC}^0$ *circuits, then there exists a positive polynomial* p *such that* $k(n) \geq n/p(\log n)$ *for all sufficiently large* n.

This result rules out $\mathcal{AC}^0$-extractors that either extract from entropy $k = n/\log^{\omega(1)} n$ or use a seed length r that is sublinear in the output length. However, the result leaves open the possibility that (non-trivial) $\mathcal{AC}^0$-extractors exist for other settings of parameters. Indeed, when making only the non-triviality requirement (i.e., $m = r + 1$), two such extractors existed in the literature. First, $\mathcal{AC}^0$ circuits can extract one bit from a source of logarithmic min-entropy when using a very long seed; specifically, when $r = n = m - 1$. (This can be done by sampling input-output pairs of the inner product function [33], cf. [5, 57].) Second, non-trivial $\mathcal{AC}^0$-extractors exist for min-entropy $k \geq n/\text{poly}\log n$ (using the "sample-then-extract" paradigm developed by Nisan and

Zuckerman [45] and refined by Vadhan [54]). In fact, this $\mathcal{AC}^0$-extractor can extract logarithmically many bits.

Theorem 1.2 (following [54, Thm. 7.4]): *For every $k(n) = n/\text{poly}(\log n)$ and $\epsilon(n) = 1/\text{poly}(n)$, there exist* (non-explicit) *$\mathcal{AC}^0$ circuits that compute a* strong *$(k(n), \epsilon(n))$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Theta(\log n)}$. Furthermore, the circuits have depth $4 + \left\lceil \frac{\log(n/k(n))}{\log\log n} \right\rceil$.*

The proof of Theorem 1.2 follows the proof of [54, Thm. 7.4], which combines an adequate sampler with an adequate extractor using the (sample-then-extract) composition theorem of [54, Thm. 6.3]. We note that both the sampler and the extractor used in the original proof of [54, Thm. 7.4] are non-explicit; furthermore, the (optimal) extractor used there is probably not computable by constant-depth circuits of poly(n)-size (let alone explicit ones). Instead, we shall use the (non-optimal) explicit extractor of [28, Sec. 5], which is computable by (uniform) constant-depth circuits of size poly(n). The resulting $\mathcal{AC}^0$-extractor inherits the non-explicitness of the sampler used in the proof of [54, Thm. 7.4]. Jumping ahead, we mention that we provide an explicit version of Theorem 1.2 (see Theorem 3.1) by using a new explicit sampler (see Theorem 3.2).

Hence, while a non-explicit $\mathcal{AC}^0$-extractor for $k \geq n/\text{poly}(\log n)$ is implicit in prior work, the explicit version (as stated in Theorem 3.1) relies on our new sampler (i.e., Theorem 3.2). In any case, the foregoing results do not refer to the general trade-offs between the parameters k, r and m that allow extraction to be performed in $\mathcal{AC}^0$.

1.2 Our main results

We study the following general question.

> Parameters enabling extraction in $\mathcal{AC}^0$: *For which values of k, r and m are (k, n^{-3})-extractors $E : \{0,1\}^n \times \{0,1\}^{r(n)} \to \{0,1\}^{m(n)}$ computable in $\mathcal{AC}^0$?*

Recall that, for logarithmic seed length (i.e., $r(n) = O(\log n)$), a min-entropy bound of $k(n) > n/\text{poly}(\log n)$ is a necessary condition (even for $m(n) = r(n) + \Omega(\log n)$ (see Theorem 1.1)), whereas $m(n) = r(n) + O(\log n)$ is achievable (by Theorem 1.2). These results mark the boundaries (between impossible and possible) as a function of k when $r(n) = O(\log n)$ and $m(n) = r(n) + \Theta(\log n)$. Indeed, this boundary refers to a line $(k, r = O(\log n), m = r + \Theta(\log n))$ in the three dimensional space $(k, r, m) \in [n]^3$. Our work is aimed at mapping the entire space, and it achieves this goal for $k < n/\log^{\omega(1)} n$. *In this setting, we show that $\mathcal{AC}^0$-extraction is possible if and only if $\frac{m}{r} \leq 1 + \text{poly}(\log n) \cdot \frac{k}{n}$; that is, the extraction rate m/r exceeds 1 by an additive amount that is proportional to the min-entropy rate k/n.* The region of $k \geq n/\text{poly}(\log n)$ remains partially unmapped (as indicated in Problem 1.6 below).

In general, our impossibility results are for constant ϵ, and the possibility results support $\epsilon = 1/\text{poly}(n)$. The impossibility results are for (possibly non-uniform) $\mathcal{AC}^0$, whereas all but one of our possibility results hold for uniform $\mathcal{AC}^0$ (assuming that the relevant functions (i.e., k, r, m and ϵ) are poly(n)-time computable). *All impossibility results hold even for the model of bit-fixing sources*, where k coincides with the number of non-fixed (i.e., random) bits.

When we write $k(n) < n/\text{poly}(\log n)$ we mean that, for every positive polynomial p and all sufficiently large n, it holds that $k(n) < n/p(\log n)$. When we write $k(n) \geq n/\text{poly}(\log n)$ we mean that there exists a polynomial p such that, for all sufficiently large n, it holds that $k(n) \geq n/p(\log n)$.

With these preliminaries in place, we turn to describe our main results. In the case of strong extraction we obtain a very clear dichotomy.

Theorem 1.3 (strong extraction in $\mathcal{AC}^0$):

- impossibility: *Strong extraction, even of a single bit, is impossible in $\mathcal{AC}^0$ for $k(n) < n/\text{poly}(\log n)$, regardless of the length of the seed.*
- possibility: *Strong extraction of $m_0 = \Omega(\log n)$ bits is possible in uniform $\mathcal{AC}^0$ for any $k(n) \geq n/\text{poly}(\log n)$, using a seed of length $O(\log n)$. Furthermore, in this case, for every $t < k/2m_0$, strong extraction of $t \cdot m_0$ bits is possible using a seed of length $O(t \cdot \log n)$.*

[The impossibility result follows by Theorem 5.3. The possibility result follows by Part 2 of Corollary 6.4, which in turn is based on Theorem 3.1 (combined with Theorem 6.3 for the furthermore part).]

We comment that, for $k(n) \geq n/\text{poly}(\log n)$, one can extract poly-logarithmically many bits in $\mathcal{AC}^0$ using a seed of logarithmic length but at an error rate of $1/\text{poly}(\log n)$; this can be done by using Trevisan's extractors [53] (insead of the extractor of [28, Sec. 5]). We now turn to ordinary (i.e., non-strong) extraction, starting with the minimal case of non-trivial extraction.

Theorem 1.4 (ordinary extraction in $\mathcal{AC}^0$, the case of $m = r + 1$):

- impossibility: *Extraction of $r(n) + 1$ bits is impossible in $\mathcal{AC}^0$ for $r(n) \cdot k(n) < n/\text{poly}(\log n)$.*
- possibility: *There exists a constant $c > 2$ such that, for every $k(n) \geq c \cdot \log(n/\epsilon)$ and every $r(n) \geq (n \cdot \log^3 n)/k(n)$, extraction of $r(n) + \min(\text{poly}(\log n), k(n)/2)$ bits is possible in uniform $\mathcal{AC}^0$.*

[The impossibility result follows by Part 1 of Theorem 5.4, whereas the possibility result follows by Corollary 6.2.]

Theorem 1.5 (ordinary extraction in $\mathcal{AC}^0$, the case of $m = r + \Theta(r)$):

- impossibility: *Extraction of $r(n) + \Omega(r(n))$ bits is impossible in $\mathcal{AC}^0$ for $k(n) < n/\text{poly}(\log n)$, regardless of r.*
- possibility: *There exists a constant $c > 0$ such that, for every $k(n) \geq n/\text{poly}(\log n)$ and every $r(n) \in [\Omega(\log n), k(n)/c]$, extraction of $(1 + c) \cdot r(n)$ bits is possible in uniform $\mathcal{AC}^0$ using a seed of length $r(n)$.*

[The impossibility result follows by Part 2 of Theorem 5.4, whereas the possibility result (of Theorem 1.5) follows by the possibility result of Theorem 1.3.]

Note that the impossibility result of Theorem 5.4 establishes the same bound as Viola's [56, Thm. 6.4] (see Theorem 1.1), but does so for bit-fixing sources.

Let us restate the message of Theorem 1.5: It says that extracting $m(n) = r(n) + \Theta(r(n))$ bits is impossible if $k(n) < n/\text{poly}(\log n)$ (regardless of the size of r), whereas if $k(n) \geq n/\text{poly}(\log n)$ then using a seed of length $r(n) = \Omega(\log n)$ we can extract $r(n) + \Omega(r(n))$ bits. However, it is not clear whether we cannot extract significantly more bits in the latter case.

Open Problem 1.6 (extracting more bits at rate of at least $1/\mathrm{poly}(\log n)$): *Can one extract more that* $\mathrm{poly}(\log n) \cdot r(n)$ *bits in* $\mathcal{AC}^0$ *using a seed of length* $r(n) = \Omega(\log n)$*, when* $k(n) > n/\mathrm{poly}(\log n)$*? In particular, can one extract more that* $\mathrm{poly}(\log n)$ *bits using a seed of logarithmic length? For starters, what about the special case of constant min-entropy rate, that is,* $k(n) = \Omega(n)$*?*

We conjecture that the answer is negative and provide some evidence for this conjecture in Section 4.2. Recall that for $k(n) \geq n/\mathrm{poly}(\log n)$, we can extract poly-logarithmically many bits using a seed of logarithmic seed but at an error rate of $1/\mathrm{poly}(\log n)$; see Part 2 of Corollary 3.4. Indeed, a minor open problem regarding this case is to reduce the error rate to $1/\mathrm{poly}(n)$.

Theorems 1.4 and 1.5 can be interpolated, and they indeed follow as special cases of the following generalization (which is our main result).

Theorem 1.7 (ordinary extraction in $\mathcal{AC}^0$, the general case of $m = r + m'$):

- impossibility: *For any* $m'(n) \geq 1$*, extraction of* $r(n) + m'(n)$ *bits is impossible in* $\mathcal{AC}^0$ *if* $k(n) < \frac{m'(n)}{r(n)+m'(n)} \cdot \frac{n}{\mathrm{poly}(\log n)}$ (equiv., if $(r(n) + m'(n)) \cdot k(n) < m'(n) \cdot n/\mathrm{poly}(\log n)$).
- possibility: *There exists a constant* $c > 1$ *such that for every* $k(n)$*,* $m'(n)$*, and* $r(n)$ *such that* $k(n) \geq c \cdot (m'(n) + \log(n/\epsilon))$*, extraction of* $r(n) + m'(n)$ *bits is possible in uniform* $\mathcal{AC}^0$ *in each of the following two cases.*
  1. *For* $r(n) \cdot k(n) \geq \lceil m'(n)/\mathrm{poly}(\log n)\rceil \cdot O(n \log^2 n)$ *and* $k(n) > \mathrm{poly}(\log n)$*;*
  2. *For* $r(n) = n$*. Furthermore, for* $r(n) \cdot k(n) \geq n/\mathrm{poly}(\log n)$*, we have* $m'(n) = \Omega(k(n)) - \mathrm{poly}(\log n)$*.*

[The impossibility result follows by Theorem 5.4, whereas the possibility result follows by Corollary 6.4.]

The result of Theorem 1.7 is almost tight for $k(n) < n/\mathrm{poly}(\log n)$: Extraction of $r(n) + m'(n)$ bits is impossible in $\mathcal{AC}^0$ if $r(n) \cdot k(n) + m'(n) \cdot k(n) < m'(n) \cdot n/\mathrm{poly}(\log n)$, which is equivalent (in this case) to $r(n) \cdot k(n) < m'(n) \cdot n/\mathrm{poly}(\log n)$, but is possible if $r(n) \cdot k(n) \geq m'(n) \cdot n/\mathrm{poly}(\log n)$ (provided that $\mathrm{poly}(\log n) < m'(n) \leq k(n) - O(\log n)$). Hence, what we really do not know refers to the range of $k(n) \geq n/\mathrm{poly}(\log n)$; that is, to Problem 1.6.

A different perspective on Theorem 1.7 is obtained by considering the relation between k/n and m'/r. Theorem 1.7 asserts that for $m'/r \in [0, \Theta(1)]$, extraction in $\mathcal{AC}^0$ is possible if $k/n \geq f(n) \cdot m'/r$ for some $f(n) = 1/\mathrm{poly}(\log n)$ and impossible if $k/n < f'(n) \cdot m'/r$ for every $f'(n) = 1/\mathrm{poly}(\log n)$. Recall that Theorems 1.1 and 1.2 only refer to the case of $m'/r = \Theta(1)$, whereas Theorem 1.7 covers all $m'/r \in [0, \Theta(1)]$. Problem 1.6 refers to $m'/r = \omega(1)$ (or actually to $m'/r = (\log n)^{\omega(1)}$).

We highlight the fact that non-trivial extraction (i.e., $m'(n) = 1$) is possible in $\mathcal{AC}^0$ if and only if $k \cdot r > n/\mathrm{poly}(\log n)$. In contrast, the threshold for "significant" $\mathcal{AC}^0$-extraction, that is $m' = \Omega(r)$, is $k > n/\mathrm{poly}(\log n)$. Hence, for $r > \mathrm{poly}(\log n)$, there is a gap between the min-entropy bound that allows non-trivial $\mathcal{AC}^0$-extraction and the min-entropy required for extracting $r + \Omega(r)$ bits in $\mathcal{AC}^0$.

Extraction with respect to restricted sources. In relation to Problem 1.6, we mention that both in the model of block sources and in the model of bit-fixing sources, we can *extract more than poly-logarithmically many bits using a seed of logarithmic length*, where in both cases the min-entropy rate is at least $1/\mathrm{poly}(\log n)$ (and extraction is in $\mathcal{AC}^0$). In fact, in both cases, $n/\mathrm{poly}(\log n)$ bits are extracted.

Theorem 1.8 (extraction in $\mathcal{AC}^0$ for bit-fixing sources and block sources): *For any* $k(n) \geq n/\text{poly}(\log n)$*, extraction of* $n/\text{poly}(\log n)$ *bits using a seed of length* $O(\log n)$ *is possible in uniform* $\mathcal{AC}^0$ *for bit-fixing sources in which* $k(n)$ *of the* n *bits are not fixed. Ditto for block sources with* $\Theta(k(n)/\log n)$ *blocks such that each block has conditional min-entropy* $\Omega(\log n)$*.*

[See Corollary 4.3 and Theorem 5.2.]

Recall that the bit-fixing model allows for deterministic extractors, which work even for lower min-entropy rates, but these extractors are not computable by $\mathcal{AC}^0$ (which is to be expected in light of the fact that our impossibility results hold for the bit-fixing model). Still, it is possible that whenever extraction in $\mathcal{AC}^0$ is possible for bit-fixing sources, it is also possible via deterministic extractors. We show that this is essentially the case.

Theorem 1.9 (deterministic extraction in $\mathcal{AC}^0$ for bit-fixing sources): *For any* $k(n) \geq n/\text{poly}(\log n)$*, deterministic extraction of* $n/\text{poly}(\log n)$ *bits is possible in* $\mathcal{AC}^0$ *for bit-fixing sources in which* $k(n)$ *of the* n *bits are not fixed.*

[See Theorem 5.7.] Unlike all previously mentioned results, this possibility result only claims the existence of $\mathcal{AC}^0$ circuits (but does not provide an explicit construction). We mention (see Theorem 5.13) that we can construct explicit $\mathcal{AC}^0$ circuits that compute a deterministic disperser for this class of sources (i.e., we present a circuit that is not constant on any such source).

A circuit complexity perspective (w.r.t random restrictions). A (deterministic) extractor for bit-fixing sources in which $k(n)$ of the n bits are random constitutes a circuit that is not trivialized (i.e., does not simplify to a constant) under any restriction that keeps $k(n)$ of the variables alive.[1] Hence, Hastad's analysis [31] of the random restriction method [1, 23, 59], which implies that any depth d circuit of size $s(n)$ trivializes under a random restriction that keeps $n/O(\log^{d-1} s(n))$ variables alive, is optimal in a very strong sense: Not only that there exist $\mathcal{AC}^0$ circuits that do not trivialize *under a random restriction* that keeps $n/\text{poly}(\log n)$ variables alive, but these circuits are not trivialized *under any restriction* that keeps $n/\text{poly}(\log n)$ variables alive. In other words, *a restriction that is carefully selected based on the target circuit cannot achieve significantly better parameters than a random restriction* (i.e., cannot trivialize the circuit while leaving significantly more variables alive). There are contexts in which circuit-dependent restriction yields stronger lower bounds than its randomized counterpart. These contexts include $\mathcal{AC}^0$ circuits of nearly-linear size [15], and of threshold circuits of nearly-linear size [34].

Deterministic extraction from several independent sources. Another model allowing for deterministic extractors is the model of two or more independent sources each having min-entropy at least $k(n)$ (cf., e.g., [16, 6, 7, 37]). While this model allows for deterministic extractors, which work even for min-entropy rates below $1/\text{poly}(\log n)$, the known extractors are not computable by $\mathcal{AC}^0$ (which is to be expected in light of the fact that our impossibility results hold also for this model). We show that deterministic extraction in $\mathcal{AC}^0$ is possible also in this model.

Theorem 1.10 (deterministic extraction in $\mathcal{AC}^0$ for the multi-source model): *For any constant* $\delta > 0$*, there exist explicit* $\mathcal{AC}^0$*-extractors for* $\text{poly}(1/\delta)$ *independent sources that are each of min-entropy rate* δ*. For* $\delta = 0.99$*, four sources suffice.*

See Section 7. In the two-source model, we only obtain such extractors for rates that approach 1 (i.e., $\delta \geq 1 - \log^{-4} n$).

[1] In fact, the same holds for dispersers, which are functions that map any such source to a non-trivial distribution.

1.3 Techniques

Our results build on known results and known techniques. These are augmented by several new constructions of various pseudorandom objects including

- two alternative constructions of averaging samplers (see Sections 3.2 and 3.4, respectively);
- an $\mathcal{AC}^0$-extractor for bit-fixing sources extracting $n/\mathrm{poly}(\log n)$ bits using a logarithmically long seed (see Section 5.1);
- a deterministic $\mathcal{AC}^0$-extractor for bit-fixing sources extracting $\mathrm{poly}(\log n)$ bits (see Section 5.3);
- a deterministic two-source $\mathcal{AC}^0$-extractor (see Section 7.1);
- deterministic many-source $\mathcal{AC}^0$-extractors for lower entropy rate (see Sections 7.2 and 7.3).

These and other contributions are mentioned in Section 1.5, *where we emphasize interesting aspects that are not mentioned here*. In the current section, we focus on a few common themes that re-occur in several proofs. Similar ideas were used before, but we found the current incarnations useful and worthy of highlighting.

Generating pseudorandom partitions. Loosely speaking, the problem is to generate a pseudorandom partition of $[n]$ into m equal-sized sets such that each set has a strong hitting or sampling property. We wish to do this using a logarithmic amount of randomness and for $m = n/\mathrm{poly}(\log n)$. Specifically, Lemma 5.2.1 asserts a pseudorandom partition generator for $n/m = O(\rho^{-1}\log(1/\epsilon))^2$ such that each set of density ρ is hit by each subset of the partition with probability at least $1-\epsilon$.

The idea is to use a fixed partition of $[n]$ into n/m disjoint m-cycles and a standard hitter of sample complexity $\sqrt{n/m}$. Hoping that this hitter generates a set that hits each cycle at most once, we augment this set to a cover of all n/m cycles, and use the m "shifts" of this augmented set as a partition. Using a suitable implementation, the aforementioned hope does materialize with constant probability, and we augment the construction so to obtain a good partition with overwhelmingly high probability. (The proof of Lemma 5.2.1 presents a specific instantiation of this idea that is implementable in uniform $\mathcal{AC}^0$.)

Somewhat related problems arise in the proofs of Theorems 3.2 and 3.6. In these proofs, we need to construct a sampler that generates a very large set (say of size $n^{1/3}$) of *distinct elements*. This is relatively easy if the elements of the sample are pairwise independent. Getting sets of size greater than $\sqrt{n}$ requires additional ideas, which appear in the proof of Theorem 3.6.

In the latter case we rely on the fact that the number of occurrences of each element in the sample is not large, and that this number can be computed using a high quality hashing scheme. We then include each element in the final sample with probability that is proportional to the number of occurrences in the first sample, while noting that this random sieving preserves the sampling property of the first sample. (We warn that the implementation of this procedure in $\mathcal{AC}^0$ is not straightforward.)

Combining various pseudorandom properties. As hinted above, we may want to have a good sampler that uses samples that are uniformly distributed in the domain in a $O(1)$-wise independent manner. The problem is that good samplers use random walks on expander graphs, and in this case the samples are each uniformly distributed but they are not even pairwise independent.

The solution is to XOR an $O(1)$-wise independent sequence with the vertices visited in the random walk (see Claim 3.2.2). We show that the combined sampler inherits the properties of each of the original samplers. Indeed, such combinations were used before for different properties (e.g., Impagliazzo and Wigderson [35] de-randomized Yao's XOR Lemma by XORing the output of the "projected seed generator" of [44] with the output of a random walk generator).

1.4 The perspective of error reduction

As articulated by Zuckerman [62], there is a close relation between randomness extractors that are computable in a natural complexity class such as $\mathcal{AC}^0$ and error-reduction procedures *computable in that class*. The error-reduction procedures referred to here are confined to generating several inputs, applying the original circuit to each of these inputs, and ruling by majority. Hence, these procedures are closely related to averaging samplers (as defined implicitly in the next paragraph).

Specifically, a $(k, 0.1)$-extractor $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ yields a sampler $S : \{0,1\}^n \to (\{0,1\}^m)^{2^r}$ such that for every $f : \{0,1\}^m \to \{0,1\}$ with probability at least $1 - 2 \cdot 2^{-(n-k)}$ it holds that $2^{-r} \cdot \sum_{s \in S(U_n)} f(s) = (1 \pm 0.1) \cdot \mathbf{E}[f(U_m)]$. (Just use $S(x) = \{E(x,\sigma) : \sigma \in \{0,1\}^r\}$.) The converse holds too (by using $E(x,\sigma) = S(x)_\sigma$): A sampler $S : \{0,1\}^n \to (\{0,1\}^m)^{2^r}$ that satisfies $\mathbf{Pr}[2^{-r} \cdot \sum_{s \in S(U_n)} f(s) = (1 \pm 0.1) \cdot \mathbf{E}[f(U_m)]] > 1 - 0.1 \cdot 2^{-(n-k)}$ for every $f : \{0,1\}^m \to \{0,1\}$, yields a $(k, 0.2)$-extractor $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$. For simplicity, let us ignore the small slackness (i.e., 0.1 vs 0.2, and 2 vs 0.1) in the following discussion.

In light of the tight relationship between the extractor and the sampler in the foregoing paragraph it holds that, for $r = O(\log n)$ and any length parameter m, having $(k, 0.1)$-extractors in $\mathcal{AC}^0$ and having samplers with error $2^{-(n-k)}$ computable in $\mathcal{AC}^0$ is equivalent. Note that efficient error-reduction requires $m(n) = n^{\Omega(1)}$, since n represents the length of the input to the sampler and $m(n)$ the length of the sampled strings, which means that $n = \text{poly}(m(n))$ must hold.

Recall that Theorem 1.1 refers to *relatively weak extractors*; that is, ones that extracts a constant factor more bits than the length of the seed (i.e., $m(n) \geq (1 + \Omega(1)) \cdot r(n)$). It asserts that $\mathcal{AC}^0$ circuits cannot compute such extractors for min-entropy rate that is smaller than $1/\text{poly}(\log n)$. In contrast, recall that $\mathcal{AC}^0[2]$ (i.e., $\mathcal{AC}^0$ with parity gates) circuits can compute very good extractors (e.g., Trevisan's [53]); for example, such circuits can compute (k, ϵ)-extractors with a logarithmically long seed for $k(n) = m(n)^2 = \sqrt{n}$ (and $\epsilon(n) = 1/\text{poly}(n)$).

Nevertheless, Theorem 1.1 says nothing about $\mathcal{AC}^0$-extraction from sources of higher min-entropy rate (i.e., rate at least $1/\text{poly}(\log n)$). Actually, Theorem 1.2 says that $\mathcal{AC}^0$-extraction is possible in this case, but it only provides for extracting logarithmically many bits (i.e., $m(n) = O(\log n)$), whereas efficient error-reduction requires $m(n) = n^{\Omega(1)}$. Furthermore, Vadhan's approach [54], which underlies the proof of Theorem 1.2, seems to yield $\mathcal{AC}^0$-extractors of logarithmic seed length only when the output length is polylogarithmic, even when the min-entropy rate is a constant. The question (see Problem 1.6) is whether one can extract more randomness under these conditions (i.e., using a source of constant min-entropy and a logarithmically long seed).

While a positive resolution regarding $m(n) = n^{\Omega(1)}$ would imply efficient error-reduction for $\mathcal{AC}^0$, a bypass was found recently. Specifically, a recent revision of [29] (posted in June 2014) establishes error-reduction for $\mathcal{AC}^0$ at the same level that would have been implied by the best $\mathcal{AC}^0$-extractors that are not ruled out by Theorem 1.1. That is, it offers error-reduction at a level that corresponds to min-entropy $k(n) = n/\text{poly}(\log n)$, output length $m(n) = n^{\Omega(1)}$, and logarithmic seed length (i.e., $r(n) = O(\log n)$).

This was obtained by observing that we do not really need information-theoretic extractors (computable in $\mathcal{AC}^0$), but rather extractors (computable in $\mathcal{AC}^0$) that output distributions that fool all $\mathcal{AC}^0$ circuits. Alternatively, it suffices to extract randomness for auxiliary circuits obtained by shrinking the input of the original $\mathcal{AC}^0$ circuits by using the pseudorandom generator of Nisan [43, 44]. Hence, randomness-efficient error-reduction for $\mathcal{AC}^0$ was obtained without presenting an $\mathcal{AC}^0$-extractor with the corresponding parameters. In other words, there is a gap (at least in our knowledge) between *general error-reduction implementable in* $\mathcal{AC}^0$ (via samplers as reviewed above) and *error-reduction for* $\mathcal{AC}^0$, which may be defined as obtaining *samplers that satisfy the sampling requirement only with respect to functions that f that are computable in $\mathcal{AC}^0$.*

This gap in our knowledge provides a new motivation for resolving Problem 1.6, but now a resolution in the negative direction would be more interesting. Such a result would mean that, for a natural choice of parameters, randomness-efficient error-reduction for $\mathcal{AC}^0$ exists while a corresponding $\mathcal{AC}^0$-extractor does not exist (where the correspondence between parameters is as in the standard relation articulated by Zuckerman [62]).

1.5 A roadmap and additional comments on the technical contents

Following the preliminaries, this write-up proceeds as follows. In Section 3, we prove Theorem 3.1, which is based on a non-explicit construction of Vadhan [54, Thm. 7.4]. Our improvement boils down to presenting an explicit "averaging sampler" with parameters that are comparable to the non-explicit construction. One key observation underlying the new construction is that *the relaxed notion of averaging sampler as defined by Vadhan in [54, Def. 6.1] can be composed and manipulated in ways that are not possible with the standard definition of averaging samplers.* The reason is that Vadhan's notion is actually a hybrid of the notions of hitters and (standard) averaging samplers. Using the new sampler one can also improve the parameters in some of Vadhan's explicit constructions of local extractors [54].

In general, Section 3 demonstrates the relevance of the study of local extractor to extraction in $\mathcal{AC}^0$. In particular, local extractors yield constant-depth circuits of size that is exponential in the seed length and in the locality. In some cases, the size can be made even smaller (e.g., sub-exponential in the locality).

In Section 4 we consider block-sources. In Section 4.1 we show that applying an extractor to individual blocks of a block-source, while using the same seed in all applications, yields an extractor. This result seems to be folklore, but we believe that it is a very useful one, since we think that this extraction strategy is a natural thing to do when actually having a block-source. This is relevant to the context of $\mathcal{AC}^0$, because the resulting extractor preserve the computational complexity of the original extractor. In Section 4.2 we consider *the difficulty of converting an arbitrary high min-entropy source into a block-source: Loosely speaking, we show that two natural approaches to this task fail.*

In Section 5 we consider $\mathcal{AC}^0$-extractors for bit-fixing sources. In Section 5.1, we present our first construction, which uses a randomized procedure that outputs a partition of $[n]$ into small subsets that intersect any set of sufficient density (with high probability). The procedure uses a logarithmic amount of randomness and is explicit (and hence is implementable by uniform $\mathcal{AC}^0$ circuits). It yields an $\mathcal{AC}^0$-extractor that uses a logarithmically long seed and extracts $n/\text{poly}(\log n)$ bits from bit-fixing sources of min-entropy $n/\text{poly}(\log n)$. In Section 5.3 we combine the latter extractor with a new deterministic extractor (which extracts poly-logarithmically many bits), and obtain a deterministic extractor that essentially matches the performance of the seeded extractor.

The former deterministic extractor is based on *a new $\mathcal{AC}^0$-reduction of the task of extraction from (oblivious) bit-fixing sources of entropy rate that tends to 0* (i.e., $1/\text{poly}(\log n)$) *to the task of extraction from non-oblivious bit-fixing sources of entropy rate that tends to 1* (i.e., $1-1/\text{poly}(\log n)$), whereas the latter task was treated by Ajtai and Linial [4].[2] Indeed, the reduction is from a more restricted type of sources to a broader type of sources, but the sources of the more restricted type have significantly lower entropy.

The parameters obtained by these extractors are essentially optimal, with respect to $\mathcal{AC}^0$-extraction from bit-fixing sources. Indeed, in Section 5.2, we present impossibility results that extend those that are stated in Theorem 1.1: One result asserts that that $\mathcal{AC}^0$ circuits cannot compute a strong extractor for bit-fixing sources when the number of "unfixed" (i.e., random) bits is $n/(\log n)^{\omega(1)}$ (regardless of the seed length). In general, Section 5.2 provides proofs of all our impossibility results, *showing that the relevant bounds hold even for extractors that should only work for bit-fixing sources.*

In Section 5.4 we show that the foregoing impossibility results regarding extraction from bit-fixing sources do not hold for a restricted class of such sources, called zero-fixing sources [18], consisting of bit-fixing sources in which all fixed bits are set to zero. We show that $\mathcal{AC}^0$ *circuits, which use a seed of logarithmic length, can extract from zero-fixing sources that contain only a logarithmic number of random bits.*

Turning back to general sources of min-entropy k, in Section 6 we consider extraction with a seed of linear length. In Section 6.1 we generalize Viola's construction [57, Lem. 4.3] of a non-trivial extractor in $\mathcal{AC}^0$ to one that outputs $\text{poly}(\log n)$ additional bits (rather than one). In Section 6.2 we show that independent applications of an extractor (i.e., with independently distributed seeds) yield an extractor, provided that the total number of extracted bits does not exceed the min-entropy bound. This is a naive result, which relies on well-known ideas, but it is advantageous in the context of $\mathcal{AC}^0$ because the resulting extractor preserve the computational complexity of the original extractor. Indeed, this simple observation is pivotal in establishing the general upper bound of Theorem 1.7, which presents a trade-off between the seed length and the number of (additional) bits extracted.

In Section 7, we consider deterministic $\mathcal{AC}^0$-extractor for several independent sources. Leaving open the question of extraction from *pairs* of sources of some constant min-entropy rate, we prove the existence of $\mathcal{AC}^0$-extractors for pairs of sources of min-entropy rate $1-\log^{-4} n$. This is obtained by presenting a *uniform $\mathcal{AC}^0$-reduction of the task at hand to the task of extraction from non-oblivious bit-fixing sources of entropy rate $1-O(\log^{-3} n)$*. (Unlike in Section 5.3 here the reduction is between incomparable types of sources and the target sources have lower entropy.) We also present, for any constant $\delta > 0$, *explicit $\mathcal{AC}^0$-extractors for* $\text{poly}(1/\delta)$ *independent sources that are each of min-entropy rate δ, whereas for $\delta = 0.99$ four sources suffice.*

Finally, in Section 8, we list and restate open problems that are mentioned in various parts of the paper.

[2] In non-oblivious bit-fixing sources the fixed bits may be determined as a function of the values of the random bits, whereas in (oblivious) bit-fixing sources the fixed bits are set indepedently of the values of the random bits. The connection between influence of sets as studies in [4] and deterministic extraction from non-oblivious bit-fixing sources was made in [40].

2 Preliminaries

By "circuits" we mean Boolean circuits with negations and unbounded fan-in `and` and `or` gates. Since we deal with a very low complexity class (i.e., $\mathcal{AC}^0$), it is important to be careful about the representation of objects. In particular, elements of $[n] = \{1, 2, ..., n\}$ are represented as $(\lfloor \log_2 n \rfloor + 1)$-bit long strings, and subsets of $[n]$ are represented as (unsorted) sequences over $[n]$. This convention is important for supporting an efficient implementation of the projection operation; that is, for a sequence $x = (x_1, ..., x_n) \in \{0,1\}^n$ and $I \subseteq [n]$, we let X_I denote the projection of x on I (i.e., if $I = (i_1, ..., i_t)$, then $X_I = (x_{i_1}, ..., x_{i_t})$). Indeed, the mapping $(x, (i_1, ..., i_t)) \mapsto x_{(i_1,...,i_t)}$ is computable in uniform $\mathcal{AC}^0$ (e.g., by having the j^{th} output bit equal $\bigvee_{i\in[n]}(x_i \wedge (i = i_j))$).[3]

By $\log n$ we mean $\log_2 n$, whereas by $\text{poly}(n)$ we mean any (unspecified) positive polynomial. In several statements (e.g., Theorem 1.1), we preferred to use 0.999 (resp., 0.499) rather than "for every constant smaller than one" (resp., "for every constant smaller than half").

Another general convention is that multiple occurrences of the same random variable mean that the same random value is assigned in all occurrences; that is, if X is a random variable, then (X, X, X) represents the random variable obtained by selecting x according to X and outputting (x, x, x). By U_ℓ we denote a random variable that is uniformly distributed over $\{0,1\}^\ell$.

Below we review the basic definitions regarding extractors as well as some results regarding locally computable extractors.

2.1 General extractors

We recommend Shaltiel's survey [52] for a general introduction to randomness extractors.

Definition 2.1 (min-entropy and (n,k)-sources): *The* min-entropy *of a random variable X, denoted $\mathbf{H}_\infty(X)$, is $\min_x\{\log_2(1/\mathbf{Pr}[X = x])\}$. An (n,k)*-source *is a random variable X assuming values in $\{0,1\}^n$ such that $\mathbf{H}_\infty(X) \geq k$.*

(Indeed, $\max_x\{|\mathbf{Pr}[X = x]\} \leq 2^{-\mathbf{H}_\infty(X)}$.) In the following definition, $\mathbf{\Delta}[X\,; Y]$ denotes the statistical difference (a.k.a variation distance) between X and Y; that is,

$$\mathbf{\Delta}[X\,; Y] \stackrel{\text{def}}{=} \frac{1}{2} \cdot \sum_v |\mathbf{Pr}[X = v] - \mathbf{Pr}[Y = v]| = \max_S\{\mathbf{Pr}[X \in S] - \mathbf{Pr}[Y \in S]\}$$

Definition 2.2 ((seeded) randomness extractors): *The function $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ is called an* ϵ-error extractor for a class of sources $\mathcal{C}$ *if for every X in $\mathcal{C}$ it holds that $\mathbf{\Delta}[E(X, U_r)\,; U_m] \leq \epsilon$. It is called a* strong ϵ-error extractor for $\mathcal{C}$ *if for every X in $\mathcal{C}$ it holds that $\mathbf{\Delta}[E(X, U_r) \circ U_r\,; U_m \circ U_r] \leq \epsilon$. When $\mathcal{C}$ is the class of (n,k)-sources, we call E a* (k,ϵ)-extractor.

Note that

$$\mathbf{\Delta}[E(X, U_r) \circ U_r\,; U_m \circ U_r] = \mathbf{E}_{s\leftarrow U_r}[\mathbf{\Delta}[E(X, s)\,; U_m]].$$

We say that an extractor is explicit if a circuit computing it can be constructed in $\text{poly}(n)$-time. Indeed, to make sense of this definition, one should consider a family of extractors parameterized

[3] In contrast, if $I \subseteq [n]$ is represented by the n-bit long indicator vector $\chi = (\chi_1, ..., \chi_n)$ such that $\chi_i = 1$ if $i \in I$ and $\chi_i = 0$ otherwise, then the mapping $(x, I) \mapsto x_I$ is not computable in $\mathcal{AC}^0$. The reason is that counting (i.e., computing $\sum_{i\in[n]} b_i$) is $\mathcal{AC}^0$-reducible to the foregoing operation by setting $x = 1^n$ and measuring the length of x_I, where $I = \{i \in [n] : b_i = 1\}$, while relying on $(b_1, ..., b_n)$ being an admissible representation of I.

by n; that is, we actually consider the family $\{E_n : \{0,1\}^n \times \{0,1\}^{r(n)} \to \{0,1\}^{m(n)}\}_{n\in\mathbb{N}}$ such that E_n is an $(k(n),\epsilon(n))$-extractor. Likewise, when using asymptotic notation in reference to an extractor $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$, we actually refer to a family as above; that is, we always view the source length (i.e., n) as a varying parameter (which determines all other parameters; e,g., $r = r(n)$, $m = m(n)$, etc). Furthermore, we assume that $k, r, m : \mathbb{N} \to \mathbb{N}$ are monotonically non-decreasing and that $\epsilon : \mathbb{N} \to [0,1]$ is monotonically non-increasing. These conventions are used extensively throughout this write-up.

2.2 Local extractors

Locally computable extractors were systematically studied by Vadhan [54]. These constructs as well as some of Vadhan's techniques (see especially [54, Sec. 6]) are used by us towards constructing extractors that can be computed in $\mathcal{AC}^0$.

Definition 2.3 (t-local extractors): *The extractor $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ is called t-*local *if of every $s \in \{0,1\}^r$ the residual function $f_s(x) \stackrel{\rm def}{=} E(x,s)$ depends on at most t bits of x.*

For the case of constant min-entropy rate, we have the following result.

Theorem 2.4 (special case of [54, Thm. 8.5]): *For every constants $\delta, \beta > 0$ and $\epsilon(n) = 1/\text{poly}(n)$, there exists an explicit $O(\log n)$-local strong $(\delta n, \epsilon(n))$-extractor $E : \{0,1\}^n \times \{0,1\}^{\beta\cdot m(n)} \to \{0,1\}^{m(n)}$ for $m(n) = \Theta(\log n)$. Furthermore, the extractor is computable in uniform $\mathcal{AC}^0$.*

The furthermore claim is not stated in [54], but it follows from the main claim by combining the circuits that compute the residual functions (obtained by fixing the seed). Specifically, we combine depth-two circuits that compute the residual functions (where each function depends on $O(\log n)$ bits), with depth-two circuits that select the adequate function, obtaining depth-three circuits. For the case of min-entropy rate $1/\text{poly}(\log n)$, we mention the following non-explicit construction.

Theorem 2.5 (special case of [54, Thm. 7.4]): *For every $k(n) = n/\text{poly}(\log n)$ and $\epsilon(n) = 1/\text{poly}(n)$, there exists an* (non-explicit) *$(2n/k(n))$-local strong $(k(n),\epsilon(n))$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Omega(\log n)}$.*

(Note that here the seed is longer than the output, but the result is non-trivial since the extractor is strong.) While the relevance of Theorem 2.4 to our study was demonstrated in its furthermore-part, the relevance of Theorem 2.5 is less clear and arises from the technique used in its proof. The point is that the proof of Theorem 2.5 is based on the composition theorem of [54, Thm. 6.3], which implies that combining an adequate sampler with an adequate extractor yields an extractor that can be computed *more efficiently than standard extractors.* Loosely speaking, the sampler is used to sample relatively few bits in the source, and the extractor is applied to the resulting sequence of bits, which approximately maintains the entropy rate of the source. Thus, the complexity of the resulting extractor is related to the complexity of sampling and to the complexity of extraction from a much shorter source. (This fact will be extensively used in Section 3.)

The sample-and-extract paradigm goes back to the work Nisan and Zuckerman [45], but the point here is using it in order to reduce the computational complexity of extraction. Furthermore, better parameters are obtained by using the following relaxed notion of an averaging sampler, introduced by Vadhan [54], which is a hybrid of a sampler and a hitter (see discussion following the definition).

Definition 2.6 (averaging samplers, relaxed [54, Def. 6.1]):[4] *A function $S : \{0,1\}^r \to [n]^t$ is called a (μ, μ', γ)-*averaging sampler *if for every $f : [n] \to [0,1]$ such that $\rho(f) \overset{\text{def}}{=} \mathbf{E}_{i\in[n]}[f(i)] \geq \mu$ it holds that*

$$\mathbf{Pr}_{I\leftarrow S(U_r)}\left[\frac{1}{t}\sum_{i\in I} f(i) < \mu'\right] \leq \gamma. \tag{1}$$

Furthermore, it is required that $|S(u)| = t$ for every $u \in \{0,1\}^r$; that is, the sampler must always generate t distinct *elements.*

The standard notion of a (δ, γ)-averaging sampler is obtained from Definition 2.6 by requiring that S is an $(\mu, \mu - \delta, \gamma)$-averaging sampler for every $\mu \in [0,1]$. (Note that in such a case an upper bound on the probability that $\frac{1}{t}\sum_{i\in I} f(i) > \rho(f) + \delta$ follows by considering the function $1 - f$.) The definition of a (μ, γ)-hitter is obtained from Definition 2.6 by replacing Eq. (1) with $\mathbf{Pr}_{I\leftarrow S(U_r)}[\sum_{i\in I} f(i) = 0] \leq \gamma$. (Indeed, the latter definition remains intact if one only considers Boolean functions $f : [n] \to \{0,1\}$ (such that $\rho(f) \geq \mu$).)[5] (In Appendix A.1 we prove two useful features of such averaging samplers, although these features are not essential to this write-up.)

The relevance of averaging samplers to our project is captured by the following composition theorem of Vadhan [54].

Theorem 2.7 (sample-then-extract [54, Thm. 6.3]):[6] *For any $0 < 3\tau < \delta \leq 1$, let $\mu = (\delta - 2\tau)/\log(1/\tau)$ and $\mu' = (\delta - 3\tau)/\log(1/\tau)$. Suppose that the following two conditions hold.*

1. *$S : \{0,1\}^r \to [n]^t$ is a (μ, μ', γ)-averaging sampler;*

2. *$E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$ is a $((\delta - 3\tau) \cdot t, \epsilon_0)$-extractor.*

Then, $E : \{0,1\}^n \times \{0,1\}^{r_0+r} \to \{0,1\}^m$ defined by $E(x, (s_0, s)) = E_0(x_{S(s)}, s_0)$ is a $(\delta \cdot n, \epsilon_0 + \gamma + \exp(-\Omega(\tau n)))$-extractor. Furthermore, if E_0 is strong then so is E.

For any $\beta \in (0,1)$, setting $\tau = (1 - \beta)\delta/3$ and assuming that $\delta = \omega(\log(1/\epsilon)/n)$, we obtain the following simplified form.

Corollary 2.8 (Theorem 2.7, specialized): *For any $\beta \in (0,1)$ and $\delta = \omega(\log(1/\epsilon)/n)$, let $\mu' = \Theta(\delta/\log(1/\delta))$ and $\mu = \frac{1+2\beta}{3\beta} \cdot \mu = (1 + \Omega(1)) \cdot \mu'$. Suppose that the following two conditions hold.*

1. *$S : \{0,1\}^r \to [n]^t$ is a (μ, μ', ϵ)-averaging sampler;*

2. *$E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$ is a $(\beta\delta \cdot t, \epsilon)$-extractor.*

Then, $E : \{0,1\}^n \times \{0,1\}^{r_0+r} \to \{0,1\}^m$ defined by $E(x, (s_0, s)) = E_0(x_{S(s)}, s_0)$ is a $(\delta \cdot n, 3\epsilon)$-extractor. Alternatively, if $\delta = \Omega(\log(1/\gamma)/n)$ and S is a (μ, μ', γ)-averaging sampler, then E is a $(\delta \cdot n, \epsilon + \gamma^{\Omega(1)})$-extractor. Furthermore, if E_0 is strong then so is E.

Indeed, when using Corollary 2.8 one should artificially set the error parameter of both constructs in the hypothesis to be the maximum of their actual values. Note that *if both S and E_0 are computable in* (uniform) $\mathcal{AC}^0$*, then so is E.*

[4] The formulation in [54, Def. 6.1] is slightly different: Firstly, the current (μ, μ', γ)-averaging sampler corresponds to a $(\mu, \mu - \mu', \gamma)$-averaging sampler in [54, Def. 6.1]. Secondly, the "distinct element condition" is an integral part of Definition 2.6, whereas it is an additional feature in [54, Def. 6.1].

[5] See the proof of [27, Thm. 5.10], which is adapted in the proof of Claim A.2.

[6] The formulation in [54, Thm. 6.3] is slightly different: See Footnote 4.

3 Local extractors and extraction in $\mathcal{AC}^0$

In this section, we use local extractors and the ideas underlying their construction to obtain extractors computable in $\mathcal{AC}^0$. This approach was already used in proving Theorem 2.4, and we stated our intention to use it towards proving Theorems 1.2 and 3.1. Let us spell out how Theorem 2.4 was proved, in order to clarify the connection between local extraction and extraction in $\mathcal{AC}^0$.

Theorem 2.4 asserts an (explicit) extractor of logarithmic locality and logarithmic seed length. This means that for any possible seed, the residual extraction function depends only on logarithmically many bits in the source, which implies that these residual functions can be computed by depth-two circuits of polynomial size. Combining the polynomially many circuits that correspond to all possible fixing of the seed, we obtain the desired $\mathcal{AC}^0$-extractor.

Note that the foregoing argument can be turned around: It implies that if $\mathcal{AC}^0$ cannot compute extractors with logarithmic seed length for certain parameters, then such extractors cannot have logarithmic locality. We mention that Bogdanov and Guo [10] proved a logarithmic lower bound on the locality of extractors, which (unlike our lower bounds) applies also to the case of $k(n) = \Omega(n)$, but this does not rule out $\mathcal{AC}^0$-extractors (see Theorem 2.4).

Turning to Theorem 1.2 and its explicit version (captured by Theorem 3.1), recall that its proof is based on Vadhan's sample-then-extract technique (as stated in [54, Thm. 6.3] and restated in Corollary 2.8). The starting point is the proof of Theorem 2.5, which uses a sampler (of a logarithmically long seed) that samples poly-logarithmically many bits of the source. Unlike Vadhan, we cannot afford an arbitrary extractor here, so instead we use an extractor that can be computed by (explicit) constant-depth circuits of poly(n)-size (see Section 3.1 for details). In addition, we make the entire construction explicit by using a new explicit sampler (instead of the non-explicit sampler used originally in [54]). Thus, we obtain:

Theorem 3.1 (an explicit version of Theorem 1.2): *For every* $k(n) = n/\mathrm{poly}(\log n)$ *and* $\epsilon(n) = 1/\mathrm{poly}(n)$, *there exist* explicit $\mathcal{AC}^0$ *circuits that compute a* strong $(k(n), \epsilon(n))$*-extractor* $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Theta(\log n)}$*. Furthermore, the circuits have depth* $4 + \left\lceil \frac{\log(n/k(n))}{\log\log n} \right\rceil$.

The new explicit sampler, presented in Section 3.2, uses a seed of logarithmic length, and so it is trivially implemented by uniform $\mathcal{AC}^0$ circuits. As stated in Corollary 3.3, combining this sampler with Corollary 2.8, the construction of $\mathcal{AC}^0$-extractors for (n, k)-sources reduces to the construction of poly(n)-circuits of constant depth for extraction from $(t, \Omega(k/n) \cdot t)$-sources, where $t = \mathrm{poly}(n/k)$. Since $k = n/\mathrm{poly}(\log n)$, we can afford constant-depth circuits of size $\exp(t^c)$ for a sufficiently small constant $c > 0$.

The applications of the new sampler are spelled out in Section 3.3. Since the new extractor works only for $t = \widetilde{O}(n^{1/3})$ (or for constant error $\gamma > 0$), it does not suffice for the application in Section 6, and so we present (in Section 3.4) an alternative sampler that works essentially for any t and any error $\gamma = 1/\mathrm{poly}(n)$, which suffices for the latter application. The alternative version uses a seed of polylogarithmic length, which forces us to detail its implementation in $\mathcal{AC}^0$.

3.1 Proving Theorem 1.2

As a warm-up, we prove Theorem 1.2, which asserts that *for every* $k(n) = n/\mathrm{poly}(\log n)$ *and* $\epsilon(n) = 1/\mathrm{poly}(n)$*, there exist* $\mathcal{AC}^0$ *circuits that compute a strong* $(k(n), \epsilon(n))$*-extractor* $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{O(\log n)}$. (Furthermore, the circuits have depth $4 + \frac{\log(n/k(n))}{\log\log n}$.)

Proof Sketch: We slightly modify the proof of [54, Thm. 7.4], which combines an averaging sampler with an extractor (using [54, Thm. 6.3] (see Corollary 2.8)), while setting $\delta = k(n)/n = 1/\text{poly}(\log n)$, $m = \Theta(\log n)$ and $t = O(m/\delta)$. Specifically, instead of using the (optimal) non-explicit extractor asserted in [54, Lem. 7.1], which may not be computable in $\mathcal{AC}^0$, we shall use the strong (k_0, ϵ)-extractor $E_0 : \{0,1\}^t \times \{0,1\}^{O(\log n)} \to \{0,1\}^{m(n)}$ asserted in [28, Sec. 5], where $k_0 = O(m + \log(1/\epsilon))$. Recall that this extractor uses a seed of length $O(m + \log(t/\epsilon)) = O(\log n)$. (At this point we use the same non-explicit averaging sampler as in the original proof; that is, the sampler of [54, Lem. 7.2].)

The key observation is that E_0 can be computed by (uniform) poly(n)-size circuits of depth $2 + \log_m t = 3 + \frac{\log(n/k(n))}{\log m}$, since the residual computation (resulting when fixing the seed) amounts to a linear combination (over $\text{GF}(2^{O(\log t)})$) of $O(\log t)$-bit long blocks of the t-bit source viewed as a sequence of $t/O(\log t)$ elements over a field of size poly(t). The sampler itself uses a logarithmically long seed, and thus can be computed by depth two circuits of polynomial size. ■

Towards the explicit version. The foregoing proof, which follows the proof of [54, Lem. 7.1], combines an adequate sampler with an adequate extractor using the composition theorem of [54, Thm. 6.3]. Recall that we replaced the extractor used in the original proof by an alternative extractor, which happens to be explicit. In order to obtain an explicit version of the foregoing construction (i.e., prove an explicit version of Theorem 1.2), we also need to replace the non-explicit sampler used in the original proof. This will be done in Section 3.3, after designing an explicit sampler (in Section 3.2).

3.2 A new averaging sampler

As stated in Section 3.1, a central ingredient in our constructions is a new sampler that uses a seed of logarithmic length regardless of the density (i.e., μ) of the functions that it semi-approximates, where by "semi-approximates" we refer to the relaxed notion of an averaging sampler (as reviewed in Definition 2.6).

Theorem 3.2 (an averaging sampler with logarithmic seed length): *Let $\alpha \in (0,1)$ be an arbitrary constant. Then, for every $\mu > n^{-1/3}/\text{poly}(\log n)$ and $\gamma \geq 1/\text{poly}(n)$, there exists an explicit $(\mu, \alpha\mu, \gamma)$-averaging sampler $S : \{0,1\}^{O(\log n)} \to [n]^t$ for any $t \in [\Theta(\mu^{-1}\log(n/\gamma)), \widetilde{O}(n^{1/3})]$. An analogous result holds for any $\mu > 1/n$, $t = \Omega(1/\mu)$, and any constant $\gamma > 0$.*

That is, the main claim requires $\mu > n^{-1/3}$ and allows $\gamma \geq 1/\text{poly}(n)$, whereas the alternative allows any $\mu > 0$ (equiv., $\mu > 1/n$) but provide only for constant $\gamma > 0$. (The constant 1/3, in the exponent, can be replaced by any constant smaller than 1/2.)

Proof: We first establish the result for a constant γ, and then extend it to arbitrary $\gamma \geq 1/\text{poly}(n)$. Our starting point is the observation that the pairwise-independence generator yields a $(\mu, \alpha\mu, \gamma)$-averaging sampler with $t = \Theta(1/\gamma\mu)$ samples. Basically, the behavior of this generator with respect to the relaxed notion of averaging samplers (as in Definition 2.6) is similar to its behavior with respect to the notion of hitters (i.e., the sample size is linearly related to $O(1/\mu)$).

Claim 3.2.1 (the pairwise independence generator as an averaging sampler): *Let F be a finite field, $t < |F|$, and $\phi_1, ..., \phi_t$ be t distinct non-zero elements of F. Consider $G : F^2 \to F^t$ such that,* *for every $r, s \in F$,*

$$G(r,s) = (r+\phi_1 s, r+\phi_2 s, ..., r+\phi_t s).$$

Then, for any constants $\alpha, \beta \in (0,1)$, selecting uniformly and independently $r \in F$ and $s \in F \setminus \{0\}$, and outputting $G(r,s)$ yields a $(\mu, \alpha\mu, \beta)$-averaging sampler for $\mu = \Omega(1/t)$.

We shall also use the fact that each of the elements in the sample is uniformly distributed in F. An alternative construction that enjoys all the above features is the expander neighborhood generator (see, e.g., [27, Apdx. C.2]), when instantiated with Ramanujan graphs (see [41]).

Proof: The proof is by a straightforward adaptation of the proof of the hitting property (see, e.g., [27, Apdx. C.2]). Specifically, for r and s selected uniformly in F, and any function $f : F \to [0,1]$ such that $\rho(f) \geq \mu$, denoting by ζ_j the value of $f(r+\phi_j s)$, we have

$$\begin{aligned}\mathbf{Pr}\left[\sum_{j=1}^{t}\zeta_j < \alpha\cdot\rho(f)\cdot t\right] &\leq \mathbf{Pr}\left[\left|t\cdot\rho(f) - \sum_{j=1}^{t}\zeta_j\right| > (1-\alpha)\cdot t\cdot\rho(f)\right] \\ &\leq \frac{t\cdot(1-\rho(f))\cdot\rho(f)}{((1-\alpha)\cdot t\cdot\rho(f))^2} \\ &< \frac{1}{(1-\alpha)^2\cdot\mu\cdot t}\end{aligned}$$

where the second inequality uses Chebyshev's Inequality. Avoiding the choice of $s = 0$ guarantees that the t elements in $G(r,s)$ are indeed disjoint, while skewing the probability by at most $1/|F| = o(1)$. ■

Establishing the alternative claim (of the theorem) and moving on. Associating $[n]$ with the non-zero elements of a finite field (of size approximately n)[7] and using any $t \geq O(1/\mu)$, we obtain an $(\mu, \alpha\mu, \beta)$-averaging sampler that uses a seed of length $2\log n$ (as in the alternative claim).

In order to reduce the error probability to $\gamma = 1/\text{poly}(n)$, we shall use a random walk of length $\ell = O(\log(n/\gamma))$ on a constant-degree expander over the vertex set $[n^2]$ to generate seeds for the generator G of Claim 3.2.1. Actually, in order to guarantee that these seeds do not generate samples that intersect, we will shift these seeds using a $O(1)$-wise independent sequence over $[n]^2$, and discard a small part of the resulting sample. The point is that this combination yields a sequence that maintains the sampling features of the expander random walk and the property that each $O(1)$ elements of the sequence are independently and uniformly distributed in the set. (At this point, we shall discard the few repetitions.)

Claim 3.2.2 (a randomized version of the "random walk on an expander" hitter): *For every constants $\epsilon > 0$ and $c \in \mathbb{N}$ and a varying m, let $\ell = O(\log m)$ be sufficiently large, and consider the following generator, denoted G', that uses a seed of length $O(\log m)$.*

1. *The generator uses the first part of the seed to generate a random walk of length ℓ on a $\text{poly}(1/\epsilon)$-regular expander with vertex set $\mathbb{Z}_m$. Denote the sequence of visited vertices by $(v_1, ..., v_\ell)$.*

2. *The generator uses the second part of the seed to generate a $2c$-wise independent sequence of ℓ elements in $\mathbb{Z}_m$, denoted $(s_1, ..., s_\ell)$.*

[7] The discrepancy between the field size and n can be ignored (e.g., by using a slightly bigger field and mapping elements out of $[n]$ is some standard manner).

3. *For every $i \in [\ell]$, the generator computes $v'_i \leftarrow v_i + s_i \bmod m$.*

4. *The generator outputs the sequence of the first $\ell' = \ell - c$ distinct elements in the sequence $(v'_1, ..., v'_\ell)$, and outputs $(1, .., \ell')$ if $|\{v'_i : i \in [\ell]\}| < \ell - c$.*

Then, the foregoing generator is a $(1-\epsilon, 1-O(\sqrt{\epsilon}), \exp(-\Omega(\ell))$-averaging sampler.

In other words, for every constants $\epsilon > 0$ and $c \in \mathbb{N}$, the generator $G' : \{0,1\}^{O(\log m)} \to (\mathbb{Z}_m)^{O(\log m)}$ is a $(1-\epsilon, 1-O(\sqrt{\epsilon}), m^{-c})$-averaging sampler.

Proof: We first upper bound the probability that $|\{v'_i : i \in [\ell]\}| < \ell - c$. Fixing any sequence of v_i's (as selected in Step 1), we consider the probability space generated by Step 3. For the bad event to occur, there must be a set of $2c$ indices, denoted I, such that the set $S_I \stackrel{\rm def}{=} \{v_i + s_i \bmod m : i \in I\}$ has cardinality smaller than c. Since for every $2c$-subset I it holds that $\mathbf{Pr}[|S_I| < c] < \binom{m}{c} \cdot (c/m)^{2c} < (c^2/m)^c$, the bad event occurs with probability at most $\binom{\ell}{2c} \cdot (c^2/m)^c < (c^2\ell^2/m)^c < 2^{-\Omega(\ell)}$, provided that $\ell / \log m$ is a sufficiently large constant.

We now analyze the sampling property of the sequence of all v'_i's, while noting that any subsequence of length ℓ' will do almost as well (since the omission causes a loss of at most $c = o(\sqrt{\epsilon}\ell)$ units). Fixing an arbitrary function $f : \mathbb{Z}_m \to [0,1]$ such that $\rho(f) \geq 1 - \epsilon$, we now fix an arbitrary sequence $(s_1, ..., s_\ell)$ as selected in Step 2. This selection induces ℓ auxiliary functions $f_1, ..., f_\ell$ such that $f_i(x) = f(x + s_i \bmod m)$, since under this definition $f(v'_i) = f_i(v_i)$. Setting $\epsilon' = 6\sqrt{\epsilon}$, we shall prove that

$$\mathbf{Pr}\left[\sum_{i \in [\ell]} f_i(v_i) < \ell \cdot (1 - \epsilon')\right] = \exp(-\Omega(\ell)), \tag{2}$$

where $(v_1, ..., v_\ell)$ is generated as in Step 1. We recall the *general* Expander Random Walk Theorem (see [27, Thm. A.4]), which asserts that for Boolean functions $b_i : [m] \to \{0,1\}$ it holds that $\mathbf{Pr}[\sum_{i \in [\ell]} b_i(v_i) = \ell] < \prod_{i \in [\ell]} \min(1, \rho(b_i) + \epsilon)^{1/2}$, where ϵ upper bounds the (square of) the spectral gap of the expander (i.e., the expander was chosen so that this upper bound holds). Hence, Eq. (2) follows for *Boolean* f_i's by considering all $\epsilon'\ell$-subsets $I \subset [\ell]$ and setting $b_i = 1 - f_i$ if $i \in I$ and $b_i = 1$ otherwise. Specifically, the probability that $\sum_{i \in [\ell]} f_i(v_i) < (1 - \epsilon') \cdot \ell$ is upper bounded by $\binom{\ell}{\epsilon'\ell} \cdot (2\epsilon)^{\epsilon'\ell/2}$, since this event can occur only if there exists a $\epsilon'\ell$-subset I such that for every $i \in I$ it holds that $f_i(v_i) = 0$ (whereas $\rho(f_i) \geq 1 - \epsilon$).[8] Now, using

$$\begin{aligned} \binom{\ell}{\epsilon'\ell} \cdot (2\epsilon)^{\epsilon'\ell/2} &< (3\ell/\epsilon'\ell)^{\epsilon'\ell} \cdot (2\epsilon)^{\epsilon'\ell/2} \\ &= (1/2)^{\epsilon'\ell/2}, \end{aligned}$$

where the equality uses $\epsilon' = 6\sqrt{\epsilon}$, and the claim follows (for Boolean f).

Observing that any averaging sampler for Boolean functions also works for general functions (see Claim A.2), the claim is established. In this case, we can establish that the generator is a $(1 - 0.5\epsilon, 1 - O(\sqrt{\epsilon}), \exp(-\Omega(\ell))$-averaging sampler. An alternative argument for the current setting may proceed by defining auxiliary Boolean functions f'_i such that $f'_i(x) = 1$ if and only

[8] Note that, for this I, we have $\rho(b_i) \leq \epsilon$ for $i \in I$ and $b_i \equiv 1$ otherwise. Hence, $\sum_{i \in I} f_i(v_i) = 0$ if and only if $\sum_{i \in I} b_i(v_i) = |I|$, which holds if and only if $\sum_{i \in [\ell]} b_i(v_i) = \ell$, whereas the probability of the latter event is upper bounded by $\prod_{i \in [\ell]} \min(1, \rho(b_i) + \epsilon)^{1/2} \leq (2\epsilon)^{|I|/2}$.

if $f_i(x) \geq 1 - \epsilon^{1/3}$, noting that $\rho(f_i') \geq 1 - \epsilon^{2/3}$, and using $\epsilon' = O(\epsilon^{1/3})$. (This alternative only establishes that the generator is a $(1-\epsilon, 1-O(\epsilon^{1/3}), \exp(-\Omega(\ell))$-averaging sampler, which is good enough for our application.) ■

Establishing the main claim of the theorem. The theorem follows by combining the samplers asserted in Claims 3.2.1 and 3.2.2. Specifically, we use G' of Claim 3.2.2 to generate a sequence of ℓ' seeds for the generator G of Claim 3.2.1 (i.e., we set $m = n^2$ and associate $\mathbb{Z}_m$ with $(F \setminus \{0\})^2 \equiv [n]^2$). Actually, we might as well use the sequence of all ℓ seeds defined in Step 3 of G' (since we are going to make our own omissions anyhow). We set the constant parameter β in Claim 3.2.1 so to fit ϵ in Claim 3.2.2 (i.e., $\beta = \epsilon$), while picking the constant parameter $\alpha < 1$ as large as we wish (and ditto w.r.t $t \in [\Theta(\mu^{-1}\log(n/\gamma)), \widetilde{O}(n^{1/3})]$). That is, the combined generator maps its seed $s \in \{0,1\}^{O(\log n)}$ to the sequence of sets $G(s_1), ..., G(s_\ell)$, where $(s_1, ..., s_\ell) \leftarrow G'(s)$.

The combined generator satisfies the average sampling property of Eq. (1); that is, for every $f : [n] \to [0,1]$ such that $\rho(f) \geq \mu$, with probability at least $1-\gamma$, the $\ell \cdot t$-sized sample, denoted I, satisfies $(1/|I|) \cdot \sum_{i \in I} f(i) \geq (1 - O(\sqrt{\epsilon})) \cdot \alpha\mu$. This is the case since at least $1 - O(\sqrt{\epsilon})$ fraction of the seeds generated by G' produce samples that have an f-average of at least $\alpha\mu$. (Indeed, the one-sided flavor of Eq. (1) allows to discard the exceptional seeds.) Since $\epsilon > 0$ and $1 - \alpha > 0$ can be made arbitrarily small constants, we can have $(1 - O(\sqrt{\epsilon})) \cdot \alpha$ be arbitrarily close to one. We note that the sampling guarantee remains valid also if we omit any $o(\ell)$ of the ℓ samples (produced by the ℓ seeds generated by G').

The question is whether the sample I contains no repetitions. Recall that G generates samples that have no repetitions, and that the t elements in each sample are each uniformly distributed in $[n]$. For any desired c, the seeds generated by G' are $2c$-wise independent. Now, consider a random graph R with vertices corresponding to the ℓ seeds and edges connecting a pair of seeds if and only if the corresponding samples intersect. We claim that, with probability at least $1 - (\ell^2 t^2/n)^c$, this graph has an independent set of size $\ell - 2c$.

To prove the last claim note that if the graph has no vertex cover of size $2c$, then it must have a matching of size greater than c. We shall show that even having a matching of size c is unlikely. Denoting the number of possible matching of size c in an ℓ-vertex graph by $M < \binom{\ell}{2c} \cdot (2c!) < \ell^{2c}$, note that the probability that the graph R has a matching of size c is upper bounded by $M \cdot (t^2/n)^c < (\ell^2 t^2/n)^c$, because we consider $M \cdot (t^2)^c$ events, and each event occurs with probability $(1/n)^c$. Specifically, each event corresponds to a choice of c disjoint pairs of seeds and a choice of a pair of samples per each pair of seeds, and this sequence of $2c$ samples is uniformly distributed in $[n]^{2c}$. (Here we use the fact that the seeds generated by G' are uniformly distributed in an $2c$-wise independent manner, and that each sample generated by G is uniformly distributed in $[n]$.)

Recalling that $t = \widetilde{O}(n^{1/3})$ and $\ell = O(\log n)$ (and $\gamma = 1/\text{poly}(n)$), and that c is a constant chosen at our discretion, we conclude that with probability $1-\gamma$ the graph contains an independent set of size $\ell - 2c$, which means that the ℓ seeds generate at least $\ell - 2c$ disjoint samples. The final sampler outputs the union of $\ell - 2c$ disjoint samples, and an arbitrary sequence of $(\ell - 2c) \cdot t$ elements otherwise (i.e., if such a collection of disjoint samples does not exist). By the foregoing analysis this final sampler also satisfies the average sampling property of Eq. (1). ■

3.3 Applications to explicit constructions of extractors

We first describe the implication of the new averaging sampler on randomness extraction in $\mathcal{AC}^0$ (from (n,k)-sources). We start by spelling out the construction obtained by instantiating Corollary 2.8 with the averaging sampler of Theorem 3.2.

Corollary 3.3 (using the averaging sampler of Theorem 3.2): *Let $\beta \in (0,1)$ be a constant, $\delta > n^{-1/3}$ and $t \in [\Theta(\delta^{-1}\log^2 n), \widetilde{O}(n^{1/3})]$. Suppose that $E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$ is a $(\beta\delta \cdot t, \epsilon)$-extractor that is computable by* (uniform) *constant-depth circuits of* poly(n)*-size. Then, there exists a $(\delta \cdot n, 3 \cdot \max(\epsilon, \mathrm{poly}(1/n)))$-extractor $E : \{0,1\}^n \times \{0,1\}^{r_0+O(\log n)} \to \{0,1\}^m$ that is computable by* (uniform) *$\mathcal{AC}^0$ circuits. Furthermore, the depth of the circuits for E is only one unit more than the depth of the circuits for E_0, and if E_0 is strong then so is E.*

(The constant 1/3 can be replaced by any constant smaller than 1/2.)

Proof: Towards invoking Corollary 2.8, we set $\mu' = \beta\delta/O(\log(1/\delta))$ and $\mu = \frac{1+2\beta}{3\beta} \cdot \mu'$. We use the $(\mu, \mu', 1/\mathrm{poly}(n))$-averaging sampler $S : \{0,1\}^{O(\log n)} \to [n]^t$ of Theorem 3.2, while noting that $\mu'/\mu = 3\beta/(1+2\beta)$ is a constant smaller than 1, whereas $\mu > n^{-1/3}/\mathrm{poly}(\log n)$ and $t = \Omega(\delta^{-1}\log^2 n) = \Omega(\mu^{-1}\log n)$. Invoking Corollary 2.8, while noting that $\delta = \omega((\log n)/n)$ and that S is computable by uniform DNFs (resp., CNFs) of poly(n)-size, we obtain the desired extractor. ■

Next, combining Corollary 3.3 with known extractors, which can be computed by constant-depth circuits of poly(n)-size, we derive the following –

Corollary 3.4 (extraction in $\mathcal{AC}^0$ with seed of logarithmic length):

1. (Theorem 3.1, restated): *For every $k(n) \geq n/\mathrm{poly}(\log n)$ and $\epsilon(n) = 1/\mathrm{poly}(n)$, there exist explicit $\mathcal{AC}^0$ circuits that compute a* strong *$(k(n), \epsilon(n))$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Theta(\log n)}$. Furthermore, the circuits have depth $4 + \left\lceil \frac{\log(n/k(n))}{\log\log n} \right\rceil$.*
2. *For every $k(n) \geq n/\mathrm{poly}(\log n)$, $m(n) = \mathrm{poly}(\log n)$, and $\epsilon(n) = 1/\mathrm{poly}(\log n)$, there exist explicit $\mathcal{AC}^0$ circuits that compute a* strong *$(k(n), \epsilon(n))$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{m(n)}$. Furthermore, the circuits have depth $3 + \left\lceil \frac{\log(m(n) \cdot n/k(n))}{\log\log n} \right\rceil$.*

The extractor of Part 2 has longer output (i.e., $m(n) = \mathrm{poly}(\log n)$ rather than $m(n) = O(\log n)$), but weaker quality than the extractor of Part 1 (i.e., $\epsilon(n) = 1/\mathrm{poly}(\log n)$ rather than $\epsilon(n) = 1/\mathrm{poly}(n)$). We cannot afford the error-reduction procedures of [48], since these procedures seem sequential in nature. But it may be possible to implement the extractor of [30] (or another adequate extractor) by constant-depth circuits of sub-exponential size; in general, for a parameter ℓ (to be set to $O(\log n)$) and any $\delta > 1/\mathrm{poly}(\ell)$ and $t = \mathrm{poly}(\ell)$, we merely need $\exp(-\Omega(\ell))$-error extractors for $(\delta t, t)$-sources that are implementable by constant-depth circuits of $\exp(\ell)$-size and extract $(\delta t)^{\Omega(1)}$ bits.

Proof: Starting with Corollary 3.3, the two parts are proved by providing corresponding extractors. Specifically, we set $\beta = 1/2$, $\delta = k(n)/n$ and $t = \Theta((\log n)^c \cdot \delta^{-1})$, for a constant $c \geq 1$ selected below, and pick an adequate extractor $E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$.

For Part 1 we set $c = 1$ and use the strong $(\delta t/2, \epsilon)$-extractor $E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$ asserted in [28, Sec. 5], which supports $r_0 = O(m)$ for $m = \Theta(\delta t) - \log(1/\epsilon)$. Using $\epsilon = 1/\text{poly}(n)$ and $t = \Omega(\delta^{-1} \log n)$, we have $m = \Theta(\log n)$, and all claims follow, since (as detailed next) E_0 can be computed by (uniform) poly(n)-size circuits of depth $2 + \lceil \log_m t \rceil = 3 + \lceil (\log(n/k(n)))/\log m \rceil$.

To see that E_0 can be computed by (uniform) poly(n)-size circuits of depth $2 + \lceil \log_m t \rceil$, we consider all poly$(n)$ possible fixings of its seed, and observe that E_0 can be computed by a depth-two circuit that selects the appropriate residual circuit. As noted in Section 3.1, viewing the t-bit source as a sequence of $t/O(\log t)$ elements over a field F of size poly(t), the residual computation (for each fixing of the seed to E_0) amounts to a linear combination (over F) of these field elements. Such linear combinations can be computed by depth $1 + \lceil \log_m t \rceil$ circuits of poly(n)-size.

Turning to Part 2, we set $c = \frac{\log m}{\log \log n} + 1$ (i.e., $m = (\log n)^{c-1}$) and use the strong $(\delta t/2, \epsilon)$-extractor $E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$ asserted by Trevisan [53], which supports $r_0 = O(\log(t/\epsilon))$ and $m = (\delta t/2)^{(c-1)/c}$, provided that $\delta t/2 > t^{\Omega(1)}$ and $\epsilon = 1/\text{poly}(t)$. Recalling that the residual computations corresponding to Trevisan's extractor amounts to a linear combination of the t bits, we conclude that these linear combinations can be computed by depth $1 + \lceil \log_{\log n} t \rceil = 1 + \lceil c + \log_{\log n}(n/k(n)) \rceil$ circuits of poly(n)-size. ■

Application to explicit constructions of local extractors. Combining Theorem 3.2 with [54, Thm. 6.3] (see Corollary 2.8), yields improved parameters for explicit local extractors. Further improvement is obtained by using better extractors (i.e., the current state-of-art extractors of Guruswami *et al.* [30, Thm. 1.5]) than those available to Vadhan [54].[9]

Corollary 3.5 (implications to locally computable extractors): *For every constant $\alpha \in (0,1)$, $k = \Omega(\text{poly}(\log n))$ and $\epsilon = 1/\text{poly}(n)$, and every $t \geq \min(\Theta((n/k)\log^2 n), n)$, there exists an explicit construction of a t-local (k, ϵ)-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\alpha \cdot (k/n) \cdot t}$.*

Proof Sketch: Assume that $t < n$, since otherwise the claim is trivial (using $t = n$ and [30, Thm. 1.5]). Now, setting $\delta = k(n)/n$, $\mu = \delta/O(\log(1/\delta))$ and $t \geq \Theta(\mu^{-1} \log n)$, combine the $(\mu, \alpha\mu, 0.3\epsilon)$-averaging sampler of Theorem 3.2 (which maps $\{0,1\}^{O(\log n)}$ to $[n]^t$) with the $(\alpha\delta t, 0.3\epsilon)$-extractor of Guruswami *et al.* [30, Thm. 1.5] (set to map $\{0,1\}^t \times \{0,1\}^{O(\log(t/\epsilon))}$ to $\{0,1\}^{\alpha^2 \delta t}$), where the combination is via Corollary 2.8. This yields extraction of $\alpha^2 \delta t$ bits. ■

3.4 An alternative averaging sampler

The drawback of the averaging sampler presented in Theorem 3.2 is that it does not apply to the case that both $\mu < n^{-1/2}$ and $\gamma < 1/n^2$ (or so). This was good enough for the applications to extraction in $\mathcal{AC}^0$ presented in Section 3.3, but is not sufficient for other applications (see, e.g., Section 6, and specifically Corollary 6.2).

Recall that the reason for the lower bound on μ is that we needed $t^2 > \mu^{-2}$ to be (significantly) smaller than n, so that we may expect two random t-subsets of $[n]$ to be disjoint. The latter condition was used for establishing the claim that the sampler always generate distinct elements (as required in the furthermore clause of Definition 2.6). We now achieve the latter goal while

[9] The latter improvement is in the output length and in some cases in the deviation parameter ϵ. Specifically, using only extractors that were available to Vadhan (i.e., [53, 48]), one can extract $\Omega((k/n) \cdot t)^{0.999}$ bits (rather than $\Omega((k/n) \cdot t)$ bits) for $\epsilon = \max(\text{poly}(1/n), \exp(-t^{1-o(1)}))$.

using a seed of length $O(\log n)^2$, rather than $O(\log n)$, which suffices for our application (i.e., Corollary 6.2).

Theorem 3.6 (an averaging sampler for all densities): *Let $\alpha \in (0,1)$ be an arbitrary constant. Then, for every $\mu > 1/n$ and $\gamma \geq 1/\text{poly}(n)$, there exist explicit constant-depth* poly(n)*-size circuits $C_n : \{0,1\}^{O(\log n)^2} \to [n]^t$ that compute a $(\mu(n), \alpha\mu(n), \gamma(n))$-averaging sampler for any $t \in [\Theta(\mu^{-1}\log^2 n), 0.1n]$.*

Proof Sketch: Our starting point is the simple (pairwise independence) averaging sampler G asserted in Claim 3.2.1. Setting $\ell = O(\log n)$, we consider ℓ independent invocations of G and the sequence of $\ell \cdot t$ elements generated. Denoting this sequence (or multi-set) by S, note that each element occurs in S at most ℓ times, since the sets generated by G have no repetitions. Denoting the number of occurrences of i (in the multi-set S) by q_i, we use an ℓ-wise independent sequence of length n over $[\ell]$ in order to select each $i \in [n]$ with probability q_i/ℓ, resulting in a set S' (with no repetitions).

Using an ℓ^{th} moment inequality (see Appendix A.2) it can be shown that the probability that the number of sampled elements is $(1 \pm \epsilon) \cdot t$ is greater than $1 - (\epsilon^{-2}\ell^2/t)^{\ell/2}$, since the expected number is $\ell \cdot t/\ell$ (whereas $t \geq 2\epsilon^{-2}\ell^2 = O(\log^2 n)$, since we refer to any constant $\epsilon > 0$). Fixing any $f : [n] \to [0,1]$ such that $\rho(f) \geq \mu$, and assuming that $\sum_{i\in S} f(i) \geq \alpha\mu \cdot \ell t$ (which occurs with probability at least $1-\gamma$ (cf. the proof of Theorem 3.2)), we apply the same reasoning to the f-value of the random set S'. Specifically, since $t = \Omega(\mu^{-1}\log^2 n)$, it follows that for any constant $\epsilon > 0$ the probability that $\sum_{i\in S'} f(i) < (1-\epsilon)\cdot\alpha\mu\cdot t$, is at most $(\epsilon^{-2}\ell^2/\alpha\mu t)^{\ell/2} < \gamma$ (where here we use $t \geq 2\epsilon^{-2}\alpha^{-1}\mu^{-1}\ell^2 = O(\mu^{-1}\log^2 n)$). Finally, we augment the set S' so to obtain a set of size exactly $(1+2\epsilon)\cdot t$.

It is left to show that the foregoing sampler can be implemented by uniform constant-depth circuits of poly(n)-size. This is not straightforward. Hence, the foregoing description is merely an intuitive motivation and the actual implementation proceeds as follows.

1. Obtaining the multi-set S (by using ℓ invocations of G), and generating an ℓ-wise independent sequence $(p_1, ..., p_n) \in [\ell]^n$.

   The implementation of this step by uniform constant-depth circuits of size poly(n) is straightforward for G and quite standard for the ℓ-wise independent sequence. The construction of an ℓ-wise independent sequence over $[\ell]$ is actually performed over GF$(2^{\log n})$, and involves taking various linear combinations of ℓ field elements, which constitute the seed of the ℓ-wise independence generator.

2. Computing the number of occurrences of each element (in S) and producing the set S'. Specifically, the element $e \in [n]$ in included in S' if and only if it occurs in S at least p_e times (i.e., $q_e \geq p_e$), where $(p_1, ..., p_n)$ is the sequence produced in Step 1.

   This can be done by uniform constant-depth circuits of size poly(n) because the number of occurrences of element $e \in [n]$ in S equals the cardinality of the maximal set $I \subseteq [\ell]$ such that each index in I corresponds to an invocation of G that produced a sample that contains e. Specifically, the condition to check for each set I is that *for every $i \in I$ there exists $j \in [t]$ such that $r_i + \phi_j s_i = e$, where (r_i, s_i) is the seed used in the i^{th} invocation of G.* (Alternatively, we can count the number of occurrences of each $e \in [n]$ in S by using a counter for small values, as in Step 3(b).) Indeed, we obtain a representation of S as an n-long sequence over

$\{0, 1, ..., \ell\}$, and produce (or sieve out) the set S' using the same representation, which means that S' is represented as an n-bit long sequence.

The above actions can be performed by uniform constant-depth circuits of size $2^\ell \cdot \text{poly}(n) = \text{poly}(n)$. Note, however, that the set S' is not in the standard format that we use throughout this paper, which is a format that seems essential for having $\mathcal{AC}^0$ circuits compute a mapping of the form $(x, S') \mapsto x_{S'}$. The next steps are aimed at putting S' in the standard format.

3. Making progress towards obtaining a standard representation of S' (as a sequence of elements of $[n]$) is done as follows.

   (a) First, we select an ℓ-wise independent hash function $h : [n] \to [t/\ell^3]$ and hash $S' \subset [n]$ to $[t/\ell^3]$ such that, with probability at least $1-\gamma$, each image $i \in [t/\ell^3]$ is assigned $(1\pm2\epsilon)\cdot\ell^3$ elements of S'. (The probabilistic claim holds by virtue of the ℓ-wise independent hash function we use, while noting that $(1 - e) \cdot \ell^3 > 2\epsilon^{-2} \cdot \ell^2$.)

   (b) Next, we rank the elements of S' that are hashed by h to each image $i \in [t/\ell^3]$. This is done by using a (uniform $\mathcal{AC}^0$) counter for small values (i.e., a uniform $\mathcal{AC}^0$ circuits that counts the number of ones a string of length n, when guaranteed that this number does not exceed $\text{poly}(\log n)$).[10] An explicit construction of such a counter was presented in [47], improving over [3, Sec. 5] (and subsequent works).[11] Here, we count for each $e \in (S' \cap h^{-1}(i))$ the number of elements in S' that are smaller than e and are hashed by h to $h(e)$ (i.e., we compute $\sum_{e'\in[n]} \chi_{e'}$, where $\chi_{e'} = 1$ if $e' \in S'$ and $h(e') = h(e)$).

   At the end of the current step we obtain a representation of S' as a sequence of t/ℓ^3 (sorted) sets that are each of size $(1 \pm 2\epsilon) \cdot \ell^3$, where the latter claim holds with probability $1 - 2\gamma$. This is closed to the desired format, but is not quite there.

   Note that the current step can be performed by uniform constant-depth circuits of size $\text{poly}(n)$, where the key observation is that such circuits can rank $\text{poly}(\log n)$ many elements that reside in an array of length n. (The implementation of that ℓ-wise independent hash functions from $[n]$ to $[n/\ell^3]$ is similar to the implementation of ℓ-wise independent sequences over $[\ell]$ discussed in Step 1.)

4. Obtaining a standard representation of the augmented set S' (as a sequence of elements of $[n]$) is done as follows. Indeed, obtaining the standard representation is linked to augmenting the set S' to a set of size exactly $(1 + 2\epsilon) \cdot t$.

   Using an ℓ-wise independent sequence of length t over $[n]$, we first select t elements of $[n]$, while noting that (with probability at least $1 - \gamma$) we obtain at least $t/2$ distinct elements that are not in S'. (Here we use $t \leq 0.1n$.) Furthermore, the number of additional elements mapped to each image of the (ℓ-wise independent) hash function h (used in Step 3) is at most $(1 + \epsilon) \cdot \ell^3$ and the number of additional elements not in S' that are mapped to this image under h is at least $\ell^3/2$.

   Denoting the multi-set of additional elements by A and fixing a good hash function h, note that for every $i \in [t/\ell^3]$ it holds that (i) $|S' \cap h^{-1}(i)| = (1\pm2\epsilon)\cdot\ell^3$, (ii) $|A \cap h^{-1}(i)| \leq (1+\epsilon)\cdot\ell^3$, and (iii) $|(A \setminus S') \cap h^{-1}(i)| \geq \ell^3/2$. Now, we rank the $O(\ell^3)$ elements of A that are mapped

[10] These circuit can also detect the case that the guarantee is violated.

[11] An alternative construction is presented in Appendix A.3.

by h to each image $i \in [t/\ell^3]$, just as we did with S' in Step 3, and use the two rankings in order to obtain a list of $(1+2\epsilon) \cdot \ell^3$ elements that are mapped (by h) to i such that this list contains all elements in $S' \cap h^{-1}(i)$. (Specifically, we place the $j^{\rm th}$ elements of $S' \cap h^{-1}(i)$ in position j, and the $j^{\rm th}$ element of $A \cap h^{-1}(i)$ in position $|S' \cap h^{-1}(i)| + j \leq (1+2\epsilon)\ell^3$.)[12] At this point we have the desired format.

If any of these steps failed, which happened with probability $O(\gamma)$, then we just output a fixed sequence. ■

Corollary 3.7 (using the averaging sampler of Theorem 3.6): *Let $\beta \in (0,1)$ be a constant, $\delta = \Omega(\log n)/n$ and $t \in [\Theta(\delta^{-1}\log^3 n), 0.1n]$. Suppose that $E_0 : \{0,1\}^t \times \{0,1\}^{r_0} \to \{0,1\}^m$ is a $(\beta\delta \cdot t, \epsilon)$-extractor that is computable by* (uniform) *constant-depth circuits of* poly(n)*-size. Then, there exists a $(\delta \cdot n, \epsilon + \text{poly}(1/n))$-extractor $E : \{0,1\}^n \times \{0,1\}^{r_0 + O(\log n)^2} \to \{0,1\}^m$ that is computable by* (uniform) *$\mathcal{AC}^0$ circuits.*

Proof Sketch: As in the proof of Corollary 3.3, towards invoking Corollary 2.8, we set $\mu' = \beta\delta/O(\log(1/\delta))$ and $\mu = \frac{1+2\beta}{3\beta} \cdot \mu'$. But here we use the $(\mu, \mu', 1/\text{poly}(n))$-averaging sampler $S : \{0,1\}^{O(\log n)^2} \to [n]^t$ of Theorem 3.6, while noting that $\delta = \Omega((\log n)/n)$ and $t = \Omega(\mu^{-1}\log^2 n)$. Invoking the alternative part of Corollary 2.8, while noting that $\delta = \Omega((\log n)/n)$, the claim follows. ■

4 Extraction from Block Sources

We consider block sources as defined by Chor and Goldreich [16], and not their generalization as used by Zuckerman [60, 61] (see also [45]) and subsequent works.

Definition 4.1 (block sources [16]):[13] *An $(n, (\ell, b))$-*block source *is a sequence of* (possibly dependent) *random variables $X = (X_1, ..., X_n) \in \{0,1\}^{n\ell}$ such that for every $i \in [n]$ and $x_1, ..., x_i \in \{0,1\}^\ell$ it holds that*

$$\mathbf{Pr}[X_i = x_i | X_1 \circ \cdots \circ X_{i-1} = x_1 \circ \cdots \circ x_{i-1}] \leq 2^{-b}$$

It follows that $\mathbf{Pr}[X_1 \circ \cdots \circ X_n = x_1 \circ \cdots \circ x_n] \leq (2^{-b})^n$, which means that X is an $(n\ell, nb)$-source.

A short discussion. Zuckerman and subsequent works considered a generalized definition of block sources with varying block-length, where typically the blocks lengths decrease drastically (e.g., $|X_i| < |X_{i-1}|/2$). This was used as a methodological step towards constructing extractors for general min-entropy sources. While the study of randomness extractors for general min-entropy sources proved to have numerous applications (most of which were not envisioned originally), we believe that the original motivation of *extracting high-quality randomness out of low-quality sources*

[12] Alternatively, we can just rank the the $O(\ell^3)$ elements of $S' \cup A$ that are mapped by h to each image $i \in [t/\ell^3]$, while considering the elements of S' as smaller than those of A. Either way, the fact that A is a multi-set is irrelevant to our analysis, and the procedure just uses each element of A as if it has appeared with multiplicity 1. In contrast, one should not ignore multiplicity when considering the multi-set S, since multiplicity may have an effect on the sampling properties of S.

[13] Actually, block sources were defined in [16] as being an infinite sequence of random variables that satisfy the (conditional) min-entropy bound.

of randomness is extremely important. Furthermore, we believe that block sources are a very realistic model of poor sources of randomness, and hence we believe that extracting high-quality randomness from such sources is of great importance.

Of course, one can extract randomness from block sources by using a corresponding extractor for general min-entropy sources, since any ϵ-error extractor for $(n\ell, nb)$-sources is an ϵ-error extractor for $(n, (\ell, b))$-sources. Yet, more advantageous constructions may be possible for block sources, because the latter are a very restricted special case. In particular, it may be desirable to extract randomness on-the-fly (i.e., in a block-by-block manner), rather than wait for the entire source outcome, and it may be desirable to do so without storing much information. (Of course, this is impossible for general min-entropy sources.)

4.1 A simple extractor

The following construction extract randomness separately from each block, using the same random seed, and without sharing any other information among the (block-based) steps of the extraction process. We stress that using the same seed for extraction from all the blocks harms the quality of the output in a small and minimal manner.

Theorem 4.2 (a simple extractor for block sources): *Let* $E : \{0,1\}^\ell \times \{0,1\}^r \to \{0,1\}^m$ *be a strong* (b, ϵ)*-extractor. Then,* $E' : \{0,1\}^{n\ell} \times \{0,1\}^r \to \{0,1\}^{nm}$ *defined by* $E'(x_1 \circ \cdots \circ x_n, s) = E(x_1, s) \circ \cdots \circ E(x_n, s)$ *is a strong* $n \cdot \epsilon$*-error extractor for* $(n, (\ell, b))$*-block sources.*

Theorem 4.2 seems to be folklore. In particular, it follows as a special case of Lemma 5.7 of Guruswami *et al.* [30], but this fact is not transparent because their result refers to a "block chaining" construction; that is, in their construction, the original seed is used on the last block, and extraction from the i^{th} block (via E_i) is used both for obtaining a part of the output and a seed for extraction from the $i - 1^{\text{st}}$ block.[14] Given a strong extractor E as above, one should first define an auxiliary extractor $E_1(x, s) = (s, E(x, s))$, and then apply [30, Lem. 5.7] with $E_i = E_1$ (for all i's). (For sake of self-containment, we provide a direct and detailed proof of Theorem 4.2.)

Proof: We need to upper bound the statistical distance between U_{nm+r} and $E(X_1, U_r) \circ \cdots \circ E(X_n, U_r) \circ U_r$, where all occurrences of U_r represent the same outcome and $X = (X_1, ..., X_n) \in \{0,1\}^{n\ell}$ is an $(n, (\ell, b))$-block source. We prove the claim by induction on n, where the base case (of $n = 1$) is immediate by the hypothesis regarding E. In the induction step we proceed as follows, using the notation $X_{[j,k]} = (X_j, ..., X_k)$ and $E'(X_{[j,k]}, U_r) = E(X_j, U_r) \circ \cdots \circ E(X_k, U_r)$, where the choice of $i \in [n-1]$ is immaterial (i.e., any $i \in [n-1]$ will do):

$$
\begin{aligned}
&\mathbf{\Delta}[E'(X, U_r) \circ U_r \,; U_{nm} \circ U_r] \\
&\quad= \mathbf{\Delta}[E'(X_{[1,i]}, U_r) \circ E'(X_{[i+1,n]}, U_r) \circ U_r \,; U_{im} \circ U_{(n-i)m} \circ U_r] \\
&\quad\le \mathbf{\Delta}[E'(X_{[1,i]}, U_r) \circ E'(X_{[i+1,n]}, U_r) \circ U_r \,; E'(X_{[1,i]}, U_r) \circ U_{(n-i)m} \circ U_r] \\
&\qquad + \mathbf{\Delta}[E'(X_{[1,i]}, U_r) \circ U_{(n-i)m} \circ U_r \,; U_{im} \circ U_{(n-i)m} \circ U_r] \\
&\quad\le \mathbf{\Delta}[X_{[1,i]} \circ E'(X_{[i+1,n]}, U_r) \circ U_r \,; X_{[1,i]} \circ U_{(n-i)m} \circ U_r] \qquad (3) \\
&\qquad + \mathbf{\Delta}[E'(X_{[1,i]}, U_r) \circ U_r \,; U_{im} \circ U_r] \qquad (4)
\end{aligned}
$$

[14] Specifically, in [30, Lem. 5.7], $E'(x_1 \circ \cdots \circ x_n, s) = y_1 \circ \cdots \circ y_n$ (or rather $E'(x_1 \circ \cdots \circ x_n, s) = s_0 \circ y_1 \circ \cdots \circ y_n$), where $s_n = s$ and $(s_{i-1}, y_i) \leftarrow E_i(x_i, s_i)$ for $i = n, ..., 1$.

where the last inequality uses the fact that $\mathbf{\Delta}[\Pi(Y)\,;\Pi(Z)] \leq \mathbf{\Delta}[Y\,;Z]$ holds for any random process Π. Using the induction hypothesis (regarding extraction from the $(i,(\ell,b))$-source $X_{[1,i]}$), Eq. (4) is upper bounded by $i \cdot \epsilon$. So we turn to analyze Eq. (3). Letting X'_x denote the distribution of $X_{[i+1,n]}$ conditioned on $X_{[1,i]} = x$, we get

$$\begin{aligned}
&\mathbf{\Delta}[X_{[1,i]} \circ E'(X_{[i+1,n]}, U_r) \circ U_r\,; X_{[1,i]} \circ U_{(n-i)m} \circ U_r] \\
&\quad = \quad \mathbf{E}_{x \leftarrow X_{[1,i]}}[\mathbf{\Delta}[E'(X'_x, U_r) \circ U_r\,; U_{(n-i)m} \circ U_r] \\
&\quad \leq \quad (n-i) \cdot \epsilon
\end{aligned}$$

where the inequality uses the induction hypothesis regarding extraction from the $(n-i,(\ell,b))$-source X'_x. The claim follows. ■

Relevance to the study of extraction in $\mathcal{AC}^0$. Theorem 4.2 reduces the complexity of extraction from $(n,(\ell,b))$-block sources to the complexity of extraction from (ℓ,b)-sources. In particular, we get –

Corollary 4.3 ($\mathcal{AC}^0$ extractors for block sources): *For every $\ell(n) = \text{poly}(\log n)$ and $\epsilon(n) = 1/\text{poly}(n)$ there exist $b(n) = O(\log n)$ and explicit $\mathcal{AC}^0$ circuits that compute a strong $\epsilon(n)$-error extractor $E : \{0,1\}^{n\cdot\ell} \times \{0,1\}^{O(\log n)} \to \{0,1\}^{n\cdot\Omega(\log n)}$ for $(n,(\ell,b))$-block sources. Furthermore, the circuits have depth $4 + \frac{\log(\ell(n)/b(n))}{\log\log n}$.*

Note that $b(n)/\ell(n)$ is the min-entropy rate of the source.

Proof: Wishing to use the construction of Theorem 4.2, we (again) use the (b,ϵ)-extractor $E : \{0,1\}^{\ell(n)} \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Omega(\log n)}$ asserted in [28, Sec. 5]. As noted in the proof of Theorem 3.1, this extractor can be computed by (uniform) poly(n)-size circuits of depth $3+\log_{m(n)} \ell(n) = 4 + (\log(\ell(n)/b(n)))/\log\log n$. ■

4.2 On converting min-entropy sources into block sources

In light of the relative ease of extracting randomness from block sources, it is natural to try to convert general min-entropy sources into block sources. Such conversion is intended to be easier than extraction; specifically, it amounts to sampling bits of the original source and arranging them in blocks.

This idea goes back to Zuckerman's work [60, 61], but in all known incarnations each block is sampled using fresh randomness. This is typically not a problem when the number of blocks is small, but in our context we wish the number of blocks to be large. The question addressed here is *whether conversion is possible using randomness complexity that is significantly smaller than the number of desired blocks.*

Following is a definition that captures what we mean by conversion, which we call *blocking* (i.e., converting into a block source). Loosely speaking, a blocker is given a general $(n,\delta n)$-source, and needs to "produce" a block source with $m > 1$ blocks such that each blocks has min-entropy rate at least δ' (conditioned on prior blocks). Of course, we consider $\delta' \in (0,\delta)$ and blocks of length $s \leq n/m$. In terms of the following definition of blockers, we ask whether we may have $r = o(m)$.

Definition 4.4 (converting general min-entropy sources to block sources): *For $n, m, s, r \in \mathbb{N}$ and $\epsilon, \delta, \delta' \in [0, 1]$, consider a generator $S : \{0, 1\}^r \to ([n]^s)^m$, which on input a seed $u \in \{0, 1\}^r$ outputs an m-long sequence of s-subsets of $[n]$, denoted $(S_1(u), ..., S_m(u))$. We say that S is a $(\delta, \delta', \epsilon)$-*blocker *if for every $(n, \delta n)$-source X, for at least $1 - \epsilon$ of the choices of $u \in \{0, 1\}^r$, it holds that $(X_{S_1(u)}, ..., X_{S_m(u)})$ is an $(m, (s, \delta' s))$-block source.*

Definition 4.4 is analogous to the strong version of extraction (i.e., strong extractors). A milder requirement is that $(X_{S_1(U_r)}, ..., X_{S_m(U_r)})$ is ϵ-close to a $(m, (s, \delta' s))$-block source. For simplicity, we consider the stronger notion first and postpone the consideration of the milder notion to later (see Remarks 4.7 and 4.9).

While we do not resolve the foregoing question, we present two illustrations for its difficulty. First, we show that the requirements from a blocker must be more demanding than the requirements from a generator of a sequence of disjoint sets in which each set in the sequence has good sampling properties. This is shown for the case of $m = 2$, albeit using a somewhat contrived construction. Next, we show that a natural construction fails too; this is shown for $m = n^{\Omega(1)}$ and $r = O(\log n)$. In both cases, the reader may think of small but constant values of $\delta > \delta' > 0$ and of $s = \text{poly}(\log n)$. (Recall that randomness extraction from block sources can be performed in $\mathcal{AC}^0$ when the block length is poly-logarithmic.)

1st illustration: Sampling does not suffice (even for two blocks)

Here we show that a "two-set sampler" is not necessarily a good blocker. We shall show this by considering a very general and natural definition of a two-set sampler, and show that there exists such constructs that fails as blockers.

We shall refer to a rather generic notion of a two-set sampler that generates pairs of sets such that some property $\mathcal{P}$ is satisfied for each individual set. The property $\mathcal{P}$ is actually a collection of conditions (or properties), and it is required that each condition is satisfied (with specified probability). Of course, it is also required that the two generated sets are disjoint.

Definition 4.5 (two-set sampler w.r.t a class of properties $\{\mathcal{P}_i\}$): *For $n, s, r \in \mathbb{N}$, let $S : \{0, 1\}^r \to [n]^s \times [n]^s$ and denote $(S_1(u), S_2(u)) = S(u)$. For $\epsilon \in [0, 1]$ and $\mathcal{P}_i \subseteq [n]^s$ for every $i \in [t]$, we say that S is a* two-set sampler w.r.t *$(\epsilon, \{\mathcal{P}_i : i \in [t]\})$ if the following two conditions hold:*

1. *For each $\sigma \in \{1, 2\}$ and $i \in [t]$, it holds that $\mathbf{Pr}[S_\sigma(U_r) \in \mathcal{P}_i] \geq 1 - \epsilon$.*
2. *For every $u \in \{0, 1\}^r$, the sets $S_1(u)$ and $S_2(u)$ are disjoint.*

A natural class of properties are those that correspond to a (δ, ϵ)-averaging sampler. In this case, the properties correspond to subsets of $[n]$ and the property corresponding to a set $T \subseteq [n]$ consists of all s-subsets having $(\rho(T) \pm \delta) \cdot s$ elements in T, where $\rho(T)$ is the density of T in $[n]$.

We could have stated the result only for averaging samplers (i.e., for the aforementioned properties that correspond to them), but we believe that it good to state it in greater generality. Since $\{\mathcal{P}_i\}$ is totally generic, it is not clear that two-set samplers w.r.t it exist. For this reason, the following theorem assumes the existence of a two-set sampler w.r.t $\{\mathcal{P}_i\}$. Furthermore, we also require that $\{\mathcal{P}_i\}$ is closed under relabeling of $[n]$; that is, for every permutation $\pi : [n] \to [n]$ and every i there exists a j such that $\mathcal{P}_j = \{\pi(A) : A \in \mathcal{P}_i\}$. Note that the properties corresponding to averaging samplers are indeed closed under relabeling of $[n]$. The following theorem says that whenever two-set samplers (w.r.t $\mathcal{P}$) exist at all, there exists such samplers that fail as blockers (for the case of $m = 2$).

Theorem 4.6 (two-set samplers are not necessarily blockers): *Let $n, s \in \mathbb{N}$ such that $n = \omega(s^2)$, $\epsilon \in [0, 1]$, and $\mathcal{P} = \{\mathcal{P}_i \subseteq [n]^s\}$ be a collection of properties that is closed under relabeling of $[n]$. If there exists a two-set sampler w.r.t $(\epsilon, \mathcal{P})$, then there exists a two-set sampler w.r.t $(2\epsilon, \mathcal{P})$ that is not a $(0.5, o(1), 1 - o(1))$-blocker. Furthermore, there exists an $(n, 0.5n)$-source X such that given X, with probability at least $1 - 2s^2/n$, this two-set sampler outputs a source of two blocks such that the second block is totally determined by the first block.*

Proof: We start with an arbitrary two-set sampler w.r.t $(\epsilon, \mathcal{P})$, denoted $S : \{0, 1\}^r \to [n]^s \times [n]^s$. Consider a random matching $\pi : [n] \to [n]$ (i.e., a random bijection π such that $\pi(i) \neq i$ and $\pi(\pi(i)) = i$ for every $i \in [n]$), and note that, w.v.h.p, for an $1 - O(s^2/n)$ fraction of the u's it holds that $\pi(S_1(u)) \cap S_1(u) = \emptyset$. Let us call such u's good for π, and note that there exists a matching π for which the fraction of good u's is at least $1 - O(s^2/n) = 1 - o(1)$. Fix such a π and define a new two-set sampler S', which uses a seed $(u, u') \in \{0, 1\}^{2r}$, as follows:

1. If $\pi(S_1(u)) \cap S_1(u) = \emptyset$, let $S'(u, u') = (S_1(u), \pi(S_1(u)))$ and call u good;
2. Otherwise (i.e., $\pi(S_1(u)) \cap S_1(u) \neq \emptyset$) let $S'(u, u') = S(u')$.

Note that $1 - o(1)$ of the u's are good, and that S' is a two-set sampler w.r.t $(2\epsilon, \mathcal{P})$. To prove the latter assertion, let us first consider $S'_1(U_{2r})$, and denote the set of good r-bit strings by G. Then, for every $\mathcal{P}_i$,

$$\begin{aligned}\mathbf{Pr}[S'_1(U_{2r}) \notin \mathcal{P}_i] &= \mathbf{Pr}_{u \leftarrow U_r}[S_1(u) \notin \mathcal{P}_i \wedge u \in G] + \mathbf{Pr}_{(u,u') \leftarrow U_{2r}}[S_1(u') \notin \mathcal{P}_i \wedge u \notin G] \\ &< \mathbf{Pr}_{u \leftarrow U_r}[S_1(u) \notin \mathcal{P}_i] + \mathbf{Pr}_{(u,u') \leftarrow U_{2r}}[S_1(u') \notin \mathcal{P}_i].\end{aligned}$$

The same analysis applies to the second set (i.e., $S'_2(U_{2r})$), except that here $S_1(u)$ is replaced by $S_1(\pi(u))$ and the "closure under relabeling" hypothesis is used. Specifically, suppose that $\mathcal{P}_j = \{\pi^{-1}(A) : A \in \mathcal{P}_i\}$, then:

$$\begin{aligned}\mathbf{Pr}[S'_2(U_{2r}) \notin \mathcal{P}_i] &= \mathbf{Pr}_{u \leftarrow U_r}[S_1(\pi(u)) \notin \mathcal{P}_i \wedge u \in G] + \mathbf{Pr}_{(u,u') \leftarrow U_{2r}}[S_2(u') \notin \mathcal{P}_i \wedge u \notin G] \\ &< \mathbf{Pr}_{u \leftarrow U_r}[S_1(\pi(u)) \notin \mathcal{P}_i] + \mathbf{Pr}_{(u,u') \leftarrow U_{2r}}[S_2(u') \notin \mathcal{P}_i] \\ &= \mathbf{Pr}_{u \leftarrow U_r}[S_1(u) \notin \mathcal{P}_j] + \mathbf{Pr}_{u' \leftarrow U_r}[S_2(u') \notin \mathcal{P}_i].\end{aligned}$$

Hence, S' is a two-set sampler w.r.t $(2\epsilon, \mathcal{P})$. In contrast to this fact, as shown next, it turns out that S' fails miserably as a blocker.

Let X be uniform over the set of strings $\{x : (\forall i \in [n])\ x_i = x_{\pi(i)}\}$, which has cardinality $2^{n/2}$. This means that X has min-entropy $n/2$. On the other hand, whenever u is good, we have that $X_{S'_2(u)}$ is determined by $X_{S'_1(u)}$, since in this case $S'_2(u) = \pi(S'_1(u))$, which implies $X_{S'_2(u)} = X_{\pi(S'_1(u))} = X_{S'_1(u)}$. Hence, with probability at least $1 - 2s^2/n = 1 - o(1)$ (i.e., whenever u is good), the second block in the source $(X_{S'_1(u)}, X_{S'_2(u)})$ has min-entropy zero conditioned on the first block. ■

Remark 4.7 (non-strong blocking fails too): *Actually, the proof of Theorem 4.6 applies also to the weaker notion of blocking in which it is only required that $(X_{S'_1(U_r)}, X_{S'_2(U_r)})$ is ϵ-close to a*

$(2,(s,\delta's))$-block source. The proof implies that $(X_{S'_1(U_r)},X_{S'_2(U_r)})$ is $2s^2/n$-close to a source in which the second block equals the first block, and thus has no conditional entropy at all. It follows that this sampled source is far from any block source in which the second block has even just few bits of min-entropy.

Digest. Note that Theorem 4.6 does not refer to the randomness complexity of the two-set sampler. In such a setting, we know that blockers (let alone for $m=2$) do exist. Hence what Theorem 4.6 asserts is only that requirements regarding the sampling features of individuals sets generated by a two-set samplers do not imply that this two-set sampler is a blocker.

The reason that two-set samplers may fail as blockers is that their definition makes too mild requirements regarding the relation between the two generated sets. Indeed, Definition 4.4 only requires that these two sets be disjoint. The proof of Theorem 4.6 capitalizes on this fact and uses a fixed matching of elements between all pairs of sets (i.e., for a fixed matching π, the elements of the second set are typically the π-mates of the elements of the first set).

2nd illustration: A natural candidate that fails

The proof of Theorem 4.6 relies on a contrived example and shows that such an example exists no matter what "sampling property" (regarding individual sets) is considered. Here we take an opposite approach: We consider a natural construction (and do not specify the sampling properties that it satisfies). Specifically, we shall present a natural multi-set sampler, which we believe most readers may find a natural candidate for a good blocker, and show that it actually fails as a blocker. The multi-set sampler (and candidate blocker) refers to the case of $m=n^{\Omega(1)}$, $s=\text{poly}(\log n)$, and $r=O(\log n)$. (Recall that these are the parameters that we want in order to extract $n^{\Omega(1)}$ bits in $\mathcal{AC}^0$ via conversion to block sources.)

A rhetoric question: *What is more natural than trying the standard $O(1)$-wise independent generator?* Indeed, let us consider it.

Construction 4.8 (constant-wise independent multi-set sampler): *Let F be a finite field of size n and $c\geq 2$ be a constant, and consider the c-wise independent generator, denoted $G:F^c\to F^n$, such that $G(u)=(g_1(u),...,g_n(u))$ and $g_j(u_1,...,u_c)=\sum_{i\in[c]}u_i\alpha_j^{i-1}$, where $\alpha_1,...,\alpha_n$ are distinct field elements. For $m=n^{\Omega(1)}$ and $s=\text{poly}(\log n)$, consider the sampler $S(u)=(S_1(u),...,S_m(u))$ such that $S_i(u)=\{g_{(i-1)s+1}(u),...,g_{is}(u)\}$.*

Note that $S:F^c\to(F^s)^m$, which means that this sampler has a seed of length $c\log_2 n$. *Is this sampler not a natural candidate for a blocker?* Well, as we show next, it fails badly.

- *Preliminaries*: We assume that for some $d\in\mathbb{N}$ it holds that $ms\in(0.9\pm0.1)\cdot n^{1/d}$ (or so).[15] Let $F=K^d$, where K is a finite field, and let $H\subset K$ of sufficiently small constant density (say, density $1/3cd$). Hence, $k\overset{\text{def}}{=}|H^d|=\Omega(n)$, since $n=|K^d|$ and $|H|=\Omega(|K|)$.
- *A class of (affine) sources*: Next, for any $f:H^d\to K$, consider its low-degree extension $f':K^d\to K$, and let $X\in K^n$ be uniformly distributed among all d-variate polynomials of individual degree $|H|$; that is, for a uniformly distributed $f:H^d\to K$, let $X_\alpha=f'(\alpha)$ for

[15] After all, if the above sampler is a good blocker for $m=n^{\Omega(1)}$ and $s=\text{poly}(\log n)$, then it should be a good blocker also when n is an integer power of ms (or so).

every $\alpha \in K^d \equiv [n]$. So X has "min-entropy" k (in units of symbols in K). Indeed, this is a source over the alphabet K (rather than over $\{0,1\}$).

By the way, it is an affine source, since the values of all X_α's (i.e., the polynomial f') are determined as linear combination of the values at $\alpha \in H^d$ (i.e., the function f).

- *The foregoing sampler S as a candidate blocker*: Now, for any $u = (u_1, ..., u_c) \in [n]^c \equiv (K^d)^c$, the sampler $S : (K^d)^c \to ((K^d)^s)^m$ produces the sampled source $(X_{S_1(u)}, ..., X_{S_m(u)})$. Recall that $ms \in (0.9 \pm 0.1) \cdot |K|$ and $|H| = |K|/3cd$.

Let X be a generic source from the above class. Plugging in the definition of S_i, note that the j^{th} element in the i^{th} block of the sampled source is $X_{g_{(i-1)s+j}(u)} = X_{C_u(\alpha_{(i-1)s+j})}$, where $C_u(\alpha) = \sum_{\ell \in [c]} u_\ell \alpha^{\ell-1}$. Now, suppose that $\alpha_1, ..., \alpha_{ms} \in F = K^d$ are all in the base field K (which is possible since $ms < |K|$). Then, it suffices to define C_u on K (i.e., $C_u : K \to K^d$), which means that C_u is a $(c-1)$-degree curve over K^d (i.e., it is a curve over K^d, defined based on $u \in (K^d)^c$, with a free parameter in K).

Recalling that X is defined in term of the low-degree extension f' of a random function $f : H^d \to K$, we have $X_{C_u(\alpha_{(i-1)s+j})} = f'(C_u(\alpha_{(i-1)s+j}))$, where $f' \circ C_u : K \to K$ (via K^d) is a degree $(c-1)d|H|$ univariate polynomial over K (since f' is a d-variate polynomial of individual degree $|H|$ and C_u is a curve of degree $c-1$). Hence, for every seed u of the sampler, less than $cd|H|$ symbols of the sampled source determine all other symbols (of the sampled source); in particular, *the first $cd|H|/s$ blocks of the sampled source $(X_{S_1(u)}, ..., X_{S_m(u)})$ fully determine the remaining $m - (cd|H|/s)$ blocks.* Recall that $cd|H|/s = |K|/3s < m/2$ (by our choice of H and K). ■

Remark 4.9 (non-strong blocking fails too): *Since there are only polynomially many curves C_u, using few additional symbols* (let alone few additional blocks)*, we can determine which curve is used, and determine the remaining blocks based on this. Typically, each additional symbol read from the sampled source, beyond the $cd|H|$ symbols that suffice for extrapolation, cuts the number of possible curves by a factor of $|K|$.*

Critique: One may raise two reasonable objections to our example. (1) We considered sources over a large (non-binary) alphabet K; (2) We made the sampler use the $|K|$ elements of $K \subset K^d$ as the coefficients in the c-wise independent sequence (of length $|K|$). We believe that both objections are not acute.

Objection (1) can be addressed by encoding the elements of K by binary strings of length $\ell = \log |K|$. This requires an analogous modification of the sampler in which indices in $[n]$ are replaced by ℓ-sets of $[n\ell]$ (i.e., i is replaced by $\{(i-1)\ell + 1, ..., i\ell\}$).[16] (Alternatively, one may argue that the question for arbitrary alphabets is as natural.) Objection (2) can be addressed by claiming that the construction should work regardless of the choice of field elements, let alone that the same argument holds when choosing field elements that reside on any line (or low degree curve) in K^d (instead of residing in K). Indeed, we do not recall any application of c-wise independence that insists on an "non-structured" choice of the field elements (whenever there is a choice at all). Actually, it is quite natural to use a structured choice of field elements.

[16] Indeed, this merely moves the problem from the alphabet to the sampler (which now samples indices by taking all indices in each sampled block), and one may object to viewing such a sampler as natural.

Discussion

Note that both counterexamples utilize affine sources, whereas it is possible to convert affine sources (having min-entropy at least $k = n/\text{poly}(\log n)$) into block sources (using a seed of logarithmic length and obtaining blocks of length $s = \text{poly}(n/k)$). Specifically, as hinted up-front, it is well known that conversion into m blocks is possible if one is willing to use a seed of length $r = O(m \log n)$ (by repeated sampling with fresh random seeds). Since the number of affine sources is less than 2^{n^2}, we may infer that there exists a set of $O(n^2/\delta\epsilon)$ such that for an affine source of min-entropy δn at least a $1 - \epsilon$ fraction of these seeds yield an $(m, (s, 0.5\delta s))$-block source.

The forgoing argument cannot be applied to general $(n, \delta n)$-sources, since their number is too large. But it is not inconceivable that a more refine counting argument may work. More generally, we ask –

Open Problem 4.10 (blockers of logarithmic randomness): *Does there exist a $(\delta, \delta', o(1))$-blockers $S : \{0,1\}^{O(\log n)} \to ([n]^s)^m$ for constant $\delta' \in (0, \delta) \subset (0, 1)$ and $m = \omega(1)$? What about $m = n^{\Omega(1)}$ and $s = \text{poly}(\log n)$?*

A positive answer to the latter question is a sufficient but possibly not necessary condition for a positive resolution of Problem 1.6. Of course, a negative resolution of Problem 1.6 would imply a negative answer to Problem 4.10.

5 Extraction from Block-Fixing Sources

In this section, we consider $\mathcal{AC}^0$ extractors for bit-fixing sources, a model first considered by Chor *et al.* [17]. In this model the min-entropy bound k denotes the number of bits that are random in the source, whereas the other $n - k$ bits are fixed *obliviously* of the values of the random bits.[17] Actually, we consider a generalization to block-fixing sources, first considered in [40].

Definition 5.1 (block-fixing sources [17, 40]): *An (n, k, ℓ)-*block-fixing source *is a sequence of random variables $X = (X_1, ..., X_n) \in \{0,1\}^{n\ell}$ such that there exists a set of at least k indices $I \subseteq [n]$ and a sequence $(x_1, ..., x_n) \in \{0,1\}^{n\ell}$ such that X_I is uniformly distributed over $\{0,1\}^{|I|\ell}$ and $X_i = x_i$ for every $i \in [n] \setminus I$. The special case of $\ell = 1$ is referred to as a (n, k)-*bit-fixing source*; that is, a $(n, k, 1)$-block-fixing source is called a (n, k)-bit-fixing source.*

Note that extractors for block-fixing sources need not preserve the block structure in their output, although the extractors presented in Theorem 5.2 do preserve this structure. In general, for every ℓ, an ϵ-error extractor for $(n\ell, k\ell)$-bit-fixing sources is an ϵ-error extractor for (n, k, ℓ)-block-fixing sources, and ditto for strong extractors. It is also easy to see that if $E : \{0,1\}^n \times \{0,1\}^{r(n)} \to \{0,1\}^{m(n)}$ is a strong ϵ-error extractor for (n, k)-bit-fixing sources, then $E' : \{0,1\}^{\ell \cdot n} \times \{0,1\}^{r(n)} \to \{0,1\}^{\ell \cdot m(n)}$ given by

$$E'((x_{1,1}, ..., x_{n,\ell}), s) = E((x_{1,1}, ..., x_{n,1}), s) \circ \cdots \circ E((x_{1,\ell}, ..., x_{n,\ell}), s)$$

is a strong $\ell\epsilon$-error extractor for $(n\ell, k\ell)$-bit-fixing sources. (An analogous statement for ordinary extractors seems to require using ℓ different seeds, and so the seed length becomes $\ell \cdot r(n)$.)

[17] Indeed, such sources are sometimes called *oblivious bit-fixing sources*, in order to distinguish them from non-oblivious bit-fixing sources [40] in which the remaining $n - k$ bits are fixed as a function of the values of the k random bits.

In Section 5.1, we present (strong) extractors for $(n, n/\mathrm{poly}(\log n), \ell)$-block-fixing sources that use a seed of logarithmic length and extract $n\ell/\mathrm{poly}(\log n)$ bits, whereas in Section 5.3 we present deterministic extractors of similar performance. Both extractors work in $\mathcal{AC}^0$, which is optimal in light of the results presented in Section 5.2 (assuming that $\ell \leq \mathrm{poly}(\log n)$). Specifically, in Section 5.2, we prove that there exist no strong $\mathcal{AC}^0$ extractors for min-entropy lower than $n/\mathrm{poly}(\log n)$ (regardless of the seed length). We also show that the same holds for ordinary extractors that output $(1+\Omega(1)) \cdot r(n)$ bits when using a seed of length $r(n)$.

5.1 Extraction with a logarithmically long seed

Although extraction from bit-fixing sources is possible without using a random seed (see [17, 40]), the known deterministic (i.e., seedless) extractors are not computable in $\mathcal{AC}^0$. In this section we present seeded extractors (for bit-fixing sources) that are computable in $\mathcal{AC}^0$ (and use seeds of logarithmic length). In Section 5.3, following ideas of Gabizon *et al.* [25], we shall use the aforementioned seeded extractors to obtain deterministic extractors of similar performance, based on new deterministic extractors that extract poly-logarithmically many bits.

Theorem 5.2 (seeded extractor for block-fixing sources): *Let $k : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0,1]$ be such that $k(n) \geq n/\mathrm{poly}(\log n)$ and $\epsilon(n) \geq 1/\mathrm{poly}(n)$. Then, there exists a function $E : \{0,1\}^{n\ell} \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\mathrm{poly}(k(n)/n) \cdot n\ell}$ that is computable in uniform $\mathcal{AC}^0$ and constitutes a strong ϵ-error extractor for (n,k,ℓ)-block-fixing sources.*

Theorem 5.2 is related to results of Gabizon *et al.* [25, Sec. 6], but the parameters are quite different: Most importantly, their extractors are not in $\mathcal{AC}^0$, although they could have obtained such extractors with a different setting of the parameters in their construction. Likewise, they extract only $m(n) = k(n)^{\Omega(1)}$ bits. On the other hand, they treat any $k(n) \geq \mathrm{poly}(\log n)$ and use a seed of length $O(\log k(n))$. The pivot of our solution is the use of a new pseudorandom partition generator, captured by Lemma 5.2.1.

Proof: For sake of simplicity, we consider the case of $\ell = 1$, and denote the bits of the source by $x_1 \circ \cdots \circ x_n$. The basic idea, which goes back to Gabizon *et al.* [25, Sec. 6], is to use a random partition of the source into many short sources, and extract bits from each of these short sources by XORing the corresponding bits. That is, we use the seed of the extractor to generate (pseudo)random subsets, denoted $S_1, ..., S_m \subset [n]$, and output the bits $\oplus_{j \in S_i} x_j$ for $i = 1, ..., m$. This works provides that each of the S_i's contains a location of a random (i.e., non-fixed) bit of the original source, and that these (non-fixed) locations are distinct. Furthermore, the S_i's should each be of size $\mathrm{poly}(\log n)$ so to allow for the XOR to be implemented by constant-depth $\mathrm{poly}(n)$-size circuits.

The first idea that comes to mind is to generate the subsets by repeatedly invoking an ordinary sampler (cf. [27]), using related seeds, but this seems to require that $m < \sqrt{n}$. An alternative approach, which also goes back to Gabizon *et al.* [25, Sec. 6], is to generate a pseudorandom partition of $[n]$ into m subsets of equal size. Unfortunately, the techniques used in [25, Sec. 5] also seems to require that $m < \sqrt{n}$, whereas we seek $m = n/\mathrm{poly}(\log n)$. Furthermore, we need to generate such a partition using a seed of logarithmic length and each subset in the partition should have a strong hitting property (i.e., as stated in the second condition of the following claim). (We comment that similar problems arise in the proof of Theorems 3.2 and 3.6, but the parameters and hitting requirements there are different.)

Lemma 5.2.1 (pseudorandom partitions with a strong hitting property): *For $\delta, \gamma > 0$ and $n, t \in \mathbb{N}$ such that $t = \Theta(\delta^{-1} \log(1/\gamma))^2$ divides n, let $m = n/t$ and $r = O(\log(n/\gamma))$. Then, there exists an explicit function $G : \{0,1\}^r \to ([n]^t)^m$ such that the following two conditions hold.*

1. *For every $u \in \{0,1\}^r$ the m-sequence $G(u)$ is a partition of $[n]$; that is, for every $j_1 \neq j_2$ it holds that $G(u)_{j_1}$ and $G(u)_{j_2}$ are disjoint t-subsets of $[n]$.*
2. *For every $T \subseteq [n]$ of density δ, with probability at least $1 - \gamma$, each subset in $G(U_r)$ hits T; that is, $\mathbf{Pr}[\exists j \in [m] \text{ s.t. } G(U_r)_j \cap T = \emptyset] \leq \gamma$.*

Proof: Our starting point is an ordinary hitter $H : \{0,1\}^{r'} \to [n]^s$, where $r' = O(\log(n/\gamma')$ and $s = O(\delta^{-1} \log(1/\gamma'))$, that hits each set of density δ with probability at least $1 - \gamma'$. We further assume that every two sample points of this hitter are uniformly distributed independently of one another. Note that the combined hitter of [27, Apdx. C.3] satisfies the first requirement, whereas the second requirement can be achieved by "randomizing" the original sample via a sequence of pairwise independent "shifts" (cf. Claim 3.2.2).[18] Associating $[n]$ with $[t] \times \mathbb{Z}_m$ and using $t = s^2$, it follows that with probability at least half, the sample $((i_1, j_1), ..., (i_s, j_s)) \leftarrow H(U_{r'})$ contains no collision on the first coordinate (i.e., for every $a \neq b$ it holds that $i_a \neq i_b$). Hence, conditioned on this event, for every set T of density δ, the probability that $H(U_{r'})$ hits T is at least $1 - 2\gamma'$. Consider the following generator $G' : \{0,1\}^{r'} \to ([n]^t)^m$:

1. On input $u \in \{0,1\}^{r'}$, obtain $((i_1, j_1), ..., (i_s, j_s)) \leftarrow H(u)$. If there exist $a \neq b$ such that $i_a = i_b$, then output a fixed partition $(S_0, ..., S_{m-1})$ of $[n] \equiv [t] \times \mathbb{Z}_m$ into t-subsets; for example, $S_j = \{(i, j) : i \in [t]\}$. In this case we say that u is bad.
2. Otherwise (i.e., for every $a \neq b$ it holds that $i_a \neq i_b$), for every $i \in [t]$ and $j \in \mathbb{Z}_m$, let $p_{i,j} = (i, j_a + j \bmod m)$ if $i = i_a$ and $p_{i,j} = (i, j)$ otherwise (i.e., $i \notin \{i_1, ..., i_s\}$). For every $j \in \mathbb{Z}_m$, define $G'_j(u) = \{p_{i,j} : i \in [t]\}$, and output the m-sequence $(G'_0(u), ..., G'_{m-1}(u))$. In this case we say that u is good, and it holds that

   $$G'_0(u) = \{(i_a, j_a) : a \in [s]\} \cup \{(i, 0) : i \in ([t] \setminus \{i_1, ..., i_s\})\}$$

   and $G'_j(u) = \{(i, j' + j \bmod m) : (i, j') \in G'_0(u)\}$.

Note that in each case the output sequence is a partition of $[n] \equiv [t] \times \mathbb{Z}_m$. In the "good case" (i.e., of a good u) this follows since $G'_0(u) = \{(i, g(i)) : i \in [t]\}$ for some function $g : \mathbb{Z}_m \to \mathbb{Z}_m$ (i.e., $g(i_a) = j_a$ for $a \in [s]$ and $g(i) = 0$ for $i \notin \{i_1, ..., i_s\}$) and $G'_j(u) = \{(i, g(i) + j \bmod m) : i \in [t]\}$. The foregoing properties of H imply that, with probability at least half $U_{r'}$ is good, whereas conditioned

[18] Alternatively, we can apply the sampler used in the proof of Theorem 3.2 (with $c = 1$), but set the parameters in order to satisfy hitting rather than sampling. The difference between the two versions is that shifting the samples (as proposed in the main text) is different from shifting the seeds used to generate the subsamples. Specifically, recall that the combined hitter of [27, Apdx. C.3] has the form $H''(u) = \cup_{i \in [t'']} H'_i(v_i)$, where $(v_1, ..., v_{t''}) \leftarrow W(u)$ is a random walk on an expander and H'_i is a pairwise independence generator. In the main text we suggested using pairwise independent shifts of the sample $\cup_{i \in [t'']} H'(v_i)$, whereas in the proof of Theorem 3.2 we used the sample $\cup_{i \in [t'']} H'(v'_i)$, where $(v'_1, ..., v'_{t''})$ is obtained by a pairwise independent shift of $(v_1, ..., v_{t''})$. In the analysis of the first alternative one relies on the generalized hitting property of H', whereas in the analysis of the second alternative one relies on the generalized hitting property of W, where generalized hitting refers to generating a sequence $(\sigma_1, ..., \sigma_s)$ such that for every sequence of sets $(T_1, ..., T_s)$ (each of density δ) with probability at least $1 - \gamma$ there exists an $i \in [s]$ such that $\sigma_i \in T_i$. Note that the standard analyses of both H' and W extends to this case.

on this event the probability that $G'_j(U_{r'}) \in T$ is at least $1-2\gamma'$, for every set T of density δ and for every $j \in \mathbb{Z}_m$.

Let us reflect for a moment on the structure of the generator G'. It uses a fixed partition of $[n] \equiv [t] \times \mathbb{Z}_m$ into t cycles, each of length m, where the i^{th} cycle consists of $\{(i,j) : j \in \mathbb{Z}_m\}$. Assuming that the sample $H(u) \in ([t] \times \mathbb{Z}_n)^s$ hits exactly s cycles, we augmented the sample to a cover of all cycles by adding (dummy) elements of the form $(i,0)$ for every uncovered cycle. (Of course, this cannot damage the hitting property.) Finally, we use the m "shifts" of the resulting set S_0 as a partition, where the j^{th} shift of the set $S_0 \subset [t] \times \mathbb{Z}_m$ equals $\{(i, j'+j) : (i,j') \in S_0\}$. In case the initial sample hits less than s cycles (i.e., some cycle is hit by more than a single point in the sample), the generator outputs a fixed partition. The only problem is that the latter bad event may occur with constant probability (of at most 1/2), whereas we want it to occur (in the final generator) with probability smaller than γ.

The desired generator G is obtained by taking a random walk of length $t' = O(\log(1/\gamma))$ on a constant degree expander with vertex set $\{0,1\}^{r'}$, and outputting $G'(v)$ where v is the first vertex on the walk that is good. If no such vertex exists, then we just output the fixed partition, but this event occurs with probability at most $\gamma/2$.

Denoting the random walk by $(v_1, ..., v_{t'})$, recall that the probability that some v_j is good is at least $1 - \exp(-\Omega(t')) > 1 - (\gamma/2)$. We show that conditioned on this event, for every T of density δ, with probability at least $1 - (3\gamma/4) - (8t'm\gamma'/\gamma)$, each set in the partition output by G hits the set T. This is shown by considering only the indices $j \in [t']$ such that with probability at least $\gamma/4t'$ vertex v_j is the first vertex in the walk that is good. Specifically, denote by $\mathcal{G}_j$ the event that v_j is good, and let $\mathcal{G}'_j$ denote the event $\mathcal{G}_j \wedge \neg(\vee_{j'<j}\mathcal{G}_{j'})$. Let $J = \{j \in [t'] : \mathbf{Pr}[\mathcal{G}'_j] \geq \gamma/4t'\}$ and note that $\mathbf{Pr}[\vee_{j\in J}\mathcal{G}'_j] > 1 - (\gamma/2) - t' \cdot (\gamma/4t') = 1 - 3\gamma/4$. On the other hand, for each $j \in J$, the probability that each set in $G(v_j)$ hits T is at least $1 - 2m\gamma'$. Let use denote this event by $\mathcal{H}_j$. Then, the probability that all sets output by G hit T is at least

$$\begin{aligned}\sum_{j\in J} \mathbf{Pr}[\mathcal{G}'_j] \cdot \mathbf{Pr}[\mathcal{H}_j|\mathcal{G}'_j] &\geq \mathbf{Pr}[\vee_{j\in J}\mathcal{G}'_j] \cdot \min_{j\in J}\{\mathbf{Pr}[\mathcal{H}_j|\mathcal{G}'_j]\} \\ &\geq \left(1 - \frac{3\gamma}{4}\right) \cdot \min_{j\in J}\left\{1 - \frac{\mathbf{Pr}[\neg\mathcal{H}_j]}{\mathbf{Pr}[\mathcal{G}'_j]}\right\}.\end{aligned}$$

Using $\mathbf{Pr}[\neg\mathcal{H}_j] \leq 2m\gamma'$ and $\mathbf{Pr}[\mathcal{G}'_j] \geq \gamma/4t'$ (for any $j \in J$), we obtained the claimed lower bound of $(1 - 3\gamma/4) \cdot (1 - (2m\gamma')/(\gamma/4t')) > (1 - 3\gamma/4) + (8t'm\gamma'/\gamma)$.

Setting $\gamma' = \gamma^2/32t'm$, it follows that each set output by G hits a set T of density δ with probability at least $1 - \gamma$. Noting that G uses a seed of length $r = r' + O(t') = O(\log(n/\gamma))$, the lemma follows. ■

Wrapping up. We set $\delta = k(n)/n = 1/\text{poly}(\log n)$ and $\gamma = \epsilon = 1/\text{poly}(n)$, and use Lemma 5.2.1 with $s = O(\delta^{-1}\log(1/\gamma))$ and $t = s^2 = O(\delta^{-1}\log n)^2$, which implies $m = n/t = \Omega(\delta^2 n/\log^2 n) = n/\text{poly}(\log n)$ and $r = O(\log n)$. The extractor $E(x,u)$ outputs $y_1 \circ \cdots \circ y_m$ such that $y_j = \oplus_{i\in G(u)_{j+1}} x_i$. Since G can be computed by uniform depth-two circuits of size poly(n) and $t = \text{poly}(\log n)$, it follows that E is in uniform $\mathcal{AC}^0$. In analyzing the performance of E on an arbitrary (n,k)-bit-fixing source, let T denote the set of k unfixed bits in the source, and note that the output bits (i.e., y_j's) depend on disjoint sets of bits in the source and that with probability at least $1-\gamma$ each output bit depends on some unfixed bit of the source. The theorem follows for $\ell = 1$, and the argument for general ℓ is identical. ■

5.2 Impossibility results

The following impossibility result asserts the optimality of Theorem 5.2 with respect to the parameter k (i.e., the number of random blocks in the source): While Theorem 5.2 asserts strong extractors that are computable in $\mathcal{AC}^0$ for any $k(n) = n/(\log n)^{O(1)}$, the following result asserts that this is not possible for any $k(n) = n/(\log n)^{\omega(1)}$.

Theorem 5.3 (impossibility of strong extraction in $\mathcal{AC}^0$): *Suppose that $E : \{0,1\}^{n\ell} \times \{0,1\}^r \to \{0,1\}$ is computable by $s(n)$-size circuits of depth $d = d(n)$. If E is a strong 0.499-error extractor for (n,k,ℓ)-block-fixing sources, then $k > n/(\ell \cdot O(\log s(n))^{d-1})$.*

Recall that $\ell = 1$ corresponds to bit-fixing sources. Note that the current result regarding (n,k)-bit-fixing sources implies that *if a strong $(0.499/\ell)$-error extractor for (n,k,ℓ)-block-fixing sources is computable in $\mathcal{AC}^0$, then $k \geq n/\text{poly}(\log n)$*. We conjecture that this holds even for 0.499-error extractor for (n,k,ℓ)-block-fixing sources; that is, we *conjecture that the linear dependence on ℓ, in the foregoing results, can be eliminated.*

Before proving Theorem 5.3 we note that it is incomparable but related to Theorem 1.1 (i.e., Viola's [56, Thm. 6.4]): Theorem 5.3 refers to strong extractors that output a single bit, whereas Theorem 1.1 applies to ordinary extractors that output a constant factor more bits than their seed length. (A version that refers to ordinary extractors is presented later; see Theorem 5.4.) An important advantage of Theorem 5.3 over Theorem 1.1 is that it refers to a much more restricted class of sources (i.e., $(n,k,1)$-block-fixing sources rather than (n,k)-sources).

Proof: For simplicity, we start with the case of $\ell = 1$. Fixing any value $\sigma \in \{0,1\}^r$ of the seed, consider the residual depth d circuit of size $s = s(n)$, denoted C_σ, that computes the mapping $x \mapsto E(x,\sigma)$. Invoking the "average sensitivity bound" of Linial *et al.* [38], as improved by Boppana [11], for every C_σ we have

$$\sum_{i\in[n]} \mathtt{I}_i(C_\sigma) < B \stackrel{\text{def}}{=} O(\log s)^{d-1}, \tag{5}$$

where for any Boolean function $f : \{0,1\}^n \to \{0,1\}$

$$\mathtt{I}_i(f) \stackrel{\text{def}}{=} \mathbf{Pr}_{x\leftarrow U_n}[f(x) \neq f(x \oplus 0^{i-1}10^{n-i})]. \tag{6}$$

(See background in [36, Sec. 12.4] and [46, Chap. 2].)

It follows that there exists a set of $\epsilon n/B$ indices $I \subseteq [n]$ such that $\mathbf{E}_{\sigma\leftarrow U_r}[\sum_{i\in I} \mathtt{I}_i(C_\sigma)] < \epsilon$, since the expectation over all $|I|$-sized subsets is smaller than ϵ. Furthermore, there exists a string $z \in \{0,1\}^n$ such that

$$\mathbf{E}_{\sigma\leftarrow U_r}\left[\sum_{i\in I} \mathbf{Pr}_{x\leftarrow U_n}\left[C_\sigma(x) \neq C_\sigma(x \oplus 0^{i-1}10^{n-i})\middle| x_{[n]\setminus I} = z_{[n]\setminus I}\right]\right] < \epsilon \tag{7}$$

and there exists a set $G \subseteq \{0,1\}^r$ of density at least $1 - \sqrt{\epsilon}$ such that for every $\sigma \in G$ it holds that

$$\sum_{i\in I} \mathbf{Pr}_{x\leftarrow U_n}\left[C_\sigma(x) \neq C_\sigma(x \oplus 0^{i-1}10^{n-i})\middle| x_{[n]\setminus I} = z_{[n]\setminus I}\right] < \sqrt{\epsilon}. \tag{8}$$

Fixing I and z as above, consider an $(n,|I|,1)$-block-fixing source $X = (X_1,...,X_n)$ such that $X_i = z_i$ if $i \in [n] \setminus I$ and X_i is random otherwise. We next show that, for every $\sigma \in G$ there exists a bit y_σ so that $\mathbf{Pr}[C_\sigma(X) = y_\sigma] > 1 - \sqrt{\epsilon}$, To prove the above claim, assume that $p \stackrel{\rm def}{=} \mathbf{Pr}[C_\sigma(X) = y_\sigma] \leq 1 - \sqrt{\epsilon}$ and $p \geq 1/2$ (w.l.o.g.). Then, $\mathbf{Pr}_{x,y \leftarrow U_n}[C_\sigma(x) \neq C_\sigma(y) | x_{[n]\setminus I} = y_{[n]\setminus I} = z_{[n]\setminus I}] = 2p(1-p) \geq \sqrt{\epsilon}$. This implies that there exists $s \in \{0,1\}^n$ such that $s_{[n]\setminus I} = 0^{n-|I|}$ and $\mathbf{Pr}[C_\sigma(X) \neq C_\sigma(X \oplus s)] \geq \sqrt{\epsilon}$, which contradicts Eq. (8), since

$$\begin{aligned}\mathbf{Pr}[C_\sigma(X) \neq C_\sigma(X \oplus s)] &\leq \sum_{i:s_i=1} \mathbf{Pr}[C_\sigma(X) \neq C_\sigma(X \oplus 0^{i-1}10^{n-i})] \\ &\leq \sum_{i \in I} \mathbf{Pr}[C_\sigma(X) \neq C_\sigma(X \oplus 0^{i-1}10^{n-i})].\end{aligned}$$

(The first inequality uses the fact that $X \equiv X \oplus s'$ for every $s' \in \{0,1\}^n$ such that $s'_{[n]\setminus I} = 0^{n-|I|}$.)

We have established that for every $\sigma \in G$, it holds that $\mathbf{Pr}[E(X,\sigma) = y_\sigma] > 1 - \sqrt{\epsilon}$, for some bit y_σ, whereas $\mathbf{Pr}[U_1 = y_\sigma] = 1/2$. It follows that $\mathbf{\Delta}[E(X,U_r) \circ U_r \,\mathbf{;}\, U_1 \circ U_r]$, which equals $\mathbf{E}_{u \leftarrow U_r}[\mathbf{\Delta}[E(X,u) \,\mathbf{;}\, U_1]]$, is greater than $\mathbf{Pr}[U_r \in G] \cdot (1 - \sqrt{\epsilon} - 0.5) \geq (1-\sqrt{\epsilon}) \cdot (0.5 - \sqrt{\epsilon}) > 0.5 - 2\sqrt{\epsilon}$. Hence, if E is a strong $(0.5 - 2\sqrt{\epsilon})$-error extractor for $(n,k,1)$-block-fixing sources, then $k > \epsilon n/B$.

The argument for general $\ell > 1$ proceeds analogously, except that here we have $n \cdot \ell$ variables/indices, which are partitioned into n blocks. We first consider the set L of all indices in $[n] \times [\ell]$ such that for every $(i,j) \in L$ it holds that $\mathbf{E}_{\sigma \leftarrow U_r}[\mathtt{I}_{i,j}(C_\sigma)] < 2B/n$, where $B = O(\log s)^{d-1}$ (as in Eq. (5)). Recalling that $\sum_{(i,j) \in [n]\times[\ell]} \mathbf{E}_{\sigma \leftarrow U_r}[\mathtt{I}_{i,j}(C_\sigma)] < B$, it follows that $|L| \geq n\ell - n/2$. We consider the set $L' \subseteq [n]$ of blocks such that $i \in L'$ if for every $j \in [\ell]$ it holds that $(i,j) \in L$. Then, $|L'| \geq n/2$. We now select an arbitrary set $I' \subseteq L'$ of size $\epsilon n / 2\ell B$, let $I = I' \times [\ell]$, and proceed as before, while noting that (as before) it holds that $\mathbf{E}_{\sigma \leftarrow U_r}[\sum_{(i,j) \in I} \mathtt{I}_{i,j}(C_\sigma)] < \epsilon$, since $|I| \cdot 2B/n = \epsilon$. That is, we fix I, $z = (z_1, ..., z_n) \in (\{0,1\}^\ell)^n$ and G (as before), and note that for every $\sigma \in G$ it holds that

$$\sum_{(i,j) \in I} \mathbf{Pr}_{x \leftarrow U_{n\ell}} \left[C_\sigma(x) \neq C_\sigma(x \oplus 0^{\mathtt{idx}(i,j)-1}10^{n-\mathtt{idx}(i,j)}) \middle| x_{[n]\setminus I} = z_{[n]\setminus I} \right] < \sqrt{\epsilon}, \tag{9}$$

where $\mathtt{idx}(i,j) = (i-1) \cdot \ell + j$. Now, consider an $(n, |I'|, \ell)$-block-fixing source $X = (X_1, ..., X_n)$ such that $X_i = z_i$ if $i \in [n] \setminus I'$ and X_i is random (i.e., distributed as U_ℓ) otherwise. Then, for every $\sigma \in G$ there exists a bit y_σ so that $\mathbf{Pr}[C_\sigma(X) = y_\sigma] > 1 - \sqrt{\epsilon}$, since otherwise Eq. (9) is contradicted. The claim follows as before, but note that $|I'| = |I|/\ell = \epsilon n / 2\ell B$. ■

Theorem 5.4 (impossibility of ordinary extraction in $\mathcal{AC}^0$): *Suppose that $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ is computable by $s(n)$-size circuits of depth $d = d(n)$, and let $m' = m - r$. Suppose that $\delta > 0$ satisfies $\binom{m}{\lfloor \delta m' \rfloor} < 1 + \delta \cdot 2^{m'}$, which is satisfied whenever either $\delta < 1/m'$ or $\delta \leq 1/3 \log m$. Then, if E is a $(1 - 2\delta - 2^{-m'})$-error extractor for $(n,k,1)$-block-fixing sources, then $k > \frac{\delta^3 m' n}{m \cdot O(\log s(n))^{d-1}}$. In particular:*

1. *If $m = r + 1$ (i.e., $m' = 1$) and E is a 0.499-error extractor for $(n,k,1)$-block-fixing sources, then $k > \frac{n}{m \cdot O(\log s(n))^{d-1}}$.*

2. *If $m = r + \Omega(r)$ (i.e., $m' = \Omega(r)$) and E is a 0.999-error extractor for $(n,k,1)$-block-fixing sources, then $k > n/O(\log s(n))^{d-1}$.*

Setting $\delta = \min(o(1), 1/3\log m)$, the general case implies that *if E is a $(1-2^{-m'}-o(1))$-error extractor for $(n,k,1)$-block-fixing sources, then $k > \frac{m'n}{\widetilde{O}(m)\cdot O(\log s(n))^{d-1}}$*. Theorem 5.4 generalizes Theorem 1.1, which only refers to the case of $m' = \Omega(r)$. Another important advantage of Theorem 5.4 over Theorem 1.1 is that it refers to a much more restricted class of sources (i.e., $(n,k,1)$-block-fixing sources rather than (n,k)-sources).[19]

Proof: The proof is very similar to the proof of Theorem 5.3, except that we consider $2^r \cdot m$ residual circuits $C_{\sigma,j}$ such that $C_{\sigma,j}(x)$ computes the $j^{\rm th}$ bit of $E(x,\sigma)$. Specifically, we again derive a set of $\epsilon n/B$ indices $I \subseteq [n]$ and a string $z \in \{0,1\}^n$ such that

$$\mathbf{E}_{\sigma\leftarrow U_r, j\in_R[m]}\left[\sum_{i\in I}\mathbf{Pr}_{x\leftarrow U_n}\left[C_{\sigma,j}(x) \neq C_{\sigma,j}(x\oplus 0^{i-1}10^{n-i})\Big| x_{[n]\setminus I} = z_{[n]\setminus I}\right]\right] < \epsilon, \qquad (10)$$

where $j \in_R [m]$ denotes that j is distributed uniformly in $[m]$. We shall again fix I and z as above, and consider an $(n,|I|,1)$-block-fixing source $X = (X_1, ..., X_n)$ such that $X_i = z_i$ if $i \in [n]\setminus I$ and X_i is random otherwise. We set $\epsilon = \delta^3 m'/m$, and denote $C_\sigma(x) = E(x,\sigma)$. Now, letting $\texttt{dist}(y,z)$ denote the Hamming distance between the m-bit long strings y and z (i.e., $\texttt{dist}(y_1\cdots y_m, z_1\cdots z_m) = |\{i\in[m] : y_i \neq z_i\}|$), we get

$$\mathbf{E}_{\sigma\leftarrow U_r}\left[\sum_{i\in I}\mathbf{E}_{x\leftarrow X}\left[\texttt{dist}(C_\sigma(x), C_\sigma(x\oplus 0^{i-1}10^{n-i}))\right]\right] < m\cdot\epsilon = \delta^3 m'. \qquad (11)$$

Hence, there exists a set $G \subseteq \{0,1\}^r$ of density at least $1-\delta$ such that for every $\sigma \in G$ it holds that

$$\sum_{i\in I}\mathbf{E}_{x\leftarrow X}\left[\texttt{dist}(C_\sigma(x), C_\sigma(x\oplus 0^{i-1}10^{n-i}))\right] < \delta^2 m'. \qquad (12)$$

Then, for every $\sigma \in G$ there exists a string $y_\sigma \in \{0,1\}^m$ so that $\mathbf{E}[\texttt{dist}(C_\sigma(X), y_\sigma)] < \delta^2 m'$, because for two independent samples x and x' drawn from X, it holds that

$$\begin{aligned}\mathbf{E}_{x,x'}\left[\texttt{dist}(C_\sigma(x), C_\sigma(x')\right] &\le \max_{s\in\{0,1\}^n : s_{[n]\setminus I} = 0^{n-|I|}}\Big\{\mathbf{E}_{x\leftarrow X}\left[\texttt{dist}(C_\sigma(x), C_\sigma(x\oplus s)\right]\Big\} \\ &\le \max_{s : s_{[n]\setminus I} = 0^{n-|I|}}\left\{\sum_{i:s_i=1}\mathbf{E}_{x\leftarrow X}\left[\texttt{dist}(C_\sigma(x), C_\sigma(x\oplus 0^{i-1}10^{n-i}))\right]\right\}\end{aligned}$$

which (by Eq. (12)) is smaller than $\delta^2 m'$. Hence, for every $\sigma \in G$, it holds that $\mathbf{Pr}[\texttt{dist}(C_\sigma(X), y_\sigma) \le \delta m'] \ge 1-\delta$. Defining $S = \{y_\sigma : \sigma \in G\}$, we note that, with probability at least $(1-\delta)^2$, the Hamming distance between $E(X,U_r)$ and S (i.e., the distance to the closest string in S) is at most $\lfloor \delta m'\rfloor$. On the other hand, the probability that U_m is at Hamming distance at most $\lfloor\delta m'\rfloor$ from S is upper bounded by $\frac{\binom{m}{\lfloor\delta m'\rfloor}\cdot|S|}{2^m} < 2^{-m'} + 4\delta$, since $|S| \le 2^r = 2^{m-m'}$ and $\binom{m}{\lfloor\delta m'\rfloor} < 1+\delta\cdot 2^{m'}$. It follows that $\mathbf{\Delta}[E(X,U_r)\,;U_m] > (1-\delta)-(2^{-m'}+\delta)$, whereas X has $\delta^3 m'n/mB$ random bits. ■

[19] We mention that the proceeding version of [56] contains a result that is more closely related to Theorem 5.4: The (n,k)-source used in the proof there also belongs to a very restricted class; specifically, the source is a randomized process that produces an output by starting with 0^n and taking $k/36$ steps such that at each step a random position is selected (with repetitions) and its value is flipped.

5.3 Deterministic extractors

Recall that the bit-fixing model allows for deterministic extractors (e.g., $E(x) = \oplus_{i\in[n]}x_i$), which work for even lower min-entropy rate than those that are impossible for $\mathcal{AC}^0$, but indeed these extractors are not computable in $\mathcal{AC}^0$. Still, it is possible that whenever extraction in $\mathcal{AC}^0$ is possible for bit-fixing sources, this is also possible via deterministic extractors. We show that this is indeed the case. Our proof proceeds in three steps: First, we present $\mathcal{AC}^0$ circuits that extract a single bit (see Theorem 5.7), next we use them to present $\mathcal{AC}^0$ circuits that extract poly-logarithmically many bits (see Theorem 5.11), and finally we combine these with the extractor of Theorem 5.2 to extract $n/\text{poly}(\log n)$ bits.

Recall that a deterministic extractor (a.k.a seedless extractor) is one that gets no seed (i.e., has seed length $r(n) \equiv 0$). By definition, any deterministic extractor is strong. An obvious deterministic extractor for (n, k)-bit-fixing sources, which is implementable in $\mathcal{AC}^0$ when $k(n) \geq n - \text{poly}(\log n)$, is $E(x) = \oplus_{i\in[n-k(n)+1]}x_i$. A first indication that one can do much better is provided by Ajtai and Linial's non-explicit construction of $\mathcal{AC}^0$ circuits in which "large sets have small influence" [4, Sec. 5]. As shown in [40, Lem. 6.1], such circuits are deterministic extractors for *non-oblivious bit-fixing sources.*

Definition 5.5 (non-oblivious bit-fixing sources [40]): *A* non-oblivious (n, k)-bit-fixing source *is a sequence of random variables $X = (X_1, ..., X_n) \in \{0, 1\}^n$ such that there exists a set of at least k indices $I \subseteq [n]$ and a sequence of functions $f_1, ..., f_n : \{0, 1\}^{|I|} \to \{0, 1\}$ such that X_I is uniformly distributed over $\{0, 1\}^{|I|}$ and $X_i = f_i(X_I)$ for every $i \in [n] \setminus I$.*

Bit-fixing sources as in Definition 5.1 are a special case in which the f_i's are constant functions. Such sources are sometimes called *oblivious bit-fixing sources.* Clearly, any ϵ-extractor for non-oblivious (n, k)-bit-fixing sources is an ϵ-extractor for (oblivious) (n, k)-bit-fixing sources.

Theorem 5.6 (deterministic extraction in $\mathcal{AC}^0$ for non-oblivious bit-fixing sources [4, Sec. 5]): *There exist $\mathcal{AC}^0$ circuits $C : \{0, 1\}^n \to \{0, 1\}$ such that, for every k, it holds that C is a deterministic $O((n - k) \cdot \log^2 n)$-error extractors for non-oblivious (n, k)-bit-fixing sources.*

As shown in [40, Lem. 6.1&6.2], the error probability of extractors of the foregoing type is captured by the notion of *influence of sets.* For a Boolean function $f : \{0, 1\}^n \to \{0, 1\}$ and a set $S \subset [n]$, the influence of S on f, denoted $\mathtt{I}_S(f)$, is defined as the maximum of $\mathbf{Pr}_{x\leftarrow U_n}[f(x) \neq f(g(x))]$, taken over all functions $g : \{0, 1\}^n \to \{0, 1\}^n$ that satisfy $g(x)_{[n]\setminus S} = x_{[n]\setminus S}$ for every $x \in \{0, 1\}^n$ (i.e., g only changes the values of x at location in S and does so depending on the entire input).[20] Ajtai and Linial [4, Sec. 5] proved that there exist balanced $\mathcal{AC}^0$ circuits $C : \{0, 1\}^n \to \{0, 1\}$ such that the influence of every set of density ρ on C is $O(\log^2 n) \cdot \rho$, where f is balanced if $\mathbf{Pr}[f(U_n) = 1] = 0.5$. Hence, these circuits are deterministic $O(\rho \log^2 n)$-error extractors for the corresponding set of sources (i.e., non-oblivious $(n, n - \rho n)$-bit-fixing sources).

Theorem 5.6 is meaningful only for min-entropy rate approaching 1; that is, it is only meaningful for non-oblivious (n, k)-bit-fixing sources with $k \geq n - O(n/\log^2 n)$. This is not an artifact of the proof (nor even of the fact that the extractor is in $\mathcal{AC}^0$): As shown by Kahn, Kalai, and Linial [39] any deterministic ϵ-extractor for *non-oblivious* (n, k)-bit-fixing sources must satisfy $k \geq$

[20] The influence of a single variable, as defined in Eq. (6), is a special case: Indeed, when considering the influence of the variable i, it suffices to consider $g(x) = x \oplus 0^{i-1}10^{n-i}$.

$n - \Omega(\epsilon^{-1}n/\log n)$. However, for *oblivious* (n,k)-bit-fixing sources, one can achieve a min-entropy rate that approaches 0; that is, a deterministic $\mathcal{AC}^0$-extractor for oblivious (n,k)-bit-fixing sources with any $k \geq n/\text{poly}(\log n)$.

Theorem 5.7 (deterministic extraction in $\mathcal{AC}^0$ for bit-fixing sources): *For every $k(n) \geq n/\text{poly}(\log n)$ and every $\epsilon(n) > 1/\text{poly}(\log n)$, there exist deterministic ϵ-error extractors $E : \{0,1\}^n \to \{0,1\}$ for (n,k)-bit-fixing sources such that the extractors are computable in $\mathcal{AC}^0$.*

Our proof of Theorem 5.7 builds upon Theorem 5.6, and thus inherits the non-uniformity of the latter. In addition, our own reduction of Theorem 5.7 to Theorem 5.6 is non-explicit, due to our use of a probabilistic analysis of the rank of rectangular matrices [9].

Proof: Our starting point is the extractor of non-oblivious bit-fixing sources asserted in Theorem 5.6. Denoting the corresponding $\mathcal{AC}^0$ circuit by $C : \{0,1\}^n \to \{0,1\}$, our plan is to construct a new circuit $C' : \{0,1\}^{n'} \to \{0,1\}$, where $n' = \widetilde{O}(n)$, by feeding each input of C with the parity of a random subset of $\text{poly}(\log n)$ inputs of C'. Specifically, $C'(x) = C(L_1(x), ..., L_n(x))$, where each L_i is a random linear function generated by selecting each x_i with probability $p = \text{poly}(\log n)/n$. Indeed, these L_i's can be computed by a constant-depth circuits of size $\text{poly}(n)$.

Suppose that, for any choice V of $\delta n' = k(n') \geq n$ variables (i.e., x_i's), at least $n - \rho \cdot n$ of the linear functions (i.e.,. L_i's) are linearly independent as functions in the variables in V. Then, any fixing of $n' - k(n')$ of the x-variables, leaves at least $n - \rho n$ of the functions (i.e., the L_i's) linearly independent, which means that assigning random values to $k(n')$ of the inputs of C' (and setting the rest arbitrarily but obliviously of the random values) yields a random assignment to $n - \rho n$ of the inputs of C.

In anticipation of considering the influence of sets of ρn inputs on C, we set $\rho = \epsilon/\Theta(\log n)^2 = 1/\text{poly}(\log n)$, which guarantees that the influence of such sets is at most ϵ. Next, we set $m = \rho \cdot n$ and $n' = \widetilde{O}(n)$, and seek a sparse n-by-n' Boolean matrix M such that (1) each row of M has at most $\text{poly}(\log n)$ one-entries, and (2) any n-by-n sub-matrix of M has rank at least $n - m$. If we had such a matrix, then we can let its rows serve as the L_i's and be done (for $k(n') = n = n'/\text{poly}(\log n')$).

While we believe that such a matrix exists, we were not able to prove this conjecture. Instead, we take a somewhat longer route. First, we construct a matrix with properties as above over a finite field of quasi-polynomial (in n) size. Then, we use this construction to get a matrix with such properties for a finite field of poly-logarithmic (in n) size. Lastly, we show how to use the latter in our context. These three steps are captured by the following three claims.

Claim 5.7.1 (a desired matrix over a finite field of quasi-polynomial size): *For any $m \in [n/\text{poly}(\log n), n]$ and $n' \in [\Omega(n \log^2 n), \widetilde{O}(n)]$, and every finite field F of cardinality $q \geq \exp(n'/m)$, there exist an n-by-n' matrix M over F that satisfies the following two properties.*

1. *Each row of M has at most $\text{poly}(\log n)$ non-zero entries.*
2. *Each n-by-n sub-matrix of M has rank at least $n - m$.*

Proof: Setting $p = (\log n)/n$, consider selecting a random sparse n-by-n' matrix M over F by setting each entry to 0 with probability $1 - p$, and letting it be a uniformly distributed nonzero value otherwise. (The choices for the various entries are independent of one another.) Then, with probability at least $1 - n \cdot \exp(-\Omega(pn')) \geq 1 - \exp(-\Omega(\log n)^3)$, each row of M has $\Theta(pn') = \text{poly}(\log n)$ non-zero entries.

In proving the second property, we use a result of Blomer *et al.* [9] that asserts that an n-by-n matrix distributed as above has rank smaller than $n-m$ with probability $O(q^{-m})$. Applying a union bound, we infer that the second property fails with probability at most

$$\begin{aligned}\binom{n'}{n} \cdot O(q^{-m}) &< (n^2)^n \cdot \exp(-\Omega(n\log^2 n)/m)^m \\ &= \exp(O(n\log n)) \cdot \exp(-\Omega(n\log^2 n)),\end{aligned}$$

where the inequality uses $\log q \geq \Omega(n'/m) = \Omega(n\log^2 n)/m$. The claim follows. ■

Claim 5.7.2 (a desired matrix over a finite field of poly-logarithmic size): *For any $m \in [n/\text{poly}(\log n), o(n)]$ and $n' \in [\Omega(n\log^2 n), \widetilde{O}(n)]$, every finite field F' of cardinality $q' = \text{poly}(n'/m)$, and $n'' = \text{poly}(q) \cdot n'$, there exist an n-by-n'' matrix M' over F' that satisfies the following two properties.*

1. *Each row of M' has at most* $\text{poly}(\log n)$ *non-zero entries.*

2. *Each n-by-$(2n/n') \cdot n''$ sub-matrix of M' has rank at least $n-m$.*

Proof: Let F be a finite field of size $\exp(\Theta(n'/m))$ and M be a matrix as guaranteed by Claim 5.7.1. Let $\ell = \lceil \log |F| \rceil = \Theta(n'/m) = \text{poly}(\log n)$. For some $q' = \text{poly}(n'/m)$ and $\ell' = \text{poly}(q')$, consider a *linear* error correcting code mapping $\text{GF}(q')^\ell$ to $\text{GF}(q')^{\ell'}$ such that this code has relative distance at least $1 - (n/n')$. (For example, the Reed-Solomon code of degree ℓ over $\text{GF}(q')$ uses $\ell' = q'$ and has relative distance $1-(\ell/q')$, whereas $\ell/q' < n/n'$ provided that $q' \geq (n'/m)^2$.) Now, encode each element of F (viewed as an ℓ-long sequence over $\text{GF}(q')$) by the corresponding codeword, obtaining an n-by-$n'\ell'$ matrix M' over $F' = \text{GF}(q')$. We may assume that F' is a sub-field of F; in fact, we should pick F to satisfy this condition (as well as the other conditions stated above).

Letting $n'' = n' \cdot \ell' = \text{poly}(\log n) \cdot n$, note that the number of nonzero entries in each row of M' is at most $\text{poly}(\log n) \cdot \ell' = \text{poly}(\log n)$, since each row of M has at most $\text{poly}(\log n)$ non-zeros. This establishes the first property of M'.

To establish the second property of M', consider an arbitrary choice of $(2n/n') \cdot (n'\ell') = 2n\ell'$ columns of M', and let M'' denote the corresponding sub-matrix. Considering the partition of the columns of M' into n' blocks of length ℓ' (each encoding a symbol of F), we infer that at least n of these blocks contain more than $\ell' n/n'$ chosen columns. Let us call these blocks heavy. Recalling that the linear code has absolute distance $\ell' - (\ell' n/n')$, we infer that any linear combination of the rows of the sub-matrix M'' that yields the zero vector must yield zero in the column of M that corresponds to each of the heavy blocks (because a codewords with more than $\ell' n/n'$ zeros must encode the zero of F (viewed as the all-zero sequence of $\text{GF}(q')^\ell$)). Recalling that there are n heavy blocks, it follows that an F'-linear combination of the rows of M'' that yields the zero vector must yield zero in at least n columns of M. Using the second property of M, it follows that this F'-linear combination must contain more than $n-m$ rows, and the claim follows. ■

Construction 5.7.3 (a kind of condenser):[21] *Let m, n', q', n'' and M' be as in Claim 5.7.2, and suppose that q' is a power of two. Let $\delta = 4n/n'$ and $s = \Theta((q'\log n)^2/\delta)$, and consider the following transformation of $(z_{1,1}, ..., z_{n'',s}) \in \{0,1\}^{n''s}$ into an n-bit long string $x = (x_1, ..., x_n)$.*

[21] For any $\delta \geq 1/\text{poly}(\log n)$ and any $\rho \geq 1/\text{poly}(\log n)$, this construction "condenses" an $(n''s, \delta n''s)$-bit-fixing source, into a *non-oblivious* $(n, (1-\rho) \cdot n)$-bit-fixing source. Thus, the min-entropy rate is significantly increased (from δ to $1-\rho$), but the output source belong to a wider class of sources.

1. *For each $i \in [n'']$, compute $z_i = \sum_{j\in[s]} z_{i,j} \bmod q'$.*

   Viewing each z_i as an element of $\mathrm{GF}(q') \equiv \mathbb{Z}_{q'}$, let $z = (z_1, ..., z_{n''}) \in \mathrm{GF}(q')^{n''}$.

2. *Compute $(y_1, ..., y_n) = M'z \in \mathrm{GF}(q')^n$. For each $i \in [n]$, let x_i be the result of applying some balanced predicate to y_i* (e.g., x_i is the least significant bit of y_i)*.*[22]

Note that these computation can be carried out by constant-depth circuits of size $\mathrm{poly}(n)$*, since* $q' = \mathrm{poly}(\log n)$*,* $n' = \widetilde{O}(n)$ *and each row of* M' *has at most* $\mathrm{poly}(\log n)$ *non-zero entries.*

Claim 5.7.4 (analysis of Construction 5.7.3): *If the input to Construction 5.7.3 is taken from an $(n''s, \delta n''s)$-bit-fixing source, then there exist $n - m$ bits in the output with a joint distribution that is $n^{-\omega(1)}$-close to U_{n-m}.*

Proof: If a δ fraction of the input bits are random, then for at least a $\delta/2$ fraction of $i \in [n'']$, called good, at least a $\delta/2$ fraction of the bits $z_{i,1}, ..., z_{i,s}$ are random. (Recall that random bits are independent of one another, whereas the other bits are fixed.) As shown by Kamp and Zuckerman [40], the sum modulo q' of the bits of a $(s, \delta' s)$-bit-fixing source is $\exp(-\Omega(\delta' s/(q')^2))$-close to the uniform distribution on $\mathbb{Z}_{q'}$. Hence, for each good i it holds that z_i is $\exp(-\Omega(\delta s/(q')^2))$-close to the uniform distribution on $\mathbb{Z}_{q'}$ (and the z_i's are independent of one another). By the choice of $s = \Omega(q' \log n)^2/\delta$, it follows that z_I is $\exp(-\Omega(\log^2 n))$-close to be uniform over $\mathrm{GF}(q')^{|I|}$, where I denotes the set of good i's and $|I| \geq \delta n''/2$.

Consider the column of M' that correspond to the good i's. By the second property of M' (and using $\delta = 4n/n'$), it holds that the rank of the corresponding n-by-$(2n/n') \cdot n''$ sub-matrix is at least $n - m$. Denoting a set of $n - m$ linearly independent rows by R, it follows that y_R is $\exp(-\Omega(\log^2 n))$-close to be uniform over $\mathrm{GF}(q')^{n-m}$, and a corresponding statement holds for x_R with respect to $\{0,1\}^{n-m}$. The claim follows. ■

Wrapping up. The final construction is as follows. Let $C : \{0,1\}^n \to \{0,1\}$ be the $\mathcal{AC}^0$ circuit guaranteed by Theorem 5.6. Setting $\rho = \epsilon/O(\log n)^2 \geq 1/\mathrm{poly}(\log n)$ and $\delta = k(n^2)/n^2 \geq 1/\mathrm{poly}(\log n)$, we use $n'' = 4n/\delta = \widetilde{O}(n)$ and invoke Construction 5.7.3. Let $C'' : \{0,1\}^{n''s} \to \{0,1\}^n$ be the $\mathcal{AC}^0$ circuit provided by Construction 5.7.3, and note that $\delta \leq k(n''s)/n''s$ (since we may assume that $k(t)/t$ is non-increasing in t). We define the final circuit $C' : \{0,1\}^{n''s} \to \{0,1\}$ as the composition of C and C''; that is, $C'(z) = C(C''(z))$. Using Claim 5.7.4, we infer that for any $(n''s, \delta n''s)$-bit-fixing source Z, it holds that $C''(Z)$ is $n^{-\omega(1)}$-close to a source in which $n - \rho n$ of the bits are totally random (and the rest may be determined as a function of them). By Theorem 5.6, in this case $C(C''(Z))$ is ϵ-close to a random bit.[23] Hence, C' is an ϵ-error extractor for for $(n''s, k(n''s))$-bit-fixing sources, which establishes the claim of the theorem. ■

Remark 5.8 (Construction 5.7.3, revisited): *While the output of Construction 5.7.3 is* not *an affine combination of its input bits, it is $n^{-\omega(1)}$-close to an affine source of min-entropy at least $n - m$* (cf. [24])*. This is due to the following two facts:*

1. *The vector $(z_1, ..., z_{n''}) \in (\{0,1\}^{\ell''})^{n''}$ produced in Step 1 is $n^{-\omega(1)}$-close to a $(n'', \delta n''/2, \ell'')$-block-fixing source, where $2^{\ell''} = q'$.*

[22] For this reason we need q' to be even.

[23] Recall that $\epsilon(n) \geq 1/\mathrm{poly}(\log n)$, which is much larger than the $n^{-\omega(1)}$ deviation created by C''.

2. *The vector x is a* GF(2)*-linear combination of the bits of z, since y is a* GF$(2^{\ell''})$*-linear combination of the blocks of z* (viewed as elements of GF$(2^{\ell''})$).

Hence, the bits of x are affine combinations of the non-fixed bits of z', where z' is the $(n'', \delta n''/2, \ell'')$-block-fixing source that is $n^{-\omega(1)}$-close to $(z_1, ..., z_{n''})$. Since the affine transformation of z' to x has rank at least $n - m$ in the non-fixed variables of z', our claim follows.

We now improve the construction asserted by Theorem 5.7 in two ways. First we show that the error of the extraction can be reduced (to a negligible in n level), and then we show that poly$(\log n)$ bits can be extracted (rather than a single one). We start by observing that XORing values extracted from disjoint portions of a bit-fixing source yields an extractor of smaller error (alas it is guaranteed to work only for bit-fixing sources of a larger amount of min-entropy).[24]

Theorem 5.9 (error reduction for deterministic extraction from bit-fixing sources): *Suppose that $E : \{0,1\}^n \to \{0,1\}$ is an ϵ-error extractor for (n,k)-bit-fixing sources. Then, for every $t \in \mathbb{N}$, the function $E' : \{0,1\}^{tn} \to \{0,1\}$, given by $E'(x_1, ..., x_t) = \oplus_{i\in[t]} E(x_i)$ is an $\epsilon^{\lceil tk/n \rceil}$-error extractor for $(tn, 2tk)$-bit-fixing sources.*

Indeed, as detailed in Corollary 5.10 below, applying Theorem 5.9 to Theorem 5.7 yields a similar deterministic $\mathcal{AC}^0$ extractor but for error rates that are smaller than $1/\text{poly}(n)$.

Proof: Letting $\delta = k/n$, we note that the existence of $2\delta tn$ random bits in the source $(X_1, ..., X_t) \in (\{0,1\}^n)^t$ implies that for at least a δ fraction of the indices $i \in [t]$ the source X_i is a $(n, \delta n)$-bit-fixing source (whereas the t sources are independent of one another). Since each of these $\lceil \delta t \rceil$ extractions yields a bit that is ϵ-close to uniform (whereas the t bits are independent of one another), the claim follows. ■

Corollary 5.10 (improved error in deterministic $\mathcal{AC}^0$-extractors for bit-fixing sources): *For every $k(n) \geq n/\text{poly}(\log n)$ and every $\epsilon(n) \geq \exp(-\text{poly}(\log n))$, there exist deterministic ϵ-error extractors $E : \{0,1\}^n \to \{0,1\}$ for (n,k)-bit-fixing sources such that the extractors are computable in $\mathcal{AC}^0$.*

Proof: Let $E_0 : \{0,1\}^{n_0} \to \{0,1\}$ be the 0.1-error extractor for $(n_0, k(k_0)/2)$-bit-fixing sources provided by Theorem 5.7. Letting $t = \Theta((n_0/k(n_0)) \log(1/\epsilon(n_0))) = \text{poly}(\log n_0)$ and $n = t \cdot n_0$, and applying Theorem 5.9, the claim follows, since t-wise XOR can be computed by constant-depth circuits of poly(n)-size. ■

Extracting slightly more bits. Applying the same idea as underlying the proof of Theorem 5.9, we can extract poly-logarithmically many bits rather than one.

Theorem 5.11 (Corollary 5.10, revisited): *For every $k(n) \geq n/\text{poly}(\log n)$ and every $\epsilon(n) \geq \exp(-\text{poly}(\log n))$, there exist deterministic ϵ-error extractors $E : \{0,1\}^n \to \{0,1\}^{\text{poly}(\log n)}$ for (n,k)-bit-fixing sources such that the extractors are computable in $\mathcal{AC}^0$.*

[24] Indeed, there is a trade-off, which we do not elaborate, between the amount of error reduction and the increase in the required min-entropy.

Proof: We proceed in two steps, first obtaining an extractor that outputs double-logarithmically many bits, and next using it to establish the claim of the theorem. As in the proof of Corollary 5.10 (and other results), we shall use the fact that t-wise sums can be computed by constant-depth circuits of $\exp(t^c)$-size, for any constant $c > 0$.

Claim 5.11.1 (Corollary 5.10, revisited): *For every $k(n) \geq n/\text{poly}(\log n)$, $\epsilon(n) \geq \exp(-\text{poly}(\log n))$, and $\ell_1(n) = O(\log\log n)$, there exist deterministic ϵ-error extractors $E : \{0,1\}^n \to \{0,1\}^{\ell_1(n)}$ for (n,k)-bit-fixing sources such that the extractors are computable in $\mathcal{AC}^0$.*

Proof: Letting $\delta = k(n^2)/n^2$ and $t = 2^{2\ell_1}\delta^{-1}\log(1/\epsilon(n^2)) = \text{poly}(\log n)$, we note that the existence of δtn random bits in the source $(X_1, ..., X_t) \in (\{0,1\}^n)^t$ implies that for at least a $\delta/2$ fraction of the indices $i \in [t]$ the source X_i is a $(n, \delta n/2)$-bit-fixing source (whereas the t sources are independent of one another). Applying the extractor of Corollary 5.10 to each of the t sources, we obtain a $(t, \delta t/2)$-bit-fixing source (or rather a t-bit string that is $\exp(-\text{poly}(\log n))$-close to such a source). Computing the sum of these t outputs modulo 2^{ℓ_1} (and invoking again the result of Kamp and Zuckerman [40]), we obtain a value that is $\exp(-\Omega(\delta t/2^{2\ell_1}))$-close (i.e., $\epsilon/2$-close) to uniform over $\mathbb{Z}_{2^{\ell_1}}$. This yields an ϵ-error extractor of ℓ_1 bits for $(tn, \delta tn)$-bit-fixing sources, and the claim follows. ■

Extracting $\text{poly}(\log n)$ many bits. For any $\ell(n) = \text{poly}(\log n)$, we will use an 0.1-biased sample space $S \subset \{0,1\}^\ell$ of size $\text{poly}(\ell)$ (cf., e.g., [26, Sec. 8.5.2]), and let $\ell_1 = \log|S| = O(\log\log n)$. For any $\epsilon(n) \geq \exp(-\text{poly}(\log n))$, and $k(n) \geq n/\text{poly}(\log n)$, we again let $\delta = k/n$. This time we set $t = \Theta(\ell + \log(1/\epsilon))/\delta = \text{poly}(\log n)$, and consider a $(tn, \delta tn)$-bit-fixing source, denoted $(X_1, ..., X_t) \in (\{0,1\}^n)^t$, inferring that at least $\delta/2$ fraction of the n-bit long X_i's are $(n, \delta n/2)$-bit-fixing sources. Applying the extractor of Claim 5.11.1 to each of these t sources, we obtain t strings, each of length ℓ_1, such that at least $\delta t/2$ of these strings are ϵ-close to being uniformly distributed in $\{0,1\}^{\ell_1}$. In other words, we obtain a $(t, \delta t/2, \ell_1)$-block fixing source. Denoting the i^{th} block in this source by y_i, and viewing it as an element of $[2^{\ell_1}] \equiv \{0,1\}^{\ell_1}$, we just output $\oplus_{i\in[t]} s_{y_i}$, where s_j is the j^{th} string in S. This ℓ-bit output is $0.1^{\delta t/2}$-biased, since for any $\alpha \in \{0,1\}^\ell$ it holds that $\langle \alpha, \oplus_{i\in[t]} s_{y_i}\rangle_2 = \oplus_{i\in[t]}\langle \alpha, s_{y_i}\rangle_2$, where $\langle\cdot,\cdot\rangle_2$ denotes inner product mod 2. It follows that the output is $(2^{\ell/2} \cdot 0.1^{\delta t/2})$-close to the uniform distribution over $\{0,1\}^\ell$, and the theorem follows (using sufficiently large $t = O(\ell + \log(1/\epsilon))/\delta$). ■

Extracting many more bits. Using the composition theorem of Gabizon *et al.* [25, Thm. 7.1], we obtain deterministic extractors that are computable in $\mathcal{AC}^0$ and extract $n/\text{poly}(\log n)$ bits. The aforementioned composition result requires three ingredients: (1) an adequate deterministic extractor (provided by Theorem 5.11), (2) an adequate seeded extractor (provided by Theorem 5.2), and (3) an adequate averaging sampler (provided by Theorem 3.6). We shall rely on the fact that the composition theorem of [25, Thm. 7.1] preserves $\mathcal{AC}^0$ complexity.

Theorem 5.12 (deterministic $\mathcal{AC}^0$ extraction of $n/\text{poly}(\log n)$ bits from bit-fixing sources): *For every $k(n) \geq n/\text{poly}(\log n)$ and every $\epsilon(n) \geq 1/\text{poly}(n)$, there exist deterministic ϵ-error extractors $E : \{0,1\}^n \to \{0,1\}^{n/\text{poly}(\log n)}$ for (n,k)-bit-fixing sources such that the extractors are computable in $\mathcal{AC}^0$.*

Proof: We start by reviewing the composition theorem of Gabizon *et al.* [25, Thm. 7.1], which is pivotal to our proof. The composition theorem of Gabizon *et al.* [25, Thm. 7.1] requires three ingredients (for some parameters $\mu, \mu', \epsilon', \epsilon''$ etc):[25]

1. A deterministic ϵ'-error extractor $E' : \{0,1\}^n \to \{0,1\}^{r+r''}$ for $(n, \mu' t)$-bit-fixing sources;

2. A seeded ϵ''-error extractor $E'' : \{0,1\}^n \times \{0,1\}^{r''} \to \{0,1\}^m$ for $(n, \mu n - t)$-bit-fixing sources;

3. An (μ, μ', γ)-averaging sampler $S : \{0,1\}^r \to [n]^t$.

It yields a deterministic ϵ-error extractor $E : \{0,1\}^n \to \{0,1\}^m$ for $(n, \mu n)$-bit-fixing sources and $\epsilon = \epsilon'' + 2^{r+3} \cdot \epsilon' + 3\gamma$ that operates as follows. Denoting the first r (resp., last r'') bits of E' by $E'_1(x)$ (resp., $E'_2(x)$), it holds that $E(x) = E''(x_{[n]\setminus S(E'_1(x))} \circ 0^t, E'_2(x))$. Hence, if each of the three ingredients is computable by constant-depth poly(n)-size circuits, then E is in $\mathcal{AC}^0$.

Setting $\mu = k(n)/n = 1/\text{poly}(\log n)$, we shall use $\mu' = \mu/2$ and $t = k(n)/2$ so that $\mu' t = n/\text{poly}(\log n)$ and $\mu n - t = n/\text{poly}(\log n)$. Furthermore, we shall use $\epsilon' = 2^{-\log^3 n}$, $\epsilon'' = \gamma = 1/\text{poly}(n)$, $m = n/\text{poly}(\log n)$, $r'' = O(\log n)$ and $r = O(\log n)^2$. The required deterministic extractor is provided by Theorem 5.11, the seeded extractor is provided by Theorem 5.2, and the required averaging sampler is provided by Theorem 3.6. Hence we obtain $\mathcal{AC}^0$ circuits computing $E : \{0,1\}^n \to \{0,1\}^m$, which is a deterministic ϵ-error extractor for $\epsilon = 2^{r+3} \cdot \epsilon' + 1/\text{poly}(n) = 1/\text{poly}(n)$. $\blacksquare$

An explicit disperser. Recall that a deterministic disperser for a class of sources is a function that defines an onto mapping from the support of each source in the class to the range of the function. That is, $D : \{0,1\}^n \to \{0,1\}^m$ is a deterministic disperser for a class of sources if for every source X in the class and over every $v \in \{0,1\}^m$ it holds taht $\mathbf{Pr}[D(X)=v] > 0$.

Theorem 5.13 (deterministic disperser in $\mathcal{AC}^0$ for bit-fixing sources): *For every $k(n) \geq n/\text{poly}(\log n)$, there exist explicit $\mathcal{AC}^0$ circuits $D : \{0,1\}^n \to \{0,1\}$ that constitute a disperser for (n,k)-bit-fixing sources.*

Proof: The first observation is that the Hamming weight of the outcome of any (n,k)-bit-fixing source is spread (unevenly) over an interval of k values. Hence, if we partition $[n]$ into consecutive intervals of length $k/10$, then at least ten of these intervals will be assigned non-zero weight.[26] Our disperser outputs the least significant bit of the index of the interval in which the source's outcome resides. The second observation is that for outcomes that reside in the middle portion of the interval we can determine the relevant index by approximate counting. Details follow.

[25] Actually, we present a special case of [25, Thm. 7.1], whereas the original version refers to a generalization of the notion of an averaging sampler. Loosely speaking, a $(\mu, \mu', \mu'', \gamma)$-averaging sampler is defined as in Definition 2.6, except that it refers to functions $f : [n] \to [0,1]$ such that $\rho(f) = \mu$ and also requires that $\mathbf{Pr}[\sum_{i \in S(U_r)} f(i) > t \cdot \mu''] \leq \gamma$. Indeed, such an $(\mu, \mu', 1, \gamma)$-averaging sampler is an (μ, μ', γ)-averaging sampler as in Definition 2.6. We comment that using the original form of [25, Thm. 7.1] allows to obtain somewhat better parameters, by showing that a small variant on the construction establishing Theorem 3.6 yields the required $(\mu, \mu', \mu'', \gamma)$-averaging sampler. (Specifically, we shall set the parameters of the pairwise independent sampler, used in the latter proof, so that its error probability is $o(\mu)$ rather than a constant, and establish the assertion for $\mu'' = 2\ell \cdot \mu$ and $\gamma = 1/\text{poly}(n)$, where $\ell = O(\log(1/\gamma))$.)

[26] Indeed, it is likely that almost all the probability mass is concentrated in one interval (since the probability mass is concentrated in an interval of length $o(k^{2/3})$, but this does not contradict the foregoing.

Let $\rho = k/n = 1/\text{poly}(\log n)$ and $\epsilon = \rho/50$, and recall that Ajtai [2] (see also [58]) provided explicit $\mathcal{AC}^0$ circuits that compute the Hamming weight of n-bit strings up to an additive deviation of ϵn. Denoting this circuit by C, consider the circuit $C'(x)$ that computes $\lfloor C(x)/\epsilon n \rfloor \in \{0, 1, ..., 1/\epsilon\}$. Then, $C'(x) \in [\lfloor \text{wt}(x)/\epsilon n \rfloor \pm 2]$, where $\text{wt}(x) \stackrel{\text{def}}{=} \sum_{i\in[n]} x_i$. Hence, for every $v \in [10/\rho]$ it holds that

$$\sum_{i=5v}^{5v+4} \mathbf{Pr}[C'(X) = i] \geq \mathbf{Pr}\left[\lfloor \text{wt}(X)/\epsilon n \rfloor = 5v + 2\right],$$

which means that if the latter term is positive then so if the former. Our disperser D outputs the least significant bit of $\lfloor C'(x)/5 \rfloor$.

For an arbitrary (n, k)-bit fixing source X, let $u = \lfloor 50\mathbf{E}[\text{wt}(X)]/k \rfloor$. Then, for every $u' \in [u \pm 10]$, it holds that $\mathbf{Pr}[\lfloor 50\text{wt}(X)/k \rfloor = u'] > 0$. It follows that $\mathbf{Pr}[\lfloor C'(X)/5 \rfloor = \lfloor u/5 \rfloor] > 0$ and $\mathbf{Pr}[\lfloor C'(X)/5 \rfloor = \lfloor u/5 \rfloor \pm 1] > 0$, whereas in these two cases D outputs different values (since the least significant bit of $\lfloor u/5 \rfloor$ is different from the least significant bit of $\lfloor u/5 \rfloor \pm 1$). ■

5.4 Extraction from zero-fixing sources

The impossibility results regarding bit-fixing sources (presented in Section 5.2) do not hold for a restricted class of such sources that was recently introduced by Cohen and Shinkar [18]. This class, called zero-fixing sources, consists of bit-fixing sources in which all fixed bits are set to zero.

Definition 5.14 (zero-fixing sources [18]): *An (n, k)-zero-fixing source is a sequence of random variables $X = (X_1, ..., X_n) \in \{0, 1\}^n$ such that there exists a set of at least k indices $I \subseteq [n]$ such that X_I is uniformly distributed over $\{0, 1\}^{|I|}$ and $X_i = 0$ for every $i \in [n] \setminus I$.*

As shown next, there exist strong $\mathcal{AC}^0$-extractors for zero-fixing sources of *logarithmic min-entropy*, which was shown to be impossible for bit-fixing sources (see Theorem 5.3).

Theorem 5.15 (seeded $\mathcal{AC}^0$-extractor for zero-fixing sources): *Let $k, m : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0, 1]$ be such that $m(n) = O(\log n)$ and $k(n) = 2m(n) + O(\log(1/\epsilon)) \leq \text{poly}(\log n)$. Then, there exists a function $E : \{0, 1\}^n \times \{0, 1\}^{O(\log n)} \to \{0, 1\}^{m(n)}$ that is computable in uniform $\mathcal{AC}^0$ and constitutes a strong ϵ-error extractor for (n, k)-zero-fixing sources.*

Proof: On input $x \in \{0, 1\}^n$ and a seed $s \in \{0, 1\}^{O(\log n)}$, the proposed extractor operates in two stages. In the first stage, which is deterministic, the extractor determines the first $k = k(n)$ locations that are assigned the value 1 in $x' = (x'_1, ..., x'_{n+k}) \stackrel{\text{def}}{=} x1^k$; that is, it determines $i_1 < \cdots < i_k \leq n+k$ such that the i_k-bit long prefix of x' equals $0^{i_1-1}10^{i_2-i_1-1}1 \cdot 0^{i_k-i_{k-1}-1}1$ In other words, for each $j \in [k]$, the extractor determines i_j as the smallest i such that $\sum_{\ell\in[i]} x'_\ell = j$. (This is done using counters that count up to $k + 1 \leq \text{poly}(\log n)$, while applying such counters to the strings $x'_1 \cdots x'_i$, for $i \in [n + k]$.)

In the second stage, which uses the seed s, we apply a (k, ϵ)-extractor to the sequence $(i_1, ..., i_k)$, which is viewed as a string of length $k \log n$. Here, again, we can use the extractor of [28, Sec. 5], while relying on the fact that $k \log n = \text{poly}(\log n)$.

Note that if X is a (n, k)-zero-fixing extractor, then the sequence determined by the first stage has min-entropy at least k. To verify the claim, consider the set I of the first k non-fixed (i.e., random) bits in X. Then, the random set $\{i \in I : X_i = 1\}$ has min-entropy k, since each of the 2^k possible subsets is equally likely. ■

Deterministic extraction. While our deterministic $\mathcal{AC}^0$-extractors for bit-fixing sources are not explicit (see Section 5.3), we show a very simple explicit construction for the case of zero-fixing sources (of comparable min-entropy rate). In fact, we prove a stronger result.

Theorem 5.16 (deterministic $\mathcal{AC}^0$-extractors for zero-fixing sources):

1. *Let $k : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0,1]$ be such that $k(n) \geq n/\mathrm{poly}(\log n)$ and $\epsilon(n) \geq 1/\mathrm{poly}(n)$. Then, there exists a function $E : \{0,1\}^n \to \{0,1\}^{n/\mathrm{poly}(\log n)}$ that is computable in uniform $\mathcal{AC}^0$ and constitutes an ϵ-error extractor for (n,k)-zero-fixing sources.*

2. *Let $k : \mathbb{N} \to \mathbb{N}$ and $\epsilon : \mathbb{N} \to [0,1]$ be such that $k(n) = \Theta(\epsilon(n)^{-3} \log n) = \mathrm{poly}(\log n)$. Then, there exists a function $E : \{0,1\}^n \to \{0,1\}$ that is computable in uniform $\mathcal{AC}^0$ and constitutes an ϵ-error extractor for (n,k)-zero-fixing sources.*

Proof Sketch: Starting with Part 1 and letting $\delta = k(n)/n \geq 1/\mathrm{poly}(\log n)$, we first present a simple extractor that outputs a single bit (and later apply the transformations underlying the proofs of Theorems 5.11 and 5.12). We consider a partition of the source into consecutive $3t/\delta$-bit long blocks, where $t = \Theta(\epsilon_0^{-3} \log n)$ for any desired constant $\epsilon_0 > 0$. The extractor picks the first block that contains at least t ones, and outputs the parity of the bits in that block. (Recall that the block length is $3t/\delta = \mathrm{poly}(\log n)$.)

Towards the analysis, we fix an arbitrary (n,k)-zero-fixing source, and let k_i denote the number of random bits in the i^{th} block, where $i = 1, ..., \delta n/3t$. We consider corresponding random variables, $Y_1, ..., Y_{\delta n/3t}$, such that Y_i denotes the sum of the bits in the i^{th} block. Clearly, there exists a block i such that $k_i \geq 3t$. Note that if $k_i < t$, then the i^{th} block will never be selected. On the other hand, if $k_i \geq t$, then, with probability at least $1 - o(1/n)$, it holds that $Y_i = (0.5 \pm \epsilon_0) \cdot k_i$. Hence, with probability at least $1 - o(1)$, some block i is selected, and it holds that $k_i \geq (2 - 2\epsilon_0) \cdot t$. Furthermore, for such a block i (i.e., $k_i \geq 2t - 2\epsilon_0 t$), the parity of the bits in the block (i.e., $Y_i \bmod 2$) has bias $O(\epsilon_0)$, also when conditioned on $Y_i \geq t$. To verify the last claim consider a pairing of the odd-parity strings of length $k_i \geq 2t - 2\epsilon_0 t$ with the even-parity strings obtained by flipping the last bit. Now omit the strings that have Hamming weight smaller than t, and note that only an $O(\epsilon_0)$ fraction of the remaining strings are left unmatched (since this omission may leave unmatched only strings of Hamming weight t, whereas $\mathbf{Pr}[Y_i{=}t] = (1 + O(\epsilon_0))^j \cdot \mathbf{Pr}[Y_i{=}t+j]$ for every $j \in [\epsilon_0^{-1}]$).

Part 1 follows by applying the transformations that underlie the proofs of Theorems 5.9, 5.11 and 5.12. Each of these transformations improves some parameter (i.e., error or output length) of deterministic $\mathcal{AC}^0$-extractors for bit-fixing sources of min-entropy $n/\mathrm{poly}(\log n)$. The point is that these transformation only use the fact that certain sub-sources derived from the given bit-fixing source are bit-fixing sources with certain parameters (where sub-sources are projections of the source on some locations).[27] The same holds for zero-fixing sources.

Turning to Part 2, we consider the following extractor, which, for every $\ell = 0, 1, ..., \log(n/k(n))$, uses a family of n^2 pairwise independent hashing functions $h : [n] \to [2^\ell]$. On input $x \in \{0,1\}^n$, the extractor finds ℓ such that for most $h : [n] \to [2^\ell]$ it holds that $S_h \stackrel{\mathrm{def}}{=} \{i \in [n] : x_i{=}1 \,\&\, h(i){=}1\}$ has cardinality at least $(1 - \epsilon(n)) \cdot k(n)$ and at most $(2 + \epsilon(n)) \cdot k(n)$. It then picks the first of these (majority) h's, and outputs the parity of $|S_h|$. (This is done by using a counter that counts till $3k(n) = \mathrm{poly}(\log n)$; note that hashing functions in these families can be computed by depth-two circuits of size $\mathrm{poly}(n)$.)

[27] This holds also for [25, Thm. 7.1], which is used within the proof of Theorem 5.12.

Towards the analysis, we fix an arbitrary (n,k)-zero-fixing source X, and let I denote the set of random (i.e., non-fixed) bit locations in it. Then, there exists an ℓ such that $|I|$ is in $[2^{\ell+1}\cdot k, 2^{\ell+2}\cdot k)$, where $k=k(n)$, and a hashing function $h:[n]\to[2^\ell]$ such that $|\{i\in I: h(i)=1\}|\in(2k-\epsilon k, 4k+\epsilon)$, where $\epsilon=\epsilon(n)$. Using an analysis as in Part 1, we infer that with high probability the pair (ℓ,h) chosen by the extractor satisfies $|\{i\in I: h(i)=1\}|\in(2k-2\epsilon k, 4k+2\epsilon k)$, where here we use the fact that for the majority of the h's it holds that $|S_h|\in[(1-\epsilon)\cdot k, (2+\epsilon)\cdot k]$. Similarly, we conclude that, for the selected h, the parity of $|S_h|$ has bias $O(\epsilon)$, and Part 2 follows. ■

Remark 5.17 (zero-block-fixing sources, generalizing Definition 5.14): *An (n,k,ℓ)-*zero-block-fixing source *is a sequence of random variables $X=(X_1,...,X_n)\in\{0,1\}^{n\ell}$ such that there exists a set of at least k indices $I\subseteq[n]$ such that X_I is uniformly distributed over $\{0,1\}^{|I|\ell}$ and $X_i=0^\ell$ for every $i\in[n]\setminus I$. Extraction from such a source is quite easy, since each non-fixed block is clearly identified as such (and is uniformly distributed on $\{0,1\}^\ell\setminus\{0^\ell\}$).*

6 Extraction with long seeds

In this section we present extractors that use very long seeds, which is indeed uncustomary in the studies of extractors, but the point is that these extractors establish the tightness of Theorem 5.4. Recall that (Part 1 of) of Theorem 5.4 establishes lower bounds on the relationship between the min-entropy $k(n)$ and the seed-length $r(n)$ for any non-trivial extractor computable in $\mathcal{AC}^0$. Specifically, it asserts that $k(n)\cdot r(n)\geq n/\text{poly}(\log n)$ must hold (for any non-trivial extraction in $\mathcal{AC}^0$). The point of the current section is showing that these bounds are tight. Specifically, we shall show that $k(n)\cdot r(n)\geq n/\text{poly}(\log n)$ suffices for trivial extraction in $\mathcal{AC}^0$.

We shall first show (see Section 6.1) that $\mathcal{AC}^0$ circuits can extract $n+\text{poly}(\log n)$ bits using a seed of length n, provided that $k(n)\geq\text{poly}(\log n)$. Actually, we show that $r(n)+\text{poly}(\log n)$ bits can be extracted in $\mathcal{AC}^0$ provided that $k(n)\cdot r(n)\geq n/\text{poly}(\log n)$ (and $k(n)\geq\text{poly}(\log n)$). This is a special case of a more general result (presented in Section 6.2) that asserts that $r(n)+m'(n)$ bits can be extracted in $\mathcal{AC}^0$ provided that $k(n)\cdot r(n)\geq m'(n)\cdot n/\text{poly}(\log n)$ (and $m'(n)\leq k(n)/2$ and $k(n)\geq\text{poly}(\log n)$). The latter result is obtained by combining extractors presented in previous part of this write-up (including the one presented in Section 6.1) with a simple scheme that amounts to repeated extraction with independent seeds.

6.1 Extraction with a seed of linear length

The following construction generalizes an $\mathcal{AC}^0$-computable extractor presented by Viola [57, Lem. 4.3], where the original construction corresponded to the case $\ell=1$.

Theorem 6.1 (the inner product extractor): *Let $n,\ell\in\mathbb{N}$ such that $\ell/2$ is a power of 3 and $\ell\leq\text{poly}(\log n)$. Then, there exists a $(\ell+3\log(2/\epsilon),\epsilon)$-extractor, $E:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}^{n+\ell}$, computable by uniform $\mathcal{AC}^0$.*

The requirement that ℓ be of a special form (i.e., $\ell/2$ is a power of 3) can be dropped when $\ell=O(\log n)$. This special form is only used for asserting the existence of (uniform) circuits of constant depth and size poly(n) for computing inverses in GF(2^ℓ).

Proof: Our starting point is the strong extractor $E':\{0,1\}^n\times\{0,1\}^n\to\{0,1\}^\ell$ that views its input source x and seed r as n/ℓ-long sequences over GF(2^ℓ) and outputs their inner product, where the arithmetics is of the said field. That is, $E'(x,r) = \sum_{i\in[t]} x_i r_i$, where $t = n/\ell$, $x = (x_1, ..., x_t)$ and $r = (r_1, ..., r_t)$. Note that for a random r, the mapping $x \mapsto E'(x,r)$ is pairwise independent, and so E' is a strong $(\ell + 3\log(2/\epsilon), \epsilon)$-error extractor (by the Leftover Hashing Lemma, see, e.g., [26, Thm. D.4]). Hence, $E''(x,r) = r \circ E'(x,r)$ is an $(\ell+3\log(2/\epsilon), \epsilon)$-error extractor with output length $n + \ell$.

Wishing to obtain $\mathcal{AC}^0$ circuits, we define $E(x,s) = E''(x, f(x,s)) = (f(x,s), E'(x, f(x,s)))$, where $f(x,s) = (f_1(x,s), ..., f_t(x,s)) \in \mathrm{GF}(2^\ell)^t$ is defined as follows.

- Fictitiously define $s_0 = 0 \in \mathrm{GF}(2^\ell)$ and $x_0 = 1 \in \mathrm{GF}(2^\ell)$.
- Let $\mathrm{prv}_x(i) = j \in \{0,1,...,i-1\}$ denote the index of the last non-zero element that precedes i; that is, $\mathrm{prv}_x(i) = j \in \{0,1,...,i-1\}$ if $x_j \neq 0$ and $x_{j+1} = \cdots = x_{i-1} = 0$.
- Finally, define $f_i(x,s) = s_i$ if $x_i = 0$ and $f_i(x,s) = (s_i - s_{\mathrm{prv}_x(i)})/x_i$ otherwise.

In particular, $f(0^t, s) = s$ and $f_i(1^t, s) = s_i - s_{i-1}$ (for every $i \in [t]$). Note that for $\ell = 1$, we have $f_i(x,s) = s_i$ if $x_i = 0$ and $f_i(x,s) = s_i - s_{\mathrm{prv}_x(i)}$ otherwise.

Before showing that E is in $\mathcal{AC}^0$, we analyze its output distribution. Note that, for every fixed x, the mapping $s \mapsto f(x,s)$ is a bijection, and it follows that $f(x, U_n)$ is distributed identically to U_n. Hence, $E(x, U_n) = (f(x,U_n), E'(x, f(x,U_n)))$ is distributed identically to $E''(x, U_n) = (U_n, E'(x, U_n))$, since in both distributions the last ℓ bits are obtained by applying the same function (i.e., $E'(x,\cdot)$) to the first n bits. It follows that E'' is an $(\ell + 3\log(2/\epsilon), \epsilon)$-error extractor with output length $n + \ell$. It is left to show that E can be computed in uniform $\mathcal{AC}^0$.

Turning to the complexity of E, we first consider the computation of $E'(x, f(x,s))$. Note that $E'(0^t, r) = 0 \in \mathrm{GF}(2^\ell)$ for every r, whereas for $x \neq 0^t$, it holds that $E'(x, f(x,s))$ equals

$$\begin{aligned}
\sum_{i\in[t]} x_i f_i(x,s) &= \sum_{i:x_i\neq 0} x_i f_i(x,s) \\
&= \sum_{i:x_i\neq 0} x_i \cdot (s_i - s_{\mathrm{prv}_x(i)})/x_i \\
&= \sum_{i:x_i\neq 0} s_i - \sum_{i:x_i\neq 0} s_{\mathrm{prv}_x(i)}
\end{aligned}$$

which equals s_i for the largest i such that $x_i \neq 0$. Hence, $E'(x, f(x,s)) = \bigvee_{i\in[n]} \chi_i \cdot s_i$, where $\chi_i = ((x_i \neq 0) \wedge \bigwedge_{i<k\leq n}(x_k = 0))$ and $\sigma \cdot (\tau_1, ..., \tau_\ell) = (\sigma \wedge \tau_1, ..., \sigma \wedge \tau_\ell)$ for every $\sigma \in \{0,1\}$ and $(\tau_1, ..., \tau_\ell) \in \{0,1\}^\ell \equiv \mathrm{GF}(2^\ell)$.

Finally, we consider the computation of the f_i's. We first note that $\mathrm{prv}_x(i)$ can be computed by uniform $\mathcal{AC}^0$ (by computing the bits $((x_j \neq 0) \wedge \bigwedge_{j<k<i}(x_k = 0))$ for $j \in \{0,1,...,i-1\}$). Next, we note that for $\ell = \mathrm{poly}(\log n)$, addition and multiplication in $\mathrm{GF}(2^\ell)$ are computable by (uniform) constant-depth circuits of $\mathrm{poly}(n)$-size.[28] We can take inverses in $\mathrm{GF}(2^\ell)$ by raising to the power $2^\ell - 2$, but the question is whether this can be done in by (uniform) circuits of constant depth and size $\exp(\ell^c)$ for any desired $c > 0$. Fortunately, for ℓ of the form $2 \cdot 3^i$, the answer is positive – see [32, Cor. 6 (2)]. ■

[28] In particular, recall that multiplication in $\mathrm{GF}(2^\ell)$ reduces to computing inner products mod 2 of ℓ-bit long vectors, whereas such computation can be carried out by depth $d = O(1)$ circuits of size $\exp(\ell^{1/(d-1)}) = \mathrm{poly}(n)$.

An alternative construction. As an alternative generalization of the extractor presented by Viola [57, Lem. 4.3], consider using the seed (which will now have length $2n$ rather than n) as a description of a sequence of $n' = n/\ell$ Toeplitz matrices, denoted $T = (T_1, ..., T_{n'})$, where each T_i is a ℓ-by-ℓ matrix.[29] Likewise, we view the source x as a sequence $(x_1, ..., x_{n'}) \in (\{0,1\}^\ell)^{n'}$, and output $E'(x,T) = (T, Tx)$, where $Tx = \sum_{i\in[n']} T_i x_i$. Recall that E' is a strong $(\ell + 3\log(2/\epsilon), \epsilon)$-error extractor; again, this follows by the the Leftover Hashing Lemma (using the fact that $x \mapsto E'(x, U_n)$ is pairwise independent).

It is simpler to implement this extractor in $\mathcal{AC}^0$ when using affine transformations based on ℓ-by-ℓ Toeplitz matrices; that is, rather than the T_i's we use $A_i = (T_i, b_i)$'s, where $b_i \in \{0,1\}^\ell$, and output the pair $((A_1, ..., A_{n'}), v)$ such that $v = \sum_{i\in[n']}(T_i x_i + b_i)$. That is, for $A = (A_1, ..., A_{n'})$, where $A_i = (T_i, b_i)$, the extractor is defined by $E(x, A) = (A, \sum_{i\in[n']}(T_i x_i + b_i))$. Indeed, $v = \sum_{i\in[n']} T_i x_i + \sum_{i\in[n']} b_i$, but it is useful to have this redundancy. Specifically, the $\mathcal{AC}^0$ implementation of E uses its seed to determine $(A_1, ..., A_{n'})$, where $A_i = (T_i, b_i)$, but then outputs $(((T_1, b'_1), ..., (T_{n'}, b'_{n'})), T_{n'} x_{n'} + b_{n'})$ such that $b'_i = b_i + (T_{i-1}x_{i-1} + b_{i-1})$, where $T_0 x_0 + b_0 = 0^\ell$. The key observation is that $T_{n'}x_{n'} + b_{n'}$ equals the last ℓ bits of $E(x, ((T_1, b'_1), ..., (T_{n'}, b'_{n'})))$. This holds because

$$\begin{aligned}\sum_{i\in[n']}(T_i x_i + b'_i) &= \sum_{i\in[n']}(T_i x_i + b_i + T_{i-1}x_{i-1} + b_{i-1}) \\ &= T_{n'}x_{n'} + b_{n'}.\end{aligned}$$

Turning back to the construction that uses linear (rather than affine) transformations, we mention that it can be implemented by using the first column in each matrix T_i that corresponds to a 1-entry in x_i.

Using a shorter seed when the min-entropy is $\omega(\log^3 n)$. Combining the extractor of Theorem 6.1 with a suitable averaging sampler, we obtain the following.

Corollary 6.2 (on the tightness of Theorem 5.4): *For any $k = \Omega(\log n)$ and $r \geq T \stackrel{\rm def}{=} \min(n, \Theta(n\log^3 n)/k)$, there exists a $(k, {\rm poly}(1/n))$-extractor, $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^{r+\min(\Omega(k),{\rm poly}(\log n))}$, that is computable by uniform $\mathcal{AC}^0$.*

We stress the fact that Corollary 6.2 implies an extractor that outputs $m(n) = r(n) + \Omega(\log n) > r(n)$ bits using a seed of length $r(n) = O(n\log^3 n)/k(n)$ (provided that $k(n) = \Omega(\log n)$). Hence, non-trivial extraction in $\mathcal{AC}^0$ (i.e., $m(n) > r(n)$) is possible whenever $k(n)\cdot r(n) = O(n\log^3 n)$ (provided $k(n) = \Omega(\log n)$). Recall that (Part 1 of) Theorem 5.4 asserts that $k(n)\cdot m(n) = \Omega(n/{\rm poly}(\log n))$ must hold for any non-trivial extraction in $\mathcal{AC}^0$. Loosely speaking, the combination of these results implies that *non-trivial extraction in $\mathcal{AC}^0$ is possible if and only if $k(n)\cdot m(n) = \widetilde{\Theta}(n)$.*

Proof: (The case of $T = n$ follows immediately from Theorem 6.1, and we focus on the case that $T < 0.1n$ (since otherwise, we can artificially increase T to n and act accordingly).) Assuming that $T \leq 0.1n$, we combine an adequate averaging sampler with the extractor of Theorem 6.1, where the combination uses the sample-then-extract paradigm (as stated in Corollary 2.8). Actually, we plug

[29] As usual, each Toeplitz matrix is represented by its first row and its first column. For notational simplicity we view strings as either row or column vectors, according to their use.

the extractor of Theorem 6.1 into Corollary 3.7, which already incorporate the adequate sampler. Specifically, when invoking Corollary 3.7, we set $r_0 = t = r - O(\log n)^2$ and $\delta = k(n)/n$ (and use $\epsilon = 1/\text{poly}(n)$ and (say) $\beta = 1/2$).[30] ■

6.2 Repeated extraction with independent seeds

The following straightforward scheme for repeated extraction is wasteful in its use of the seed, but its appeal lies in the fact that it preserves the computational complexity of the original extractor. The key observation underlying its analysis is that (typical) conditioning on m' bits extracted from an (n, k)-source yields an $(n, k - m' - 1)$-source; this observation is at least implicit in numerous works regarding randomness extraction.

Theorem 6.3 (repeated extraction): *Let $n, k, r, m, t \in \mathbb{N}$ and $\epsilon \in [0, 1]$. If $E : \{0,1\}^n \times \{0,1\}^r \to \{0,1\}^m$ is a (k, ϵ)-extractor, then $E_t : \{0,1\}^n \times \{0,1\}^{tr} \to \{0,1\}^{tm}$ defined by $E_t(x, s_1 \circ \cdots \circ s_t) = E(x, s_1) \circ \cdots \circ E(x, s_t)$ is a $(k + (t-1) \cdot (m+1), (2t-1) \cdot \epsilon)$-extractor. An analogous statement holds for strong extractors.*

(Hence, the quality of extraction is harmed in a small manner, especially when $t \ll k/m$. Indeed, the result is meaningful only if $k + (t-1)(m+1) < n$.)

Proof: We first prove the version that refers to ordinary extractors.[31] We proceed by induction on t, proving that for any $(n, k + (t-1)(m+1))$-source X the statistical distance between U_{tm} and $E_t(X, U_{td})$ is at most $(2t-1) \cdot \epsilon$. The base case (of $t = 1$) is immediate by the hypothesis regarding E. In the induction step we proceed as follows, while writing $E_t(x, s_1 \circ \cdots \circ s_t) = E_{t-1}(x, s_1 \circ \cdots \circ s_{t-1}) \circ E(x, s_t)$:

$$\begin{aligned}
&\mathbf{\Delta}[E_t(X, U_{tr})\,;U_{tm}] \\
&\quad = \mathbf{\Delta}[E_{t-1}(X, U_{(t-1)r}) \circ E(X, U_r)\,; U_{(t-1)m} \circ U_m] \\
&\quad \le \mathbf{\Delta}[E_{t-1}(X, U_{(t-1)r}) \circ E(X, U_r)\,; U_{(t-1)m} \circ E(X, U_r)] \\
&\qquad + \mathbf{\Delta}[U_{(t-1)m} \circ E(X, U_r)\,; U_{(t-1)m} \circ U_m] \\
&\quad \le \mathbf{\Delta}[E_{t-1}(X, U_{(t-1)r}) \circ E(X, U_r)\,; U_{(t-1)m} \circ E(X, U_r)] + \mathbf{\Delta}[E(X, U_r)\,; U_m], \qquad (13)
\end{aligned}$$

where the last inequality uses $\mathbf{\Delta}[\Pi(Y)\,; \Pi(Z)] \le \mathbf{\Delta}[Y\,; Z]$ (for any random process Π). Using the hypothesis regarding E, the second term of Eq. (13) is upper bounded by ϵ. So we turn to analyze the first term of Eq. (13). We shall use the following notation:

- We let $k' = k + (t-1)(m+1)$ and recall that X is an (n, k')-source;
- We let Y denote the distribution of $E(X, U_r)$;
- For any y in the support of Y, we let X'_y denote the distribution of X conditioned on $E(X, U_r) = y$.

[30] We mention that an essentially weaker statement, which refers only to $k \ge n^{2/3}$, follows by instantiating Corollary 3.3 with the extractor of Theorem 6.1. In this case, the resulting extractor has seed length $r = t + O(\log n)$ rather than $t + O(\log n)^2$, but this gain is insignificant because in both cases $t = \Omega(\log^3 n)$.

[31] The proof of the version that refers to strong extraction is very similar, and will be presented later.

Using these notations we have

$$\begin{aligned}
&\mathbf{\Delta}[E_{t-1}(X,U_{(t-1)r})\circ E(X,U_r)\,;U_{(t-1)m}\circ E(X,U_r)] \\
&\quad = \quad \mathbf{E}_{y\leftarrow Y}[\mathbf{\Delta}[E_{t-1}(X'_y,U_{(t-1)r})\,;U_{(t-1)m}]] \\
&\quad \leq \quad \max_{y:\mathbf{H}_\infty(X'_y)\geq k'-m-1}\left\{\mathbf{\Delta}[E_{t-1}(X'_y,U_r)\,;U_{(t-1)m}]\right\} && (14)\\
&\qquad\quad +\mathbf{Pr}_{y\leftarrow Y}[\mathbf{H}_\infty(X'_y)<k'-m-1], && (15)
\end{aligned}$$

where the inequality uses the fact that $\mathbf{\Delta}[\,;]$ is upper bounded by 1. We upper bound Eq. (14) by using the induction hypothesis, while noting that in this case X'_y is an $(n,k+(t-1)(m+1)-m-1)$-source. Hence, Eq. (14) is upper bounded by $(2(t-1)-1)\cdot\epsilon$. To bound Eq. (15), we first observe that

$$\mathbf{Pr}_{y\leftarrow Y}\left[\mathbf{Pr}[E(X,U_r)=y]<0.5\cdot 2^{-m}\right]<\epsilon, \tag{16}$$

because otherwise the hypothesis regarding E is violated. (Specifically, let $B=\{y:\mathbf{Pr}[E(X,U_r)=y]<0.5\cdot 2^{-m}\}$, then $\mathbf{Pr}[U_m=y]=2^{-m}>2\cdot\mathbf{Pr}[Y=y]$ for every $y\in B$, which implies $\mathbf{Pr}[U_m\in B]>2\cdot\mathbf{Pr}[Y\in B]$, and so $\mathbf{Pr}[Y\in B]\geq\epsilon$ implies $\mathbf{\Delta}[Y\,;U_m]>\epsilon$.) Now, using Eq. (16), it follows that

$$\begin{aligned}
&\mathbf{Pr}_{y\leftarrow Y}[\mathbf{H}_\infty(X'_y)<k'-m-1] \\
&\quad\leq\quad \mathbf{Pr}_{y\leftarrow Y}\left[\mathbf{Pr}[E(X,U_r)\!=\!y]<2^{-m-1}\right] \\
&\qquad\quad +\mathbf{Pr}_{y\leftarrow Y}\left[\exists x \text{ s.t. } \mathbf{Pr}[X'_y\!=\!x]>2^{-k'+m+1}\middle|\mathbf{Pr}[E(X,U_r)\!=\!y]\geq 2^{-m-1}\right] \\
&\quad\leq\quad \epsilon+\mathbf{Pr}_{y\leftarrow Y}\left[\exists x \text{ s.t. } \mathbf{Pr}[X\!=\!x|E(X,U_r)\!=\!y]>2^{-k'+m+1}\middle|\mathbf{Pr}[E(X,u)\!=\!y]\geq 2^{-m-1}\right]_. && (17)
\end{aligned}$$

Next, note that for every $x\in\{0,1\}^n$ and $y\in\{0,1\}^m$, it holds that

$$\mathbf{Pr}[X'_y=x]=\mathbf{Pr}[X=x|E(X,U_r)=y]\leq\frac{\mathbf{Pr}[X=x]}{\mathbf{Pr}[E(X,U_r)=y]}$$

where equality holds if and only if $\mathbf{Pr}[E(x,U_r)=y]=1$. Hence, for y such that $\mathbf{Pr}[E(X,U_r)=y]\geq 2^{-m-1}$, and for every x, it holds that

$$\begin{aligned}
\mathbf{Pr}[X=x|E(X,U_r)=y] \quad &\leq\quad \frac{\mathbf{Pr}[X=x]}{\mathbf{Pr}[E(X,U_r)=y]} \\
&\leq\quad 2^{-k'+m+1}
\end{aligned}$$

which means that the second term in Eq. (17) is zero. Hence, Eq. (15) is upper bounded by ϵ, and the induction claim follows, since Eq. (13) is upper bounded by $\epsilon+(2(t-1)-1)\cdot\epsilon+\epsilon$.

The case of strong extractors. We now turn to the version that refers to strong extractors, and proceed as in the ordinary case subject to adequate modifications. (Indeed, the reader may want to skip the rest of the proof.) Denoting $E'_t(x,\overline{u})=E_t(x,\overline{u})\circ\overline{u}$ (and $E'(x,u)=E(x,u)\circ u$), we prove (by induction on t) that for any $(n,k+(t-1)(m+1))$-source X the statistical distance between U_{tm+tr} and $E'_t(X,U_{tr})$ is at most $(2t-1)\cdot\epsilon$. Here we use:

$$\begin{aligned}
&\mathbf{\Delta}[E'_t(X,U_{tr})\,\mathbf{;}\,U_{tm+tr}]\\
&\quad = \mathbf{\Delta}[E'_{t-1}(X,U_{(t-1)r})\circ E'(X,U_r)\,\mathbf{;}\,U_{(t-1)(m+r)}\circ U_{m+r}]\\
&\quad \leq \mathbf{\Delta}[E'_{t-1}(X,U_{(t-1)r})\circ E'(X,U_r)\,\mathbf{;}\,U_{(t-1)(m+r)}\circ E'(X,U_r)] + \mathbf{\Delta}[E'(X,U_r)\,\mathbf{;}\,U_{m+r}]. \qquad (18)
\end{aligned}$$

Using the (strong) hypothesis regarding E, the second term of Eq. (18) is upper bounded by ϵ. So we turn to analyze the first term of Eq. (18). We shall use the following notation:

- For any $u \in \{0,1\}^r$, we let Y_u denote the distribution of $E(X,u)$;
- For any $u \in \{0,1\}^r$ and y in the support of Y_u, we let $X'_{u,y}$ denote the distribution of X conditioned on $E(X,u)=y$.

Using these notations we have

$$\begin{aligned}
&\mathbf{\Delta}[E'_{t-1}(X,U_{(t-1)r})\circ E'(X,U_r)\,\mathbf{;}\,U_{(t-1)(m+r)}\circ E'(X,U_r)]\\
&\quad = \mathbf{E}_{u\leftarrow U_r;y\leftarrow Y_u}[\mathbf{\Delta}[E'_{t-1}(X'_{u,y},U_{(t-1)r})\,\mathbf{;}\,U_{(t-1)(m+r)}]]\\
&\quad \leq \max_{u,y:\mathbf{H}_\infty(X'_{u,y})\geq k'-m-1}\left\{\mathbf{\Delta}[E'_{t-1}(X'_{u,y},U_r)\,\mathbf{;}\,U_{(t-1)(m+r)}]\right\} \qquad (19)\\
&\qquad\quad +\mathbf{Pr}_{u\leftarrow U_r;y\leftarrow Y_u}[\mathbf{H}_\infty(X'_{u,y})<k'-m-1]. \qquad (20)
\end{aligned}$$

We upper bound Eq. (19) by using the induction hypothesis, while noting that in this case $X'_{u,y}$ is an $(n,k+(t-1)(m+1)-m-1)$-source. Hence, Eq. (19) is upper bounded by $(2(t-1)-1)\cdot\epsilon$. To bound Eq. (20), we first observe that

$$\mathbf{Pr}_{u\leftarrow U_r;y\leftarrow Y_u}\left[\mathbf{Pr}[E(X,u)=y]<0.5\cdot 2^{-m}\right]<\epsilon \qquad (21)$$

because otherwise the hypothesis regarding E is violated. (Specifically, let $B=\{(u,y):\mathbf{Pr}[E(X,u)=y]<0.5\cdot 2^{-m}\}$, then $\mathbf{Pr}[U_{r+m}\in B]>2\cdot\mathbf{Pr}[(U_r,Y_{U_r})\in B]$, since $\mathbf{Pr}[U_m=y]>2\cdot\mathbf{Pr}[Y_u=y]$ for every $(u,y)\in B$.) It follows that

$$\begin{aligned}
&\mathbf{Pr}_{u\leftarrow U_r;y\leftarrow Y_u}[\mathbf{H}_\infty(X'_{u,y})<k'-m-1]\\
&\quad \leq \mathbf{Pr}_{u\leftarrow U_r;y\leftarrow Y_u}\left[\mathbf{Pr}[E(X,u){=}y]<2^{-m-1}\right] \qquad (22)\\
&\qquad +\mathbf{Pr}_{u\leftarrow U_r;y\leftarrow Y_u}\left[\exists x \text{ s.t. } \mathbf{Pr}[X'_{u,y}{=}x]>2^{-k'+m+1}\middle|\mathbf{Pr}[E(X,u){=}y]\geq 2^{-m-1}\right], \qquad (23)
\end{aligned}$$

where Eq. (22) is upper bounded by ϵ. Next, note that for every $x\in\{0,1\}^n$, $u\in\{0,1\}^r$ and $y\in\{0,1\}^m$, it holds that

$$\mathbf{Pr}[X'_{u,y}=x]=\mathbf{Pr}[X=x|E(X,u)=y]\leq\frac{\mathbf{Pr}[X=x]}{\mathbf{Pr}[E(X,u)=y]}$$

where equality holds if and only if $E(x,u)=y$. Hence, for u and y such that $\mathbf{Pr}[E(X,u)=y]\geq 2^{-m-1}$, and for every x, it holds that

$$\begin{aligned}
\mathbf{Pr}[X=x|E(X,u)=y] &\leq \frac{\mathbf{Pr}[X=x]}{\mathbf{Pr}[E(X,u)=y]}\\
&\leq 2^{-k'+m+1}
\end{aligned}$$

which means that the the second term in Eq. (23) is zero. Hence, Eq. (20) is upper bounded by ϵ, and the induction claim follows (since Eq. (18) is upper bounded by $\epsilon+(2(t-1)-1)\cdot\epsilon+\epsilon$). ■

Applications to the study of extraction in $\mathcal{AC}^0$. Combining Theorem 6.3 with the various extractors presented in this work, we obtain.

Corollary 6.4 (on extraction in $\mathcal{AC}^0$ using a long seed): *For every $\epsilon : \mathbb{N} \to (0,1]$ such that $\epsilon(n) \geq 1/\text{poly}(n)$ and every constant $\alpha < 1$, there exist explicit $\mathcal{AC}^0$ circuits that compute (k,ϵ)-extractors, $E : \{0,1\}^n \times \{0,1\}^{r(n)} \to \{0,1\}^{m(n)}$, for the following relation between r, m and k.*

1. *For any $k(n) = \Omega(n)$, we can have any $m(n) \leq \alpha\cdot k(n)$ with any $r(n) = \max(\beta\cdot m(n), \Omega(\log n))$, for any constant $\beta > 0$. Furthermore, the extractor is strong and the $\mathcal{AC}^0$ circuits have depth three.*

2. *For any $k(n) \geq n/\text{poly}(\log n)$, we can have any $m(n) \leq \alpha \cdot k(n)$ with $r(n) = \max(\beta \cdot m(n), \Omega(\log n))$, for any constant $\beta > 0$. Furthermore, the extractor is strong and the $\mathcal{AC}^0$ circuits have depth $4 + \frac{\log(n/k(n))}{\log\log n}$.*

3. *For any constant $c > 0$ and $k(n) \geq O(\log n)^{c+3}$, we can have any $m'(n) = m(n) - r(n) \leq \alpha \cdot k(n)$ with $r(n) \cdot k(n) = m'(n) \cdot n/(\log n)^c$. In particular:*

   - *We can have $m'(n) = \alpha \cdot k(n)$ and $r(n) = n/(\log n)^c$.*
   - *We can have $m'(n) = \min(\alpha \cdot k(n), (\log n)^c)$ and $r(n) = n/k(n)$.*

   In addition, for any $k(n) = \Omega(\log n)$ we can have $m'(n) = m(n) - r(n) = \min(\alpha\cdot k(n), \text{poly}(\log n))$ and $r(n) = n$.

Note that in Part 1 the output length exceeds the seed length, whereas in the other parts the output is shorter. The extractors in Parts 1 and 2 are strong, but this is not the case (and cannot be the case) in Part 3. The additional claim (in Part 3) is merely a restating of Theorem 6.1, which also appears as a special case of Corollary 6.2.

Proof: In all parts, $t = t(n)$ denotes the number of times that the basic extractor $E_0 : \{0,1\}^n \times \{0,1\}^{r_0(n)} \to \{0,1\}^{m_0(n)}$ is invokes; that is, we will derive $m(n) = t(n)\cdot m_0(n)$ and $r(n) = t(n)\cdot r_0(m)$.

Part 1 (resp., Part 2) is obtained by combining Theorem 2.4 (resp., Theorem 3.1) with the second part of Theorem 6.3 (i.e., the part referring to strong extractors). In both parts we have $r_0(n) = \Theta(\log n)$; in Part 1 we can have any $m_0(n) = r_0/\beta = O(r_0(n))$, whereas in Part 2 we have some $m_0(n) = \Omega(r_0(n))$. Note that the extractor asserted in Theorem 2.4 can be implemented by depth-three $\mathcal{AC}^0$ circuits.

Part 3 combines Corollary 6.2 with the first part of Theorem 6.3 (i.e., the part referring to ordinary extractors). Here we use $r_0 = \min(n, O(n\log^3 n)/k(n))$ and $m_0' = m_0 - r_0 = \min(\alpha k(n), (\log n)^{c+3})$. Hence, for $k(n) \geq O(\log n)^{c+3}$, we have $r_0 \cdot k(n) = m_0' \cdot n/O(\log n)^c$, which yields $r(n) \cdot k(n) = m'(n) \cdot n/O(\log n)^c$ for $r(n) = t(n) \cdot r_0(n)$ and $m'(n) = t(n) \cdot m_0'(n) \leq \alpha k(n)$. ■

7 Extraction from several independent sources

While deterministic extraction is not possible from a single general $(n, n-1)$-source, it is known to be possible when having a constant number of independent sources (cf., e.g., [16, 6, 7, 37]). We ask whether such extraction is possible in $\mathcal{AC}^0$, provided that the min-entropy rate is at least $1/\text{poly}(\log n)$. Considering extraction from a constant number of independent sources, note that the impossibility results in Section 5.2 can be adapted to rule out min-entropy rates below $1/\text{poly}(\log n)$. Details follow.

We view the c sources as a single source of length cn in which the n-bit long parts are independent of one another. Applying the proof of Theorem 5.3 to such a generic source, yields a bit-fixing source of length cn in which ck bits are random. These ck positions are "typical" (w.r.t an averaging argument) and so we can make sure that there are approximately k random bits in each n-bit long part of the source. This proves that $\mathcal{AC}^0$-extraction from a constant number of sources is impossible if all sources have min-entropy rate below $n/\text{poly}(\log n)$. However, above that level of min-entropy, $\mathcal{AC}^0$-extraction is possible when *using a seed of logarithmic length*. Hence, it is reasonable to ask whether deterministic $\mathcal{AC}^0$-extractors can meet this performance (as in the case of bit-fixing sources).

7.1 Extraction from two independent sources

In this section, we consider $\mathcal{AC}^0$-extractors for pairs of independent sources, a model first considered by Vazirani [55] (for Santha-Vazirani sources [51]). Recall that Chor and Goldreich [16] showed that inner-product mod 2 yields a good extractor for pairs of independent sources provided that each source has min-entropy rate above half; in fact, it yields an ϵ-extractor for any pair consisting of an (n, k_1)-source and an (n, k_2)-source such that $k_1 + k_2 > n + 1 + 2\log(1/\epsilon)$. A natural question that arises is whether such a seedless extractor can operate in $\mathcal{AC}^0$. Specifically:

Definition 7.1 (a (seedless) two-source extractor): *The function $E : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is called a $((k_1, k_2), \epsilon)$*-extractor *if for every pair of independent random variables (X, Y) such that X is a (n, k_1)-source and Y is a (n, k_2)-source it holds that $\mathbf{\Delta}[E(X,Y)\,;U_m] \le \epsilon$.*

We ask whether a $((0.51n, 0.51n), 0.01)$-extractor can be computed in $\mathcal{AC}^0$. So far, we only know of $((k, k), 0.01)$-extractors in $\mathcal{AC}^0$ for $k(n) = n - (n/\log^4 n)$ and $m = 1$.

Theorem 7.2 (two-source deterministic extractor in $\mathcal{AC}^0$): *For $k(n) = n - (n/\log^4 n)$ and $\epsilon(n) = n^{-\omega(1)}$, there exists a $((k, k), \epsilon)$-extractor in $\mathcal{AC}^0$* (extracting a single bit).

The constant 4 in the exponent is an arbitrary constant greater than three.

Proof: First, we shall see how to extract a single bit with error $O(1/\log n)$ from sources of rate $1 - \log^{-3} n$. Consider a partition of each of the two sources into blocks of length $\ell = \log^2 n$; that is, let $X = (X_1, ..., X_{n/\ell})$ and $Y = (Y_1, ..., Y_{n/\ell})$ such that $|X_i| = |Y_i| = \ell$. We shall prove (see Claim 7.2.1 below) that each source is $\exp(-\Omega(\ell))$-close to a source in which at least a $1 - 10\log^{-3} n$ fraction of the blocks have min-entropy rate at least 0.7 conditioned on the previous blocks.

Next, we apply the inner-product mod 2 extractor [16] on each pair of corresponding blocks, obtaining a sequence Z of $n' \stackrel{\text{def}}{=} n/\ell$ bits such that at least $1 - 20\log^{-3} n$ of them are "quasi-random" conditioned on the previous bits; that is, Z_i is the inner-product mod 2 of X_i and Y_i and for all but at most $20n'/\log^3 n$ of the indices $i \in [n']$ it holds that $\mathbf{Pr}[Z_i = 1 | Z_{[i-1]} = \alpha] = 0.5 \pm \exp(-\Omega(\ell))$ for every $\alpha \in \{0,1\}^{i-1}$. Applying the extractor of Theorem 5.6 (which was used in the proof of Theorem 5.7) to Z, we extract a bit with bias $O(1/\log n)$. (Indeed, a sequence as above is $n \cdot \exp(-\Omega(\ell))$-close to a *non-oblivious* bit-fixing source of length n/ℓ with $20n/(\ell \log^3 n)$ bits that are fixed as an arbitrary function of the random bits.)[32]

[32] Given a sequence of $n' = n/\ell$ bits in which k' bits are quasi-random conditioned on the previous bits, note that the subsequence of quasi-random bits is close to being uniformly distributed in $\{0,1\}^{k'}$.

Finally, to reduce the error of the extractor, we use a source of higher min-entropy rate $(1 - \log^{-4} n)$, which we break into $\log n$ parts such that each part has conditional min-entropy rate at least $1 - \log^{-3} n$. Applying the foregoing extractor to each part, we get $\log n$ bits such that each bit has bias $O(1/\log n)$ conditioned on the previous ones. XORing these bits, we obtain a bit with bias $O(1/\log n)^{\log n} = n^{-\omega(1)}$. The proof is completed once we prove the following claim.

Claim 7.2.1 (on the number of blocks with high conditional min-entropy): *Let $X = (X_1, ..., X_{n'})$ be a $(n, \delta n)$-source such that $|X_i| = \ell = n/n'$, and $\delta', \epsilon' \in (0, 1)$ such that $\epsilon' \ell \geq \log n$. Then, X is $\exp(-\Omega(\epsilon' \ell))$-close to a source X' for which there exists a set $I \subseteq [n']$ of cardinality at least $\frac{\delta - \delta'}{1 - \delta'} \cdot n'$ such that for every $i \in I$ and $x \in \{0, 1\}^n$ it holds that $\mathbf{Pr}[X'_i = x_i | X'_{[i-1]} = x_{[i-1]}] \leq 2^{-(\delta' - 2\epsilon')\ell}$.*

For the above application we set $\delta = 1 - \log^{-3} n$, $\delta' = 0.9$, and $\epsilon' = 0.1$. Hence, $\frac{\delta - \delta'}{1 - \delta'} = 1 - \frac{1 - \delta}{1 - \delta'} = 1 - 10 \cdot (1 - \delta)$ and $\delta' - 2\epsilon' = 0.7$.

Proof: As in many known cases, the proof proceeds via analyzing the collision probability. Recall that the collision probability of a random variable Z, denoted $\mathbf{CP}[Z]$, equals $\mathbf{Pr}[Z^{(1)} = Z^{(2)}]$, where $Z^{(1)}$ and $Z^{(2)}$ are two independent copies of the random variable Z; that is, $\mathbf{CP}[Z] = \sum_z \mathbf{Pr}[Z = z]^2$. Using the fact that X has min-entropy δn, we infer that $\mathbf{CP}[X] \leq 2^{-\delta n}$. Hence, it holds that

$$\prod_{i \in [n']} \mathbf{Pr}[X_i^{(1)} = X_i^{(2)} | X_{[i-1]}^{(1)} = X_{[i-1]}^{(2)}] \ \leq \ 2^{-\delta n' \cdot \ell}, \tag{24}$$

where $X^{(1)}$ and $X^{(2)}$ are two independent copies of X. We call i good if $\mathbf{Pr}[X_i^{(1)} = X_i^{(2)} | X_{[i-1]}^{(1)} = X_{[i-1]}^{(2)}] \leq 2^{-\delta' \ell}$, and infer that at least a $\frac{\delta - \delta'}{1 - \delta'}$ fraction of the i's are good.[33] We next observe that, for every good i, there exists a set $S_{i-1} \subseteq \{0, 1\}^{(i-1)\ell}$ such that $\mathbf{Pr}[X_{[i-1]} \in S_{i-1}] \geq 1 - 2^{-\epsilon' \ell}$ and for every $z \in S_{i-1}$ it holds that

$$\mathbf{Pr}[X_i^{(1)} = X_i^{(2)} | X_{[i-1]}^{(1)} = X_{[i-1]}^{(2)} = z] \leq 2^{-(\delta' - \epsilon')\ell}. \tag{25}$$

Moving back to a min-entropy upper bound, for any good i and every $z \in S_{i-1}$, let $H_{i,z} \stackrel{\text{def}}{=} \{x_i : \mathbf{Pr}[X_i = x_i | X_{[i-1]} = z] > 2^{-(\delta' - 2\epsilon')\ell}\}$. Then, $\mathbf{Pr}[X_i \in H_{i,z} | X_{[i-1]} = z] \leq 2^{-\epsilon' \ell}$ holds (for any good i and $z \in S_{i-1}$).[34] Let $X' = X$ if for every good i it holds that $X_{[i-1]} \in S_{i-1}$ and $X_i \in H_{i, X_{[i-1]}}$, and X' be uniform otherwise. Then, X' is $O(n' \cdot 2^{-\epsilon' \ell})$-close to X, and for every good i and every $x \in \{0, 1\}^n$ it holds that $\mathbf{Pr}[X'_i = x_i | X'_{[i-1]} = x_{[i-1]}] \leq 2^{-(\delta' - 2\epsilon')\ell}$. The claim follows. ■

This completes the proof of the theorem. ■

Digest. Note that the sequence Z extracted at the first part (of the main step) is apparently more restricted than a non-oblivious bit-fixing source, but it is not an (oblivious) bit-fixing source. Hence, we applied the extractor of Theorem 5.6 to Z, rather than applying the $\mathcal{AC}^0$-extractors for bit-fixing sources that can handle lower min-entropy. Of course, there is room in between, and

[33] Denoting the set of good i's by G, we have $(2^{-\ell})^{|G|} \cdot (2^{-\delta' \ell})^{n' - |G|} \leq 2^{-\delta n' \cdot \ell}$, and it follows that $(1 - \delta')|G| \geq (\delta - \delta')n'$.

[34] Let Y and Y' be distributed independently and identically to X_i conditioned on $X_{[i-1]} = z$ (i.e., $\mathbf{Pr}[Y = y] = \mathbf{Pr}[X_i = y | X_{[i-1]} = z]$), and $H \stackrel{\text{def}}{=} \{y : \mathbf{Pr}[Y = y] > 2^{-(\delta' - 2\epsilon')\ell}\}$. Then, $\mathbf{Pr}[Y = Y'] \leq 2^{-(\delta' - \epsilon')\ell}$ implies $\mathbf{Pr}[Y \in H] \leq 2^{-\epsilon' \ell}$. See farther discussion in Remark 7.3.

one may capitalize on it, but currently we do not see a way of doing so. Indeed, the point is trying to obtain $\mathcal{AC}^0$-extraction for two independent sources of some constant min-entropy rate. (In Section 7.2, we achieve the corresponding goal for four sources.)

Remark 7.3 (from conditional collision probability to conditional min-entropy): *The proof of Claim 7.2.1 relies on transforming an upper bound on conditional collision probability to a lower bound on conditional min-entropy. For the benefit of future applications, we distill the corresponding claim here.*

> Let $X^{(1)}$ and $X^{(2)}$ be two independent copies of X, and let $f, g : \{0,1\}^* \to \{0,1\}^*$.
> If $\mathbf{Pr}[f(X^{(1)})=f(X^{(2)})|g(X^{(1)})=g(X^{(2)})] \leq \epsilon \cdot 2^{-k}$, then X is $2\sqrt{\epsilon}$-close to X' such that $\mathbf{Pr}[f(X')=v|g(X')=u] \leq 2^{-k}$ for every u, v.

The proof of this claim proceeds in two steps. First, we infer that there exists a set S such that $\mathbf{Pr}[g(X) \in S] \geq 1 - \sqrt{\epsilon}$ and $\mathbf{Pr}[f(X^{(1)}) = f(X^{(2)})|g(X^{(1)}) = g(X^{(2)}) = s] \leq \sqrt{\epsilon} \cdot 2^{-k}$ for every $s \in S$. Next, for every $s \in S$ and every v, we show that X is $\sqrt{\epsilon}$-close to X' that satisfies $\mathbf{Pr}[f(X')=v|g(X')=s] \leq 2^{-k}$. This is shown by considering the random variable X_s defined as $f(X)$ conditioned on $g(X) = s$, letting $H = \{v : \mathbf{Pr}[X_s=v] > 2^{-k}\}$, and noting that $\mathbf{CP}[X_s] \geq \mathbf{Pr}[X_s \in H] \cdot \min_{v\in H}\{\mathbf{Pr}[X_s=v]\}$, which implies $\sqrt{\epsilon}2^{-k} \geq \mathbf{CP}[X_s] > \mathbf{Pr}[X_s \in H] \cdot 2^{-k}$. The claim follows by letting X' be X conditioned on both $g(x) \in S$ and $f(X) \notin H$.

Remark 7.4 (reducing two-source extraction to extraction from non-oblivious block-fixing sources): *As stated above, the core of the proof of Theorem 7.2 is a transformation of a pair of independent sources, each having min-entropy rate $\delta > 0.75$, into a non-oblivious bit-fixing source of min-entropy rate at least $1 - 4 \cdot (1 - \delta) - o(1)$, where here we use Claim 7.2.1 with any constant $\delta' > 0.5$ and $\epsilon' = 1/\log n$* (rather than with $\delta' = 0.9$ and $\epsilon' = 0.1$). *Similarly, we can transform such a pair of sources into a non-oblivious block-fixing source with block length $\Omega(\log^2 n)$, by using an extractor that outputs $\Omega(\ell)$ bits [16, 19]. Hence, deterministic two-source $\mathcal{AC}^0$-extractors for some constant min-entropy rate exist if such extractors exist for a non-oblivious block-fixing source of some constant rate and super-logarithmic block length.*

Recall that there exist no deterministic extractors for non-oblivious *bit*-fixing source of any (non-trivial) constant min-entropy rate [39]. Such a result is not known for extraction from non-oblivious *block*-fixing source of (any constant min-entropy rate and) super-logarithmic block length, although it was conjectured more than a decade ago in [22, Conj. 2.2]. Hence, ruling out two-source extractors for any constant rates requires settling the latter conjecture.

7.2 Extraction from four independent sources

In this section, we consider $\mathcal{AC}^0$-extractors from a constant number of independent sources, showing that such extractor exist for *four* independent sources of *some* constant min-entropy rate. In Section 7.3, we shall show how to extract from independent sources of any constant min-entropy rate, but we shall use a larger number of sources (which depends on the rate). In both section, we shall refer to the following definition, which generalizes Definition 7.1.

Definition 7.5 (a (seedless) multi-source extractor): *For a constant integer $c \geq 2$, the function $E : (\{0,1\}^n)^c \to \{0,1\}^m$ is called a $((k_1, ..., k_c), \epsilon)$-*extractor *if for every sequence of c independent random variables $(X^{(1)}, ..., X^{(c)})$ such that $X^{(i)}$ is a (n, k_i)-source for $i = 1, .., ., c$, it holds that $\mathbf{\Delta}[E(X^{(1)}, ..., X^{(c)})\,;U_m] \leq \epsilon$.*

Indeed, Definition 7.1 corresponds to the special case of $c = 2$.

While Theorem 7.2 applies to (two) sources of min-entropy rate of the form $1-o(1)$, the following result holds for (four) sources of min-entropy rate that is bounded away from 1. Furthermore, while Theorem 7.2 does not assert *uniform* $\mathcal{AC}^0$ circuits, the following result does assert such circuits.

Theorem 7.6 (four-source deterministic extractor in $\mathcal{AC}^0$): *For $k(n) = 0.99n$ and $\epsilon(n) = n^{-\omega(1)}$, there exists a $((k, k, k, k), \epsilon)$-extractor in uniform $\mathcal{AC}^0$* (extracting poly-logarithmically many bits).

Proof Sketch: The general strategy is to use the first two sources in order to sample polylogarithmically many bits of the third and fourth sources, and then extract out of the selected sub-sources. Thus, in a sense, we construct a condenser that transforms the second pair of sources into a pair of short sources that maintains a sufficiently high min-entropy rate; specifically, the resulting sources have min-entropy rate at least 0.51 (whereas the original ones had min-entropy rate 0.99). This conversion is performed using the imperfect randomness that exists in the first pair of sources.

Specifically, we will consider a partition of each source into $n' = n/\ell$ blocks of length $\ell = \log^2 n$, denoted $X^{(i)} = (X_1^{(i)}, ..., X_{n'}^{(i)})$, and apply a two-source extractor $E' : \{0,1\}^\ell \times \{0,1\}^\ell \to [n'/\ell]$, which is computable by constant-depth poly(n)-size circuits (e.g., E' computes bilinear functions [16, 19]), to the blocks of the first two sources. We view these n' outcomes (i.e., $E'(X_j^{(1)}, X_j^{(2)})$ for $j \in [n']$) as an assignment of blocks indexed by $[n']$ into $n'' = n'/\ell$ cells (i.e., block j is assigned to cell $E'(X_j^{(1)}, X_j^{(2)})$), and pick a cell $c_{\min} \in [n'/\ell]$ with a minimal number of blocks (breaking ties arbitrarily).[35] Finally, we apply a second two-source extractor $E'' : \{0,1\}^{\ell^2} \times \{0,1\}^{\ell^2} \to \{0,1\}^{\Omega(\ell)}$ to the sub-sources $X_{j_1}^{(3)} \cdots X_{j_{\ell'}}^{(3)}$ and $X_{j_1}^{(4)} \cdots X_{j_{\ell'}}^{(4)}$, where $(j_1, ..., j_{\ell'})$ is the sequence of blocks assigned to cell $c_{\min}$ and $\ell' \le \ell$. (If $\ell' < \ell$, then we pad the said sequence of blocks to the full length of ℓ^2.)

Intuitively, the first extractor E' uses blocks in the first pair of sources (i.e., $X^{(1)}$ and $X^{(2)}$) in order to assign blocks of the second pair of sources into cells. Pairs of blocks of sufficiently high min-entropy (in $X^{(1)}$ and $X^{(2)}$) will assign the corresponding blocks (of $X^{(3)}$ and $X^{(4)}$) to a random cell, but other pairs of blocks (in $X^{(1)}$ and $X^{(2)}$) may assign blocks (of $X^{(3)}$ and $X^{(4)}$) arbitrarily. Still, each cell is assigned many blocks that have sufficiently high min-entropy in the second pair of sources. Hence, the smallest cell, denoted $c_{\min}$, contains blocks of $X^{(3)}$ and $X^{(4)}$ that have average min-entropy rate that exceeds 0.5, and applying the extractor E'' to these blocks (of $X^{(3)}$ and $X^{(4)}$) will yield an almost random output. Note that the value $c_{\min}$ may not be random; it is merely determined as the index of the smallest cell.

In order to analyze the quality of this construction, we first invoke Claim 7.2.1. Specifically, for $\delta = 0.99$ and some constant $\delta', \epsilon' \in (0,1)$, we say that a block index $j \in [n']$ is good for the i^{th} source if $\mathbf{Pr}[X'_j = x_j | X'_{[j-1]} = x_{[j-1]}] \le 2^{-(\delta' - 2\epsilon')\ell}$, for every x, where X' is $\exp(-\Omega(\epsilon'\ell))$-close to $X^{(i)}$. Recall that at least a $\frac{\delta-\delta'}{1-\delta'} = 1 - \frac{1-\delta}{1-\delta'}$ fraction of the blocks are good for each source. We call j good if it is good for all four sources, and conclude that at least a $1 - 4 \cdot \frac{1-\delta}{1-\delta'} > 1/2$ fraction of the blocks are good. Hence, with very high probability, each cell is assigned at least

[35] Indeed, this strategy is inspired by Feige's protocol for leader election [21], and it was already employed in the context of $O(1)$-source extraction by Li [37]. Here, we pick the smallest cell in order to guarantee that the total length of the blocks assigned to it is small. This guarantees both that E'' can be computed by constant-depth circuits of poly(n)-size and that the min-entropy rate of blocks (of $X^{(3)}$ and $X^{(4)}$) in this cell exceeds half. Jumping ahead, we mention that in the proof of Theorem 7.7 we shall only rely on the first implication (and the second implication will not hold).

$(1 - 4 \cdot \frac{1-\delta}{1-\delta'} - o(1)) \cdot \ell$ good blocks, and the min-entropy of the blocks assigned to each cell is at least $(1 - 4 \cdot \frac{1-\delta}{1-\delta'} - o(1)) \cdot \ell \cdot (\delta' - 2\epsilon')\ell$. The analysis is completed by selecting δ' and ϵ' such that $(1 - 4 \cdot \frac{1-\delta}{1-\delta'}) \cdot (\delta' - 2\epsilon')$ is strictly larger than $1/2$ (e.g., using $\delta' = 0.9$ and $\epsilon' = 0.01$ we get a lower bound of 0.52).

We wish to emphasize a key point regarding the foregoing probabilistic analysis. Recall that blocks are defined as "good" (for a particular source) based on their *conditional* min-entropy. That is, the j^{th} block (of the i^{th} source) is good if the conditional min-entropy of that block, conditioned on the prior $j-1$ blocks of this source, exceeds a specific threshold. It follows that if the j^{th} block is good for both the first and second sources, then the j^{th} block (of the third and fourth sources) will be assigned a random cell, conditioned on the assignment of the prior $j-1$ blocks. Similarly, the blocks assigned to the smallest cell (i.e., $c_{\min}$) are defined as good in the same sense, which implies that their total min-entropy is sufficiently high, since it is the sum of their individual conditional min-entropies. Hence, if a ρ fraction of the blocks assigned to the smallest cell are good, then each of these blocks has conditional min-entropy rate at least $\delta'' = \delta' - 2\epsilon'$ (conditioned on the prior blocks), and the total min-entropy rate of the blocks in this cell is at least $\rho \cdot \delta''$.

What remains is implementing the determination the smallest cell and its contents by constant-depth circuits of $\text{poly}(n)$-size. This can be done using techniques as in the proof of Theorem 3.6. Specifically, we refer to the ranking procedure applied in Step 3(b) of the proof.[36] Lastly, note that the foregoing extractor outputs $\Omega(\ell)$ bits, where $\ell = \log^2 n$. The proof remains intact when setting ℓ to be any larger poly-logarithmic function. ■

7.3 Extraction from $\text{poly}(1/\delta)$ sources of rate δ

Looking at the four-source extractor (presented in the proof of Theorem 7.6), one may notice that there are two main reasons for the high constant lower bound on the min-entropy rate (i.e., 0.99) used there. One is that we used a bilinear two-source extractor, whereas we have such extractors only for rate greater than 0.5 (and, in general, explicit two-source extractors are known [12] only for constant rate that is slightly smaller than 0.5).[37] The second reason is that we use blocks that are "good" (i.e., have sufficient min-entropy rate) with respect to *all* sources.

But if we are willing to use more sources, we may reach an arbitrary low constant rate. Firstly, we will use the multi-source extractors of Barak *et al.* [6], which can extract from any constant rate $\rho > 0$ using $\text{poly}(1/\rho)$-many sources of such rate. The crucial observation is that these extractors compute polynomials (of degree that is smaller than the number of sources) over a finite field, and they can be computed by constant-depth circuits of sub-exponential size. Since we shall be applying these extractors to sub-sources of polylogarithmic (in n) length, we can afford this size, which is $\text{poly}(n)$.

Secondly, there is no need to insist on using only blocks that are good for all sources; we can use blocks that are good for a constant fraction of the sources. This assertion relies on the fact that the extractors of [6] work well also when applied to many (independent) sources such that only a constant fraction of these sources are good (i.e., have sufficient min-entropy). The latter fact is based on two observations:

[36] Recall that our ranking procedure amount to counting (in uniform $\mathcal{AC}^0$) the number of marked elements in an array of length n', when guaranteed that this number does not exceed $\text{poly}(\log n)$. An explicit construction of such a counter was presented in [47], improving over [3, Sec. 5] (and subsequent works).

[37] The said constant is not specified in [12], and is estimated to be higher than 0.49.

1. The extractors of [6] iterate the three-source extractor $E_3(x,y,z) = xy + z$, where x, y, z are field elements. Hence, assuming that $\mathbf{Pr}[X = 0] = 0$ and ditto for Y, *the min-entropy of $E_3(X,Y,Z)$ is at least the maximum among the min-entropy of the three sources.* (The condition can be guaranteed by redefining each of the sources so that it never assumes the value zero, while noting that this reduces the min-entropy of the source by at most one unit.)

2. By the entropy increasing lemma of [6], *the min-entropy of $E_3(X,Y,Z)$ is at least a constant factor larger than the minimum among the min-entropy of the three sources*, as long at this value does not exceed 0.9ℓ, where 2^ℓ is the size of the field. Using the following tree marking game, this implies that the iterative extraction procedure of [6], which applies E_3 at the internal nodes of a ternary tree (while the sources are placed at the leaves), produces output of min-entropy at least 0.9ℓ.

   Consider a full ternary tree of height h such that a ρ fraction of the leaves are marked 1 (corresponding to good sources of sufficiently high min-entropy rate), and the other leaves are marked 0 (corresponding to bad sources). Going from the leaves to the root, the following marking rule is applied: If the children of a node are all marked $i > 0$, then the parent is marked $i + 1$ (corresponding to an application of the entropy increasing lemma of [6]), otherwise the parent is marked by the maximum of its children (corresponding to preservation of entropy asserted in Observation 1). Then, as shown in Claim 7.7.1 (below), for every fixed $\rho > 0$ and any (allowed) marking of the leaves, the minimal possible marking of the root is unbounded as a function of h.

Plugging these observations into the framework of the proof of Theorem 7.6, while replacing each pair of sources by many independent sources, we get uniform $\mathcal{AC}^0$-extractors for poly$(1/\delta)$-sources of min-entropy rate δ, for any $\delta > 0$.

Theorem 7.7 (deterministic extractor in $\mathcal{AC}^0$ for poly$(1/\delta)$ sources of constant rate δ): *For any constants $\delta > 0$ and $\gamma > 1$, setting $t = \text{poly}(1/\delta)$, there exists a uniform $\mathcal{AC}^0$ function $E : (\{0,1\}^n)^t \to \{0,1\}^{\Theta(\log^\gamma n)}$ that constitutes a $((\delta n, ..., \delta n), n^{-\omega(1)})$-extractor.*

Proof Sketch: We partition the t sources into two sets; a set of t_1 sources will be used to select a small cell, and the remaining $t_2 = t - t_1$ sources will be used for the actual extraction based on sub-sources determined by the selected cell. That is, as in the proof of Theorem 7.6, we partition each source into $n' = n/\ell$ blocks, each of length $\ell = \log^\gamma n$, and use two *multi*-source extractors, denoted E' and E'', for sources of length ℓ and ℓ^2, respectively. However, here the first extractor (i.e., E') uses $t_1 \gg 2$ sources and the second extractor (i.e., E'') uses $t_2 \gg 2$ sources. Furthermore, here E' extracts a slightly longer string, which is interpreted as an element in $[n'] \times [n'/\ell]$. Specifically, such $(b, c) \in [n'] \times [n'/\ell]$ is interpreted as placing the b^{th} block in cell number c. That is, the pair extracted from the j^{th} block (of the first t_1 sources) does not assign block j to some cell, but rather assigns a hopefully random block to a hopefully random cell, where these hopes are materialized if extraction from the j^{th} block succeeds (i.e., if this block has sufficient min-entropy in sufficiently many sources).

Specifically, the first stage of the extractor will use the iterated extraction procedure of [6]. Recall that this procedure is based on the extractor $E_3 : F^3 \to F$ that operates over the finite field F (of prime cardinality) such that $E_3(x,y,z) = xy + z$ and $|F| \approx 2^\ell$. Actually, we shall use a minor modification of E_3 that avoids zero elements by letting $E_3'(x,y,z) = g(x)g(y) + g(z)$, where

$g(0)=1$ and $g(x)=x$ for all $x \in F \setminus \{0\}$. Note that the behaviour of E_3' with respect to sources of min-entropy k is captured by the behaviour of E_3 with respect to sources of min-entropy $k-1$, since $\mathbf{H}_\infty(g(X)) \geq \mathbf{H}_\infty(X)-1$, where $\mathbf{H}_\infty(V)$ denotes the min-entropy of the random variable V. The extractor $E' : F^{3^i} \to F$ is defined recursively (as in [6]) by $E'(x_1,...,x_{3^i}) = E_3'(y_1,y_2,y_3)$, where $y_j = E'(x_{(j-1)\cdot 3^{i-1}+1},...,x_{j\cdot 3^{i-1}})$. Actually, $E'(x_1,...,x_{t_1})$ is redefined as the first $2\log n' - \log \ell$ bits of $E'(x_1,...,x_{t_1})$ (as defined above). Recall that E' will be applied n' times, where the j^{th} application takes as input $(X_j^{(1)},...,X_j^{(t_1)})$, where $X_j^{(i)}$ is the j^{th} block of the i^{th} source. We shall detail the second stage of the final extractor at a later point.

Towards analyzing this construction, we first invoke Claim 7.2.1 (using $\delta' = \delta/2$ and $\epsilon' = \delta/12$). Saying that a block j is good for the i^{th} source if its conditional min-entropy rate exceeds $\delta/3$, we infer that at least $\delta/2$ of the blocks are good for each specific source (i.e., the i^{th} source). (Actually, the bound refers to the conditional min-entropy of a source that is close to the i^{th} source.) Hence, at least $\delta/4$ of the blocks are good for at least $\delta/4$ of the (first t_1) sources. Let us call such a block good, and analyze extraction from it using the two foregoing observations. We establish the second observation by proving the following.

Claim 7.7.1 (analysis of the tree marking game): *Consider a full ternary tree of height h such that at least $\rho \cdot 3^h$ leaves are assigned the value 1 and the other leaves are assigned the value 0. Assign an internal node the value $i+1$ if all its children are assigned the value $i>0$, and otherwise assign it the maximum value that is assigned to its children. Then, for every fixed $\rho \geq 0.9^h$, the value assigned to the root is at least $h/100$.*

Proof: Let us turn the table around and denote by $N_h(i)$ the maximal number of leaves that may be assigned the value 1 in a ternary tree in which the root is assigned a value that is smaller or equal to i. Clearly, $N_h(i) \geq N_h(i-1)$ and $N_{h+1}(i) \geq N_h(i)$. Note that for a tree consisting of a single vertex (i.e., $h=0$), it holds that $N_0(0)=0$ and $N_0(i)=1$ for every $i \geq 1$. The key observation is that

$$N_{h+1}(i) = \max(3 \cdot N_h(i-1), 2 \cdot N_h(i) + N_h(i-1)) = 2 \cdot N_h(i) + N_h(i-1).$$

It follows that $N_h(i) = 2^h + \sum_{j \in [h]} 2^{j-1} \cdot N_{h-j}(i-1)$, for every $i \geq 1$. Next, for $h, i \geq 1$, one can prove by induction on i that $N_h(i) = \sum_{j=0}^{i-1} 2^{h-j} \cdot \binom{h}{j}$. Our last technical claim is that $N_h(i) \geq 2.7^h$ implies $i \geq h/100$. Suppose, towards the contradiction that $i < h/100$. Then, $N_h(i) = \sum_{j=0}^{i-1} 2^{h-j} \cdot \binom{h}{j} < 2^h \cdot 2^{H_2(0.01)\cdot h}$, which is upper bounded by $2^{h+0.1h} < 2.2^h$, in contradiction to $N_h(i) \geq 2.7^h$. Turning back to the main claim, we note that if a $\delta \geq 0.9^h$ fraction of the leaves are assigned the value 1 and the root is assigned the value i, then $N_h(i) \geq \delta \cdot 3^h$, which implies that $i \geq h/100$. ■

Now, assuming that the j^{th} block is good, we claim that $E'(X_j^{(1)},...,X_j^{(t_1)})$ is $\exp(-\Omega(\ell))$-close to the uniform distribution over $[n'] \times [n'/\ell]$ (independently of prior blocks). This is the case since the j^{th} block is good for at least an $\delta/4$ fraction of the sources, whereas the values analyzed in Claim 7.7.1 (with $\rho = \delta/4$) reflect the min-entropy of the corresponding extracted field element. Specifically, leaves assigned the value 1 correspond to blocks with (conditional) min-entropy rate of at least $\delta/3$; internal nodes with value $i+1$ having children that have value $i>0$ correspond to the application of E_3' to random variables of min-entropy rate at least $\min((1+\Omega(1))^i \cdot \delta/3, 0.9)$ (see [6, Lem. 3.1]); whereas the other internal nodes correspond to the application of E_3' that maintain the maximum min-entropy rate of the three random variables to which E_3' is applied.[38] Hence, using $h = \Theta(\log(1/\delta))$, we obtain at level $h-1$ of the tree (i.e., at the children of the root) three independent sources such that each of them has (conditional) min-entropy rate at least 0.9, where the conditioning is over the values of prior blocks. Using [42, Lem. 13], while relying on the fact that $E_3(r, y, s)$ may be viewed as a universal hashing function mapping y to $ry + s$ (based on the key (r, s)), we are done.

Again, we wish to emphasize a key point regarding the foregoing probabilistic analysis. Recall that blocks are defined as good for a specific source based on their *conditional* min-entropy in that source. A block is globally good if it is good for sufficiently many sources (out of the first t_1 sources). Hence, extaction from a good block yields a quasi-random outcome, conditioned on the values of all prior blocks (and the corresponding outcomes that were extracted). As detailed next, a similar analysis will apply to the blocks (of the remaining t_2 sources) assigned to any specific cell.

It is time to detail the second stage of the final extractor. This stage uses the remaining $t_2 = t - t_1$ sources. First, the n' outcomes obtained in the first stage are interpreted as an assignment of blocks to cells; specifically, block b is assigned to cell c if the pair $(b, c) \in [n'] \times [n'/\ell]$ appears in this n'-long sequence of outcomes. Then, a cell with a minimal number of assigned blocks is picked, and the extractor E'' (which is analogous to the extractor E' that was used in the first stage) is applied to the corresponding t_2 sub-sources; specifically, if the selected cell was assigned the blocks $j_1, ..., j_{\ell'}$, where $\ell' \leq \ell$, then E'' is applied to the t_2 sub-sources $X_{j_1}^{(t_1+1)} \cdots X_{j_{\ell'}}^{(t_1+1)}, ..., X_{j_1}^{(t)} \cdots X_{j_{\ell'}}^{(t)}$.

Defining good blocks (for a specific source, and globally good blocks) as in the first stage (albeit for the last t_2 sources), we infer that (with very high probability) each cell is assigned at least $((\delta/4)^2 - o(1)) \cdot \ell$ good blocks, since at least $\delta/4$ of the assignments are random and the density of good blocks is at least $\delta/4$. (Recall that a block is good if it is good for at least a $\delta/4$ fraction of the last t_2 sources, and that being good for a source means having conditional min-entropy rate at least $\delta/3$.) In this case (i.e., for a typical assignment of blocks to cells), the average min-entropy rate of the blocks assigned to the smallest cell is at least $((\delta/4)^2 - o(1)) \cdot \delta/3 = \Omega(\delta^3)$, where the average is taken over the last t_2 sources. It follows that $\Omega(\delta^3)$ of the corresponding sub-sources (i.e., the sources restricted to the blocks assigned to a cell) have min-entropy rate of $\Omega(\delta^3)$. Applying Claim 7.7.1 (this time with $\rho = \Theta(\delta^3)$ and again with $h = \Theta(\log(1/\rho))$), we infer that the final output is close to uniform.

Finally, note that all operations (including the arithmetics in the finite fields, which have size $\exp(\text{poly}(\log n))$) can be implemented by constant-depth circuits of size $\text{poly}(n)$ (see [8, 49]).[39] ■

Trading-off sources for error. Following Barak *et al.* [7], we can reduce the number of sources to a fixed constant (i.e., five) independent of the constant min-entropy rate of these sources, but this comes at the expense of increasing the extraction error to a larger $o(1)$.

Theorem 7.8 (deterministic extractor in $\mathcal{AC}^0$ for five sources of constant rate δ): *For any constants $\delta > 0$ there exists a uniform $\mathcal{AC}^0$ function $E : (\{0,1\}^n)^5 \to \{0,1\}^{\omega(1)}$ that constitutes a $((\delta n, \delta n, \delta n, \delta n, \delta n), o(1))$-extractor.*

[38] Recall that the min-entropy of $g(X)g(Y) + g(Z)$ is at least the maximum of the min-entropy of these three (independent) random variables, where we use the fact that $\mathbf{Pr}[g(X)=0] = 0$ (and ditto for Y).

[39] Alternatively, one can use constructions over finite fields of characteristic two, and use the bounds stated in [Sec. 2.8]Z4 rather than [6, Lem. 3.14].

Indeed, the three-source extractor in [7, Thm. 1.1] is only claimed to extract a constant number of bits with constant error, but the proof (as is) establishes functions of the form $\log^{\Omega(1)} n$ and $\log^{-\Omega(1)} n$, respectively. Our result just inherits these bounds.

Proof Sketch: We follow the strategy of the proof of Theorem 7.7, while implementing the two stages of extraction in a different manner. Specifically, the first extraction stage (which used the t_1-sources extractor E') will be performed by the two-source *somewhere extractor* of [7, Thm. 6.2], whereas the second extraction stage (which used the t_2-sources extractor E'') will be performed by the three-source extractor of [7, Thm. 1.1]. These two extractors will be applied to poly($\log n$)-bit long blocks of the various sources, and they can be implemented by explicit constant-depth poly(n)-size circuits, since the computations involved reduce to applications of the three-source extractor E_3 of [6] and the bilinear extractor of [16, 19].[40]

Note that the two-source somewhere extractor of [7, Thm. 6.2] outputs a constant number of strings such that one of them is random (or rather an output-dependent choice of an element in the output sequence yields a distribution that is close to being uniform). Nevertheless, if we assign to cell c all blocks with index b such that (b, c) appeared in the output sequence of some invocation of the two-source somewhere extractor, then the analysis of the first stage remains valid, except that now the total size of the cells is a constant factor bigger. (Indeed, if $E' : \{0,1\}^\ell \times \{0,1\}^\ell \rightarrow ([n'] \times [n'/\ell])^{O(1)}$ is the two-source somewhere extractor in use, then we use the $O(1)$-long output sequence $E'(X_j^{(1)}, X_j^{(2)})$ for each $j \in [n']$.)

In the second stage, we shall use a three-source extractor E'' that can handle min-entropy rate that is a constant factor smaller than the bound used in the proof of Theorem 7.7. Note that this extractor only outputs a small number of bits with an $o(1)$ deviation from the uniform distribution.

In the analysis of both stages, we invoke Claim 7.2.1 and infer that in *each source* at least $\delta/2$ of the blocks have (conditional) min-entropy rate of at least $\delta/3$. The problem is that, in order for extraction to work, we need (many) blocks that have sufficient min-entropy in *each of* the first two sources (resp., *each of* the last three sources). This problem is solved for the first two sources by using poly($1/\delta$) matchings over $[n']$, rather than the single matching $\{(j, j)\}_{j \in [n']}$. Specifically, we use (partial) matchings that correspond to a partition of the edges of an expander graph into partial monotone functions. Such expanders are called monotone [20] and were constructed in [13]. Monotonicity is important here because the definition of a good block refers to the conditional min-entropy of the block, which in turn refers to a fixed order.

In light of the above, the first extraction stage is actually as follows. Let $f_1, ..., f_t : [n'] \rightarrow [n']$ be partial monotone functions such that $t = \text{poly}(1/\delta)$ and for any set S of density $\delta/2$ it holds that $\cup_{i \in [t]} f_i(S)$ has density at least $1 - \delta/4$. (The second condition follows by applying the exapander mixing lemma to the graph $([n'], \cup_{i \in [t]} \{(v, f_i(v)) : v \in [n']\}$.)[41] For every $j \in [n']$ and $i \in [t']$, where $t' = \text{poly}(1/\delta)$ is the length of the sequence output by $E' : \{0,1\}^\ell \times \{0,1\}^\ell \rightarrow ([n'] \times [n'/\log n])^{t'}$, we compute the t'-long sequence $E'(X_j^{(1)}, X_{f_i(j)}^{(2)})$, and assign block b to cell c if (b, c) appeared in that sequence.

Now, letting B_1 (resp., B_2) denote the set of blocks that are good for the first (resp., second) source, it follows that $|B_1|, |B_2| \geq \delta n'/2$. Hence, $(\cup_{i \in [t]} f_i(B_1)) \cap B_2$ has density at least $\delta/4$, and there exists $i \in [t]$ such that $\{(j, f_i(j)) : j \in B_1 \wedge f_i(j) \in B_2\}$ has density at least $\delta/4t$. For this i and

[40] For the second extractor (i.e., the three-source extractor of [7, Thm. 1.1]) the corresponding construction in [7, Thm. 1.1] also apply an optimal two-source extractor, but it is applied on strings of very short length.

[41] If f_i is undefined on v, then $f_i(v)$ is ignored in the various expressions; for example $f_i(S \cup \{v\}) = f_i(S)$ and $(v, f_i(v)) \cup W = W$.

for every $j \in B_1 \cap f_i^{-1}(B_2)$, it holds that the j^{th} block in the first source (resp., the $f_i(j)^{\text{th}}$ block in the second source) has min-entropy rate at least $\delta/3$ conditioned on the prior blocks, where the blocks are ordered according to j (and we use $f_i(j') < f_i(j)$ for $j' < j$). Hence, the corresponding assignment (based on extraction from these two blocks (i.e., $X_j^{(1)}$ and $X_{f_i(j)}^{(2)}$)) will be *somewhere random*, conditioned on the prior assignment. It follows that at least $\frac{\delta/4t}{t'} = \text{poly}(\delta)$ fraction of the blocks will be assigned at random, which guarantees that (w.v.h.p.) every cell will be assigned at least $\text{poly}(\delta) \cdot \ell$ random blocks. (Recall that extraction from at least $\delta/4t$ blocks will be *somewhere random*, which essentially means that one of the t' extracted outputs is random.)

In the second extraction stage we shall also use matchings of the blocks of one source with blocks of the second source, but here we need to match the blocks of one source with blocks of two other sources, which can be done just in the same manner. (Indeed, we shall use a Monotone expander for density $\delta/2$ and monotone degree $t = \text{poly}(\delta)$ to determine the matching of blocks of $X^{(3)}$ and blocks of $X^{(4)}$ (as done for $X^{(1)}$ and $X^{(2)}$), but we shall use a Monotone expander for density $\delta/4t$ and monotone degree $t'' = \text{poly}(\delta/t)$ to determine the matching of blocks of $X^{(3)}$ and blocks of $X^{(5)}$.) ■

8 Open Problems (collected and restated)

In this section we collect and restate open problems that are mentioned in various parts of the paper. We start with problems regarding general min-entropy sources.

Extraction from general min-entropy sources. The main open problem was stated as Problem 1.6. It refers to extracting more than poly-logarithmically many bits from a source of constant (or even $1/\text{poly}(\log n)$) min-entropy rate using a logarithmically long seed. More generally, we have –

Open Problem 8.1 (Problem 1.6, restated): *Can extractors computable in $\mathcal{AC}^0$ achieve an extraction rate that is greater than a poly-logarithmic function* (i.e., $m(n)/r(n) > \text{poly}(\log n)$) *when the min-entropy rate is at least* $1/\text{poly}(\log n)$ *and* $r(n) = \Omega(\log n)$*?*

Recall that for $k(n) \geq n/\text{poly}(\log n)$ and $r(n) = O(\log n)$, we have $\mathcal{AC}^0$-extractors either for the case of $m(n) = r(n) + \Omega(\log n)$ and $\epsilon(n) = 1/\text{poly}(n)$ (see Part 1 of Corollary 3.4) or for the case of $m(n) = \text{poly}(\log n)$ and $\epsilon(n) = 1/\text{poly}(\log n)$ (see Part 2 of Corollary 3.4). We left open the following problem.

Open Problem 8.2 (closing a gap w.r.t the error parameter): *For* $k(n) \geq n/\text{poly}(\log n)$ *and* $r(n) = O(\log n)$*, can $\mathcal{AC}^0$-extractors extract more that logarithmically many bits with* $1/\text{poly}(n)$ *error? Moreover, for any polynomial p, can they extract* $p(\log n)$ *bits with error* $\epsilon(n) = 1/p(n)$*?*

Turning back to Problem 8.1, recall that a positive resolution of Problem 4.10 (which refers to converting general sources to block sources) is a sufficient but unnecessary condition for a positive resolution of a version of Problem 8.1 (when both are qualified with respect to constant min-entropy rate).

Open Problem 8.3 (Problem 4.10, restated): *Does there exist a* $(\delta, \delta', o(1))$*-blockers* $S : \{0,1\}^{O(\log n)} \to ([n]^s)^m$ *for constants* $\delta \in (0,1)$ *and* $\delta' \in (0,\delta)$ *and some unbounded* $m = \omega(1)$*? What about* $m = n^{\Omega(1)}$ *and* $s = \text{poly}(\log n)$*? And what about* $\delta \geq 1/\text{poly}(\log n)$ *and* $\delta - \delta' \geq 1/\text{poly}(\log n)$*?*

Extraction from restricted sources. While we proved the existence of deterministic $\mathcal{AC}^0$-extractors for bit-fixing sources and for pairs of independent sources, our results in these cases have deficiencies.

Firstly, in the two-source model we only obtained deterministic $\mathcal{AC}^0$-extractors for entropy rate that tends to 1 (see Theorem 7.2). Recall that non-explicit functions (outside of $\mathcal{AC}^0$) can extract from pairs of sources of logarithmic min-entropy [16, Thm. 7], but the impossibility results in Section 5.2 can be adapted to rule out min-entropy rates below $1/\mathrm{poly}(\log n)$. On the other hand, above that level of min-entropy $\mathcal{AC}^0$-extraction *with logarithmic seed length* is possible. Hence, it is reasonable to ask whether deterministic $\mathcal{AC}^0$-extractors can meet this performance (as in the case of bit-fixing sources).

Open Problem 8.4 (two-source deterministic extraction in $\mathcal{AC}^0$): *Does there exists a* $((k,k),\epsilon)$*-extractor in* $\mathcal{AC}^0$ (even extracting just a single bit) *for any* $k(n) = n/\mathrm{poly}(\log n)$ *and* $\epsilon(n) = 1/\mathrm{poly}(n)$*?*

For starters, one may consider the case of constant min-entropy rate; that is, *is any* $(\Omega(n),\Omega(n),\Omega(1))$*-extractor computable in* $\mathcal{AC}^0$? Recall that $\mathcal{AC}^0$ circuits can extract from four sources of some constant (i.e., 0.99) min-entropy rate (cf. Theorem 7.6), and they can extract from $\mathrm{poly}(1/\delta)$-many sources of any constant min-entropy rate $\delta > 0$ (cf. Theorem 7.7).

We mention that, as shown by Chor and Goldreich [16, Sec. 4], any function that is a $(\Omega(n),\Omega(n),\Omega(1))$-extractor has $\Omega(n)$ distributional communication complexity *with respect to the uniform distribution.* Whether such a function exists in $\mathcal{AC}^0$ is open. (Note that the $\Omega(n)$ distributional communication complexity of set disjointness is not with respect to the uniform distribution.)

A second deficiency in the aforementioned results regarding deterministic extractors is that these $\mathcal{AC}^0$-extractors are non-explicit. Hence, we ask for explicit versions of these results. Since all results for the bit-fixing source model are obtained by uniform $\mathcal{AC}^0$-reductions from Theorem 5.7, it suffices to have an explicit version of the latter.

Open Problem 8.5 (explicit version of Theorem 5.7 – deterministic extraction in uniform $\mathcal{AC}^0$ for bit-fixing sources): *For every* $k(n) \geq n/\mathrm{poly}(\log n)$ *and every* $\epsilon(n) > 1/\mathrm{poly}(\log n)$*, present deterministic* ϵ*-error extractors* $E : \{0,1\}^n \to \{0,1\}$ *for* (n,k)*-bit-fixing sources such that the extractors are computable in* uniform $\mathcal{AC}^0$.

For extraction from pairs of sources, our first challenge is to provide an explicit construction that matches the parameters of Theorem 7.2. However, since we believe that Theorem 7.2 can be improved w.r.t the required min-entropy rate, we state a more general challenge.

Open Problem 8.6 (two-source deterministic extraction in uniform $\mathcal{AC}^0$): *For every* k,ϵ *such that the existence of* $((k,k),\epsilon)$*-extractor in* $\mathcal{AC}^0$ *is established, provide an explicit construction* (i.e., a construction in uniform $\mathcal{AC}^0$).

An intermediate step towards resolving Problems 8.5 and 8.6 is providing an explicit construction of a deterministic extractor for *non-oblivious* bit-fixing sources with parameters matching those in Theorem 5.6. Such an explicit construction would be of independent interest. Actually, for our current applications, it will suffice to establish the following:

Open Problem 8.7 (explicit version of Theorem 5.6 – deterministic extraction in uniform $\mathcal{AC}^0$ for non-oblivious bit-fixing sources): *For some* $\rho = 1/\text{poly}(\log n)$*, provide* uniform $\mathcal{AC}^0$ *circuits* $C : \{0,1\}^n \to \{0,1\}$ *such that* C *is balanced and the influence of every set of density* ρ *on* C *is at most* 0.1*.*

Recall that the non-explicit circuits of Ajtai and Linial [4] (i.e., Theorem 5.6) support $\rho = 1/O(\log n)^2$.

Acknowledgments

We are grateful to Ronen Shaltiel and Salil Vadhan for useful discussions.

References

[1] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1–48, 1983.

[2] Miklós Ajtai. Approximate counting with uniform constant-depth circuits. In *Advances in computational complexity theory*, pages 1–20. Amer. Math. Soc., Providence, RI, 1993.

[3] Miklos Ajtai and Michael Ben-Or. A Theorem on Probabilistic Constant Depth Computations. In *16th ACM Symposium on the Theory of Computing*, 471–474, 1984.

[4] Miklos Ajtai and Nathan Linial. The influence of large coalitions. *Combinatorica*, Vol. 13 (2), pages 129–145, 1993.

[5] László Babai. Random oracles separate PSPACE from the polynomial-time hierarchy. *Information Processing Letters*, 26(1):51–53, 1987.

[6] Boaz Barak, Russell Impagliazzo, and Avi Wigderson. Extracting Randomness Using Few Independent Sources. *SIAM Journal on Computing*, Vol. 36(4), pages 1095–1118, 2006. Preliminary version in *45th FOCS*, 2004.

[7] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. Simulating independence: new constructions of condensers, ramsey graphs, dispersers, and extractors. *Journal of the ACM*, Vol. 57(4), 2010. Preliminary version in *37th STOC*, 2005.

[8] Paul Beame, Stephen Cook, and James Hoover. Log-Depth Circuits for Division and Related Problems. *SIAM Journal on Computing*, Vol. 15, pages 994–1003, 1986.

[9] Johannes Blomer, Richard M. Karp and Emo Welzl. The rank of sparse random matrices over finite fields. *Random Struct. Algorithms*, Vol. 10(4), pages 407–419, 1997.

[10] Andrej Bogdanov and Siyao Guo. Sparse extractor families for all the entropy. In *ACM Innovations in Theoretical Computer Science conf. (ITCS)*, pages 553–560, 2013.

[11] Ravi B. Boppana. The Average Sensitivity of Bounded-Depth Circuits. *Information Processing Letters*, 63(5), pages 257–261, 1997.

[12] Jean Bourgain. More on the Sum-Product Phenomenon in Prime Fields and its Applications. *Int. J. Number Theory*, Vol. 1, pages 1–32, 2005.

[13] Jean Bourgain and Amir Yehudayoff. Monotone Expansion. In *44th ACM Symposium on the Theory of Computing*, pages 1061–1078, 2012.

[14] Larry Carter and Mark N. Wegman. Universal Hash Functions. *Journal of Computer and System Science*, Vol. 18, pages 143–154, 1979.

[15] Shiva Chaudhuri and Jaikumar Radhakrishnan. Deterministic restrictions in circuit complexity. In *28th ACM Symposium on the Theory of Computing*, pages 30–36, 1996.

[16] Benny Chor and Oded Goldreich. Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity. *SIAM Journal on Computing*, Vol. 17(2), pages 230–261, 1988. Preliminary version in *26th FOCS*, 1985.

[17] Benny Chor, Joel Friedman, Oded Goldreich, Johan Hastad, Steven Rudich, and Roman Smolensky. The Bit Extraction Problem or t-Resilient Functions. In *26th IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[18] Gil Cohen and Igor Shinkar. Zero-Fixing Extractors for Sub-Logarithmic Entropy. *ECCC*, TR14-160, 2014.

[19] Yevgeniy Dodis, Ariel Elbaz, Roberto Oliveira, and Ran Raz. Improved Randomness Extraction from Two Independent Sources. In *8th RANDOM*, Springer LNCS, pages 334–344, 2004.

[20] Zeev Dvir and Avi Wigderson. Monotone Expanders: Constructions and Applications. *Theory of Computing*, Vol. 6(1), pages 291–308, 2010.

[21] Uriel Feige. Noncryptographic Selection Protocols. In *40th IEEE Symposium on Foundations of Computer Science*, pages 142–153, 1999.

[22] Ehud Friedgut. Influences in Product Spaces: KKL and BKKKL Revisited. *Comb., Prob. & Comp.*, Vol. 13(1), pages 17–29, 2004.

[23] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[24] Ariel Gabizon and Ran Raz. Deterministic extractors for affine sources over large fields. *Combinatorica*, Vol. 28 (4), pages 415–440, 2008. Preliminary version in *46th FOCS*, 2005.

[25] Ariel Gabizon, Ran Raz, and Ronen Shaltiel. Deterministic Extractors for Bit-Fixing Sources by Obtaining an Independent Seed. *SIAM Journal on Computing*, Vol. 36 (4), pages 1072–1094, 2006. Preliminary version in *45th FOCS*, 2004.

[26] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. Cambridge University Press, 2008.

[27] Oded Goldreich. A Sample of Samplers: A Computational Perspective on Sampling. In *Studies in Complexity and Cryptography*. Springer LNCS 6650, pages 302–332, 2011.

[28] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures and Algorithms*, Vol. 11(4), pages 315–343, 1997. Preliminary version in *26th STOC*, 1994.

[29] Oded Goldreich and Avi Wigderson. Derandomizing algorithms that err extremely rarely. In *46th ACM Symposium on the Theory of Computing*, pages 109–118, 2014. Full version available from *ECCC*, TR13-152, 2013.

[30] Venkatesan Guruswami, Christopher Umans, and Salil P. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM*, Vol. 56(4), 2009. Preliminary version in *ECCC*, TR06-134, 2006.

[31] Johan Hastad. Almost Optimal Lower Bounds for Small Depth Circuits. *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pages 143–170, 1989. Extended abstract in *18th STOC*, 1986.

[32] Alexander Healy and Emanuele Viola. Constant-Depth Circuits for Arithmetic in Finite Fields of Characteristic Two. In *STACS*, pages 672–683, 2006. See also *ECCC*, TR05-087, 2005.

[33] Russell Impagliazzo and Moni Naor. Efficient cryptographic schemes provably as secure as subset sum. *J. of Cryptology*, 9(4):199–216, 1996.

[34] Russell Impagliazzo, Ramamohan Paturi, and Michael E. Saks. Size-Depth Tradeoffs for Threshold Circuits. *SIAM Journal on Computing*, Vol. 26(3), pages 693–707, 1997. Preliminary version in *25th STOC*, 1993.

[35] Russell Impagliazzo and Avi Wigderson. P=BPP if E requires exponential circuits: Derandomizing the XOR Lemma. In *29th ACM Symposium on the Theory of Computing*, pages 220–229, 1997.

[36] Statsys Jukna. *Boolean Function Complexity: Advances and Frontiers.* Algorithms and Combinatorics, Vol. 27, Springer, 2012.

[37] Xin Li. Extractors for a Constant Number of Independent Sources with Polylogarithmic Min-Entropy. In *54th IEEE Symposium on Foundations of Computer Science*, pages 100–109, 2013,

[38] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant Depth Circuits, Fourier Transform, and Learnability. *Journal of the ACM*, Vol. 40(3), pages 607–620, 1993. Preliminary version in *30th FOCS*, 1989.

[39] Jeff Kahn, Gil Kalai, and Nathan Linial. The Influence of Variables on Boolean Functions (Extended Abstract). In *29th IEEE Symposium on Foundations of Computer Science*, pages 68-80, 1988.

[40] Jesse Kamp and David Zuckerman. Deterministic Extractors for Bit-Fixing Sources and Exposure-Resilient Cryptography. *SIAM Journal on Computing*, Vo. 36(5), pages 1231–1247, 2007. Preliminary version in *44th FOCS*, 2003.

[41] Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, Vol. 8(3), pages 261–277, 1988. Preliminary version in *18th STOC*, 1986.

[42] Yishay Mansour, Noam Nisan, and Prasoon Tiwari. The Computational Complexity of Universal Hashing. *Theor. Comput. Sci.*, Vol. 107(1), pages 121–133, 1993. Preliminary version in *22nd STOC*, 1990.

[43] Noan Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, Vol. 11 (1), pages 63–70, 1991.

[44] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Science*, Vol. 49, No. 2, pages 149–167, 1994. Preliminary version in *29th FOCS*, 1988.

[45] Noam Nisan and David Zuckerman. Randomness is Linear in Space. *Journal of Computer and System Science*, Vol. 52 (1), pages 43–52, 1996. Preliminary version in *25th STOC*, 1993.

[46] Ryan O'Donnell. *Analysis of Boolean Functions Hardcover*. Cambridge University Press, 2014.

[47] Prabhakar Ragde and Avi Wigderson. Linear-Size Constant-Depth Polylog-Treshold Circuits. *Information Processing Letters*, Vol. 39(3), pages 143–146, 1991.

[48] Ran Raz, Omer Reingold, Salil P. Vadhan. Error Reduction for Extractors. In *40th IEEE Symposium on Foundations of Computer Science*, pages 191–201, 1999.

[49] John Reif. On Threshold Circuits and Polynomial Computation. In *2nd Conf. on Structure in Complexity Theory*, pages 118–123, 1987.

[50] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting Randomness via Repeated Condensing. *SIAM Journal on Computing*, Vol. 35 (5), pages 1185–1209, 2006. Preliminary version in 41st FOCS, 2000.

[51] Miklos Santha and Umesh V. Vazirani. Generating Quasi-random Sequences from Semi-random Sources. *Journal of Computer and System Science*, Vol. 33(1), pages 75–87 (1986). Preliminary version in 25th FOCS, 1984.

[52] Ronen Shaltiel. Recent Developments in Explicit Constructions of Extractors. In *Current Trends in Theoretical Computer Science: The Challenge of the New Century, Vol 1: Algorithms and Complexity*, World scietific, 2004. (Editors: G. Paun, G. Rozenberg and A. Salomaa.) Preliminary version in *Bulletin of the EATCS 77*, pages 67–95, 2002.

[53] Luca Trevisan. Extractors and Pseudorandom Generators. *Journal of the ACM*, Vol. 48 (4), pages 860–879, 2001. Preliminary version in *31st STOC*, 1999.

[54] Salil P. Vadhan. Constructing Locally Computable Extractors and Cryptosystems in the Bounded-Storage Model. *Journal of Cryptology*, Vol. 17(1), pages 43–77, 2004.

[55] Umesh V. Vazirani. Towards a Strong Communication Complexity Theory or Generating Quasi-Random Sequences from Two Communicating Slightly-random Sources (Extended Abstract). In the *17th ACM Symposium on the Theory of Computing*, pages 366–378, 1985.

[56] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, Vol. 13 (3-4), pages 147–188, 2005. Preliminary version in *18th CCC*, 2003.

[57] Emanuele Viola. On Constructing Parallel Pseudorandom Generators from One-Way Functions. In the *20th* IEEE Conference on Computational Complexity, pages 183–197, 2005.

[58] Emanuele Viola. On Approximate Majority and Probabilistic Time. *Computational Complexity*, Vol. 18 (3), pages 337–375, 2009. Preliminary version in *22nd CCC*, 2007.

[59] Andrew Yao. Separating the polynomial-time hierarchy by oracles. In *26th IEEE Symp. on Foundations of Computer Science (FOCS)*, pages 1–10, 1985.

[60] David Zuckerman. General Weak Random Sources. In *31st IEEE Symposium on Foundations of Computer Science*, pages 534–543, 1990.

[61] David Zuckerman. Simulating BPP Using a General Weak Random Source. In *32nd IEEE Symposium on Foundations of Computer Science*, pages 79–89, 1991.

[62] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, Vol. 11(4), pages 345–367, 1997. Preliminary version in *28th STOC*, 1996.

[63] David Zuckerman. Linear Degree Extractors and the Inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, Vol. 3, pages 103–128, 2007.

Appendices

Appendix A.1 presents some observations made in passing, regarding a notion used in this work. In contrast, Appendix A.2 presents a known (but not well-known) result that is used in this work. Lastly, Appendix A.3 presents a somewhat different solution to the problem of counting upto $\mathrm{poly}(\log n)$ in $\mathcal{AC}^0$.

A.1 On the robustness of averaging samplers

We prove two "robustness claims" regarding the notion of (relaxed) averaging samplers (as in Definition 2.6). These claims are not essential to the current write-up, but we consider it worthy to present them. The first claim asserts that samplers that perform well for Boolean functions also perform well for general functions.

Definition A.1 (Boolean averaging samplers, a restriction of Definition 2.6): *A function $S : \{0,1\}^r \to [n]^t$ is called a* Boolean (μ,μ',γ)-averaging sampler *if* Eq. (1) *holds for every $f : [n] \to \{0,1\}$ such that $\rho(f) \stackrel{\rm def}{=} \mathbf{E}_{i\in[n]}[f(i)] \geq \mu$.*

The aforementioned claim, stated and proved next, is analogous to the corresponding claim for standard averaging samplers (cf. [27, Thm. 5.10]).

Claim A.2 (Boolean vs general averaging samplers): *Let $\mu,\mu',\epsilon \in (0,1]$ and $n > t = \Omega(\mu\epsilon^{-2}\log(1/\gamma))$. If $S : \{0,1\}^r \to [n]^t$ is a Boolean (μ,μ',γ)-averaging sampler, then S is a $(\mu+\epsilon,\mu'-\epsilon,3\gamma)$-averaging sampler.*

The hypothesis $n > t = \Omega(\mu\epsilon^{-2}\log(1/\gamma))$ is not a real restriction when $\epsilon = \Omega(\mu - \mu')$, since any (μ,μ',γ)-sampler must satisfy it (cf., e.g., the discussion in [27]).[42]

Proof: The proof mimics the proof of the corresponding claim for standard averaging samplers (cf. [27, Thm. 5.10]). For any function $f : [n] \to [0,1]$ such that $\rho(f) \geq \mu + \epsilon$, we consider a mental experiment in which the sampler is invoked on a random Boolean function that reflects f. Specifically, we consider a random Boolean function $b : [n] \to \{0,1\}$ such that $b(i) = 1$ with probability $f(i)$ and $b(i) = 0$ otherwise, for every $i \in [n]$ and independently of the setting of all other values. Then, with probability at least $1 - \exp(-(\epsilon/\mu)^2 \cdot \mu n) > 1 - \gamma$ (over the choice of b), it holds that $\rho(b) \geq \rho(f) - \epsilon \geq \mu$. In this case, with probability at least $1 - \gamma$ (over the choice of I when b is fixed arbitrarily), the sample I chosen by the Boolean averaging sampler satisfies $\sum_{i\in I} b(i) \geq \mu'$. Lastly, with probability at least $1 - \exp(-(\epsilon/\mu)^2 \cdot \mu t) > 1 - \gamma$ (over the choice of b when I is fixed arbitrarily), it holds that $\sum_{i\in I} f(i) \geq \sum_{i\in S} b(i) - \epsilon \geq \mu' - \epsilon$. The claim follows. ■

The second claim asserts an upwards translation of the performance guarantee of (relaxed) averaging samplers. Note that the translation (from (μ,μ',γ) to $(m\cdot\mu, m\cdot\mu', m\cdot\gamma)$) preserves the relative error (i.e., $(\mu-\mu')/\mu$) rather that the absolute error (i.e., $\mu - \mu'$).

Claim A.3 (upward translation of the performance guarantee): *If $S : \{0,1\}^r \to [n]^t$ is a (μ,μ',γ)-averaging sampler, then for every $m \in \mathbb{N}$ it holds that S is a $(m\cdot\mu, m\cdot\mu', m\cdot\gamma)$-averaging sampler. The same holds with respect to Boolean averaging samplers.*

[42] Indeed, one alternative ending of the proof of Claim 3.2.2 uses Claim A.2; in that application, μ and μ' are both constants in $(0,1)$, and $\epsilon = (1 - mu)/2$.

The claim holds trivially for any (μ,μ',γ)-averaging sampler that is actually a standard averaging sampler (i.e., one that approximates the average value of any function up to an addition term of $\mu-\mu'$ with error probability $1-\gamma$).

Proof: The claim is proved by considering auxiliary functions that share the "weight" of the target function f such that each auxiliary function takes approximately an equal share of the weight of f. Specifically, given $f:[n]\to[0,1]$ such that $\mathbf{E}_{i\in[n]}[f(i)]\geq m\cdot\mu$, consider m auxiliary functions $f_j:[n]\to[0,1]$ such that $f(i)=\sum_{j\in[m]}f_j(i)$ and $\mathbf{E}_{i\in[n]}[f_j(i)]\geq\mu$ for every $j\in[m]$. Note that if f is Boolean then (w.l.o.g.) so are the f_j's. Now, by Eq. (1), for every $j\in[m]$ it holds that

$$\mathbf{Pr}_{I\leftarrow S(U_r)}\left[\frac{1}{t}\sum_{i\in I}f_j(i)<\mu'\right]\leq\gamma$$

and the claim follows by a union bound. ■

A.2 A standard high moment inequality

The following concentration bound for somewhat independent random variables is well known to the experts, but is hard to find in standard texts.

Lemma A.4 (folklore): *Let $\zeta_1,...,\zeta_n$ be identical random variables that are distributed in $[0,1]$ in a $2k$-wise independent manner, and let $M=\mathbf{E}[\sum_{i\in[n]}\zeta_i]$. Then, for any $\epsilon>0$, it holds that*

$$\mathbf{Pr}\left[\left|\sum_{i\in[n]}\zeta_i-M\right|>\epsilon\cdot M\right]<\left(\epsilon^{-2}k^2/M\right)^k_.$$

Proof: Let $\overline{\zeta}_i=\zeta_i-\mathbf{E}[\zeta_i]$ and note that $\mathbf{E}[\overline{\zeta}_i]=0$. By Markov's inequality and linearity of expectation, we have

$$\begin{aligned}\mathbf{Pr}\left[\left|\sum_{i\in[n]}\zeta_i-M\right|>\epsilon\cdot M\right] &\leq \frac{\mathbf{E}\left[\left(\sum_{i\in[n]}\overline{\zeta}_i\right)^{2k}\right]}{(\epsilon\cdot M)^{2k}}\\ &= \frac{\sum_{i_1,...,i_{2k}\in[n]}\mathbf{E}\left[\prod_{j\in[2k]}\overline{\zeta}_{i_j}\right]}{(\epsilon\cdot M)^{2k}} \qquad (26)\end{aligned}$$

Now, the key observation is that terms in which some variable appears exactly once do not contribute to the sum in Eq. (26). In general, by virtue of $2k$-wise independence, for any $t\leq 2k$, a term in which variables indexed $j_1,...,j_t$ occur with multiplicities $e_1,...,e_t$ contributes $\mathbf{E}\left[\prod_{\ell\in[t]}\overline{\zeta}_{j_\ell}^{e_\ell}\right]=\prod_{\ell\in[t]}\mathbf{E}[\overline{\zeta}_{j_\ell}^{e_\ell}]$. Denoting by $S(n,2k,j)$ the set of $2k$-long sequences over $[n]$ in which exactly j variables appear and each of these variables appears with multiplicity at least two, we have

$$\begin{aligned}\sum_{i_1,...,i_{2k}\in[n]}\mathbf{E}\left[\prod_{j\in[2k]}\overline{\zeta}_{i_j}\right] &= \sum_{j\in[k]}\sum_{(i_1,...,i_{2k})\in S(n,2k,j)}\mathbf{E}\left[\prod_{j\in[2k]}\overline{\zeta}_{i_j}\right]\\ &\leq \sum_{j\in[k]}|S(n,2k,j)|\cdot(M/n)^j\end{aligned}$$

where the inequality uses the fact that for every $e \geq 2$ it holds that $\mathbf{E}[\overline{\zeta}_i^e] \leq \mathbf{E}[\zeta_i^e] \leq \mathbf{E}[\zeta_i] = M/n$. Using $|S(n,2k,j)| < \binom{n}{j} \cdot j^{2k} < n^j \cdot k^{2k}/2$, we get

$$\begin{aligned} \mathbf{Pr}\left[\left|\sum_{i\in[n]} \zeta_i - M\right| > \epsilon \cdot M\right] &< \frac{0.5k^{2k} \cdot \sum_{j\in[k]} n^j \cdot (M/n)^j}{(\epsilon \cdot M)^{2k}} \\ &< \frac{k^{2k}}{(\epsilon^2 \cdot M)^k} \end{aligned}$$

and the lemma follows. ■

A.3 Counting few ones in a long string

We consider the problem of counting the number of ones in a string, when guaranteed that this number is small. Specifically, for $\ell = \text{poly}(\log n)$, given an n-bit string x such that $s \stackrel{\text{def}}{=} \sum_{i\in[n]} x_i \leq \ell$, we seek explicit $\mathcal{AC}^0$-circuits that compute s. A solution to this problem has been known for decates; see [47], improving over [3, Sec. 5] (and subsequent works). For sake of elegancy, we present a somewhat different solution here.

We use a small (i.e., $\text{poly}(n)$-sized) familty of pairwise independent hash functions mapping $[n]$ to $[\ell^2]$. Such functions can be described by string of length $2\log n$, and so they can be generically evaluated by depth-two circuits of $\text{poly}(n)$ size. A random function in such family H shatters any set of size ℓ with constant probability; that is, for every set $I \subset [n]$ such that $|I| \leq \ell$ it holds that $\mathbf{Pr}_{h\in H}[|h(I)| = |I|] = \Omega(1)$ (since the collision probability is $1/\ell^2$).

Now, on input x as above, we enumerate all $h \in H$, compute for every h the value of $\sum_{v\in[\ell^2]}(\bigvee_{i\in h^{-1}(v)} x_i)$, and take the maximum value. Indeed, if h shatters $I = \{i : x_i = 1\}$, then the value we computed equals $\sum_{i\in[n]} x_i$. The above computation is in $\mathcal{AC}^0$ since we compute a ℓ^2-wise sum (of some unbounded ors).[43]

[43] Indeed, $\bigvee_{i\in h^{-1}(v)} x_i$ can be computed as $\bigvee_{i\in[n]}(x_i \vee (h(i) = v))$.