Almost Chor-Goldreich Sources and Adversarial Random Walks[*]

Dean Doron
deand@bgu.ac.il
Ben Gurion University of the Negev
Be'er Sheva, Israel

Dana Moshkovitz
danama@cs.utexas.edu
University of Texas at Austin
Austin, Texas, USA

Justin Oh
sjo@cs.utexas.edu
University of Texas at Austin
Austin, Texas, USA

David Zuckerman
diz@cs.utexas.edu
University of Texas at Austin
Austin, Texas, USA

ABSTRACT

A Chor–Goldreich (CG) source is a sequence of random variables $X = X_1 \circ \ldots \circ X_t$, where each $X_i \sim \{0,1\}^d$ and X_i has δd min-entropy conditioned on any fixing of $X_1 \circ \ldots \circ X_{i-1}$. The parameter $0 < \delta \leq 1$ is the entropy rate of the source. We typically think of d as constant and t as growing. We extend this notion in several ways, defining *almost* CG sources. Most notably, we allow each X_i to only have conditional *Shannon entropy* δd.

We achieve pseudorandomness results for almost CG sources which were not known to hold even for standard CG sources, and even for the weaker model of Santha–Vazirani sources: We construct a *deterministic condenser* that on input X, outputs a distribution which is close to having *constant entropy gap*, namely a distribution $Z \sim \{0,1\}^m$ for $m \approx \delta dt$ with min-entropy $m - O(1)$. Therefore, we can simulate any randomized algorithm with small failure probability using almost CG sources with *no* multiplicative slowdown. This result extends to randomized *protocols* as well, and any setting in which we cannot simply cycle over all seeds, and a "one-shot" simulation is needed. Moreover, our construction works in an online manner, since it is based on random walks on expanders.

Our main technical contribution is a novel analysis of random walks, which should be of independent interest. We analyze walks with adversarially correlated steps, each step being entropy-deficient, on good enough lossless expanders. We prove that such walks (or certain interleaved walks on two expanders), starting from a fixed vertex and walking according to $X_1 \circ \ldots \circ X_t$, accumulate most of the entropy in X.

CCS CONCEPTS

• **Theory of computation → Pseudorandomness and derandomization**; *Random walks and Markov chains*; *Expander graphs and randomness extractors*; Complexity classes.

STOC '23, June 20–23, 2023, Orlando, FL, USA
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9913-5/23/06.
https://doi.org/10.1145/3564246.3585134

KEYWORDS

condensers, expander Graphs, extractors, random Walks, randomized algorithm, Santha–Vazirani sources

ACM Reference Format:
Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. 2023. Almost Chor-Goldreich Sources and Adversarial Random Walks. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3564246.3585134

1 INTRODUCTION

Randomness is an incredibly useful resource. The use of randomness is sometimes provably essential (e.g., in cryptography or property testing), and sometimes we conjecture it is not, prominently in time-bounded randomized algorithms. Yet, it is often the case that randomized algorithms outperform deterministic ones. However, true randomness is scarce, and often we may only be able to access a weak, defective source of randomness. This motivates the problem of simulating randomized algorithms that expect to receive true randomness, using only weak sources of randomness.

The most natural way to use a weak random source is to convert it into a high quality random source. An extractor does exactly this. Specifically, a (deterministic) extractor for a class of sources $\mathcal{X}$ over n bits is a function $\mathsf{Ext}\colon \{0,1\}^n \to \{0,1\}^m$ such that for any $X \in \mathcal{X}$ it holds that $\mathsf{Ext}(X)$ is close, in total variation distance, to U_m, the uniform distribution on m bits. Deterministic extractors are only possible for some restricted classes of sources.

For general sources $\mathcal{X}$, randomness extraction is possible with the addition of a short random seed $Y \sim \{0,1\}^\ell$, independent of X. It is not hard to see that simulation of randomized algorithms given a weak randomness source can be done by cycling over all seeds; see the well known [24, Lemma 2.10]. For a running time T, that simulation takes $2^\ell(T + t_{\mathsf{Ext}})$ time, where t_{Ext} is the time it takes to compute the extractor. Since typically $t_{\mathsf{Ext}} \leq T$, we denote by 2^ℓ the simulation's *slowdown*, and naturally we want to minimize it. Generally, the distributions that we could hope to extract from are modeled as an arbitrary probability distribution with some amount of min-entropy [15, 45], also known as k-sources.[1] Unfortunately,

[*] The full version of the paper is given in [24].

[1] We say that X is a k-source if its min-entropy is at least k, i.e., if every sequence x occurs in X with probability at most 2^{-k}.

we have a lower bound of $\ell \geq \log n + O(1)$ on the seed length of extractors for arbitrary k-sources over n bits, so simulating **BPP** with weak sources using extractors must incur at least $\Omega(n)$ slowdown.[2]

Previous research focused on two extremes: sources where deterministic extraction is possible, and hence there's a negligible slowdown, and simulations giving an $\Omega(n)$ slowdown. A basic natural question is to ask whether anything can be done in between these extremes.

1. Are there natural weak sources where deterministic extraction is impossible, but where an $o(n)$ or even constant slowdown is possible?

It turns out that an affirmative answer to this question can be inferred from previous results, as we will discuss later. However, for some applications, such as in one-shot scenarios like cryptography and interactive proofs, one cannot cycle over all seeds. In other applications, even a constant slowdown is undesirable. In such settings, a deterministic transformation is essential. We therefore ask what is feasible deterministically.

2. Are there natural weak sources where deterministic extraction is impossible, but nevertheless it is possible to deterministically transform the source into a random variable that is essentially as useful as uniform randomness in many settings?

We answer this question in the affirmative for Santha-Vazirani (SV) and Chor-Goldreich (CG) sources, and generalizations of such sources, which we call Shannon CG sources and almost CG sources, by giving constructions of deterministic condensers with constant entropy gap.

Additionally, in some situations one may not know the ultimate length of a weak random source, or one may wish to extend the length of a given transformed random variable while preserving its useful properties. This leads us to ask:

3. Can the deterministic transformations from Question 2 be computed in an online manner?

This online extraction question is of interest in cryptography [18, 19]. We also answer this question in the affirmative for our generalized notions of CG sources.

Our algorithms take a very natural approach: perform a random walk using the source as a sequence of instructions. For arbitrary sources with entropy rate $1/2$, a random walk may not mix at all: each random step may be followed by an adversarial step that reverses the random step. This raises the question:

4. Do random walks mix well in some sense for any natural weak sources with entropy rate below $1/2$?

We show that indeed it is possible to get *good mixing properties for random walks using SV sources and their generalizations.* That is, for an adversarial random walk on a sufficiently high quality expander, it suffices that each step has a small amount of fresh entropy for the walk to mix quite well. We give an overview of our analysis, which is readily applicable even beyond the scope of pseudorandomness, in section 6.

2 SANTHA–VAZIRANI SOURCES AND CHOR–GOLDREICH SOURCES

Santha–Vazirani (SV) sources [39] are sequences of random bits in which the conditional distribution of each bit given the previous ones can be partially controlled by an adversary. Namely, $X = X_1 \circ \ldots \circ X_t$, each $X_i \sim \{0,1\}$, is a δ-SV source if for any i and any prefix $a \in \{0,1\}^{i-1}$ and $b \in \{0,1\}$, it holds that $\Pr[X_i = b | X_{[1,i-1]} = a] \leq 1 - \delta/2$.[3] Chor and Goldreich [15] generalized the SV model by considering each $X_i \sim \{0,1\}^d$ and assuming that no sequence of d bits has too high a probability of being output. Formally, X is a δ-CG source if for any i and any prefix $a \in \{0,1\}^{d(i-1)}$, it holds that $H_\infty(X_i | X_{[1,i-1]} = a) \geq \delta d$, where H_∞ denotes the min-entropy. We typically think of d being constant and t growing.[4]

Santha and Vazirani showed that there is no deterministic extractor for SV sources that's better than outputting the first bit[5] [39] (see also [38]). Chor and Goldreich showed an even stronger result for CG sources.

Theorem 2.1 ([15]). *The class of δ-CG sources does not admit deterministic extraction.*

We first observe that a constant-length seed suffices to extract from CG sources (and thus SV sources). The proof is actually given in [33, Lemma 10], although there is no theorem statement to this effect (because the focus in [33] was on general min-entropy sources).

Theorem 2.2 (follows from [33]). *For any constants $0 < \varepsilon, \delta \leq 1$, there exists an ε-error extractor for δ-CG sources, with seed length $\ell = O(1)$.*

This was improved to CG sources with subconstant δ in [41, Lemma 5.3], but again there is no theorem statement. Since we believe many are not aware of this result, for completeness, we include a proof in [24, Appendix A.2] that puts it in a more general framework.

By the previously mentioned connection, Theorem 2.2 gives a simulation using CG sources with constant slowdown.[6] However, there are scenarios where even constant seed is undesirable. This work shows that there is a way to *deterministically transform* such generalized CG sources, in an online manner, into a random variable that is *essentially as useful* as a nearly uniform random variable in many scenarios. In a bit more detail, surprisingly, we show that one can simulate low-error randomized algorithms, and in general *biased distinguishers*, in a "one-shot" manner. In particular, we have the following theorem.

[2] Note that the slowdown is (at least) linear in n, and the number of random coins is $m < n$. The difference between n and m naturally depends on the entropy k that the source has. For the precise lower bounds on the parameters of extractors for arbitrary k-sources, see [34]. In terms of explicit results, for $k = \Omega(n)$, a simulation with linear slowdown follows from [46], and for arbitrary k-s we can get a polynomial slowdown (e.g., from [29, 32]).

[3] We denote $X_{[1,i-1]} = X_1 \circ \ldots \circ X_{i-1}$. Note that the X_i-s are not assumed to be independent.

[4] This is in contrast with "block-sources", which is the term often used when t is very small and d is large.

[5] We note that some variations of SV sources do admit better deterministic extraction. See [6].

[6] We give a brief overview of the construction of Theorem 2.2. Given $X_1 \circ \ldots \circ X_t$, we use a constant-sized seed Y to extract, in a "strong" sense (say, using universal hashing) a uniform Z_1 from $X_{[1,a]}$ where $a = O(1)$. Then, we use Z_1 as a seed to extract from $X_{[a+1,b]}$ to get Z_2, where $[a+1, b]$ is roughly twice as long as $[1, a]$. Continuing this way for $s = O(\log t)$ times, we use Z_s as a seed to extract from a suffix of X of length $\Omega(dt)$. The output of the final extraction is the output of the extractor.

Theorem 2.3 (informal; follows from Theorem 3.2). *There exists a deterministic, efficient, function* Cond *such that the following holds. Given a δ-CG source $X = X_1 \circ \ldots \circ X_t$, each $X_i \sim \{0,1\}^{d=O(1)}$, for any randomized algorithm A and any input w such that $A(w, y)$ errs with probability $O_{\delta,d}(\varepsilon^2)$ (over a uniform $y \sim U$), it holds that $A(w, \mathsf{Cond}(x))$ errs with probability ε (over $x \sim X$).*

The one-shot simulation via CG sources (and later we will see that such a simulation is possible with a much richer class of sources) is possible in light of our deterministic condensers, which are overviewed in section 3 (see also the discussion in section 4). We continue with the very natural generalization of CG sources that we study.

Shannon *CG Sources.* Instead of requiring that each X_i, conditioned on every prefix, has at least δd min-entropy, we only require the conditional X_i have δd Shannon entropy.[7]

While Shannon CG sources seem more general than the *almost* CG sources we define next, it turns out that strong enough results for almost CG sources imply results for Shannon CG sources. Thus, much of the technical focus of this work is on almost CG sources, with the case of Shannon CG sources following as a corollary.

Almost *CG Sources.* Instead of requiring that each X_i, conditioned on every prefix, has at least δd min-entropy, we only require the conditional X_i to be γ*-close* to some source with entropy rate δ.

Definition 2.4 (almost CG source, I). We say that $X = X_1 \circ \ldots \circ X_t$, each $X_i \sim \{0,1\}^d$, is a γ-almost δ-CG source if for any i and any prefix $a \sim X_{[1,i-1]}$, it holds that $X_i | \left\{X_{[1,i-1]} = a\right\}$ is γ-close, in total variation distance, to a source with δd min-entropy.

The definition of almost CG sources is also quite natural. In particular, considering γ-s which can be much larger than 2^{-d} is very natural and has several advantages. In particular, it is often the case that the X_i-s are a result of some prior transformations, which almost always incur some error. In fact, we already demonstrate such an example in this work. In subsection 6.2, we will see that in order to condense from an (almost) δ-CG source, we will first "condense" the original source into a γ-almost δ' CG source with $\delta' > \delta$, and some $\gamma > 0$. In Theorem 5.3 we will further extend our definition of almost CG sources.

The techniques of [33] also work to give a constant-seeded extractor for almost CG sources as defined in Theorem 2.4.

Theorem 2.5 (see Section A.3 of [24]). *For any constants $0 < \varepsilon, \delta, \gamma \le 1$, and $\gamma \ge 0$, there exists an ε-error extractor for γ-almost δ-CG sources, with seed length $\ell = O(1)$.*

For the formal statement, see [24, Corollary A.8]. Although this generalization is not hard, we stress that it was not known, and in particular requires some observations about almost CG sources provided in this work (see [24, Lemma 3.3]). Later on, we'll discuss even further extensions of CG-sources, for which the techniques of [33] completely fail, while ours do not.

3 DETERMINISTIC CONDENSING FROM ALMOST CG SOURCES

Recall that we have the following parameters:

(1) d is the length of each block, and t is the number of blocks (so X is distributed over $n = dt$ bits.);
(2) Each block X_i is γ-close to having δ entropy rate; and,
(3) m denotes the output length of our extractor (and later condenser).

Later, we will study two additional extensions for CG sources: Those with some λ-fraction of *damaged* blocks, for which we have no guarantee, and those in which for every good block, it is only guaranteed that all but some ρ-fraction of prefixes give rise to a (close to) high-entropic block.

While an extractor aims to purify a weak source X into a nearly-uniform source, a *condenser* aims to improve the source's quality, namely by increasing the entropy rate [35]. Formally,

$$\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^m$$

is a (k', ε) condenser for a class of sources $\mathcal{X}$ distributed over $\{0,1\}^n$ if for any $X \in \mathcal{X}$ and an independent and uniform $Y \sim \{0,1\}^\ell$, it holds that $\mathsf{Cond}(X, Y)$ is ε-close to a source with k' min-entropy. When $\ell = 0$, we say the condenser is *deterministic* (or seedless), and that $\mathcal{X}$ admits deterministic condensing.

The entropy *rate* of a condenser is $\frac{k'}{m}$, and we want it to be larger than $\frac{k}{n}$, where k is the min-entropy in each $X \in \mathcal{X}$. When the rate is very close to 1, i.e., when k' is very close to m, it makes sense to measure the additive difference $m - k'$.

Definition 3.1 (entropy gap). The *entropy gap* of a random variable $Z \sim \{0,1\}^m$ is $\Delta = m - H_\infty(Z)$. We say that a (k', ε) condenser Cond has entropy gap Δ if its output is ε-close to a source with entropy gap Δ. (Note that an extractor has entropy gap 0.)

Condensers were proven incredibly useful as building blocks for extractors (e.g., in [5, 29, 36, 42, 46]). Regardless, they are also of great independent interest, because:

(1) They can achieve parameters that are *unattainable* for extractors, and in particular,
(2) There are classes of sources that admit deterministic condensing and (provably) do not admit deterministic extraction.

For item 1, we give as an example the fact that for arbitrary weak sources, condensers can achieve smaller entropy loss[8] and a smaller seed length. The latter fact was used for the construction of full-fledged extractors and pseudorandom generators (see [8, 23]).

Our focus in this work is on the intriguing phenomenon described in item 2. Recall that the class of CG sources do not admit deterministic extraction. Our main result is that not only do CG sources, and even almost CG sources, admit deterministic condensing, but we are able to construct explicit condensers for such sources with *constant entropy gap*!

[7] Recall that one always have that $H(X) \ge H_\infty(X)$, for $H(\cdot)$ being the Shannon entropy. In fact, one can easily find X-s with nearly maximal Shannon entropy, but extremely low min-entropy, or even smooth min-entropy.

[8] The entropy loss of a condenser or an extractor is the difference between the input entropy and the output entropy. When $\mathcal{X}$ is the set of all k-sources, the entropy loss of a seeded extractor $\mathsf{Ext}\colon \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^m$ is $k + d - m$, and the entropy loss of a (k', ε) seeded condenser $\mathsf{Cond}\colon \{0,1\}^n \times \{0,1\}^\ell \to \{0,1\}^m$ is $k + d - k'$. In seeded condensers, the entropy loss can be zero, which is impossible for extractors (see [2, 34]).

Theorem 3.2 (see also Theorem 6.1 in [24]). *For any constants $\delta, \varepsilon, \gamma > 0$, any constant integer $d \geq 1$, the following holds. For any positive integer t, there exists an explicit function*

$$\mathsf{Cond}\colon \{0,1\}^{n=dt} \to \{0,1\}^{m=\Omega(\delta dt)}$$

such that given an almost δ-CG source X with smoothness parameter γ, $\mathsf{Cond}(X)$ is ε-close to an $(m - O(\log \frac{1}{\varepsilon}))$-source.

We view Theorem 3.2 as quite striking. It states that even a stream of constant-length random strings where each element locally appears *essentially deterministic* (for example, consider $d = 1000$ and $\delta d = 0.01$), can be readily transformed, without any additional resources, into a random variable that is almost as useful as nearly uniform randomness in many applications.

Deterministic extraction (and thus condensing) is known for several classes of sources. Some have more algebraic structure, such as uniform distributions on affine subspaces or varieties (see [12, 25] and references therein), where others are arguably better models of random sources obtained from natural physical phenomena, such as bit-fixing sources, samplable sources, small-space sources or local sources ([11, 16, 30, 43, 44] are just few examples). Our study of CG sources and almost CG sources adds to the very short list of natural classes of sources which admit deterministic condensing (even explicitly) but do not admit deterministic extraction. In [3], Ball, Goldreich, and Malkin considered the problem of condensing and extracting from two *somewhat dependent* sources. They showed that if X and Y are weak sources such that each source has bounded influence on the outcome of the other source[9], or that the mutual information $I(X, Y)$ is bounded, then condensing from X and Y is possible, whereas extraction is not. A more contrived example is a certain type of block sources which appear in [7].

4 SIMULATING TRUE RANDOMNESS WITH ALMOST CG SOURCES

The deterministic condenser guaranteed by Theorem 3.2 implies a constant-seed extractor as in Theorem 2.5. This is because there are explicit extractors for sources with constant entropy gap Δ that have seed length $O(\Delta)$ [28] (see [24, Theorem 2.12]; there are even explicit extractors with seed length $O(\log(\Delta/\varepsilon))$ [37], but they don't further improve our seed length asymptotically). We now state our more general constant-seed extractor that works even for almost CG sources.

Theorem 4.1 (see also Theorem 7.1 in [24]). *For any constants $\delta, \varepsilon, \gamma > 0$, any constant integer $d \geq 1$, the following holds. For any positive integer t there exists an explicit function*

$$\mathsf{Ext}\colon \{0,1\}^{n=dt} \times \{0,1\}^{\ell=O(1)} \to \{0,1\}^{m=\Omega(\delta dt)}$$

such that given an almost δ-CG source X with smoothness parameter γ, and an independent uniform $Y \sim \{0,1\}^{\ell}$, it holds that $\mathsf{Ext}(X, Y) \approx_{\varepsilon} U_m$.[10]

We now focus on ways in which our deterministic condenser is better than the constant-seed extractor (even for exact CG sources). We give a *one-shot simulation* of randomized protocols with almost CG sources for biased distinguishers, and particularly, a no-overhead simulation of **BPP** algorithms that err with small probability. This wasn't known even for CG sources, or even for SV sources. We discuss this next.

The Usefulness of Constant Entropy Gap. While constant seed is needed to simulate a **BPP** algorithm with error $\frac{1}{3}$ using CG sources, what if we start with an algorithm that has a very small constant error? What if we wish to simulate a *protocol* rather than an algorithm, and we cannot simply cycle over all seeds? Our next discussion is devoted to what can be done with nonzero, yet very small, entropy gap.

Consider the following simple observation.

Proposition 4.2 (see, e.g., [20]). *Let $Z \sim \{0,1\}^m$ be $\frac{\varepsilon}{2}$-close to some random variable with $m - \Delta$ min-entropy. Then, for any $\mathsf{BAD} \subseteq \{0,1\}^m$ with density at most $\rho(\mathsf{BAD}) \leq 2^{-\Delta-1}\varepsilon$, it holds that $\Pr[Z \in \mathsf{BAD}] \leq \varepsilon$.*

Thus, Theorem 3.2 implies that we can sample roughly $\frac{m}{\delta}$ bits from an almost CG source, apply our condenser, and simulate a randomized algorithm that uses m bits of randomness. As long as the algorithm's error is small enough compared to our condenser's entropy gap, we can simulate it to within a (larger) error ε, and the *only overhead we have is computing the condenser.* This is the essence of Theorem 2.3. We note that sources with small entropy gap were recently used to simulate algorithms that err rarely in the computational setting, where computational entropy is used rather than the min-entropy of Theorem 4.2 (see [23]).

Additionally, we observe that Theorem 4.2 and Theorem 2.3 suggest an *alternative* method for simulating **BPP** algorithms with constant overhead. Given a randomized algorithm A that errs with probability at most $\frac{1}{3}$, simply amplify the algorithm to error probability $2^{-\Delta-1}\varepsilon$ by considering A' that repeats A on fresh randomness a constant number of times and takes the majority vote. Then, one can simply run A' using Z as the randomness. Note this method is different than the standard one as it does not require computing an extractor at all. In other words, modulo different constant error probabilities, a source with constant entropy gap is essentially as useful as a nearly uniform source for **BPP** algorithms.

Sources with small Δ have found applications in cryptography (see, e.g., [4, 20–22]), and our one-shot generation of constant-gap sources from almost CG sources make the latter useful for those applications. In [20], Dodis, Pietrzak, and Wichs considered the notion of *biased distinguishers*, which is well-motivated in cryptography, and studied extractors that are only guaranteed to fool biased distinguishers rather than arbitrary ones. (This is also related to "slice extractors.")

Definition 4.3 (unpredictability extractor, [20]). A function

$$D\colon \{0,1\}^m \times \{0,1\}^{\ell} \to \{0,1\}$$

is a μ-distinguisher if $\mathbb{E}[D(U_m, Y)] \leq \mu$, where (U_m, Y) is uniform over $\{0,1\}^m \times \{0,1\}^{\ell}$. A function $\mathsf{UExt}\colon \{0,1\}^n \times \{0,1\}^{\ell} \to \{0,1\}^m$ is a (k, μ, ε)-unpredictability extractor if for any k-source

[9] For a discussion about the notion of bounded influence, see [3, Section 2.2], or Definition 4.1 in the ECCC version of [3].

[10] We remark that the output length $m = \Omega(\delta dt)$ can in fact be stated as $m = (1-\theta)\delta dt$ where θ is an arbitrary small constant, by slightly strengthening the constraints on the constructions' parameters. For simplicity and readability, we do not give the constraints' dependence on θ.

$X \sim \{0,1\}^n$ and any μ-distinguisher D, we have that

$$\mathbb{E}[D(\mathsf{UExt}(X,Y),Y)] \le \varepsilon,$$

where Y is uniform over $\{0,1\}^\ell$ and independent of X.

Dodis et al. showed that condensers with small entropy gap are equivalent to unpredictability extractors [20].[11] This follows from the connection between sources with small entropy gap and biased distinguishers, essentially rephrasing 4.2: For any $Z \sim \{0,1\}^m$ which is ε-close to having $m-\Delta$ min-entropy, and a μ-distinguisher $D\colon \{0,1\}^m \to \{0,1\}$, it holds that $\mathbb{E}[D(Z)] \le \varepsilon + 2^\Delta \mu$. While Dodis et al. discussed seeded primitives and arbitrary weak source, the connection between constant entropy gap and biased distinguishers readily follows to our setting as well. Concretely, Theorem 3.2 gives deterministic unpredictability extractors for almost CG sources.[12] We believe the notion of a deterministic unpredictability extractor is a very natural one and may find applications beyond the ones that stem from [20].

To conclude this section, we mention a work by Gavinsky and Pudlák on deterministic condensers for SV sources [26]. There, they studied the less-standard notion of errorless condensers, and showed that no such determinstic condenser exists for (standard) SV sources. We do allow error, which evidently does enable deterministic condensing. (Allowing error also enables seeded extraction from general weak sources, and is the standard model in pseudorandomness.) They also gave a seedless condenser for a more restrictive model than SV sources, although it doesn't have constant entropy gap.

5 ON ALMOST CG SOURCES AND THE SMOOTHNESS PARAMETER

Before presenting our technique, let us further discuss the smoothness parameter γ. Towards this end, let us introduce the notion of smooth min-entropy, which we implicitly used above. For a smoothness parameter $\alpha > 0$, we let

$$H_\infty^\alpha(X) = \max_{X' : |X - X'| \le \alpha} H_\infty(X').^{13}$$

Using this terminology, the i-th block in our almost CG source satisfies $H_\infty^\gamma(X_i | X_{[1,i-1]} = a) \ge \delta d$ for any prefix $a \sim X_{[1,i-1]}$, and the output of the condenser satisfies $H_\infty^\varepsilon(\mathsf{Cond}(X)) \ge m - O(1)$.

One could imagine the the setting of $\gamma > 0$ to be a technical extension, but successfully handling this regime draws highly nontrivial consequences. First, note that we *cannot* reduce the $\gamma > 0$ setting to the $\gamma = 0$ case via a union-bound type argument, since $\gamma t \gg 1$. It turns out that this is not simply a matter of proof technique.

Proposition 5.1 (informal; see Claim 3.14 in [24]). *There exists an almost δ-CG source with smoothness parameter γ which is far from any $(1-2\gamma)\delta$-CG source.*

Despite this, our technique does handle constant γ-s. Moreover, we emphasize that an almost CG source with $\gamma > 0$ over $dt = n$ bits may not even have $\Omega(\delta n)$ bits of entropy. To see this, consider the source $X = X_1 \circ \ldots \circ X_t$ such that for each $i \in [t]$, X_i is zero with probability γ, and an arbitrary δd-source over $\{0,1\}^d \setminus \{0\}$. Thus, $\Pr[X = 0] = \gamma^t$ and so $H_\infty(X) \le t \log \frac{1}{\gamma}$. Still, our condenser outputs a source which is close to having roughly δn bits of entropy! This implies that such an X must have ample *smooth* min-entropy. Indeed, this is the case.

Proposition 5.2 (informal; see Claim 3.13 in [24]). *Every almost δ-CG source over n bits with smoothness parameter γ has smooth min-entropy $(1-2\gamma)\delta n$.*

Such a claim follows from a technique similar to "entropy flattening" (see, e.g., [27]), where the min-entropy of a distribution X is improved by taking multiple independent copies of X.

Handling Shannon Entropy. Handling $\gamma > 0$ enables us to extend our results to Shannon CG sources. Given a Shannon δ-CG source, we show that by grouping every $O(1)$ consecutive blocks, we get an almost $\Omega(\delta^2)$-CG sources with smoothness parameter γ that is exponentially-small in the number of grouped blocks (see [24, Corollary 3.11]). Then, we can easily apply our results for almost CG sources. See [24, Theorems 6.4, 4.3] for the precise condensing and extraction results. Note that the transition from Shannon entropy to min-entropy necessarily induces error, so $\gamma > 0$ is crucial here.

Handling Damaged Blocks. Our random-walks based condensing method is flexible enough to handle damaged blocks too. Namely, we allow some λ-fraction of the i-s to have *completely arbitrary* conditional distributions.

Definition 5.3 (almost CG source, II). A (γ,λ)-almost δ-CG source is a sequence of random variables $X = X_1 \circ \ldots \circ X_t$, each $X_i \sim \{0,1\}^d$, such that for at least $(1-\lambda)t$ of the i-s, it holds that $H_\infty^\gamma(X_i | X_{[1,i-1]} = a) \ge \delta d$ for any prefix $a \sim X_{[1,i-1]}$.

When the damage pattern is arbitrary, we can condense to within $O(\lambda d t)$ entropy gap (i.e., we lose d bits of entropy for each damaged block). [24, Corollary 4.14] handles the $\lambda > 0$ setting as well. We remark that the [33, 41] technique would fail for even one damaged block. Moreover, when the damaged locations are "nicely distributed", our technique regains the $O(1)$ entropy gap. We elaborate it on this more in subsection 6.4, and give the technical details in [24, Theorems 5.4, 6.4, 7.3, 7.4].

6 OUR TECHNIQUE: A NEW ANALYSIS OF ADVERSARIAL RANDOM WALKS

Our main technical contribution is a new analysis of adversarial random walks. Let's begin our discussion with exact Chor-Goldreich sources. Spectral analysis has been the main tool to analyze random walks on expanders. However, it doesn't seem to work for CG sources with rate below 1/2. This is because there is no specialized method for CG sources; existing spectral methods that work for CG sources also work for general min-entropy sources, and general sources with rate below 1/2 do not mix at all (recall that each random step may be followed by an adversarial step that reverses the random step). Moreover, even for general sources with rate above 1/2 a random stopping time is required, which amounts to a linear number of seeds. We hope to condense without a seed or extract with a constant number of seeds.

[11] The use of biased distinguishers is also explicit in the recents works of [13, 40].

[12] We note that [20] cared about the entropy *loss*. Our condensers lose roughly a small constant fraction of the entropy, which is much more that what is attainable for seeded condensers with small entropy gap.

[13] The distance here is the total variation distance.

Furthermore, spectral methods generally exploit the Markovian nature of random walks. However, an adversarial random walk is not Markovian. That is, the distribution of the next step depends not only on the walk's current node, but also on the path it took to get there. Indeed, although it is true that the distribution of the next step from a given node v is a convex combination of instruction distributions over all the paths that end at v, the memory in the walk still presents a challenge.

Our approach uses expansion directly. We therefore use the highest quality expanders: bipartite lossless expanders.

Definition 6.1 (balanced lossless expander). We say that a D-left-regular bipartite graph $G = ([M], [M], E)$ is a $(K_{\max}, \varepsilon)$ lossless expander if for all subsets $S \subseteq [M]$ of size at most $K_{\max}$, the neighborhood set $\Gamma_G(S)$ has size at least $(1-\varepsilon)D|S|$.

For technical purposes, we will actually require that the right degree of the lossless expander be small as well. For a high-level understanding of our work, it suffices to assume that the expander is biregular.

For numerous applications a modest vertex expansion is not enough, and lossless expansion is essential.[14] An explicit construction of balanced (and somewhat imbalanced) constant-degree lossless expanders was given by Capalbo, Reingold, Vadhan, and Wigderson [10].[15] As a pseudorandomness primitive, it is instructive to think of $\Gamma_G\colon \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^m$, the neighborhood function of G, as a *lossless conductor* (where we use $\{0,1\}^m \equiv [M]$).

Definition 6.2 (balanced lossless conductor). A function

$$\mathsf{LC}\colon \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^m$$

is a $(k_{\max}, \varepsilon)$ lossless conductor if for any $k \le k_{\max}$, a k-source X, and an independent and uniform $Y \sim \{0,1\}^d$, it holds that $H^{\varepsilon}_{\infty}(\mathsf{LC}(X,Y)) \ge k + d$.[16]

That is, the output distribution "absorbs" the d bits of entropy from the seed, up to an ε error. Intuitively, the larger the vertex expansion, the less freedom the adversary has to skew the distribution over the next step. We soon make this intuition more concrete.

Our first construction, which works for large δ-s, goes as follows. Given an almost CG source $X = X_1 \circ \ldots \circ X_t$, each $X_i \sim [D]$, we walk, from a fixed node, along a $(t+1)$-partite graph with a copy of G between each two layers (the graph's size M is chosen as a function of the source's parameters). Namely, we start at some fixed $Z_0 \in [M]$, and for each $i \in [t]$, let

$$Z_i = \Gamma_G(Z_{i-1}, X_i),$$

and output $\mathrm{Cond}(X) = Z_t$.

For an exact δ-CG source, this amounts to a random walk where an adversary, after seeing previous steps, chooses D^{δ} nodes among the D neighbors, and the random walker steps to a random node among these D^{δ} nodes. We are able to show:

Theorem 6.3 (informal; see Theorem 4.8 in [24]). *Let $X_1 \circ \ldots \circ X_t$ be a δ-CG source, with each $X_i \sim \{0,1\}^d$. Let G be a sufficiently good $D = 2^d$-regular expander. Then, for any $\eta > 0$, the last step Z_t of a random walk on G, performed as above, is η-close to a $k - O(d + \log \frac{1}{\eta})$-source.*

The proof is nontrivial, and we discuss it next.

Evading the Union Bound. The naive approach to analyze the output distribution after t steps is to follow the definition of conductors. However, conductors only guarantee that the output distribution is ε-close to a distribution with appropriate entropy. Thus, even disregarding the correlation between source and seed, such an argument naturally forces us to union bound over the error of each step. Indeed, one can even show that if each instruction comes from a δd-source, and one wishes to add exactly δd entropy, then such a union bound is necessary. Our ultimate solution avoids this union bound issue, and in doing so, only argues that the entropy gain at each step is $0.9\delta d$ instead.[17]

Expansion of Weight Functions. As usual in analyzing random walks, we need to handle real nonnegative probabilities. It is standard to do this using eigenvalues, but there is a loss in going from expansion to eigenvalues, or other analytic tools such as hypercontractivity. These analytic methods don't seem to capture lossless expansion.

We give a simple way to capture lossless expansion by directly generalizing the combinatorial definition of expansion to nonnegative real numbers, which doesn't seem to have been considered before. Specifically, let 1_S denote the indicator function of a set S. Then $1_{\Gamma(S)}(v) = \vee_{w \in \Gamma(v)} 1_S(w)$. To generalize this to weight functions (nonnegative real valued functions), we replace the OR with a max. We then show that the expansion of weight functions with support size at most K exactly equals the expansion of sets with size at most K. This enables us to capture the effect of lossless expansion. We can even generalize this weighted notion to unique neighbor expansion, although it is not necessary for the proof.

6.1 The ℓ_q Norm as a Progress Measure

Recall that spectral analysis typically uses the ℓ_2 norm as a measure of progress. While the ℓ_2 norm doesn't appear to work in our setting, we manage to use the ℓ_q norm as a progress measure, for some suitable $q = 1 + \alpha$. That is, we show that the ℓ_q norm of the vertex distribution decreases by a suitable multiplicative factor at each step.

Theorem 6.4 (informal; see Lemma 4.6 in [24]). *Let $G = (U = [M], V = [M], E)$ be a bipartite D-regular (K, ε) lossless expander with error $\varepsilon = \frac{1}{D^{\beta}}$. For any $0 < \alpha < \beta$, set $q = 1 + \alpha$ and let $\delta \ge 1 - \beta + \alpha$.*

Let p_U be a probability distribution over U and let r_u, for each $u \in U$, be a distribution over $\{0,1\}^d \equiv [D]$, each being a δd source. For any $u \in U$ and $v \in V$ let $r_u(u,v)$ denote the probability that

[14] Examples can be found in coding theory, data structures, algorithms, storage models, and proof complexity (see the references in [10], and [9, 14, 17, 31] for more recent works).

[15] For very small sets, Alon showed that lossless expansion follows from high girth. See also [1]. In the regime where $M \ll N$, the degree needs to be super-constant, and explicit constructions for this regime are known (e.g., [29, 42]).

[16] The correct equivalence would be to lossless *condensers* if we allow the construction itself to depend on k (see [42]). For the sake of our discussion, this difference won't matter, and in the technical sections we will not use the lossless condensers/conductors terminology.

[17] Or $(1-\theta)\delta d$ for an arbitrary constant θ close to 0, at the expense of modifying some constraints in the construction.

the edge leading from u to v is chosen under r_u. Namely, for G's labelling function $\ell\colon E \longrightarrow [D]$ we denote $r_u(u,v) \equiv r_u(\ell(u,v))$. Define p_V as the induced probability distribution on V. That is, $p_V(v) = \sum_{u\in\Gamma(v)} r_u(u,v)p_U(u)$. Then,

$$\|p_V\|_q^q \le \frac{8}{D^{\delta\alpha}} \cdot \|p_U\|_q^q,$$

as long as $\|p_U\|_q^q$ is not already smaller than $1/K^{\alpha}$.

The ℓ_q-norm is a *proxy measure for min-entropy*, since any distribution p such that $\|p\|_q^q \le 2^{-\alpha k}$ is ε-close to a distribution with entropy $k - \frac{1}{\alpha}\log\frac{1}{\varepsilon}$ (see [24, Corollary 2.3]). Thus, Theorem 6.4 implies that every step on a lossless expander, according to a δd source, adds roughly δd bits of entropy to the vertex distribution, up to a "saturation" point of roughly $k = \log K$ bits of entropy. Since we have explicit constructions wherein $k = m - O(1)$, a saturated vertex distribution already has constant entropy gap.

One advantage of using the q-norm is that it allows us to better control the error term corresponding to the small lossy part of the lossless expander. For example, certain nodes on the right may have high degree, causing their probability after a step of a random walk to be large. This problem is exacerbated by the adversarial nature of a random walk via an almost-CG-source, which can assign up to γ probability to edges leading to high degree right nodes. By considering the q-norm for a sufficiently small α, we have a measure of entropy that is less sensitive to such error, all while still ensuring that the entropy gained at each step is roughly the same as the entropy in each instruction.

To prove Theorem 6.4, since the distribution of the random walk's vertex may not be uniform, we generalize set expansion and unique neighbor expansion to apply to "weight functions" and probability distributions. We then apply Jensen's inequality with a nonstandard choice of coefficients that heavily weights the term where we gain. This gives a simple analysis of adversarial random walks that uses expansion directly.

Overall, our analysis gives a "spectral-like" analysis of random walks even when such techniques cannot be directly applied. In addition to its application in deterministic condensing, we believe that this analysis of entropy gain via random walks from correlated and nonuniform steps is interesting on its own.

Handling Smoothness. Up until now, we did not address the smoothness parameter γ thoroughly. Quite surprisingly, it turns out that our technique based on the ℓ_q-norm analysis is flexible enough to support constant γ-s without substantial changes. Indeed, when dealing with such instructions, we extend Theorem 6.4 and show that the ℓ_q- norm decrease factor is now roughly $\frac{1}{D^{\delta\alpha}} + D^{\alpha}\gamma$. In fact, there are cases where this factor is tight. This seems unfortunate, because we are now seemingly only gaining less than $\log\frac{1}{\gamma}$ min-entropy at each step, or in other words, lose the vast majority of the desired δd bits.

The trick to overcome this is to simply pick α sufficiently small in the ℓ_q-norm analysis (recall that we set $q = 1 + \alpha$). Indeed, by choosing $\alpha \approx \frac{1}{d}\log\frac{1}{\gamma}$, we see that γ is then comparable to $\frac{1}{D^{\delta\alpha}}$. Under the assumption that $\gamma \le 2^{-O(1/\delta)}$, the decrease factor can be made to be $D^{-0.9\delta\alpha}$. Thus, we once again gain roughly 90% of the entropy at each step. Setting α this small only results in a loss of roughly $O(d)$ bits of entropy over the entire walk.[18] For the precise norm evolution with an arbitrary γ, in [24, Corollaries 4.11, 4.12], we set α accordingly.

Additionally, we observe that the assumption $\gamma \le 2^{-O(1/\delta)}$ is quite mild, as γ only depends on δ and not d. Thus, for sufficiently large d-s, $\gamma \gg D^{-O(1)}$. We note that setting α to be a small constant, say $\alpha = 1/6$, *would* require $\gamma \le D^{-O(1)}$ in order to argue that $0.9\delta d$ bits of entropy is gained at each step. We view our setting of parameter α as a way that allows us to avoid treating each instruction source as pessimistically as a $\log\frac{1}{\gamma}$-source.

The Limit of Our First Construction. We now explain why our first construction only works for large enough δ. For concreteness, assume that we are at some $Z_{i-1} \sim \{0,1\}^m$ with $H_\infty(Z_{i-1}) = k$, and walk according to $X_i \sim [D]$ having entropy δd (assume for now that $\gamma = 0$). For simplicity, assume that Z_i is flat over some set $S \subseteq [M]$ of size $K = 2^k \le K_{\mathsf{max}}$, recalling that we walk over a sequence of $(K_{\mathsf{max}}, \varepsilon)$ bipartite lossless expanders with M vertices, arranged in series. It may be informative to simply think of the walk as over a single $(K_{\mathsf{max}}, \varepsilon)$ undirected lossless expander.

While any large enough subset of S or of the edges leaving S has nice properties (for example, at least $1 - 2\varepsilon$ fraction of the vertices in S have a *unique neighbor*), there can still be ε-fraction of the KD edges leaving S that behaves badly. In particular, εKD of the edges may lead to vertices that have *many* incoming edges from S. Assume for simplicity that each node in S has the same number of bad edges, namely εD edges from each node in S lead to heavy vertices. When $D^{\delta} \le \varepsilon D$, an adversarial X_i can potentially, for each node, be supported *only* on instructions that lead to bad edges. In this case, Z_{i+1} may have accumulate neither min-entropy, nor smooth min-entropy. Thus, we must consider the case where $D^{\delta} \gg \varepsilon D$.

This raises the question of how small can we take ε to be as a function of D, or alternatively, how large can we take δ to be given an existing lossless expander. Non-explicitly, we have Γ_G-s with a great seed length, namely $d = 1 \cdot \log\frac{1}{\varepsilon} + O(1)$, in which case we can take $\varepsilon \ll D^{-(1-\delta)}$ even when $\delta > 0$ is arbitrarily small. In [10], however, the required seed length is $d = \frac{1}{\beta}\log\frac{1}{\varepsilon}$ for some constant $\beta < \frac{1}{2}$.[19] Denoting $\delta_{\mathsf{thr}} = 1 - \beta$, we see that we can only hope to handle almost δ-CG sources with $\delta > \delta_{\mathsf{thr}}$, and we do indeed achieve this. We note that both in [10] and in an optimal lossless expander, $K_{\mathsf{max}} = \Omega_D(M)$, which is good enough to lead to constant entropy gap.

6.2 Our Two-Level Construction

We handle general $\delta > 0$ via a two-level process: We first walk over a small, *optimal* lossless expander in order to simulate an instruction with sufficiently large δ, and then "flush" it as a step in the big CRVW graph over M vertices.

We are given $X = X_1 \circ \ldots \circ X_t$, each $X_i \sim \{0,1\}^d \equiv [D]$. We let $H = ([D_{\mathsf{crvw}}], [D_{\mathsf{crvw}}], E)$ be an optimal lossless expanders with degree D, and we can choose its error ε to be very close to $1/D$. The number of vertices in H corresponds to the degree of our standard

[18] A key point here is that the closer α is to 1, the larger we can allow our ℓ_q-norm bound to be in order to get high entropy. See [24, Corollary 2.3].

[19] The actual β is around $\frac{1}{6}$, and $\beta < \frac{1}{2}$ is an inherent barrier for their construction.

CRVW graph G over M vertices, and we choose D_{crvw} to be quasi-polynomial in D.[20] For the exact choice of parameters for G and H, see [24, Section 5]. Now:

- For some parameter $b = \mathrm{poly}(d)$, we group consecutive blocks of X into "super-blocks" $X'_1 \circ \ldots X'_{t/b}$, each X'_i containing b blocks of length d each.
- For each $i \in [t/b]$, we use X'_i as instructions to a separate random walk on H, starting from some fixed node each time. Denote by Z_i the final node reached after the b steps.
- We show that $Z = Z_1 \circ \ldots \circ Z_{t/b}$ is itself an almost CG source, but this time with $\delta > \delta_{\mathsf{thr}}$. Thus, we can use Z as instructions for G!

Fortunately, H is constant-sized, so we can find it in constant time. Using optimal constant-sized ingredients is also a key idea in the [10] construction itself.

6.3 Removing the Constraints on d and γ

So far, we discussed how to condense from a γ-almost δ-CG source when $\gamma < 2^{O(1/\delta)}$ and $d > \mathrm{poly}(1/\delta)$.[21] To obtain Theorem 3.2, which has no such constraints, we observe that grouping the instructions of the CG source into blocks of length $\mathrm{poly}(1/\delta)$ yields a new CG source with roughly the same entropy rate, but with sufficiently large instruction length, and smoothness error exponentially small in $1/\delta$. The fact that grouping instructions into blocks improves the smoothness error follows quite easily from the observation that sampling a heavy instruction (one whose probability is at most γ) at step i is independent of sampling heavy instructions in previous steps. Thus, the number of heavy instructions sampled over many i follows Chernoff-like tail bounds. See [24, Lemma 3.3] for details.

6.4 Suffix-Friendliness

While our technique is flexible enough to recover from damaged blocks and suffer only the expected decrease in entropy per damaged block, it cannot achieve constant entropy gap, if, say, all the damaged blocks are at the end. However, if at any step we can guarantee that we won't encounter too many damaged blocks from now on, we *can* regain constant entropy gap. Roughly speaking, the favorable case is that the λ-fraction of bad blocks is nicely distributed in the sense that each suffix contains at most λ-fraction of bad blocks (up to an additive term). We call this property *suffix friendliness* (see the precise definition in [24, Definition 3.4]), and show that we can deterministically condense from such sources to within constant entropy gap in [24, Section 4.3.3]. Moreover, we observe that given an almost CG source with $\lambda = 0$, a *random* pattern that damages each block with probability roughly λ, leads to a suffix friendly almost CG source with "bad blocks" parameter λ (see [24, Lemma 3.5]).

6.5 The Construction's Runtime

Recall that the simulation slowdown is also affected by the time it takes to compute the extractor, or condenser (in the "one-shot" simulation setting). Our online manner of condensing, together with the fact that the primitives we use (namely, the CRVW expander and the GW extractor) are efficient, makes our construction efficient as well. In particular, in we achieve a near-quadratic runtime in the TM model. See [24, Appendix C]. In the RAM model, in which each machine word can store integers up to $N = 2^n$ and perform arithmetic in $\mathbb{F}_q$ for a prime $q \le N$ at unit cost, our construction takes *linear* time.

7 ON SUPPORTING BAD PREFIXES

We extended δ-CG sources to handle smoothness γ and λ fraction of bad blocks. One can also try and further relax the notion of CG sources in the following way: Instead of requiring that for each non-damaged block X_i, for *any* prefix $a \sim X_{[1,i-1]}$ it holds that $X_i|\{X_{[1,i-1]} = a\}$ is γ-close to having entropy rate δ, we require it only for *most* prefixes. Concretely, what if we allow some ρ-fraction of the prefixes to lead to instructions having low entropy? (See [24, Definitions 8.3, 8.6], also for the Shannon-entropy variant.)

That extension seems *too* permissive, at least in some regime of parameters. We show that any random variable $X \sim \{0,1\}^n$ with $H(X) \ge (1-\zeta)n$ is already an almost $\Omega(1)$-CG source with error parameters γ, λ, and ρ, all roughly equal to $\zeta^{\Omega(1)}$. Moreover, with a constant seed, we show that we can even increase the (smooth) entropy rate from an arbitrary $\Omega(1)$ to $1-\zeta$, at the cost of increasing λ and ρ. Thus, since we provably cannot condense or extract from high Shannon entropy with constant seed, we have an inherent barrier to handling $\rho > 0$ alongside a comparable, nonzero λ. We discuss it further, and give the precise details, in [24, Section 8].

8 EXTENSION: ONLINE CONDENSING AND MAINTAINING CONSTANT ENTROPY GAP

Unlike other condensers, our construction is an "online" one. That is, the construction makes one pass over the randomness stream $X_1 \circ \ldots \circ X_t$ in order to form the required instructions, and never needs to store more than a constant number of bits in memory before updating the location in the big graph. Moreover, we don't even need to know the number of blocks ahead of time![22]

As given above, it is easy to see that the construction does not ensure constant entropy deficiency in the output distribution *throughout* the random walk, but only at the end, even if there are no corrupted instructions at all ($\lambda = 0$). However, one can *easily adapt* our approach to also work in such a "completely online" fashion. The idea is to walk on graphs of gradually increasing size. Namely, after every constant number of steps (for some fixed constant), we map the current vertex to a vertex in a graph that is a constant times larger (but with the same degree) and continue the walk from there. Although we do not give such a result in full formality, in [24, Appendix B] we present an informal theorem and a brief discussion sketching its proof.

ACKNOWLEDGMENTS

We wish to thank Oded Goldreich for valuable comments and suggestions on a preliminary version of this paper, and Salil Vadhan and Yevgeniy Dodis for very helpful discussions. We also thank an

[20] One can also think of H as an ε-error optimal lossless conductor $H\colon \{0,1\}^{\mathrm{poly}(d)} \times \{0,1\}^d \to \{0,1\}^{\mathrm{poly}(d)}$ with seed length $d = \log\frac{1}{\varepsilon} + O(1)$.

[21] We did not mention the constraint on d explicitly, however the intuition is clear: the raw amount of entropy in an instruction, δd, should be at least 1.

[22] In the two-level construction of subsection 6.2, we first computed all Z_i-s just for the simplicity of exposition. Clearly we can compute Z_i, implement it on the big graph, and continue to compute Z_{i+1} without the need to keep storing Z_i.

anonymous reviewer for useful comments and for pointing out a minor flaw in a previous version of [24].

Dana Moshkovitz was supported in part by NSF Grant CCF-1705028 and CCF-2200956. Justin Oh was supported in part by NSF Grant CCF-1705028 and the David L. Miller and Mary H. Miller Graduate Fellowship. David Zuckerman was supported in part by NSF Grant CCF-1705028 and CCF-2008076 and a Simons Investigator Award (#409864).

REFERENCES

[1] Noga Alon and Michael Capalbo. 2002. Explicit unique-neighbor expanders. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 73–79.

[2] Nir Aviv and Amnon Ta-Shma. 2019. On the entropy loss and gap of condensers. *ACM Transactions on Computation Theory (TOCT)* 11, 3 (2019), 1–14.

[3] Marshall Ball, Oded Goldreich, and Tal Malkin. 2022. Randomness extraction from somewhat dependent sources. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[4] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. 2011. Leftover hash lemma, revisited. In *Annual Cryptology Conference*. Springer, 1–20.

[5] Boaz Barak, Guy Kindler, Ronen Shaltiel, Benny Sudakov, and Avi Wigderson. 2010. Simulating independence: New constructions of condensers, ramsey graphs, dispersers, and extractors. *Journal of the ACM (JACM)* 57, 4 (2010), 20.

[6] Salman Beigi, Omid Etesami, and Amin Gohari. 2017. Deterministic Randomness Extraction from Generalized and Distributed Santha–Vazirani Sources. *SIAM J. Comput.* 46, 1 (2017), 1–36.

[7] Avraham Ben-Aroya, Gil Cohen, Dean Doron, and Amnon Ta-Shma. 2019. Two-source condensers with low error and small entropy gap via entropy-resilient functions. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2019)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[8] Avraham Ben-Aroya, Dean Doron, and Amnon Ta-Shma. 2019. An efficient reduction from two-source to nonmalleable extractors: achieving near-logarithmic min-entropy. *SIAM J. Comput.* 0 (2019), STOC17–31.

[9] Radu Berinde, Anna C. Gilbert, Piotr Indyk, Howard Karloff, and Martin J. Strauss. 2008. Combining geometry and combinatorics: A unified approach to sparse signal recovery. In *Proceedings of the 46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 798–805.

[10] Michael Capalbo, Omer Reingold, Salil Vadhan, and Avi Wigderson. 2002. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the 34th Annual Symposium on Theory of Computing (STOC)*. ACM, 659–668.

[11] Eshan Chattopadhyay and Jesse Goodman. 2022. Improved extractors for small-space sources. In *Proceedings of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 610–621.

[12] Eshan Chattopadhyay, Jesse Goodman, and Jyun-Jie Liao. 2022. Affine extractors for almost logarithmic entropy. In *Proceedings of the 62nd Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 622–633.

[13] Lijie Chen and Roei Tell. 2021. Simple and fast derandomization from very hard functions: eliminating randomness at almost no cost. In *Proceedings of the 53rd Annual Symposium on Theory of Computing (STOC)*. ACM, 283–291.

[14] Xue Chen, Kuan Cheng, Xin Li, and Minghui Ouyang. 2022. Improved Decoding of Expander Codes. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

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

[16] Anindya De and Thomas Watson. 2012. Extractors and lower bounds for locally samplable sources. *ACM Transactions on Computation Theory (TOCT)* 4, 1 (2012), 1–21.

[17] Domingos Dellamonica Jr. and Yoshiharu Kohayakawa. 2008. An algorithmic Friedman–Pippenger theorem on tree embeddings and applications. *The Electronic Journal of Combinatorics* (2008), R127–R127.

[18] Yevgeniy Dodis, Siyao Guo, Noah Stephens-Davidowitz, and Zhiye Xie. 2021. No Time to Hash: On Super-Efficient Entropy Accumulation. In *CRYPTO (Lecture Notes in Computer Science, Vol. 12828)*. Springer, 548–576.

[19] Yevgeniy Dodis, Siyao Guo, Noah Stephens-Davidowitz, and Zhiye Xie. 2021. Online linear extractors for independent sources. In *Proceedings of the 2nd Conference on Information-Theoretic Cryptography (ITC)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[20] Yevgeniy Dodis, Krzysztof Pietrzak, and Daniel Wichs. 2014. Key derivation without entropy waste. In *Advances in Cryptology–EUROCRYPT 2014*. Springer, 93–110.

[21] Yevgeniy Dodis, Thomas Ristenpart, and Salil Vadhan. 2012. Randomness condensers for efficiently samplable, seed-dependent sources. In *Theory of Cryptography Conference*. Springer, 618–635.

[22] Yevgeniy Dodis and Yu Yu. 2013. Overcoming weak expectations. In *Theory of Cryptography Conference*. Springer, 1–22.

[23] Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. 2020. Nearly optimal pseudorandomness from hardness. In *Proceedings of the 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 1057–1068.

[24] Dean Doron, Dana Moshkovitz, Justin Oh, and David Zuckerman. 2022. Almost Chor–Goldreich Sources and Adversarial Random Walks. In *Electronic Colloquium on Computational Complexity (ECCC)*.

[25] Zeev Dvir. 2012. Extractors for varieties. *computational complexity* 21, 4 (2012), 515–572.

[26] Dmitry Gavinsky and Pavel Pudlák. 2020. Santha-Vazirani sources, deterministic condensers and very strong extractors. *Theory of Computing Systems* 64, 6 (2020), 1140–1154.

[27] Oded Goldreich and Salil Vadhan. 1999. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference)(Cat. No. 99CB36317)*. IEEE, 54–73.

[28] Oded Goldreich and Avi Wigderson. 1997. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms* 11, 4 (1997), 315–343.

[29] Venkatesan Guruswami, Christopher Umans, and Salil Vadhan. 2009. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. *Journal of the ACM (JACM)* 56, 4 (2009), 20.

[30] Jesse Kamp, Anup Rao, Salil Vadhan, and David Zuckerman. 2006. Deterministic extractors for small-space sources. In *Proceedings of the 38th Annual Symposium on Theory of Computing (STOC)*. ACM, 691–700.

[31] Ting-Chun Lin and Min-Hsiu Hsieh. 2022. Good quantum LDPC codes with linear time decoder from lossless expanders. *arXiv preprint arXiv:2203.03581* (2022).

[32] Chi-Jen Lu, Omer Reingold, Salil Vadhan, and Avi Wigderson. 2003. Extractors: Optimal up to constant factors. In *Proceedings of the 35th Annual Symposium on Theory of computing (STOC)*. 602–611.

[33] Noam Nisan and David Zuckerman. 1996. Randomness is Linear in Space. *J. Comput. System Sci.* 52, 1 (1996), 43–52.

[34] Jaikumar Radhakrishnan and Amnon Ta-Shma. 2000. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics* 13, 1 (2000), 2–24.

[35] Ran Raz and Omer Reingold. 1999. On recycling the randomness of states in space bounded computation. In *Proceedings of the 61st Annual Symposium on Theory of Computing (STOC)*. ACM, 159–168.

[36] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. 2006. Extracting randomness via repeated condensing. *SIAM J. Comput.* 35, 5 (2006), 1185–1209.

[37] Omer Reingold, Salil Vadhan, and Avi Wigderson. 2002. Entropy waves, the zig-zag graph product, and new constant-degree expanders. *Annals of Mathematics* (2002), 157–187.

[38] Omer Reingold, Salil Vadhan, and Avi Wigderson. 2004. A note on extracting randomness from Santha-Vazirani sources. Manuscript. In *Electronic Colloquium on Computational Complexity (ECCC)*.

[39] Miklos Santha and Umesh V. Vazirani. 1986. Generating quasi-random sequences from semi-random sources. *J. Comput. System Sci.* 33, 1 (1986), 75–87.

[40] Ronen Shaltiel and Emanuele Viola. 2022. On Hardness Assumptions Needed for "Extreme High-End" PRGs and Fast Derandomization. In *Proceedings of the 13th Innovations in Theoretical Computer Science Conference (ITCS)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

[41] Aravind Srinivasan and David Zuckerman. 1999. Computing with very weak random sources. *SIAM J. Comput.* 28, 4 (1999), 1433–1459.

[42] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. 2007. Lossless condensers, unbalanced expanders, and extractors. *Combinatorica* 27 (2007), 213–240.

[43] Luca Trevisan and Salil Vadhan. 2000. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2000)*. IEEE, 32–42.

[44] Emanuele Viola. 2014. Extractors for circuit sources. *SIAM J. Comput.* 43, 2 (2014), 655–672.

[45] David Zuckerman. 1990. General weak random sources. In *Proceedings of the 31st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 534–543.

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

Received 2022-11-07; accepted 2023-02-06