

# Lossless Condensers, Unbalanced Expanders, and Extractors

Amnon Ta-Shma\*      Christopher Umans†      David Zuckerman‡

March 23, 2006

## Abstract

Trevisan showed that many pseudorandom generator constructions give rise to constructions of explicit extractors. We show how to use such constructions to obtain explicit *lossless condensers*. A lossless condenser is a probabilistic map using only  $O(\log n)$  additional random bits that maps  $n$  bits strings to  $\text{poly}(\log K)$  bit strings, such that any source with support size  $K$  is mapped almost injectively to the smaller domain. Our construction remains the best lossless condenser to date.

By composing our condenser with previous extractors, we obtain new, improved extractors. For small enough min-entropies our extractors can output all of the randomness with only  $O(\log n)$  bits. We also obtain a new disperser that works for every entropy loss, uses an  $O(\log n)$  bit seed, and has only  $O(\log n)$  entropy loss. This is the best disperser construction to date, and yields other applications. Finally, our lossless condenser can be viewed as an unbalanced bipartite graph with strong expansion properties.

---

\*Computer Science Department, Tel-Aviv University, Israel, 69978. email: amnon@post.tau.ac.il. Much of this work was done while the author was in the Computer Science Division, University of California, Berkeley, and supported in part by a David and Lucile Packard Fellowship for Science and Engineering and NSF NYI Grant No. CCR-9457799. The work was also supported in part by an Alon fellowship and by the Israel Science Foundation.

†Computer Science, California Institute of Technology, 1200 E. California Blvd, Pasadena CA 91125. email: umans@cs.caltech.edu. Much of this work was done while the author was a graduate student in the Computer Science Division, University of California, Berkeley. Supported in part by NSF Grants CCR-9820897, CCF-0346991, and an Alfred P. Sloan Research Fellowship.

‡Department of Computer Science, University of Texas, 1 University Station C0500, Austin, TX 78712. email: diz@cs.utexas.edu. Much of this work was done while the author was on leave at the Computer Science Division, University of California, Berkeley. Supported in part by a David and Lucile Packard Fellowship for Science and Engineering, NSF Grants CCR-9912428 and CCR-0310960, NSF NYI Grant CCR-9457799, and an Alfred P. Sloan Research Fellowship.

# 1 Introduction

## 1.1 History and Background

Sipser [28] and Santha [26] were the first to realize that extractor-like structures can be used to save on randomness. Their structure is now known as a “disperser<sup>1</sup>.” They showed that good dispersers exist and left open the problem of actually constructing them. In the early period, there was a lot of research on special cases of the problem. The general extractor problem was first defined by Nisan and Zuckerman [18]:

**Definition 1.1 (extractor, min-entropy).** *Function  $E : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^m$  is a  $(k, \epsilon)$ -extractor if for every distribution  $X$  having  $k$  min-entropy, the distribution obtained by drawing  $x$  from  $X$ ,  $y$  uniformly from  $\{0, 1\}^t$  and evaluating  $E(x, y)$ , is within statistical distance  $\epsilon$  from the uniform distribution on  $\{0, 1\}^m$ . The min-entropy of a distribution  $X$  is  $H_\infty(X) = \min_a \{-\log_2 X(a)\}$ .*

In other words, extractors get an input from an unknown source distribution  $X$  having min-entropy  $k$ , use few ( $t$ ) truly random bits that are independent of the source, and extract  $m$  output bits that are  $\epsilon$ -close to uniform. While random functions are good extractors, useful extractors are *explicit*, i.e., computable in polynomial time. Not only do such extractors help save on randomness in various contexts, but they have had many applications to seemingly unrelated areas. See the survey [16] for more details.

The goal of explicit extractor constructions is to simultaneously maximize the output length  $m$  (ideally,  $m = k + t - 2 \log \epsilon^{-1} - O(1)$ ) and minimize the seed length  $t$  (ideally  $t = \log n + 2 \log \epsilon^{-1} + O(1)$ ). Often, constructions work well for only certain values of  $k$ , and obtaining a construction that works for all min-entropies  $k$  has been a challenge.

The progress on this problem is summarized in Table 1 for the case of constant error  $\epsilon$ . Early work (e.g., [41, 42, 18, 30, 32, 43]) used hashing and pairwise independence in various forms, and viewed extractors as sophisticated hash functions. Departing from previous techniques, Trevisan [35] showed a connection between pseudorandom generators for small circuits and extractors. Thus, Trevisan’s approach viewed extractors as pseudorandom generators against all statistical tests. Trevisan then used the Nisan-Wigderson pseudorandom generator [17] to construct a simple and elegant extractor that uses  $t = O(\frac{\log^2 n}{\log k})$  truly random bits. As long as the source has at least  $k = n^{\Omega(1)}$  min-entropy, this uses only  $t = O(\log n)$  truly random bits. However, if the min-entropy  $k$  is smaller, then the number of truly random bits  $t$  is  $\omega(\log n)$  and approaches  $\log^2 n$ .

A series of papers attacked this bottleneck. Impagliazzo, et al. [10, 11] used sophisticated (and complex) recursive techniques building on Trevisan’s construction. Reingold, et al. [24] improved their result by combining the old hashing techniques with the new extractors, together with new ideas. The actual parameters achieved are stated in Table 1. There was still a tradeoff however: if one insisted on an extractor that extracted a *constant* fraction of the min-entropy using the asymptotically optimal  $O(\log n)$  truly random bits, the situation was not good. The only constructions that achieved this were [30] (for extremely small  $k = O(\log n)$ ), and [43] (for very large  $k = \Omega(n)$ ). Our results extend the range of min-entropies  $k$  for which these parameters can be achieved to all  $k \leq 2^{\log^{1-o(1)} n}$ . Our work was recently improved by Lu, et al. [14] constructing extractors for any min-entropy  $k$  using only  $O(\log(n))$  truly random bits, and outputting  $\Omega(k)$  randomness.

---

<sup>1</sup>For a definition see Subsection 1.4.

	required entropy	no. of truly random bits $t$	no. of output bits	reference
Lower bound and non-explicit	Any $k$	$\log n + \Theta(1)$	$k + t - \Theta(1)$	[20]
Early work	$\Omega(n)$	$O(\log^2 n)$	$\Omega(k)$	[18]
	$\Omega(n)$	$O(\log n)$	$\Omega(k)$	[43]
	Any $k$	$\text{polylog}(n)$	$m = k$	[32]
Following Trevisan	Any $k$	$O(\log^2 n / \log k)$	$k^{1-\alpha}$	[35]
	Any $k$	$O(\log n)$	$k / \log n$	[24]
	Any $k$	$O(\log n)$	$k^{1-\alpha}$	Cor. 5.8 (1)
	Any $k$	$O(\log n)$	$\Omega(k)$	[14]
Optimal output length	Any $k$	$O(\log n + \log^2 k (\log \log k)^2)$	$k + t - O(1)$	Cor. 5.8 (2)

Table 1: Milestones in building explicit extractors. The error  $\epsilon$  is a constant;  $\alpha$  is an arbitrary constant.

We mention that all explicit extractor constructions up to date, lose  $\Omega(k)$  entropy (except for very low or very high min-entropies). Breaking the entropy-loss barrier for extractor (and to some extent also for disperser) constructions seems to be the next major challenge.

Our main contribution is to give a useful method for converting an extractor which works well for high min-entropies into one that works well for all min-entropies. Roughly, given an extractor using a seed of length  $t(n)$  for min-entropy  $k$ , we give an extractor using a seed of length  $t(k^2) + O(\log n)$  achieving the same output length. Remarkably, our construction is loss-less and does not lose entropy. This shows that it is enough to construct loss-less extractors for the high-entropy case. In addition, using this reduction we build a disperser with a much smaller entropy loss than previously known.

## 1.2 Our result

We show how to reduce the problem of constructing an extractor for a source with *arbitrary* min-entropy  $k$  (which has been the focus of [10, 11, 24]) to the problem of constructing an extractor for a source with *large* min-entropy (the focus of most of the earlier work on extractors, e.g., Trevisan’s work), as formalized in the following theorem (see Section 5.2 for the proof):

**Theorem 1.2.** *Suppose that there is an explicit family of  $(k = k(n) = n^{1/2}, \epsilon(n))$ -extractors,*

$$\left\{ E_n : \{0, 1\}^n \times \{0, 1\}^{t(n)} \rightarrow \{0, 1\}^{m(k)} \right\}.$$

*Then for every  $k = k(n) \leq n^{1/2}$ , there exists an explicit family of  $(k, k^{-1/2} + \epsilon(k^2))$ -extractors,*

$$\left\{ E'_n : \{0, 1\}^n \times \{0, 1\}^{O(\log n) + t(k^2)} \rightarrow \{0, 1\}^{m(k)} \right\}.$$

*Furthermore, if  $\{E_n\}$  are strong extractors, then  $\{E'_n\}$  are strong extractors.*

We achieve this by constructing “condensers”. A condenser uses a small number of auxiliary random bits to transform a weak source into a distribution on fewer bits that is close to a weak

source with about the same min-entropy. Our condenser uses  $O(\log n)$  random bits to transform a length  $n$  source with min-entropy  $k$  into a distribution on  $(k/\epsilon)^{1+\delta}$  bits that is  $\epsilon$ -close to a source with the same min-entropy  $k$ . We can then apply existing extractors to this shorter source. For example, applying Trevisan’s extractor produces an extractor with seed length  $O(\log n)$  that extracts  $m = k^{1-\alpha}$  bits from a source with min-entropy  $k$ , for *any*  $k$ . Applying better (and more complicated) constructions we obtain the additional result listed in Table 1.

We remark that Reingold, et al. [24] also build extractors by first using condensers. However, our condensers differ from theirs in that ours are *lossless*, which means that they preserve *all* of the min-entropy of the source. They therefore give a truly general reduction from the arbitrary min-entropy case to the high min-entropy case for building extractors. Also, because our condensers are lossless, they are actually unbalanced bipartite *expander graphs* with very strong expansion properties (see Section 1.5).

### 1.3 Our technique

The main contribution of this paper is a construction of the condensers that prove Theorem 1.2. We use a simplification of the approach of Impagliazzo, et.al. [10, 11] that has Trevisan’s construction at its core. In this section we give an overview of our technique; we assume some familiarity with Trevisan’s extractor.

To simplify our discussion, we will deal only with source distributions  $X$  that are uniform on sets of size  $2^k$ , instead of the more general distributions  $X$  having min-entropy  $k$ . Given such a distribution  $X$ , Trevisan’s function  $\mathbf{TR} : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^m$  uses  $t = O(\frac{\log^2 n}{\log \rho})$  random bits to produce two conceptual objects: the output distribution  $\mathbf{TR}(X, U_t)$  which has  $m$  bits, and an “advice string” for each  $x \in X$  of length  $\rho m$ . In these general terms, it is easy to understand and contrast the three lines of work: Trevisan [35], Impagliazzo, et al. [10, 11], and the present paper.

- Trevisan proved that if  $\mathbf{TR}(X, U_t)$  is not an extractor, then the advice strings constitute short descriptions of a non-negligible portion of  $X$ . For this to be a contradiction (and hence prove that  $\mathbf{TR}$  is an extractor), one needs  $k > \rho m$ , which forces  $k$  to be large ( $n^{\Omega(1)}$ ) if  $t$  is to be  $O(\log n)$ . This is the bottleneck referred to in the introduction.
- Impagliazzo, et al. argued that *either*  $\mathbf{TR}$  is an extractor, *or* the advice strings constitute short descriptions of a non-negligible portion of  $X$ . If the former is true, then one has the desired extractor; if the latter is true, then one can recursively apply an extractor to the advice strings themselves (as they retain most of the original min-entropy). There is now no restriction on  $\rho$  and so one can have  $t = O(\log n)$  for any  $k$ . But it is a delicate balancing act to get the recursion to work properly and to combine the various “candidate” extractors, and in the process one loses somewhat in various other parameters.
- In the present paper, we simply choose  $m$  much larger than  $k$ , *so that*  $\mathbf{TR}$  *cannot be an extractor*, and we output the advice strings themselves. Then, unconditionally, the advice strings constitute short descriptions of a non-negligible portion of  $X$ , and therefore retain the original min-entropy; in other words, we have condensed  $n$  bits into  $\rho m$  bits. We iterate our condenser, and in each step we only need to condense the source from  $n$  bits to  $n^\gamma$  bits, for some  $\gamma < 1$  (regardless of the min-entropy  $k$ ). Therefore, we need  $\rho m \leq n^\gamma$ , and we can easily have  $\rho = n^{\Omega(1)}$ , avoiding the bottleneck altogether.

Additional randomness $t$	Entropy loss	Reference
$O(\log n)$	polylog $n$	[33]
$O(\log n)$	$3 \log n + O(1)$	this paper
$\log n + O(1)$	$\Omega(1)$	lower bound [20]

Table 2: Explicit dispersers with constant error.

Size	Reference
$O(N \cdot 2^{\text{poly}(\log \log N)})$	[25]
$O(N \cdot \text{polylog} N)$	this paper
$O(N \cdot \frac{\log^2 N}{\log \log N})$	lower bound, [20]

Table 3: Explicit depth two super-concentrators

In retrospect, our technique may seem an obvious simplification of [10, 11]. But we do need some new ideas for it to work. For example, we need to deal with entropy instead of min-entropy for much of the proof, and we need a strengthening of Yao’s next-bit predictor lemma.

## 1.4 Applications

A disperser is the “one-sided” analog of an extractor, and it is probably best understood as a bipartite graph.

**Definition 1.3 (disperser).** *A bipartite graph  $G = (V = [N = 2^n], W = [M = 2^m], E)$  with left-degree  $D = 2^d$  is a  $(K, \epsilon)$  disperser if every subset  $A \subseteq V$  of cardinality at least  $K$  has at least  $(1 - \epsilon)M$  distinct neighbors in  $W$ . A disperser is explicit if the  $i$ -th neighbor of vertex  $v \in V$  can be computed in  $\text{poly}(n, d)$  time.*

Ideally,  $K$  vertices of a degree  $D$  graph can have  $KD$  neighbors. However, a lower bound of [20] shows that in any  $(K, \epsilon)$  disperser  $G = (V, W = [M], E)$  the size of  $W$  must be smaller than  $KD$ . The *entropy loss* of a disperser is the log of this loss, i.e.,  $\log(\frac{KD}{M}) = \log K + \log D - \log M$ . The previous best construction with degree  $D = \text{poly}(n)$  had entropy loss  $\text{poly} \log(n)$ . In this paper we construct a disperser with entropy loss only  $3 \log n + O(1)$ , as stated in the following theorem:

**Theorem 1.4.** *For every  $n, k$  and constant  $\epsilon$  there is a degree  $D = \text{poly}(n)$  explicit  $(K = 2^k, \epsilon)$  disperser  $G = (V = [N = 2^n], W = [\Omega(KD/n^3)], E)$ .*

One consequence of our disperser is an almost optimal explicit depth-2 super-concentrator, defined below.

**Definition 1.5 (depth two super-concentrator).**  *$G = ((V_1, V_2, V_3), E)$  is a depth two super-concentrator if  $G$  is a layered graph with three layers: input vertices  $V_1$ , middle layer  $V_2$ , and output vertices  $V_3$ , and for all sets  $X \subseteq V_1, Y \subseteq V_3$  of cardinality  $k$ , there are at least  $k$  vertex-disjoint paths from  $X$  to  $Y$ .*

We achieve optimal size up to polylogarithmic factors. A long line of papers had tried to solve this problem; the previous best result and our result is summarized in Table 3. We obtain this result by plugging our disperser into [40]. We also improve a hardness result of Umans [36], as described in Section 7.

## 1.5 Unbalanced expanders with near-optimal expansion

An expander graph has the property that every not-too-large subset of the vertices has many neighbors, relative to its degree. Expanders have had numerous applications in computer science including network constructions [7], sorting [1, 19], complexity theory [39], cryptography [9], and pseudorandomness [2]. Many of these applications require bipartite graphs, where only subsets on one side are required to expand.

**Definition 1.6 (expander).** *A bipartite graph  $G = (V, W, E)$  is  $(K, c)$  expanding if for every  $A \subseteq V$  of cardinality at most  $K$ ,  $|\Gamma(A)| \geq c|A|$ , where  $\Gamma(A)$  is the set of neighbors of  $A$ .*

The goal is to have the expansion factor  $c$  be as close as possible to the left-degree  $T$  ( $T$  is the degree of all vertices in  $V$ ). Random graphs have  $c \geq T - (2 \log |V|) / \log |W| - o(1)$  if  $K < |V|^{.49}$ . Yet for most applications random graphs are not useful; instead, explicit, deterministic constructions are required.

Historically, constructing explicit expanders has been quite difficult. The explicit construction of constant degree expander graphs was a major breakthrough [15, 8]. These explicit constructions relied on showing an upper bound on the second largest eigenvalue of the adjacency matrix corresponding to the graph. Kahale [13] showed that such methods cannot achieve  $c > T/2$ . Yet some applications, such as [4, 29, 5] need  $c = (1/2 + \Omega(1))T$ , as then the expander has the “unique neighbors property.” This means that for any subset  $A$  of vertices, there are  $\Omega(|A|)$  vertices that are neighbors of exactly one vertex in  $A$ .

Prior to our work the only method known for constructing graphs with such large expansion was to show that the graph has large girth [3]. However, this method doesn’t appear to help when  $|V| \gg |W|$ , which is desired in the above applications. As mentioned above, our lossless condensers are actually expander graphs with very strong expansion properties. This gives a new method for constructing *unbalanced expanders* with non-constant but relatively small degree; we believe this approach and the following theorem are of independent interest.

**Theorem 1.7.** *For every positive constant  $\alpha$  and function  $\epsilon = \epsilon(N)$  there is an explicit family of degree  $T$  graphs  $G = (V = [N], W = [M], E)$  that are  $(K = 2^k, (1 - \epsilon)T)$  expanding with either of the following parameters:*

1.  $T = \text{polylog}N$  and  $M = 2^{(k/\epsilon)^{1+\alpha}}$
2.  $T = 2^{O((\log \log N)^2)}$  and  $M = 2^{O(k/\epsilon)}$ .

Using (2) with  $\epsilon = .01$ , for example, gives graphs with  $M \leq N^c$  such that every set of size at most  $N^{c'}$  expands by  $.99T$ , where  $c$  and  $c'$  are constants.

We mention that after the publication of our work Capalbo et al. [6] constructed explicit lossless extractors for high-min-entropy and also explicit slightly unbalanced loss-less expanders with constant degree. Nevertheless Theorem 1.7 remains the best loss-less expander construction up to date for the highly unbalanced case.

## 2 Preliminaries

A probability distribution  $D$  on  $\Lambda$  is a function  $D : \Lambda \rightarrow [0, 1]$  such that  $\sum_{x \in \Lambda} D(x) = 1$ .  $U_n$  is the uniform distribution on  $\{0, 1\}^n$ . The variation distance  $|D_1 - D_2|$  between two probability distributions on  $\Lambda$  is  $\frac{1}{2} \sum_{x \in \Lambda} |D_1(x) - D_2(x)| = \max_{S \subseteq \Lambda} |D_1(S) - D_2(S)|$ . We say  $D_1$  is  $\epsilon$  close to  $D_2$  if  $|D_1 - D_2| \leq \epsilon$ . The support of a distribution  $D$  is the set of all  $x$  for which  $D(x) \neq 0$ . A distribution  $D$  is *flat* over its support  $A \subseteq \Lambda$  if  $D(a) = \frac{1}{|A|}$  for all  $a \in A$ . If  $A$  is a set, we use  $A$  to also refer to the flat distribution with support  $A$ , when this meaning is clear from context.

If  $D$  is a distribution and  $f$  a function, then  $f(D)$  denotes the distribution obtained by picking  $d$  according to the distribution  $D$  and evaluating  $f(d)$ . Thus, e.g.,  $E(X, U_t)$  denotes the distribution obtained by picking  $x$  according to the distribution  $X$ , picking  $y$  uniformly at random from  $\{0, 1\}^t$ , and evaluating  $E(x, y)$ .

### 2.1 Distinguishers, next-bit predictors and pseudorandom generators

A distinguisher is a test that distinguishes between a given distribution and the uniform distribution:

**Definition 2.1 (distinguisher).** *A function  $D : \{0, 1\}^m \rightarrow \{0, 1\}$   $\epsilon$ -distinguishes a distribution  $X$ , if*

$$\left| \Pr_{x \leftarrow X} [D(x) = 1] - \Pr_{u \leftarrow U_m} [D(u) = 1] \right| \geq \epsilon.$$

A next-bit predictor is a special distinguisher that is able to predict well the  $i$ 'th bit of  $x \in X$  given the first  $i - 1$  bits of  $x$ , i.e.,

**Definition 2.2 (next bit predictor).** *Let  $X$  be a distribution over  $\{0, 1\}^m$ . A function  $T : \{0, 1\}^{<m} \rightarrow \{0, 1\}$  is a next-bit predictor for  $X$  with success  $p$ , if*

$$\Pr_{i \in [m], x \leftarrow X} [T(x_1, x_2, \dots, x_{i-1}) = x_i] \geq p.$$

Note that a next-bit predictor (or a distinguisher) need not be efficient.

Clearly, a next-bit predictor with success  $p = 1/2 + \epsilon$  (i.e., with “ $\epsilon$  advantage”) is in particular an  $\epsilon$ -distinguisher. Somewhat surprisingly, Yao showed a converse, that every distinguisher can be converted into a predictor. However, this converse is less tight. To see that, consider a distribution that picks  $m$  bits independently, with each bit being one with probability  $1/2 + \epsilon/m$ . Then, every next-bit predictor has at most an  $\epsilon/m$  advantage, and yet there exists an  $\Omega(\epsilon)$ -distinguisher. Yao’s lemma says this is essentially the worst that can happen:

**Lemma 2.3 (Yao’s next-bit predictor lemma).** *If random variable  $Y = (Y_1, Y_2, \dots, Y_m)$  distributed over  $\{0, 1\}^m$  is not  $\epsilon$ -close to uniform, then there is a next-bit predictor for  $Y$  with success  $\frac{1}{2} + \frac{\epsilon}{m}$ .*

Thus every distinguisher can be converted into a next-bit predictor, but with a loss: an  $\epsilon$ -distinguisher translates to a next-bit predictor with only  $\frac{\epsilon}{m}$  advantage. This loss is devastating for us, and one of the crucial components of our later constructions is a method for avoiding it.

A pseudorandom generator takes a short random string and expands it to a long string that looks random to all small circuits.

**Definition 2.4 (pseudorandom generator).** A function  $G : \{0, 1\}^t \rightarrow \{0, 1\}^m$  is a pseudorandom generator against size  $s$  circuits with  $\epsilon$  error if there is no size  $s$  circuit that  $\epsilon$ -distinguishes the distribution  $G(U_t)$ .  $G$  is efficient if it runs in time polynomial in its output length  $m$ .

## 2.2 Extractors and condensers

We say a distribution  $X$  has *min-entropy*  $k$ , if no element  $x$  has probability mass larger than  $2^{-k}$ . Formally:

**Definition 2.5 (min-entropy).** The min-entropy of a distribution  $X$  is  $H_\infty(X) = \min_a \{-\log_2 X(a)\}$ .

Though we deal primarily with min-entropy, some proofs will also require the usual notion of entropy:

**Definition 2.6 (entropy).** The entropy of a distribution  $X$  is  $H(X) = \sum_a -X(a) \log_2 X(a)$ . For  $p \in [0, 1]$ , the binary entropy function is  $H(p) = -p \log p - (1 - p) \log(1 - p)$ .

For every distribution  $X$ ,  $H_\infty(X) \leq H(X)$ , with equality iff  $X$  is flat.

**Definition 2.7 (condenser).** Let  $C : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^m$ .

1. We say  $C$  is a  $(n, k_1) \rightarrow_\epsilon (m, k_2)$  condenser if for every distribution  $X$  with  $k_1$  min-entropy,  $C(X, U_t)$  is  $\epsilon$ -close to a distribution with  $k_2$  min-entropy.
2. We say  $C$  is a strong  $(n, k_1) \rightarrow_\epsilon (m, k_2)$  condenser, if for every distribution  $X$  with  $k_1$  min-entropy,  $U_t \circ C(X, U_t)$  is  $\epsilon$ -close to a distribution  $U_t \circ D$  with  $t + k_2$  min-entropy.
3. We say  $C$  is a (strong) lossless condenser if it is a (strong)  $(n, k) \rightarrow_\epsilon (m, k)$  condenser.

**Remark 2.8.** Our definition of strong condenser is essentially equivalent to Raz's definition [21]: that the average, over  $y$ , of the distance of  $C(X, y)$  to a min-entropy  $k_2$ -source is at most  $\epsilon$ . That Raz's definition implies ours is not hard. To see that ours implies Raz's, suppose  $U_t \circ C(X, U_t)$  is  $\epsilon$ -close to  $U_t \circ D$ , which has min-entropy at least  $t + k_2$ . Then conditional on  $U_t = y$ ,  $D$  still has min-entropy at least  $k_2$ . The rest follows easily.

In this language we can define an extractor as a special case of a condenser (compare with Definition 1.1).

**Definition 2.9 (extractor).** Let  $E : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^m$ . Then  $E$  is a (strong)  $(k, \epsilon)$ -extractor if it is a (strong)  $(n, k) \rightarrow_\epsilon (m, m)$  condenser.

Both extractors and condensers are *explicit* if they can be computed in polynomial time. In the definitions above, we may equivalently take the source distribution  $X$  to be a flat distribution. This follows from two standard facts: (1) any distribution  $X$  with min-entropy  $k_1$  can be written as a convex combination of flat distributions with min-entropy  $k_1$ ; and (2) a convex combination of distributions that are  $\epsilon$ -close to distributions with min-entropy  $k_2$  is  $\epsilon$ -close to a single distribution with min-entropy  $k_2$ . The observation that flat distributions suffice will be used repeatedly in the proofs to follow.



### 3 Reconstructive pseudorandom generators

In the next two subsections we discuss the notion of so-called “reconstructive” pseudorandom generators, first informally, and then formally to establish the framework around which our condensers are built.

#### 3.1 An informal discussion

In this subsection we omit details and parameters, and ignore issues of worst-case vs. average-case hardness. In the next subsection we give a rigorous and formal treatment of this material.

An efficient pseudorandom generator (PRG) implies an explicit function in the complexity class  $\mathbf{E}$  that is hard for small non-uniform circuits [17]. The converse is also true, but harder to prove. The first result of this kind is the Nisan-Wigderson (NW) construction [17] (that was later improved in [12, 31, 27, 37]) that shows how to use a function in  $f \in \mathbf{E}$  that is average-case hard for small circuits, to construct a PRG.

The NW construction and later improvements are black-box constructions in the following sense. They start with an explicit function  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$  and construct from it a new function  $G^f : \{0, 1\}^t \rightarrow \{0, 1\}^m$  (where the notation is meant to indicate that  $G$  makes black-box oracle calls to  $f$ ) and prove that if  $f$  is hard, then  $G^f$  is a PRG.

Most importantly for us, this implication is proved by exhibiting a “reconstruction” algorithm. Namely, the proof describes an efficient “reconstruction” oracle Turing Machine  $R$  such that for every Boolean function<sup>2</sup>  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$ , if there is a small circuit  $C$  that  $\epsilon$ -distinguishes  $G^f(U_t)$ , then there exists a *short* advice string  $z = A(f)$  such that  $R^C(z, i)$  computes  $f(i)$ . In particular the existence of  $R$  implies:

**Lemma 3.1 (informal, [17]).** *If  $f : \{0, 1\}^\ell \rightarrow \{0, 1\}$  is suitably hard then  $G^f$  is a pseudorandom generator.*

*Proof.* (sketch) If there is a small circuit  $C$  that  $\epsilon$ -distinguishes  $G^f(U_t)$ , then by hardwiring the “correct” advice  $z = A(f)$ ,  $R^C(z, i)$  is a small circuit computing  $f$ . The contrapositive then says that if  $f$  cannot be computed by small circuits, then  $G^f(U_t)$  is a PRG.  $\square$

The above result is conditional: if  $f$  is a hard function then  $G^f$  is a PRG. Trevisan showed that reconstructive PRG are strong enough to give an *unconditional* extractor construction:

**Lemma 3.2 (informal, [35]).**  *$E : \{0, 1\}^{2^\ell} \times \{0, 1\}^t \rightarrow \{0, 1\}^m$  defined by  $E(f, y) = G^f(y)$  is an extractor.*

*Proof.* (sketch) Let  $n = 2^\ell$  and let  $X \subseteq \{0, 1\}^n$  be a large subset. We identify  $\{0, 1\}^n$  with the set of all functions from  $\{0, 1\}^\ell$  to  $\{0, 1\}$ . If  $E(X, U_t)$  is not close to uniform, then there exists a function  $C$  that  $\epsilon$ -distinguishes  $E(X, U_t)$ . By averaging, we can even say that  $C$   $\epsilon/2$ -distinguishes  $E(x, U_t)$ , for many  $x \in X$ . Therefore, for many  $x \in X$  there exists a short advice string  $z = A(x)$  for which  $R^C(z, \cdot)$  outputs  $x$ . The number of strings  $x$  with such short descriptions cannot exceed the number of possible advice strings. We conclude that if  $E(X, U_t)$  is not close to uniform, then  $X$  is small. The contrapositive says that if  $X$  is large, then  $E(X, U_t)$  is close to uniform; in other words,  $E$  is an extractor.  $\square$

---

<sup>2</sup>We treat a Boolean function and its truth-table interchangeably.

In this paper we use the same argument in a different way. Suppose we could choose the parameters so that there is a function  $C$  that  $\epsilon$ -distinguishes  $E(x, U_t)$  for almost every  $x \in X$ . The above argument shows that such strings  $x$  can be identified with their associated advice strings  $z = A(x)$ . So we have a (nearly) one-to-one mapping between  $X$  and  $A(X)$ ; in other words, the advice function  $A$  defines a lossless condenser! Clearly, the advice can not be too short now – it must be at least as long as the entropy of  $X$ . However, if  $z$  is still much shorter than  $n$ , we non-trivially condense the distribution  $X$ .

This idea almost works, except for the following technical difficulty. Current reconstruction arguments, even given a perfect distinguisher, are not able to give a perfect reconstruction (i.e. one that works for all  $x \in X$ ). Instead, they first convert the distinguisher to a next-bit predictor with a lossy conversion (see Lemma 2.3), and then use the next-bit predictor in the reconstruction. The loss in the conversion prevents us from getting a lossless condenser.

This leads us to define “reconstructive extractors” using next-bit predictors directly rather than distinguishers; i.e., the guarantee is that if  $T$  is a good next-bit predictor, then  $R^T$  is a good reconstruction procedure. We then show directly, that for a certain choice of parameters there is always a good (nearly perfect) next-bit predictor.

Summarizing, say  $G^f$  is a reconstructive PRG, with advice function  $A(f)$ . Nisan and Wigderson [17] used it to deduce that if  $f$  is a hard function, then  $G^f$  is a PRG. Trevisan [35] used it to show that  $E(f, y) = G^f(y)$  is an extractor. We use it to show that  $A$  is a lossless condenser.

### 3.2 A formal treatment: reconstructive extractors

We first define reconstructive extractors. The formalism in this section is adapted from [38].

**Definition 3.3 (reconstructive extractor).** *A triple  $(E, A, R)$  of functions where:*

- $E : \{0, 1\}^n \times \{0, 1\}^{r_E} \rightarrow \{0, 1\}^m$  is called the extractor function,
- $A : \{0, 1\}^n \times \{0, 1\}^{r_A} \rightarrow \{0, 1\}^a$  is called the advice function, and,
- $R : \{0, 1\}^a \times \{0, 1\}^{r_A} \times \{0, 1\}^{r_R} \rightarrow \{0, 1\}^n$  is called the reconstruction function

is a  $(p, q)$  reconstructive extractor if for every  $X \subseteq \{0, 1\}^n$  and every next-bit predictor  $T : \{0, 1\}^{<m} \rightarrow \{0, 1\}$  for  $E(X, U_{r_E})$  with success  $p$ , we have

$$\Pr_{x \leftarrow X, y, z} [R^T(A(x, y), y, z) = x] \geq q.$$

We now have two claims. First, we claim that we can choose  $E$  such that an almost perfect next-bit predictor exists, and second that whenever such a predictor exists,  $A$  is a lossless condenser. We begin with:

**Lemma 3.4.** *Let  $E : \{0, 1\}^n \times \{0, 1\}^{r_E} \rightarrow \{0, 1\}^m$  be a function, and let  $X \subseteq \{0, 1\}^n$  be a subset of cardinality at most  $2^k$ . Then, there exists a next-bit predictor  $T : \{0, 1\}^{<m} \rightarrow \{0, 1\}$  for  $E(X, U_{r_E})$  with success  $1 - \frac{k+r_E}{m}$ .*

The proof idea is that if  $m$  is much larger than the entropy of  $X$ , then  $E$  encodes an input  $x$  from  $X$  with much redundancy, and hence a good predictor exists. We give the formal proof in Section 3.3.

Our second claim is that if  $(E, A, R)$  is a reconstructive extractor, and if a good next-bit predictor for  $E(X, U_{r_E})$  exists, then  $A(X, U_{r_A})$  retains the entropy of  $X$ .

**Lemma 3.5.** *Let  $(E, A, R)$  be a  $(p, q = 1 - \epsilon)$  reconstructive extractor and  $X \subseteq \{0, 1\}^n$  a subset such that there exists a next-bit predictor  $T : \{0, 1\}^{<m} \rightarrow \{0, 1\}$  for  $E(X, U_{r_E})$  with success  $p$ . Then the distribution  $U_{r_A} \circ A(X, U_{r_A})$  is  $O(\epsilon)$ -close to a distribution  $U_{r_A} \circ D$  with  $r_A + \log_2 |X|$  min-entropy.*

*Proof.* Let us call a pair  $(x, y)$  with  $x \in X$  and  $y \in \{0, 1\}^{r_A}$  good if

$$\Pr_z[R^T(A(x, y), y, z) = x] > 1/2 \quad (1)$$

Let  $G$  be the set of good pairs  $(x, y)$ . Since we know  $\Pr_{x \leftarrow X, y, z}[R^T(A(x, y), y, z) = x] \geq 1 - \epsilon$ , we obtain, by an averaging argument, that  $\Pr_{x \leftarrow X, y}[(x, y) \in G] \geq 1 - 2\epsilon$ .

Now notice that Equation (1) implies that if  $(x_1, y)$  and  $(x_2, y)$  are both good, then  $A(x_1, y) \neq A(x_2, y)$ . This holds because if  $A(x_1, y) = A(x_2, y)$  then  $\Pr_z[R^T(A(x_1, y), y, z) = x_2] > 1/2$ , contradicting Equation (1). In particular, if we define  $A'(x, y) = y \circ A(x, y)$ , then  $A'$  is one-to-one on the set of good pairs  $G$ .

However, as argued above, almost every element of  $X \times \{0, 1\}^{r_A}$  is good, and so the flat distribution on the set  $G$  is  $O(\epsilon)$ -close to the distribution  $X \circ U_{r_A}$ . In particular, the probability mass on elements of  $A'(X, U_{r_A})$  with multiple preimages is at most  $O(\epsilon)$  (since  $A'$  is one-to-one on  $G$ ). By redistributing this mass, we obtain a distribution  $D \circ U_{r_A}$  with min-entropy  $\log_2 |X| + r_A$  that is  $O(\epsilon)$ -close to  $A'(X, U_{r_A})$ , which proves the lemma.  $\square$

Combining Lemmas 3.5 and 3.4 we get our main theorem that the advice function of a reconstructive extractor (with long enough output length  $m$ ) is a lossless condenser:

**Theorem 3.6.** *Assume the triple of functions*

$$\begin{aligned} E & : \{0, 1\}^n \times \{0, 1\}^{r_E} \rightarrow \{0, 1\}^m \\ A & : \{0, 1\}^n \times \{0, 1\}^{r_A} \rightarrow \{0, 1\}^a \\ R & : \{0, 1\}^a \times \{0, 1\}^{r_A} \times \{0, 1\}^{r_R} \rightarrow \{0, 1\}^n \end{aligned}$$

*is a  $(1 - \epsilon, 1 - \epsilon)$  reconstructive extractor. Then  $A$  is a strong  $(n, k) \rightarrow_{O(\epsilon)} (a, k)$  condenser, provided  $m \geq \frac{k + r_E}{\epsilon}$ .*

*Proof.* Let  $X \subseteq \{0, 1\}^n$  be an arbitrary subset of cardinality  $2^k$ . By Lemma 3.4 there exists a next-bit predictor  $T$  for  $E(X, U_{r_E})$  with  $1 - \frac{k + r_E}{m} = 1 - \epsilon$  success. By Lemma 3.5,  $U_{r_A} \circ A(X, U_{r_A})$  is  $O(\epsilon)$ -close to a distribution with min-entropy  $k + r_A$ . Using the observation regarding flat distributions at the end of Section 2.2, we find that  $A$  is the desired lossless condenser.  $\square$

### 3.3 Forcing a next-bit predictor

We now prove Lemma 3.4, showing that if the extractor's output length  $m$  is much larger than the source entropy than a good next-bit predictor exists. We begin with:

**Lemma 3.7 (strong next-bit predictor).** *If a distribution  $Y = (Y_1, Y_2, \dots, Y_m)$  over  $\{0, 1\}^m$  has entropy  $H(Y) \leq \epsilon m$ , then there is a next-bit predictor  $T$  for distribution  $Y$  with success  $1 - \epsilon$ .*

*Proof.* Let us denote

$$p_{i,y_1,\dots,y_{i-1}} = \Pr[Y_i = 1 | Y_1 = y_1, \dots, Y_{i-1} = y_{i-1}].$$

Given  $i, y_1, \dots, y_{i-1}$ , an optimal (and non-explicit) next-bit predictor  $T$  predicts 1 if  $p_{i,y_1,\dots,y_{i-1}}$  is larger than half and 0 otherwise. The error of this next-bit predictor is  $\mathbb{E}_{i \in [m], y \in Y}[\min(p_{i,y_1,\dots,y_{i-1}}, 1 - p_{i,y_1,\dots,y_{i-1}})]$ , and we now bound this term. We first notice that

$$\begin{aligned} \min(p, 1 - p) &\leq \min(p, 1 - p) \log \frac{1}{\min(p, 1 - p)} \\ &\leq p \log \frac{1}{p} + (1 - p) \log \frac{1}{1 - p} = H(p) \end{aligned}$$

It therefore follows that

$$\begin{aligned} \mathbb{E}_{i \in [m], y \in Y}[\min(p_{i,y_1,\dots,y_{i-1}}, 1 - p_{i,y_1,\dots,y_{i-1}})] &\leq \mathbb{E}_{i \in [m], y \in Y}[H(p_{i,y_1,\dots,y_{i-1}})] \\ &= \frac{1}{m} \sum_{i=1}^m H(Y_i | Y_1, Y_2, \dots, Y_{i-1}) \\ &= \frac{1}{m} H(Y) \leq \epsilon \end{aligned}$$

as required. □

We are now ready to prove Lemma 3.4

*Proof of Lemma 3.4.*  $H(E(X, U_{r_E})) \leq H(X) + H(U_{r_E}) \leq k + r_E$ . It follows by Lemma 3.7 that the optimal next-bit predictor for  $E(X, U_{r_E})$  has success at least  $1 - \frac{k+r_E}{m}$ . □

## 4 A concrete example

There are three existing constructions [35, 34, 27] meeting the requirements of Definition 3.3, and yielding lossless condensers whose parameters suffice to prove Theorem 1.2. In this section we present one of these constructions in our framework, due to Trevisan [35], based on the Nisan-Wigderson PRG construction [17], and with refinements due to [23].

### 4.1 The Trevisan reconstructive extractor

Throughout this section,  $x$  is an element of the weak random source  $X$ , which is distributed over  $\{0, 1\}^n$ . The construction requires two ingredients: an asymptotically good  $[\bar{n}, n, \bar{n}/3]$  binary code  $C$ , and a combinatorial object called a weak design, defined below:

**Definition 4.1 (weak design [23]).** *A family of sets  $\Delta = (S_1, S_2, \dots, S_m) \subseteq [t]$  is a weak  $(\ell, \rho)$  design if*

1.  $\forall i |S_i| = \ell$ , and

$$2. \forall i, \sum_{j < i} 2^{|S_i \cap S_j|} \leq \rho \cdot (m - 1).$$

We now describe the three functions  $(E, A, R)$ .

**The extractor function.** We are given input  $x$  (from the weak random source) and seed  $y$ . We first compute  $\hat{x} = C(x)$ . We view  $\hat{x}$  as a function  $\hat{x} : \{0, 1\}^\ell \rightarrow \{0, 1\}$  with  $\ell = \log \bar{n}$ .

We use a weak  $(\ell, \rho)$  design  $(S_1, S_2, \dots, S_m) \subseteq [t]$ . The seed length  $r_E$  will be  $t$ . We denote by  $y|_{S_i}$  the projection of  $y \in \{0, 1\}^t$  onto the coordinates in set  $S_i \subseteq [t]$  (so  $y|_{S_i} \in \{0, 1\}^\ell$ ).

The  $i$ -th output bit of  $E(x, y)$  is the evaluation of the function  $\hat{x}$  at  $y|_{S_i}$ . Formally,

$$E(x, y) = \hat{x}(y|_{S_1}) \circ \hat{x}(y|_{S_2}) \circ \dots \circ \hat{x}(y|_{S_m}).$$

**The advice function.** We are given  $x$  (from the weak random source) and a seed  $y$ . We first compute  $\hat{x} = C(x)$  as with the extractor. We view  $y$  as being composed of two parts:  $i \in [m]$  and  $\beta \in \{0, 1\}^{t-\ell}$  (so  $r_A = \log m + t - \ell$ ). The output of  $A$  comprises the evaluations of  $\hat{x}$  on a subset of inputs that is determined by  $i$  and  $\beta$ ; we now describe what these inputs are.

Each input in the set is an  $\ell$  bit string that is determined by  $\beta$ , together with  $j$  and an additional string  $\gamma$ , where  $j$  ranges over  $[i - 1]$  and  $\gamma$  ranges over all strings of length  $|S_i \cap S_j|$ . Such a string is obtained by first writing down the  $t$ -bit string that has  $\beta$  in coordinates  $[t] \setminus S_i$ ,  $\gamma$  in coordinates  $S_i \cap S_j$ , and zeros elsewhere, and then projecting onto coordinates  $S_j$ . We denote this final string  $w(\beta, j, \gamma)$ . Note that  $w(\beta, j, \gamma)$  coincides with  $\beta$  on  $S_j \setminus S_i$ , and it coincides with  $\gamma$  on  $S_j \cap S_i$ .

Our advice function outputs the evaluations of  $\hat{x}$  on such strings for all  $j < i$ , and all  $\gamma \in \{0, 1\}^{|S_i \cap S_j|}$ . Formally,

$$A(x; i, \beta) = (\hat{x}(w(\beta, j, \gamma)))_{j < i, \gamma \in \{0, 1\}^{|S_j \cap S_i|}}.$$

The advice function  $A$  outputs  $a = \sum_{j < i} 2^{|S_i \cap S_j|}$  bits. By the weak-design property, for every  $i \in [m]$ ,  $\sum_{j < i} 2^{|S_i \cap S_j|} \leq \rho m$  and so  $a \leq \rho m$ .

**The reconstruction function.** In this case the reconstruction algorithm is deterministic (so  $r_R = 0$ ). Its two inputs are  $A(x; i, \beta)$ , and the pair  $(i, \beta)$ . The goal of the reconstruction function is to output  $x$ .

We mirror the process used by  $A$  to compute its output. For each  $w \in \{0, 1\}^\ell$ , we write down the  $t$ -bit string  $y$  that has  $\beta$  in coordinates  $[t] \setminus S_i$  and  $w$  in the coordinates  $S_i$ . Observe that for  $j < i$ ,  $b_j = \hat{x}(y|_{S_j})$  can be found in the advice  $A(x; i, \beta)$ . We feed the bits  $b_1, \dots, b_{i-1}$  to the next-bit predictor  $T$ ; i.e., we compute  $z(w) = T(i; b_1, b_2, \dots, b_{i-1})$ . This gives a guess for the value of  $\hat{x}(y|_{S_i}) = \hat{x}(w)$ . After obtaining guesses for all  $w$ , we output  $x$  for which  $\hat{x}(\cdot)$  is closest (in Hamming distance) to  $z(\cdot)$ .

Nisan and Wigderson [17] proved:

**Lemma 4.2 ([17]).** *For every constant  $\epsilon > 0$ ,  $(E, A, R)$  described above is a  $(1 - \epsilon, 1 - 10\epsilon)$  reconstructive extractor.*

Thus, applying Theorem 3.6, we obtain

**Corollary 4.3.** *Assume there exists a weak  $(\ell = \log \bar{n} = \log n + O(1), \rho)$  design  $\Delta = (S_1, S_2, \dots, S_m) \subseteq [t]$  with  $m \geq \frac{k+t}{\epsilon}$  that can be constructed in  $\text{poly}(m, t)$  time. Then*

$$A : \{0, 1\}^n \times \{0, 1\}^{r_A} \rightarrow \{0, 1\}^a$$

*described above is an explicit strong lossless  $(n, k) \rightarrow_{O(\epsilon)} (a, k)$  condenser, with  $a = \rho m = \rho \frac{k+t}{\epsilon}$  and  $r_A \leq t$ .*

*Proof.* We may assume that  $m \leq n$  — as the identity function is a condenser for  $m \geq n$ . We can therefore verify that the seed length  $r_A = \log m + t - \ell \leq \log n + t - \ell = t$  (using that  $\ell = \log n$ ). The other parameters are immediate.  $\square$

We now want to see what parameters we get from the construction. For that we need to know how the weak-design parameters behave. Raz et al. [23] show:

**Lemma 4.4 ([23]).** *For every  $\ell, m$  and  $\rho \geq 1$ , there exists a weak  $(\ell, \rho)$  design  $\Delta = (S_1, S_2, \dots, S_m) \subseteq [t]$  that can be constructed in  $\text{poly}(m, t)$  time, where*

$$t = t(\ell, \rho) = \begin{cases} \left\lceil \frac{\ell}{\ln \rho} \right\rceil \cdot \ell & \rho \geq 3/2 \\ O(\ell^2 \log \frac{1}{\rho-1}) & 1 < \rho < 3/2 \\ O(\ell^2 \log m) & \rho = 1 \end{cases}$$

Plugging in the parameters we get the following two condensers:

**Corollary 4.5.** *For every  $n, k$  and  $\epsilon \in (0, \frac{1}{2})$ ,  $A : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^a$  described above is an explicit strong lossless  $(n, k) \rightarrow_{O(\epsilon)} (a, k)$  condenser with either of the following two choices for parameters  $a$  and  $d$ :*

1.  $t = O(\log^3 n)$  and  $a = \frac{k}{\epsilon}$ .
2.  $t = O(\log^2 n)$  and  $a = O(\frac{k + \log^2 n}{\epsilon})$ .
3. for any constant  $\alpha > 0$ ,  $t = O(\log n)$  and  $a = n^{\alpha \log e} \left( \frac{k + O(\log n)}{\epsilon} \right)$ .

*Proof.* (1) is obtained by taking  $\rho = 1$  and noting that  $m \leq n$  (otherwise we just output the original source). (2) is obtained by plugging in, say,  $\rho = 4/3$  into Corollary 4.3. (3) is obtained by plugging  $\rho = e^{\alpha \ell}$  into the same corollary.  $\square$

## 5 Composing condensers

The condenser of Corollary 4.5(3) uses  $O(\log n)$  truly random bits and shrinks the source by a polynomial factor while preserving all of the entropy. We now take such a condenser and compose it with itself several times to get a much denser source. We first define this composition.

**Definition 5.1.** *Given two condensers*

- $C_1 : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow \{0, 1\}^{m_1}$ , and

- $C_2 : \{0, 1\}^{m_1} \times \{0, 1\}^{t_2} \rightarrow \{0, 1\}^{m_2}$

we define  $(C_2 \circ C_1) : \{0, 1\}^n \times \{0, 1\}^{t_1+t_2} \rightarrow \{0, 1\}^{m_2}$  by  $(C_2 \circ C_1)(x; y_1, y_2) = C_2(C_1(x, y_1), y_2)$ .

**Lemma 5.2 (condenser composition).** *Suppose  $C_1 : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow \{0, 1\}^{m_1}$  is a (strong)  $(n, k) \rightarrow_{\epsilon_1} (m_1, k_1)$  condenser, and  $C_2 : \{0, 1\}^{m_1} \times \{0, 1\}^{t_2} \rightarrow \{0, 1\}^{m_2}$  is a (strong)  $(m_1, k_1) \rightarrow_{\epsilon_2} (m_2, k_2)$  condenser. Then  $C_2 \circ C_1$  is a (strong)  $(n, k) \rightarrow_{\epsilon_1+\epsilon_2} (m_2, k_2)$  condenser.*

*Proof.* Let  $X \subseteq \{0, 1\}^n$  be a distribution with min-entropy  $k$ .

In the non-strong case, the lemma is easy:  $C_1(X, U_{t_1})$  is  $\epsilon_1$ -close to a distribution  $X_1$  with  $k_1$  min-entropy,  $C_2(X_1, U_{t_2})$  is  $\epsilon_2$ -close to a distribution with  $k_2$  min-entropy, and therefore  $C_2(C_1(X, U_{t_1}), U_{t_2})$  is  $(\epsilon_1 + \epsilon_2)$ -close to a distribution with  $k_2$  min-entropy.

For the strong case, we start from the observation that  $U_{t_1} \circ C_1(X, U_{t_1})$  is  $\epsilon_1$ -close to a random variable  $U_{t_1} \circ D'$  distributed on  $\{0, 1\}^{t_1} \times \{0, 1\}^{m_1}$  with min-entropy at least  $t_1 + k_1$ . Note that for all  $y \in \{0, 1\}^{t_1}$ , conditioned on  $U_{t_1} = y$ ,  $D'$  still has min-entropy at least  $k_1$ . Therefore, conditioned on  $U_{t_1} = y$ , the distribution  $U_{t_1} \circ U_{t_2} \circ C_2(D', U_{t_2})$  is  $\epsilon_2$ -close to a distribution having min-entropy  $t_2 + k_2$ . Replacing  $U_{t_1} \circ D'$  with  $U_{t_1} \circ C_1(X, U_{t_1})$  we conclude that  $U_{t_1} \circ U_{t_2} \circ C_2(C_1(X, U_{t_1}), U_{t_2})$  is  $(\epsilon_1 + \epsilon_2)$ -close to a distribution with min-entropy  $t_1 + t_2 + k_1 + k_2$ .  $\square$

We now analyze iterated composition of a condenser with itself many times.

## 5.1 Iterated composition

If  $C = \{C_n : \{0, 1\}^n \times \{0, 1\}^{t(n)} \rightarrow \{0, 1\}^{m(n)}\}$  is a family of (strong) lossless  $(n, k) \rightarrow_{\epsilon} (m(n), k)$  condensers, we can compose them repeatedly. Given  $n_1, k$  and  $\epsilon > 0$ , define  $C^{(1)} = C_{n_1}$  and, for  $i > 1$ , define

$$C^{(i)} = C_{n'} \circ C^{(i-1)},$$

where  $n'$  is the output length of  $C^{(i-1)}$ . We now prove a lemma about iterated composition:

**Lemma 5.3 (iterated composition).** *Fix  $n_1, k$ , and  $\epsilon > 0$ , and let*

$$C = \{C_n : \{0, 1\}^n \times \{0, 1\}^{t(n)} \rightarrow \{0, 1\}^{m(n)}\}$$

*be a family of (strong) lossless  $(n, k) \rightarrow_{\epsilon} (m(n), k)$  condensers. Assume that  $\forall n \leq n_1$ , we have  $t(n) \leq b \log n$  (for some fixed  $b \geq 0$ ) and  $m(n) \leq n^a \Delta$  (for some fixed  $a < 1$  and  $\Delta > 0$ ). Then for all  $i \geq 1$ ,*

$$C^{(i)} : \{0, 1\}^{n_1} \times \{0, 1\}^t \rightarrow \{0, 1\}^m$$

*is a (strong) lossless  $(n_1, k) \rightarrow_{i\epsilon} (m, k)$  condenser, with*

- $m \leq \Delta^{\frac{1}{1-a}} \cdot n_1^{(a^i)}$ , and,
- $t \leq \frac{b}{1-a} \log n_1 + \frac{ib}{1-a} \log \Delta$ .

*Proof.* We use Lemma 5.2. The error accumulates additively and becomes  $i\epsilon$  as desired.

Let  $n_i$  be the input length of  $C^{(i)}$  and let  $t_i$  be the seed length of  $C^{(i)}$ . We know  $n_1$ , and for  $i > 1$ , we have  $n_i \leq \Delta \cdot n_{i-1}^a$ . Thus

$$m \leq n_i \leq \Delta \Delta^a \Delta^{a^2} \dots \Delta^{a^{i-1}} n_1^{a^i} \leq \Delta^{\frac{1}{1-a}} n_1^{a^i}.$$

For the seed lengths, we have  $t_i \leq b \log n_i$  for all  $i \geq 1$ . Therefore:

$$\begin{aligned} t = \sum_{j=1}^i t_j &\leq b \sum_{j=1}^i \log n_j \\ &\leq \frac{ib}{1-a} \log(\Delta) + b \log n_1 \sum_{j=1}^i a^j \\ &\leq \frac{ib}{1-a} \log(\Delta) + \frac{b}{1-a} \log n_1. \end{aligned}$$

□

Having that we can prove:

**Lemma 5.4.** *For every  $n, k$ , and  $\epsilon \in (0, \frac{1}{2})$ , and every constant  $\delta > 0$ , there exists an explicit strong lossless  $(n, k) \rightarrow_{\epsilon} (n', k)$  condenser  $C : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^{n'}$  with  $t = O(\log n)$  and  $n' = O(\left(\frac{k}{\epsilon}\right)^{1+\delta})$ .*

*Proof.* First, we may assume that  $k \geq \log n$ ; otherwise the condenser mentioned in [22] suffices. Now we plug Corollary 4.5(3) into Lemma 5.3 as follows.

We choose  $\epsilon' = O(\frac{\epsilon}{\log \log n})$  and  $\Delta = O(k/\epsilon')$ . We also set  $\delta' = \delta/2$  and  $a = \alpha \log e = \frac{\delta'}{2+\delta'}$ . By Corollary 4.5(3) there is a family of explicit strong lossless  $(n, k(n)) \rightarrow_{\epsilon'(n)} (m(n), k(n))$  condensers  $A : \{0, 1\}^n \times \{0, 1\}^{t(n)} \rightarrow \{0, 1\}^{m(n)}$  with  $t(n) \leq b \log(n)$  (for some constant  $b > 1$ ) and  $m(n) = n^a \Delta$ .

We now look at the composed condenser  $C = A^{(i)}$  for  $i = \log_{\frac{1}{a}} \frac{2 \log n}{\delta' \log \Delta}$ . By Lemma 5.3  $C$  is a strong lossless  $(n, k) \rightarrow_{i\epsilon'} (n', k)$  condenser  $C : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^{n'}$ , with

$$n' \leq \Delta^{\frac{1}{1-a}} \cdot n^{a^i} = \Delta^{1+\delta'/2} \Delta^{\delta'/2} = \Delta^{1+\delta'} \leq O\left(\left(\frac{k}{\epsilon'}\right)^{1+\delta'}\right) \leq O\left(\left(\frac{k}{\epsilon}\right)^{1+\delta}\right).$$

Also,

$$t = \frac{b}{1-a} \log n + \frac{ib}{1-a} \log \Delta.$$

We notice that  $\frac{b}{1-a}$  is a constant, and that  $i = O(\log \frac{\log n}{\log \Delta}) = O(\frac{\log n}{\log \Delta})$ . Therefore  $i \log \Delta = O(\log n)$  and  $t = O(\log n)$ . Finally,  $i\epsilon' = O(\epsilon' \log \log n) = O(\epsilon)$ . □

Finally, we can compose the condenser of Lemma 5.4 with the condenser of Corollary 4.5(2) to get:

**Corollary 5.5.** *For every  $n, k$ , and  $\epsilon \in (0, \frac{1}{2})$  there is an explicit strong lossless  $(n, k) \rightarrow_{O(\epsilon)} (n', k)$  condenser  $C : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^{n'}$  with  $n' = O(\frac{k}{\epsilon})$  and  $t = O(\log n + \log^2(\frac{k}{\epsilon}))$ .*



## 5.2 Extractors for low min-entropy

We now get better extractors for low-min-entropies, and also prove Theorem 1.2, by first condensing the length  $n$  input to length  $k^{1+\delta}$ , and then applying known extractors that work for sources with min-entropy that is polynomial in the source length. We start with:

**Lemma 5.6 (composing an extractor and a condenser).** *Let  $C : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^{n'}$  be a (strong) lossless  $(n, k) \xrightarrow{\epsilon} (n', k)$  condenser, and let  $E : \{0, 1\}^{n'} \times \{0, 1\}^t \rightarrow \{0, 1\}^m$  be a (strong)  $(k, \epsilon)$  extractor. Then*

$$E' : \{0, 1\}^n \times \{0, 1\}^{d+t} \rightarrow \{0, 1\}^m$$

defined by  $E'(x; y, z) = E(C(x, y), z)$  is a (strong)  $(k, \epsilon + \epsilon')$  extractor.

*Proof.* The proof is almost identical to the proof of Lemma 5.2 and we omit it.  $\square$

We can now prove Theorem 1.2.

*Proof of Theorem 1.2.* Fix  $n$  and  $k = k(n) \leq \sqrt{n}$ . Using Lemma 5.2, compose:

- A (strong) lossless  $(n, k) \xrightarrow{\epsilon=k^{-1/2}} (k^2, k)$  condenser  $C : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^{k^2}$  having  $d = O(\log n)$ , given by Lemma 5.4 and,
- A (strong)  $(k, \epsilon(k^2))$  extractor  $E : \{0, 1\}^{k^2} \times \{0, 1\}^{t(k^2)} \rightarrow \{0, 1\}^{m(k)}$  given in the statement of the theorem.

This produces the desired extractor.  $\square$

As a corollary (Corollary 5.8 below), we obtain the extractors listed in Table 1. The second extractor we produce obtains constant entropy loss, and for that we need the following slight strengthening of a lemma of [40].

**Lemma 5.7 ([23]).** *Suppose  $E_1 : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow \{0, 1\}^{m_1}$  is a  $(k, \epsilon_1)$ -extractor with entropy loss  $\Delta_1$  and  $E_2 : \{0, 1\}^{n+t_1} \times \{0, 1\}^{t_2} \rightarrow \{0, 1\}^{m_2}$  is a  $(\Delta_1 - 1, \epsilon_2)$ -extractor with entropy loss  $\Delta_2$ . Then  $E(x, y_1 \circ y_2) = E_1(x, y_1) \circ E_2(x \circ y_1, y_2)$  is a  $(k, 2\epsilon_1 + \epsilon_2)$ -extractor with entropy loss  $\Delta_2 + 1$ .*

**Corollary 5.8.** *For every  $n, k$ , and constant  $\epsilon > 0$ , there exist explicit  $(k, \epsilon)$  extractors  $E : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  with the following parameters:*

1. for an arbitrary constant  $\alpha > 0$ ,  $d = O(\log n)$  and  $m = k^{1-\alpha}$ .
2.  $d = O(\log n + \log^2 k (\log \log k)^2)$  and  $m = k + d - O(1)$ .

*Proof.* For (1), we use Trevisan's extractor [35] with seed length  $O(\log n)$  and output length  $k^{1-\alpha}$  in Theorem 1.2.

We use an extractor from [24] with seed length  $t_1 = O(\log^2 n (\log \log n)^2)$  and output length  $k$  as the first extractor  $E_1$  in Lemma 5.7. The entropy loss  $\Delta_1$  is just  $t_1$  for this extractor. We then use as the second extractor  $E_2$  in Lemma 5.7 a  $(k' = \Delta_1 - 1, O(1))$  extractor from [23] with seed length  $O(\log^3 n)$ , and constant entropy loss. After applying Theorem 1.2, the seed length of this

extractor becomes  $O(\log n + \log^3 k')$ . Altogether, we obtain a constant-error extractor with seed length

$$O(\log^2 n(\log \log n)^2) + O(\log n + \log(\log^2 n(\log \log n)^2)^3) = O(\log^2 n(\log \log n)^2)$$

and constant entropy loss. Plugging this into Theorem 1.2 one more time gives the claimed extractor.  $\square$

## 6 Dispersers with small entropy loss

In this section we construct the dispersers of Theorem 1.4.

We use a technique from Nisan and Ta-Shma [16] to prove Theorem 1.4. Nisan and Ta-Shma showed how to obtain an efficient *somewhere random extractor*, which we define soon. They then show how to get a disperser construction from a somewhere random extractor. We obtain Theorem 1.4 by plugging our improved low-entropy extractors into this construction.

### 6.1 A formal analysis

We start with the definition of somewhere random sources and somewhere random extractors. Given a random source with  $k$  min-entropy, a  $(k, \epsilon)$  extractor outputs a single distribution that is  $\epsilon$  close to uniform. In contrast, a *somewhere random extractor* outputs many distributions with the guarantee that at least one of them (and possibly only one) is  $\epsilon$ -close to uniform<sup>3</sup>. Thus, a somewhere random extractor is a weakening of the extractor notion.

**Definition 6.1 (somewhere random source).**  $B = (B_1, \dots, B_b)$  is a  $(b, m)$  somewhere random source if each  $B_i$  is a random variable over  $\{0, 1\}^m$  and there is a random variable  $Y$  over  $[b]$  such that for every  $i \in [b]$  we have  $(B_i | Y = i) = U_m$ .

**Definition 6.2 (somewhere random extractor).** A function  $E : \{0, 1\}^n \times \{0, 1\}^t \rightarrow (\{0, 1\}^m)^b$  is a  $(k, \epsilon)$  somewhere random extractor if for every distribution  $X$  with  $H_\infty(X) \geq k$  we have that  $E(X, U_t)$  is  $\epsilon$ -close to a  $(b, m)$  somewhere random source.

Nisan and Ta-Shma proved:

**Theorem 6.3 ([16]).** Suppose  $E_1 : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow \{0, 1\}^{t_2}$  is an explicit  $(k_1, \epsilon_1)$  extractor and  $E_2 : \{0, 1\}^n \times \{0, 1\}^{t_2} \rightarrow \{0, 1\}^m$  is an explicit  $(k_2, \epsilon_2)$  extractor. Then for any  $s > 0$  there is an explicit function  $E : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow (\{0, 1\}^m)^n$  that is a  $(k_1 + k_2 + s, \epsilon_1 + \epsilon_2 + 8n2^{-s/3})$  somewhere random extractor.

Plugging in our new extractors for low min-entropies we get:

**Lemma 6.4.** For every  $n, k$  and constant  $\epsilon$  there is a  $(k, \epsilon)$  somewhere random extractor  $S : \{0, 1\}^n \times \{0, 1\}^t \rightarrow (\{0, 1\}^m)^n$  with  $t = O(\log n)$  and  $m = k + t - (3 \log n + O(1))$ .

---

<sup>3</sup>The formal definition is slightly different.

*Proof.* Let  $d(n, k) = O(\log n + \log^2 k(\log \log k)^2)$  be the seed length of the extractor of Corollary 5.8 (2) for error  $\epsilon/4$ . The entropy loss of this extractor is a constant for constant  $\epsilon$ , so set  $\Delta$  to be the entropy loss for error  $\epsilon/4$ . We define:

$$\begin{aligned} s &= 3(\log n + \log \epsilon^{-1} + 4) \\ t_2 &= t(n, k) \\ t_1 &= t(n, t_2) \\ k_1 &= t_2 - t_1 + \Delta \\ k_2 &= k - k_1 - s. \end{aligned}$$

From Corollary 5.8 (2) we have the following explicit extractors:

- a  $(k_1, \epsilon/4)$  extractor  $E_1 : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow \{0, 1\}^{t_2}$ , and
- a  $(k_2, \epsilon/4)$  extractor  $E_2 : \{0, 1\}^n \times \{0, 1\}^{t_2} \rightarrow \{0, 1\}^{k_2+t_2-\Delta}$ .

Plugging these extractors into Theorem 6.3, we obtain a  $(k, \epsilon)$  somewhere random extractor:

$$E : \{0, 1\}^n \times \{0, 1\}^{t_1} \rightarrow (\{0, 1\}^{k+t_1-s-2\Delta})^n.$$

Note that  $t_1 = O(\log n)$  as required. □

Nisan and Ta-Shma proved that a somewhere random extractor is stronger than a disperser. Namely,

**Lemma 6.5 ([16]).** *Let  $\epsilon < 1$  and let  $E : \{0, 1\}^n \times \{0, 1\}^t \rightarrow (\{0, 1\}^m)^b$  be a  $(k, \epsilon)$  somewhere random extractor, where  $E(x, y) = E_1(x, y) \circ \dots \circ E_b(x, y)$ . Then the function  $D : \{0, 1\}^n \times \{0, 1\}^{t+\log b} \rightarrow \{0, 1\}^m$  defined by*

$$D(x; y, i) = E_i(x, y)$$

*is a  $(k, \epsilon)$  disperser.*

Plugging the somewhere random extractor of Lemma 6.4 into Lemma 6.5 proves Theorem 1.4.

## 7 An application to hardness of approximation

Umans [36] showed that the following  $\Sigma_2^p$  optimization problem is  $\Sigma_2^p$ -hard to approximate to within an  $s^{1/5-\epsilon}$  factor, for any constant  $\epsilon > 0$ , where  $s$  is the size of the instance:

**Succinct Set Cover:** given  $m$  subsets of  $\{0, 1\}^n$  whose union is  $\{0, 1\}^n$ , specified succinctly as the ones of 3-DNFs  $\phi_1, \phi_2, \dots, \phi_m$ , what is the minimum cardinality cover? i.e., what is the smallest set  $I \subseteq [m]$  for which  $\bigvee_{i \in I} \phi_i \equiv 1$ .

The main combinatorial objects used in the proof are dispersers. In fact, the exponent in the hardness ratio depends in a straightforward way on the parameters of the disperser used in the reduction. Specifically, Umans showed:

**Theorem 7.1 ([36]).** *Suppose there exist explicit  $(K = 2^k, 1/2)$  dispersers  $G = ([N = 2^n], W = [M = 2^m], E)$  with degree  $D = 2^d$ , and  $\Omega(\log n) \leq k, d \leq m \leq O(\log n)$ . Set  $r = 1 - (d + \log n)/m$ . Then Succinct Set Cover is  $\Sigma_2^P$ -hard to approximate to within  $s^{r-\epsilon}$ , for any constant  $\epsilon > 0$ , where  $s$  is the size of the instance.*

Prior to this work, the best inapproximability factor (of  $s^{1/5-\epsilon}$ ) was achieved using the extractors of [30], which use a seed of length  $4k + O(\log n)$  to extract  $k + d - O(1)$  bits with constant error. Picking  $k = c \log n$  in that construction gives  $r \rightarrow 1/5$  as  $c$  goes to infinity.

To achieve an inapproximability factor of  $s^{1-\epsilon}$ , which is optimal up to lower order terms, one needs an extractor (or disperser) for very low min-entropy  $k = O(\log n)$ , that extracts at least  $k$  bits, with a seed length of  $O(\log n)$  that has a *sublinear* dependence on  $k$ . Theorem 1.2 applied to an extractor from [23] gives us the required object, which allows us to prove:

**Theorem 7.2.** *Succinct Set Cover is  $\Sigma_2^P$ -hard to approximate to within  $s^{1-\epsilon}$ , for any constant  $\epsilon > 0$ , where  $s$  is the size of the instance.*

*Proof.* Let  $E : \{0, 1\}^n \times \{0, 1\}^d \rightarrow \{0, 1\}^m$  be the explicit  $(k = c \log n, 1/4)$  extractor from [23] with seed length  $d = O(\log^3 n)$  and output length  $m = k + d - O(1)$ . Applying Theorem 1.2, we obtain an explicit  $(k, 1/2)$  extractor  $E' : \{0, 1\}^n \times \{0, 1\}^{d'} \rightarrow \{0, 1\}^m$  with  $d' = O(\log n + \log^3 k)$  (the hidden constant here is independent of  $k$ ). Plugging this extractor into Theorem 7.1 gives  $r = 1 - (d' + \log n)/m \geq 1 - O(\log n + \log^3 k)/k$ , which approaches 1 as  $c$  approaches infinity.  $\square$

Plugging this new extractor into [36] in the same way as above gives improved inapproximability results for a number of other problems studied in that paper.

## 8 Unbalanced expanders

All of our constructions of (strong) lossless  $(n, k) \rightarrow_\epsilon (n', k)$  condensers  $C$  have the property that  $C$  is also a (strong) lossless  $(n, k') \rightarrow_\epsilon (n', k')$  condenser for all  $k' \leq k$ . This is because we prove our constructions are condensers using Lemma 3.4, which guarantees a predictor in certain circumstances, and Lemma 3.5. When the min-entropy of the source is less than  $k$ , the optimal predictor (and hence the reconstruction function) can only do better, leading to even more efficient preservation of the min-entropy in the output of  $C$ .

Condensers that have this extra property are *equivalent* to unbalanced expanders. We first prove this equivalence, and then prove Theorem 1.7 by plugging in specified condenser constructions from earlier in the paper.

**Theorem 8.1.** *Let  $C : \{0, 1\}^n \times \{0, 1\}^t \rightarrow \{0, 1\}^{n'}$  be a function. The bipartite graph  $G = (V = [2^n], W = [2^t \times 2^{n'}], E)$  defined by*

$$(x; y, z) \in E \Leftrightarrow E(x, y) = z$$

*is  $(K = 2^k, (1 - \epsilon)2^t)$ -expanding with degree  $2^t$  if and only if  $C$  is a strong lossless  $(n, k') \rightarrow_\epsilon (n', k')$  condenser for all  $k' \leq k$ .*

*Proof.* In the forward direction, let  $X$  be a subset of  $V$  with  $|X| \leq 2^k$ . Since  $G$  is  $(1 - \epsilon)2^t$  expanding, we know that the distribution  $D = U_t \circ C(X, U_t)$  has support of cardinality at least  $(1 - \epsilon)2^t |X|$ ,

which implies that  $D$  is  $\epsilon$ -close to a distribution with min-entropy  $t + \log_2 |X| = t + H_\infty(X)$ , as required.

In the other direction, let  $X$  be a subset of  $\{0, 1\}^n$  with  $|X| \leq 2^k$ . We know that the distribution  $D = U_t \circ C(X, U_t)$  is  $\epsilon$ -close to a distribution  $D'$  on  $\{0, 1\}^t \times \{0, 1\}^{n'}$  with min-entropy at least  $t + H_\infty(X)$ . Let  $\Gamma$  be the support of distribution  $D$ . Then

$$\epsilon \geq |D - D'| = \sum_{w \in \Gamma} (D(w) - D'(w)) = 1 - \sum_{w \in \Gamma} D'(w) \geq 1 - |\Gamma|2^{-(t+H_\infty(X))}.$$

Thus  $|\Gamma| \geq (1 - \epsilon)2^{t+H_\infty(X)} = (1 - \epsilon)2^t |X|$ , as required. □

Theorem 1.7 (1) follows from plugging the condenser of Lemma 5.4 into the above theorem. Theorem 1.7 (2) follows from plugging the condenser of Corollary 4.5 (1) into the above theorem.

**Acknowledgements.** This paper was born out of discussions held at a reading group organized by Ziv Bar-Yossef. We are indebted to Ziv for organizing the group and to all of the group members for intriguing discussions. Special thanks go to Ronen Shaltiel who helped us with the results presented in Section 6. We would also like to thank Russell Impagliazzo, Omer Reingold, Alex Russell, Ronen Shaltiel, Luca Trevisan, Umesh Vazirani and Avi Wigderson for useful discussions. Finally we thank the anonymous referees for many helpful comments that completely changed the way the material is presented.

## References

- [1] M. Ajtai, J. Komlós, and E. Szemerédi. Sorting in  $c \log n$  parallel steps. *Combinatorica*, 3:1–19, 1983.
- [2] M. Ajtai, J. Komlós, and E. Szemerédi. Deterministic simulation in Logspace. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 132–140, 1987.
- [3] N. Alon, 1999. Personal communication.
- [4] S. Arora, F. T. Leighton, and B. M. Maggs. On-line algorithms for path selection in a non-blocking network. *SIAM Journal on Computing*, 25(3):600–625, June 1996.
- [5] H. Buhrman, P.B. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bitvectors optimal? In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 449–458, 2000.
- [6] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree expansion beyond the degree / 2 barrier. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 659–668, 2002.
- [7] P. Feldman, J. Friedman, and N. Pippenger. Wide-sense nonblocking networks. *SIAM Journal on Discrete Mathematics*, 1:158–173, 1988.
- [8] O. Gabber and Z. Galil. Explicit construction of linear sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

- [9] O. Goldreich, R. Impagliazzo, L. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.
- [10] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Near-optimal conversion of hardness into pseudo-randomness. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 181–190, 1999.
- [11] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 1–10, 2000.
- [12] R. Impagliazzo and A. Wigderson.  $P=ppp$  unless  $e$  has subexponential circuits: derandomizing the xor lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 220–229, 1997.
- [13] N. Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42:1091–1106, 1995.
- [14] C. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, 2003.
- [15] G.A. Margulis. Explicit construction of concentrators. *Problems of Information Transmission*, 9:325–332, 1973.
- [16] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58:148–173, 1999.
- [17] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.
- [18] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.
- [19] N. Pippenger. Sorting and selecting in rounds. *SIAM Journal on Computing*, 16(6):1032–1038, December 1987.
- [20] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two super-concentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, 2000.
- [21] R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.
- [22] R. Raz and O. Reingold. On recycling the randomness of states in space bounded computation. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 159–168, 1999.
- [23] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan’s extractors. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 149–158, 1999.

- [24] O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 22–31, 2000.
- [25] M. Saks, A. Srinivasan, and S. Zhou. Explicit OR-dispersers with polylog degree. *Journal of the ACM*, 45:123–154, 1998.
- [26] M. Santha. On using deterministic functions in probabilistic algorithms. *Information and Computation*, 74(3):241–249, 1987.
- [27] R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM*, 52:172–216, 2005.
- [28] M. Sipser. Expanders, randomness, or time vs. space. *Journal of Computer and System Sciences*, 36:379–383, 1988.
- [29] M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.
- [30] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28:1433–1459, 1999.
- [31] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the XOR lemma. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 537–546, 1999.
- [32] A. Ta-Shma. On extracting randomness from weak random sources. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*, pages 276–285, 1996.
- [33] A. Ta-Shma. Almost optimal dispersers. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 196–202, 1998.
- [34] A. Ta-Shma, D. Zuckerman, and S. Safra. Extractors from Reed-Muller codes. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 638–647, 2001.
- [35] L. Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 141–148, 1999.
- [36] C. Umans. Hardness of approximating  $\Sigma_2^P$  minimization problems. In *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science*, pages 465–474, 1999.
- [37] C. Umans. Pseudo-random generators for all hardnesses. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 627–634, 2002.
- [38] Christopher Umans. Reconstructive dispersers and hitting set generators. In *APPROX-RANDOM*, pages 460–471, 2005.
- [39] L.G. Valiant. Graph theoretic properties in computational complexity. *Journal of Computer and System Sciences*, 13:278–285, 1976.

- [40] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.
- [41] D. Zuckerman. General weak random sources. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 534–543, 1990.
- [42] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16:367–391, 1996.
- [43] D. Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11:345–367, 1997.