# Loss-less Condensers, Unbalanced Expanders, and Extractors

Amnon Ta-Shma [*] Christopher Umans [†] David Zuckerman [‡]

## ABSTRACT

An extractor is a procedure which extracts randomness from a defective random source using a few additional random bits. Explicit extractor constructions have numerous applications and obtaining such constructions is an important derandomization goal. Trevisan recently introduced an elegant extractor construction, but the number of truly random bits required is suboptimal when the input source has low min-entropy. Significant progress toward overcoming this bottleneck has been made, but so far has required complicated recursive techniques that lose the simplicity of Trevisan's construction.

We give a clean method for overcoming this bottleneck by constructing *loss-less condensers*, which compress the $n$-bit input source without losing any min-entropy, using $O(\log n)$ additional random bits. Our condensers are built using a simple modification of Trevisan's construction, and yield the best extractor constructions to date.

Loss-less condensers also produce unbalanced bipartite expander graphs with small (polylogarithmic) degree $D$ and very strong expansion of $(1-\epsilon)D$. We give other applications of our construction, including dispersers with entropy loss $O(\log n)$, depth two super-concentrators whose size is within a polylog of optimal, and an improved hardness of approximation result.

[*] 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. This work was done while supported in part by a David and Lucile Packard Fellowship for Science and Engineering and NSF NYI Grant No. CCR-9457799.

[†] Microsoft Research, One Microsoft Way, Redmond, WA 98052. email: umans@microsoft.com. 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 Grant No. CCR-9820897.

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



*STOC'01,* July 6-8, 2001, Hersonissos, Crete, Greece.



## 1. INTRODUCTION

### 1.1 History and Background

Sipser [23] and Santha [22] were the first to realize that extractor-like structures can be used to save on randomness. Their structure is called today a "disperser[1]." They showed 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 in [15]:

DEFINITION 1.1.1 (EXTRACTOR, MIN-ENTROPY). *Function $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ 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. The goal is to construct *explicit* extractors[2] that 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 achieve good parameters only for 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 $\epsilon$. Building on [32, 33], Nisan and Zuckerman [15] constructed an extractor with $t = O(\log^2 n)$ for high min-entropy $k = \Omega(n)$. Srinivasan and Zuckerman [25] extended this solution to the case $k = n^{1/2+\epsilon}$ and Ta-Shma [26] further extended it for any min-entropy $k$. Also, Ta-Shma was the first to extract all the min-entropy from the source. Zuckerman [34] showed a construction with $t = O(\log n)$ working for high min-entropies $k = \Omega(n)$.

[1] For a definition see Subsection 1.4.

[2] An extractor $E$ is explicit if $E$ can be evaluated in polynomial time.

| required entropy | no. of truly random bits | no. of output bits | reference |
|---|---|---|---|
| Any $k$ | $\log n + \Theta(1)$ | $k + t - \Theta(1)$ | L. bound non-explicit [17] |
| $\Omega(n)$ | $O(\log^2 n)$ | $\Omega(k)$ | [15] |
| $\Omega(n)$ | $O(\log n)$ | $\Omega(k)$ | [34] |
| Any $k$ | polylog$(n)$ | $m = k$ | [26] |
| Any $k$ | $O(\log^2 n / \log k)$ | $k^{1-\alpha}$ | [28] |
| Any $k$ | $O(\log n)$ | $k/\log n$ | [20] |
| Any $k$ | $O(\log n + \log k (\log\log k)^2)$ | $(1-\alpha)k$ | Thm. 5 |
| Any $k$ | $O(\log n + \log^2 k (\log\log k)^2)$ | $k + t - O(1)$ | Thm. 5 |

**Table 1: Milestones in building explicit extractors. The error $\epsilon$ is a constant.**

Departing from previous techniques, Trevisan [28] showed a connection between pseudorandom generators for small circuits and extractors. Thus, while all previous work used hashing and pair-wise independence in various forms, and viewed extractors as sophisticated hash functions, Trevisan's approach viewed extractors as pseudorandom generators against all statistical tests. Trevisan then used the NW pseudorandom generator [14] 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. [9, 10] used sophisticated (and complex) recursive techniques building on Trevisan's construction. Reingold, et al. [20] improved their result by combining the old hashing techniques with the new extractors, and adding new tricks. 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 [25] (for extremely small $k = O(\log n)$), and [34] (for very large $k = \Omega(n)$). Our results dramatically extend the range of min-entropies $k$ for which these parameters can be achieved to all $k \leq 2^{\log^{1-o(1)} n}$.

## 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 [9, 10, 20]) 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 4 for the proof):

THEOREM 1. *Suppose that there is an explicit family of $(k = k(n) = \sqrt{n}, \epsilon(n))$-extractors,*

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

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

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

*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 shorter weak source with about the same min-entropy, with some error. Our condenser uses $O(\log n)$ random bits to transform a length $n$ source with min-entropy $k$ into a source with length $(k/\epsilon)^{1+\delta}$ that is $\epsilon$-close to a source with the same min-entropy $k$. We can then apply existing extractors, e.g., Trevisan's extractor itself, to this shorter source. With that we get the results listed in Table 1.

We remark that [20] also build extractors by first using condensers. However, our condensers differ from theirs in two important ways. Most significantly, our condensers are *loss-less* in the sense 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. Second, our condensers must work for sources having *at most* some min-entropy $k$, while in [20], the condensers must work for sources having *at least* some min-entropy $k$. This subtle difference means that our condensers 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. The condenser construction in [20] require fairly complicated recursive techniques, while we use a simplification of the approach of Impagliazzo, et.al. [9, 10] 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 $\mathbf{TR} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ 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 [28], Impagliazzo, et al. [9, 10], 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-negligable 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-negligable 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

| Additional randomness $t$ | Entropy loss | Reference |
|---|---|---|
| $O(\log n)$ | polylog$n$ | [27] |
| $O(\log n)$ | $3\log n + O(1)$ | This paper |
| $\log n + O(1)$ | $O(1)$ | L. bound [17] |

**Table 2: Explicit dispersers with constant error.**

| Size | Reference |
|---|---|
| $O(N \cdot 2^{\text{poly}(\log\log N)})$ | [21] |
| $O(N \cdot \text{polylog}N)$ | This paper |
| $O(N \cdot \frac{\log^2 N}{\log\log N})$ | Lower bound, [17] |

**Table 3: Explicit depth two super-concentrators**

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* **TR** *cannot be an extractor*, and we output the advice strings themselves. Then, unconditionally, the advice strings constitute short descriptions of a non-negligable 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 $\rho m \le n^\gamma$, and we can easily have $\rho = n^{\Omega(1)}$, avoiding the bottleneck altogether.

In retrospect, our technique may seem an obvious simplification of [9, 10]. 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 1.4.1 (disperser). *A bipartite graph $G = (V_1 = [N = 2^n], V_2 = [M = 2^m], E)$ 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 $V_2$.*

Theorem 2. *For every $n, k$ and constant $\epsilon$ there is a degree $D = \text{poly}(n)$ explicit $(K = 2^k, \epsilon)$ disperser $G = (V_1 = [N = 2^n], V_2 = [\Omega(KD/n^3)], E)$.*

Ideally, $K$ vertices of a degree $D$ graph can have $KD$ neighbors. However, a lower bound of [17] shows that in any $(K, \epsilon)$ disperser $G = (V_1, V_2 = [M], E)$ the size of $V_2$ 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 nice thing about the disperser of Theorem 2 is that it has only $3\log n + O(1)$ entropy loss, while still having a very small degree $D = \text{poly}(n)$. We compare the parameters of our disperser with the previous best construction and the optimal ones in Table 2.

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

Definition 1.4.2. *$G = ((V_1, V_2, V_3), E)$ 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 obtain this result by plugging our disperser into [31]. A long line of papers has tried to solve this problem. The previous best result and our result is summarized in Table 3. We also get an explicit $a$–expanding graph with $N$ vertices that has degree $O(\frac{N}{a} \cdot \text{polylog}N)$, and we improve a hardness result of Umans [29]. See Section 6 for more details on these applications, previous results and lower bounds.

## 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: network constructions [6], sorting [1, 16], complexity theory [30], cryptography [8], and pseudorandomness [2]. Many of these applications require bipartite graphs, where only subsets on one side are required to expand.

Definition 1.5.1 (expander). *A bipartite graph $G = (V_1, V_2, E)$ is $(K, c)$ expanding if for every $A \subseteq V_1$ of cardinality at most $K$, $|\Gamma(A)| \ge c|A|$, 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 degree $T$ ($T$ is the degree of all vertices in $V_1$). Random graphs have $c \ge T - 2\log_M N - o(1)$ if $K < N^{.49}$. 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 [12, 7]. These explicit constructions relied on showing an upper bound on the second largest eigenvalue of the adjacency matrix corresponding to the graph. Kahale [11] showed that such methods cannot achieve $c > T/2$. Yet some applications, such as [4, 24, 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 $A$.

Previously, 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 $|V_1| \gg |V_2|$, which is desired in the above applications. As mentioned above, our loss-less 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 3. *For every positive constant $\alpha$ and function $\epsilon = \epsilon(N)$ there is an explicit family of degree $T$ graphs $G =$*

$(V_1 = [N], V_2 = [M], E)$ *that is* $(K = 2^k, (1-\epsilon)T)$ *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.

# 2. PRELIMINARIES

A probability distribution $D$ on $\Lambda$ is a function $D : \Lambda \to [0,1]$ 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 $\frac{1}{2}\sum_{x\in\Lambda}|D_1(x) - D_2(x)| = \max_{S\subseteq X}|D_1(S) - D_2(S)|$. 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 $a \in A$.

Suppose $D$ is a distribution over $A \times B$. We denote $D = D_1 \circ D_2$, where $D_1$ is the distribution $D$ induces on $A$ and $D_2$ the distribution induced on $B$. We denote $(D_2|D_1 = a)$ the distribution $D$ induces on $B$ given that $D_1 = a$.

Though we deal primarily with min-entropy (defined in the Introduction), some proofs will also require the usual notion of entropy:

Definition 2.0.2 (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* $H(p) = -p\log p - (1-p)\log(1-p)$.

## 2.1 Weak Designs

The following definition and lemma are from [19]:

Definition 2.1.1 (Weak design). *A family of sets* $\Delta = (S_1, S_2, \ldots S_m) \subseteq [d]$ *is a weak* $(\ell, \rho)$ *design if*

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

Lemma 2.1.1 ([19]). *For every* $\ell, m$ *and* $\rho \geq 1$, *there exists a weak* $(\ell,\rho)$ *design* $\Delta = (S_1, S_2, \ldots S_m) \subseteq [d]$ *that can be constructed in poly*$(m,d)$ *time, where*

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

## 2.2 Condensers

In this section we define loss-less condensers and show their equivalence to unbalanced expanders. Our definitions are for flat distributions, though at the end of this section we show this is not necessary.

Definition 2.2.1 (Condenser). *Function* $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$ *is an* $(n,k) \to_\epsilon n'$ *condenser if for every* flat *distribution* $X$ *with* $H_\infty(X) \leq k$ *the distribution* $U_t \circ C(X, U_t)$ *is* $\epsilon$ *close to a distribution* $D = D_1 \circ D_2$ *over* $\{0,1\}^t \times \{0,1\}^{n'}$ *such that* $D_1 = U_t$ *and for every* $y \in \{0,1\}^t$ $H_\infty(D_2|D_1 = y) \geq H_\infty(X)$. $C$ *is* explicit *if it can be computed in polynomial time.*

Given a function $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$ we define the bipartite graph $G_C = (V_1 = [2^n], V_2 = [2^{t+n'}], E)$ with left degree $T = 2^t$ as follows: we identify $V_1$ with $\{0,1\}^n$, $V_2$ with $\{0,1\}^{t+n'}$ and we put an edge $(x,(y,z)) \in E$ iff $C(x,y) = z$. Intuitively, this graph has strong expansion if for most $y$ and all large sets $X$, $C(X,y)$ is "almost one-to-one." The next definition and lemma make this intuition precise.

Definition 2.2.2 (Inverter). $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$ *has a* $(k,\delta)$ *inverter if for all flat distributions* $X$ *with* $H_\infty(X) \leq k$, *there exists a function* $I_X : \{0,1\}^t \times \{0,1\}^{n'} \to \{0,1\}^n$ *for which*

$$\Pr_{x\in X,y}[I_X(y; C(x,y)) = x] \geq \delta.$$

Lemma 2.2.1. *Let* $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$ *and* $\epsilon > 0$. *The following are equivalent:*

1. $C$ *is an* $(n,k) \to_\epsilon n'$ *condenser.*
2. $G_C = (V_1 = [2^n], V_2 = [2^{t+n'}], E)$ *is* $(K = 2^k, (1-\epsilon)2^t)$ *expanding with degree* $2^t$.
3. $C$ *has a* $(k, 1-\epsilon)$ *inverter.*

We give the proof in Subsection 3.4.

Condensers and expanders are defined for flat distributions. However, like with extractors, one could give the following equivalent definition:

Definition 2.2.3 (condenser - alternate definition). *Let* $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$. $C$ *is an* $(n,k) \to_\epsilon n'$ *condenser if for every (flat or non-flat) distribution* $X$ *with* $H_\infty(X) \leq k$, *the distribution* $U_t \circ C(X,U_t)$ *is* $\epsilon$ *close to a distribution* $D = D_1 \circ D_2$ *over* $\{0,1\}^t \times \{0,1\}^{n'}$ *such that* $D_1 = U_t$ *and for every* $y \in \{0,1\}^t$ $H_\infty(D_2|D_1 = y) \geq H_\infty(X)$.

In Subsection 3.4 we prove:

Lemma 2.2.2. *Definitions 2.2.1 and 2.2.3 are equivalent.*

# 3. THE SUBSET CONDENSER

In this section we present an explicit loss-less condenser that is the basis of our constructions of expanders, extractors, and dispersers. Our condenser is based on the "nearly-disjoint subsets" pseudorandom generator [14] and its interpretation in the extractor setting discovered by Trevisan [28]. We begin with a key lemma.

## 3.1 A next-bit predictor lemma

An important tool used in proving that a construction is a pseudorandom generator is a "next-bit predictor". Next-bit predictors are used for the NW generator [14], and all of the extractors based on it, to construct small circuits or descriptions of certain functions. We use a next-bit predictor in a new way to prove that the output of our construction retains most of the source entropy. In this context, we can use a much stronger next-bit predictor, which we present now, and indeed this strengthening is critical for our results.

Definition 3.1.1. *A distribution* $Y = (Y_1, Y_2, \ldots, Y_m)$ *on* $\{0,1\}^m$ *has a* $\delta$ next-bit predictor *if there is a function* $T$ *such that*

$$\Pr_{I\in[m],Y}[T(I; Y_1, Y_2, \ldots, Y_{I-1}) = Y_I] \quad \geq \quad \delta$$

Note that a next-bit predictor need not be efficient. Yao noticed that:

Lemma 3.1.1 (Yao's next-bit predictor). *If distribution $Y = (Y_1, Y_2, \ldots, Y_m)$ over $\{0,1\}^m$ is not $\epsilon$ close to uniform then $Y$ has a $\delta = 1/2 + \frac{\epsilon}{m}$ next-bit predictor.*

If $m$ is much larger than the entropy of $Y$, then we can improve Yao's lemma and achieve success probability close to one.

Lemma 3.1.2 (strong next-bit predictor). *If a distribution $Y = (Y_1, Y_2, \ldots, Y_m)$ over $\{0,1\}^m$ has entropy $H(Y) \le \epsilon m$, then $Y$ has a $1-\epsilon$ next-bit predictor $T$.*

Proof. For $I$ chosen uniformly at random from $[m]$, let the random variable $Z$ be the concatenation $I \circ Y_1 \circ \ldots \circ Y_{I-1}$. Then

$$H(Y_I|Z) = \frac{1}{m}\sum_{i=1}^{m} H(Y_i|Y_1, Y_2, \ldots Y_{i-1}) = \frac{1}{m}H(Y) \le \epsilon.$$

Conditioned on $Z = z$, $Y_I$ has some probability $p_z$ of being 1. An optimal next-bit predictor will predict 1 if $p_z > 1/2$ and 0 otherwise. The error of this next-bit predictor will be $\mathrm{E}[\min(p_Z, 1-p_Z)]$. Since

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

we get that $\mathrm{E}[\min(p_Z, 1-p_Z)] \le H(Y_I|Z) \le \epsilon$, as required.

□

## 3.2 The subset condenser

To describe our condenser, we first need some notation for specifying substrings indexed by weak designs. Let $\Delta = (S_1, \ldots, S_m) \subset [d]$ be a weak $(\ell, \rho)$ design. For a string $\beta \in \{0,1\}^{d-\ell}$ and $i \in [m]$, let $\beta^{\neq i}$ be the length $d$ string that has $\beta$ everywhere except for the places indexed by $S_i$ and is undefined in the locations indexed by $S_i$. Now, given an additional $j \in [m]$ and $\gamma \in \{0,1\}^{|S_i \cap S_j|}$ let $\beta^{\neq i} \circ \gamma^{i \cap j}$ be the string $\beta^{\neq i}$ with $\gamma$ at the locations indexed by $S_i \cap S_j$. Similarly, for $a \in \{0,1\}^\ell$, let $\beta^{\neq i} \circ a^i$ denote the string $\beta^{\neq i}$ with the string $a$ inserted at the positions indexed by $S_i$.

As usual, for any string $w \in \{0,1\}^d$, the $\ell$-bit string $w_{|S_i}$ is $w$ restricted to the locations indexed by $S_i$. Notice that for any $j \neq i$, $(\beta^{\neq i} \circ \gamma^{i \cap j})|_{S_j}$ is well-defined.

By a slight abuse of this notation, for $a \in \{0,1\}^\ell$, we denote the by $a^i_{|S_j}$ the $|S_i \cap S_j|$-bit string obtained by placing $a$ in the positions indexed by $S_i$ and then restricting it to the positions indexed by $S_i \cap S_j$.

Our subset condenser $C_\Delta$ is described in Figure 1.

Theorem 4 (subset condenser). *Pick $\rho \ge 1$. For every $n$, $k$, and $\epsilon \in (0, \frac{1}{2})$, there exists an explicit $(n,k) \to_{O(\epsilon)} n'$ condenser with $n' \le \rho k/\epsilon$ and $t \le d(\ell = \log n + O(1), \rho)$.*

Proof. If $m = k/\epsilon$ is greater than $n$, then the stated bound on $n'$ is greater than $n$ (because $\rho \ge 1$). In this case the original source satisfies the theorem. Otherwise, we may assume $m \le n$, and we proceed to show that $C_\Delta$ is the desired condenser.

We first argue (via Lemma 3.1.2) that next-bit predictors exist for certain distributions related to $C_\Delta$'s output. Define the function $\mathbf{TR} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ by $\mathbf{TR}(x,y)_i = \hat{x}(y_{|S_i})$ ($\mathbf{TR}$ to suggest the Trevisan extractor). For each flat distribution $X$ over $\{0,1\}^n$ with $H(X) \le k$ and for each $y \in \{0,1\}^d$, let $D_{X,y}$ to be the distribution over $\{0,1\}^m$ induced by $\mathbf{TR}(X,y)$. Because $\mathbf{TR}$ is a deterministic function, $H(D_{X,y}) \le k$. However, $m = k/\epsilon$, and therefore by Lemma 3.1.2, $D_{X,y}$ has a $1-\epsilon$ next-bit predictor $T_{X,y}$.

Using these next bit predictors, we define an inverter family $I_X$ for $C_\Delta$.

*The inverter family $I_X$.*

**Input** : $\beta \in \{0,1\}^{d-\ell}$, $i \in [m]$, and $z \in \{0,1\}^{n'}$. $z$ is indexed by $1 \le j < i$ and $\gamma \in \{0,1\}^{|S_i \cap S_j|}$, the value that corresponds to $j$ and $\gamma$ is denoted $z_{j,\gamma}$.

**Algorithm** :

- For every $a \in \{0,1\}^\ell$ we define:
  $$w_a = T_{X,\beta^{\neq i}\circ a^i}\left(i; z_{1,a^i_{|S_1}}, \ldots z_{i-1,a^i_{|S_{i-1}}}\right)$$
  and we view $w$ as a length $\bar{n}$ string.
- If there is a unique $x \in \{0,1\}^n$ such that $\mathrm{dist}(\hat{x}, w) \le 0.1$ then output $x$, otherwise output "don't know".

Lemma 3.2.1. *The function family $I_X$ is a $(k, 1-O(\epsilon))$ inverter for $C_\Delta$.*

Proof. Fix a flat distribution $X$ over $\{0,1\}^n$ with $H_\infty(X) \le k$. Recall that for all $y$,

$$\Pr_{\alpha \in D_{X,y}, i \in [m]}[T_{X,y}(i; \alpha_1, \ldots, \alpha_{i-1}) = \alpha_i] \ge 1-\epsilon. \quad (1)$$

Averaging over $y$ and using the definition of $D_{X,z}$ and $\mathbf{TR}$,

$$\Pr_{x \in X, y, i}[T_{X,y}(i; \mathbf{TR}(x,y)_1, \ldots, \mathbf{TR}(x,y)_{i-1}) = \mathbf{TR}(x,y)_i] \ge (1-\epsilon). \quad (2)$$

Denote by $y_{|\neq S_i}$ the $(d-\ell)$-bit string obtained by restricting $y$ to those coordinates *not* in $S_i$ and denote by $y_{|S_i \cap S_j}$ the string obtained by restricting $y$ to those coordinates indexed by $S_i \cap S_j$. When $j < i$, we have $\mathbf{TR}(x,y)_j = C_\Delta(x; y_{|\neq S_i}, i)_{j, y_{|S_i \cap S_j}}$. Using this fact and the definition of $\mathbf{TR}$, we can rewrite Equation 2 to get:

$$\Pr_{x \in X, y, i}[T_{X,y}(i; C_\Delta(x; y_{|\neq S_i}, i)_{1, y_{|S_i \cap S_1}}, \ldots, C_\Delta(x; y_{|\neq S_i}, i)_{i-1, y_{|S_i \cap S_{i-1}}}) = \hat{x}(y_{|S_i})] \ge 1-\epsilon$$

If we pick $y \in \{0,1\}^d$ by first picking $\beta \in \{0,1\}^{d-\ell}$ for the positions not in $S_i$, and then $a \in \{0,1\}^\ell$ for the positions in $S_i$, we obtain, using a Markov argument,

$$\Pr_{x \in X, i, \beta}[\Pr_a[T_{X,\beta^{\neq i}\circ a^i}(i; C_\Delta(x;\beta,i)_{1,a^i_{|S_1}}, \ldots, C_\Delta(x;\beta,i)_{i-1,a^i_{|S_{i-1}}}) = \hat{x}(a)] \ge 0.9] \ge 1-10\epsilon. \quad (3)$$

**Parameters** : $n$, $k$, $\epsilon$, and $\rho \geq 1$, which controls a tradeoff between output length and the number of truly random bits.

**A binary code** : Let **BC** be an $[\bar{n}, n, 0.3]_2$ binary code, $\bar{n} = O(n)$, and denote $\hat{x} = \mathbf{BC}(x)$.

**Weak Design** : A weak $(\ell, \rho)$ design $\Delta = \{S_1, \ldots, S_m\} \subseteq [d]$, with:

- $\ell = \log \bar{n} = \log n + O(1)$,
- $m = k/\epsilon$.
- $d = d(\ell, \rho)$ as defined in Lemma 2.1.1.

**Input** : $x \in \{0,1\}^n$.

**Random coins** : A random string $\beta \in \{0,1\}^{d-\ell}$ and a random $i \in [m]$.

**Output** : The output has $n' \leq \rho(m-1)$ bits that are indexed by $1 \leq j < i$ and $\gamma \in \{0,1\}^{|S_i \cap S_j|}$. We have: $C_\Delta(x; \beta, i)_{j,\gamma} = \hat{x}((\beta^{\neq i} \circ \gamma^{i \cap j})|_{S_j})$.

**Figure 1: The subset condenser $C_\Delta$.**

On input $\beta, i, z = C_\Delta(x; \beta, i)$, inverter $I_X$ computes the string $w$ whose $a$-th bit is exactly equal to the evaluation of $T_{X,\beta^{\neq i} \circ a^i}$ in Equation 3, for $a \in \{0,1\}^\ell$. Equation 3 therefore implies that with probability at least $1 - 10\epsilon$ (over $x, \beta$ and $i$), the inverter $I_X$ outputs $x$, (because $\mathrm{dist}(\hat{x}, w) \leq 0.1$ and the code **BC** has relative distance 0.3, so $\hat{x}$ is the unique codeword close to $w$). We conclude that

$$\Pr_{x \in X, \beta, i}[I_X(\beta, i; C_\Delta(x; \beta, i)) = x] \geq 1 - 10\epsilon. \qquad (4)$$

As this holds for every flat distribution $X$ with $H_\infty(X) \leq k$, we have shown that $C_\Delta$ has a $(k, 1 - O(\epsilon))$ inverter. □

By Lemma 2.2.1, $C_\Delta$ is a $(n, k) \to_{O(\epsilon)} n'$ condenser. We just need to verify the bounds on $n'$ and $t$. By the properties of weak designs, the output length $n' \leq \rho m = \rho k/\epsilon$ as required. For $t$, we use $d - \ell$ bits to sample $\beta$, at most $\log m$ bits to sample $i$. As noted above, we can assume $m \leq n$, and we know that $\ell \geq \log n$, so $t \leq d$.

Three specific cases are:

Corollary 3.2.1. *For every $n$,$k$ and $\epsilon \in (0, \frac{1}{2})$, there exist explicit $(n, k) \to_{O(\epsilon)} n'$ condensers with any of the following three choices for parameters $n'$ and $t$:*

1. *$n' \leq k/\epsilon$ and $t \leq O(\log^3 n)$.*
2. *for any constant $\rho \geq 3/2$, $n' \leq \rho k/\epsilon$ and $t \leq \frac{1}{\ln \rho} \log^2 n + O(\log n)$.*
3. *for any $\alpha > 0$, $n' \leq k n^{\alpha \log e}/\epsilon$ and $t \leq \lceil \alpha^{-1} \rceil \cdot (\log n + O(1))$.*

Proof. (1) is obtained by plugging $\rho = 1$ into Theorem 4, and noting that in the proof we assume $m \leq n$ or else we just output the original source. (2) is obtained by plugging the specified *constant* $\rho$ into Theorem 4. (3) is obtained by plugging $\rho = e^{\alpha \ell}$ into Theorem 4. □

## 3.3 Composing Condensers

The condenser of Corollary 3.2.1(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 3.3.1. *Given two condensers*

- *$C_1 : \{0,1\}^n \times \{0,1\}^{t_1} \to \{0,1\}^{m_1}$, and*
- *$C_2 : \{0,1\}^{m_1} \times \{0,1\}^{t_2} \to \{0,1\}^{m_2}$*

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

In Subsection 3.4 we prove the following two lemmas:

Lemma 3.3.1 (condenser composition). *Suppose $C_1 : \{0,1\}^n \times \{0,1\}^{t_1} \to \{0,1\}^{m_1}$ is an $(n, k) \to_{\epsilon_1} m_1$ condenser, and $C_2 : \{0,1\}^{m_1} \times \{0,1\}^{t_2} \to \{0,1\}^{m_2}$ is an $(m_1, k) \to_{\epsilon_2} m_2$ condenser. Then $C_1 \circ C_2$ is an $(n, k) \to_{\epsilon_1 + \epsilon_2 - \epsilon_1 \epsilon_2} m_2$ condenser.*

Lemma 3.3.2. *For every $n, k$, and $\epsilon \in (0, \frac{1}{2})$, and every constant $\delta > 0$, there exists an explicit $(n, k) \to_\epsilon n'$ condenser with $n' \leq (k/\epsilon)^{1+\delta}$, and $t \leq O(\log n)$.*

Proof of Theorem 3. Lemma 3.3.2, together with the equivalence of expanders and condensers from Lemma 2.2.1 proves Theorem 3(1); for (2) we use Corollary 3.2.1. ■

Finally, we can compose the condenser of Lemma 3.3.2 with the condenser of Corollary 3.2.1(1) to get:

Corollary 3.3.1. *For every $n, k$, and $\epsilon \in (0, \frac{1}{2})$ there is an explicit $(n, k) \to_{O(\epsilon)} n'$ condenser $C : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{n'}$ with $n' \leq k/\epsilon$ and $t \leq O(\log n + \log^3(k/\epsilon))$.*

## 3.4 Condenser proofs

In this subsection, we gather some of the more technical proofs regarding condensers and condenser composition.

Proof of Lemma 2.2.1. Below, we assume that $X$ is a flat distribution over $\{0,1\}^n$ with $H_\infty(X) = h \leq k$, that $D = U_t \circ C(X, U_t)$ is the distribution induced on $\{0,1\}^t \times \{0,1\}^{n'}$ by $C$, and that $\Gamma$ is the support of distribution $D$.

(1) $\to$ (2) Since $C$ is a condenser, $D$ is $\epsilon$ close to a distribution $D' = D_1 \circ D_2$ over $\{0,1\}^t \times \{0,1\}^{n'}$, with $D_1 = U_t$ and $H_\infty(D_2 | D_1 = y) \geq h$ for all $y$. These last two conditions on $D_1$ and $D_2$ imply that for all $z$, $D'(z) \leq 2^{-t-h}$. We have:

$$\epsilon \geq |D - D'| \geq \sum_{z \in \Gamma} D(z) - D'(z) = 1 - \sum_{z \in \Gamma} D'(z) \geq 1 - \frac{|\Gamma|}{2^{t+h}}.$$

We conclude that $|\Gamma| \geq (1-\epsilon)2^{t+h}$, which implies that $G_C$ is $(2^k, (1-\epsilon)2^t)$-expanding.

(2) $\rightarrow$ (1) For every $y \in \{0,1\}^t$ pick a set $A_y$ such that $C(X,y) \subseteq A_y \subseteq \{0,1\}^m$ and $|A_y| = 2^h$. Define the distribution $D' = D_1 \circ D_2$ over $\{0,1\}^t \times \{0,1\}^{n'}$ by letting $D_1$ be $U_t$ and letting $(D_2 \mid D_1 = y)$ be the flat distribution over $A_y$. Notice that for all $z$ in its support $\Gamma$, $D(z) \geq 2^{-t-h}$, and for all $z$ in its support $\Gamma'$, $D'(z) = 2^{-t-h}$. Furthermore, we know that $|\Gamma| \geq (1-\epsilon)2^{t+h}$ and $|\Gamma'| = 2^{t+h}$. We have:

$$
\begin{aligned}
|D - D'| &= \frac{1}{2}\sum_z |D(z) - D'(z)| \\
&= \frac{1}{2}\left(\sum_{z\in\Gamma} D(z) - D'(z) + \sum_{z\notin\Gamma} D'(z)\right) \\
&= \frac{1}{2}\left(1 - \sum_{z\in\Gamma} D'(z) + \sum_{z\notin\Gamma} D'(z)\right) \\
&\leq \frac{1}{2}(1 - (1-\epsilon) + \epsilon) = \epsilon.
\end{aligned}
$$

(2) $\rightarrow$ (3) We define the inverter function $I_X : \{0,1\}^t \times \{0,1\}^{n'}$ on input $(y;w)$ to be some $x$ for which $C(x,y) = w$ if such an $x$ exists, and $0^n$ otherwise. This inverter succeeds on exactly those pairs $(y,w)$ for which $w \in C(X,y)$. Since $G_C$ is $(2^k, (1-\epsilon)2^t)$ expanding, there are at least $(1-\epsilon)2^{t+h}$ such pairs. The probability that we hit such a pair by choosing $y$ uniformly, choosing $x \in X$ (recall that $X$ is flat), and setting $w = C(x,y)$ is at least $(1-\epsilon)$. Therefore $\Pr_{x\in X,y}[I_X(y;C(x,y)) = x] \geq 1-\epsilon$.

(3) $\rightarrow$ (2) For every $y \in \{0,1\}^t$, $I_X(y;\cdot)$ is a deterministic function, and hence can be correct on at most $|C(X,y)|$ elements. Since $\Pr_{x\in X,y}[I_X(y;C(x,y)) = x] \geq 1-\epsilon$ it follows that $|\cup_y C(X,y)|/2^{t+h} \geq (1-\epsilon)$. This implies that $G_C$ is $(2^k, (1-\epsilon)2^t)$ expanding.

■

Proof of Lemma 2.2.2. The non-trivial direction is that Definition 2.2.1 implies definition 2.2.3. Indeed, let $X$ be a distribution with $H_\infty(X) \leq k$. We can represent $X$ as a convex combination of *flat* distributions with entropy exactly $H_\infty(X)$, i.e., $X = \sum_i p_i X^i$ with $p_i \in [0,1]$, $\sum p_i = 1$ and $H_\infty(X^i) = H_\infty(X)$. As $C$ is a condenser, $U_t \circ C(X^i, U_t)$ is $\epsilon$ close to a distribution $D^i = D_1^i \circ D_2^i$ with $D_1^i = U_t$ and $H_\infty(D^i) \geq t + H_\infty(X)$. We take $D = \sum_i p_i D^i$. We see that

$$
\begin{aligned}
|U_t \circ C(X,U_t) - D| &= |\sum p_i\, U_t \circ C(X^i,U_t) - \sum p_i\, D^i| \\
&\leq \sum p_i |U_t \circ C(X^i,U_t) - D^i| \leq \epsilon
\end{aligned}
$$

Also, $D_1 = \sum_i p_i D_1^i = U_t$. Finally, min-entropy is concave and hence $H_\infty(D) \geq \sum_i p_i H_\infty(D^i) \geq t + H_\infty(X)$, which implies that for any $y \in \{0,1\}^t$, $H_\infty(D_2|D_1 = y) \geq H_\infty(X)$.
■

Proof of Lemma 3.3.1. We use Lemma 2.2.1, and prove the equivalent statement for expanders,which is easier. Let $G_{C_1\circ C_2} = (V_1 = [2^n], V_2 = [2^{t_1+t_2+m_2}], E)$ be the bipartite graph defined by $C_1 \circ C_2$. We want to show that $G_{C_1\circ C_2}$ is $(2^k, (1-\epsilon_1)(1-\epsilon_2)2^{t_1+t_2})$ expanding. Let $A \subseteq V_1$ be a subset of size at most $2^k$. Then

$$\Gamma_{G_{C_1\circ C_2}}(A) \;=\; \{(y_1,y_2,z) \;:\; z = C_2(C_1(x;y_1);y_2), x \in A\}$$

For each $y_1 \in \{0,1\}^{t_1}$ let us define

$$\Gamma^{y_1}_{G_{C_1}}(A) \;=\; \{C_1(w,y_1) \;:\; w \in A\}$$

Then,

$$
\begin{aligned}
|\Gamma_{G_{C_1\circ C_2}}(A)| &= \sum_{y_1\in\{0,1\}^{t_1}} |\Gamma_{G_{C_2}}(\Gamma^{y_1}_{G_{C_1}}(A))| &(5)\\
&\geq \sum_{y_1\in\{0,1\}^{t_1}} (1-\epsilon_2)2^{t_2}|\Gamma^{y_1}_{G_{C_1}}(A)| &(6)\\
&= (1-\epsilon_2)2^{t_2}\sum_{y_1\in\{0,1\}^{t_1}} |\Gamma^{y_1}_{G_{C_1}}(A)| &(7)\\
&= (1-\epsilon_2)2^{t_2}|\Gamma_{G_{C_1}}(A)| &(8)\\
&\geq (1-\epsilon_2)2^{t_2}(1-\epsilon_1)2^{t_1}|A| &(9)
\end{aligned}
$$

where (7) is true because for all $y_1$, $|\Gamma^{y_1}_{G_{C_1}}(A)| \leq |A| \leq 2^k$, and $G_{C_2}$ is $(2^k, (1-\epsilon_2)2^{t_2})$ expanding, and (9) is true because $G_{C_1}$ is $(2^k, (1-\epsilon_1)2^{t_1})$ expanding. ■

If $C = \left\{C_n : \{0,1\}^n \times \{0,1\}^{t(n)} \rightarrow \{0,1\}^{m(n)}\right\}$ is a family of $(n,k) \rightarrow_\epsilon m(n)$ condensers, we can compose them repeatedly. Given $n_1, k$ and $\epsilon > 0$, define $C^{(1)} = C_{n_1}$ and, for $i > 1$, define $C^{(i)} = C^{(i-1)} \circ C_{n'}$, where $n'$ is the output length of $C^{(i-1)}$. We now prove a lemma about iterated composition:

Lemma 3.4.1 (iterated composition). *Fix $n_1, k$, and $\epsilon > 0$, and let*

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

*be a family of $(n,k) \rightarrow_\epsilon m(n)$ 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 an $(n_1,k) \rightarrow_{i\epsilon} m$ 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 3.3.1. The error accumulates additively. 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

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

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

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

□

Proof of Lemma 3.3.2. First, we may assume that $k \geq \log n$; otherwise the condenser mentioned in [18] will suffice. Now we plug Corollary 3.2.1(3) into Lemma 3.4.1 with

- $\epsilon' = O(\frac{\epsilon}{\log\log n})$,
- $\delta' = \delta/2$,
- $a = \alpha \log e = \frac{\delta'}{2+\delta'}$,
- $\Delta = O(k/\epsilon')$,
- $b = \lceil \alpha^{-1} \rceil = \left\lceil \frac{\log e(2+\delta')}{\delta'} \right\rceil = O(1)$, and
- $i = \log_{\frac{1}{a}} \frac{2\log n}{\delta' \log \Delta}$.

This gives us:

$$n' \leq \Delta^{\frac{1}{1-a}} \cdot {n_1}^{a^i} = \Delta^{1+\delta'/2} \Delta^{\delta'/2} = \Delta^{1+\delta'} \leq (k/\epsilon)^{1+\delta}$$

and

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

We notice that $\frac{b}{1-a}$ is a constant, and that $i = O(\log \frac{\log n}{\log \Delta}) \leq O(\frac{\log n}{\log \Delta})$. Therefore $i \log \Delta = O(\log n)$, and $t \leq O(\log n)$. ■

## 4. EXTRACTORS FOR LOW MIN-ENTROPY

In this section we prove Theorem 1 and obtain the extractors listed in Table 1, and summarized in the following theorem:

Theorem 5. *For every $n, k$, and constant $\epsilon$, there exist $(k, \epsilon)$ extractors $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ with*

1. *$t = O(\log n + \log k (\log\log k)^2)$ and $m = (1-\alpha)k$, for any constant $\alpha > 0$, and*
2. *$t = O(\log n + \log^2 k (\log\log k)^2)$ and $m = k + t - O(1)$.*

We first need a lemma regarding the composition of extractors with condensers.

Lemma 4.0.2. *Let $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{n'}$ be a $(n, k) \to_{\epsilon'} n'$ condenser, and let $E : \{0,1\}^{n'} \times \{0,1\}^t \to \{0,1\}^m$ be a (strong) $(k, \epsilon)$ extractor. Then*

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

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

Proof. We consider the case in which $E$ is a strong extractor; for the non-strong case, we simply discard the seeds. As any distribution $X$ with $H_\infty(X) \geq k$ can be written as a convex combination of distributions with min-entropy exactly $k$, it is enough to show that $U_d \circ U_t \circ E'_n(X; U_d, U_t)$ is $(\epsilon + \epsilon')$-close to $U_{d+t+m}$ for distributions $X$ over $\{0,1\}^n$ with $H_\infty(X) = k$.

Let $X$ be a distribution with $H_\infty(X) = k$. We know that $U_d \circ C(X, U_d)$ is $\epsilon'$-close to $D = D_1 \circ D_2$, where $D_1 = U_d$ and for every $y \in \{0,1\}^d$, $H_\infty(D_2 | D_1 = y) \geq k$. We now use the fact that if two distributions $A$ and $A'$ are $\epsilon$-close, then for any function $f$, $f(A)$ and $f(A')$ are also $\epsilon$-close. Consider the function $f : \{0,1\}^{n'+d+t} \to \{0,1\}^{d+t+m}$ defined by $f(x, y, z) = y \circ z \circ E(x, z)$. We see that $U_d \circ U_t \circ E(C(X; U_d); U_t) = f(C(X; U_d), U_d, U_t)$ is $\epsilon'$ close to $f(D_2, D_1, U_t) = D_1 \circ U_t \circ E(D_2; U_t)$. However, for every $y \in \{0,1\}^d$, $H_\infty(D_2|D_1 = y) \geq k$ and $E$ is an extractor, and therefore $U_d \circ U_t \circ E'(X; U_d, U_t)$ is $(\epsilon' + \epsilon)$-close to $U_{d+t+m}$, as required. □

Proof of Theorem 1. Fix $n$ and $k = k(n) \leq \sqrt{n}$. We use Lemma 3.3.2 with $\epsilon = k^{-1/2}$ and $\delta = 1/3$ to obtain an $(n, k) \to_{k^{-1/2}} k^2$ condenser $C$ that uses $d = O(\log n)$ truly random bits. Let $E_{k^2} : \{0,1\}^{k^2} \times \{0,1\}^{t(k^2)} \to \{0,1\}^{m(k)}$ be the $(k, \epsilon(k^2))$-extractor in the statement of the theorem. Applying Lemma 4.0.2 to condenser $C$ and extractor $E_{k^2}$, we obtain the required $(k, k^{-1/2} + \epsilon(k^2))$-extractor $E'_n : \{0,1\}^n \times \{0,1\}^{O(\log n)+t(k^2)} \to \{0,1\}^{m(k)}$.

■

We use extractors from [20] and [19] to obtain the results in Theorem 5.

Theorem 6 ([20]). *For every $n, k$ and constant $\epsilon > 0$, there exist explicit $(k, \epsilon)$ extractors $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ with*

1. *$t = O(\log n (\log\log n)^2)$ and $m = (1-\alpha)k$, for every constant $\alpha > 0$, and*
2. *$t = O(\log n \log k (\log\log n)^2)$ and $m = k$.*

Plugging the first extractor of Theorem 6 into Theorem 1 we obtain the extractor in Theorem 5(1). For the second extractor, we need the following results from [19].

Theorem 7 ([19]). *For every $n$, $k < n$, and $\epsilon > 0$, there exists an explicit $(k, \epsilon)$-extractor $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^{k+t-\Delta}$, where $t = O(\log^2 n \log k \log \epsilon^{-1})$ and $\Delta = 2\log \epsilon^{-1} + O(1)$.*

The following is a slight strengthening of a lemma of [31]

Lemma 4.0.3. *[19] Suppose $E_1 : \{0,1\}^n \times \{0,1\}^{t_1} \to \{0,1\}^{m_1}$ is a $(k, \epsilon_1)$-extractor with entropy loss $\Delta_1$ and $E_2 : \{0,1\}^{n+t_1} \times \{0,1\}^{t_2} \to \{0,1\}^{m_2}$ 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 $\Delta_2 + 1$.*

By plugging the extractor of [19] for constant $\epsilon$ into Theorem 1, we obtain an extractor using $t = O(\log n + \log^3 k)$ random bits with constant entropy loss. Using this extractor with $k = t_1 - 1$ as $E_2$ in Lemma 4.0.3, and the second extractor of Theorem 6 as $E_1$, we obtain an extractor using $t = O(\log n \log k (\log\log n)^2)$ random bits with constant entropy loss. Substituting this extractor into Theorem 1 gives the extractor in Theorem 5(2).

## 5. DISPERSERS AND SOMEWHERE-RANDOM EXTRACTORS

We use a technique from [13] to prove Theorem 2. They showed how to compose two extractors into a *somewhere random extractor*, which we define soon. They then show

how to get a disperser construction from a somewhere random extractor. We obtain Theorem 2 by plugging our improved extractors into this construction.

We start with definitions of somewhere random sources and somewhere random extractors, from [13]. Given a random source with $k$ min-entropy, a $(k,\epsilon)$ extractor outputs a single distribution that is $\epsilon$ close to uniform. In contrast, intuitively, a somewhere random extractor may output many distributions with the guarantee that at least one of them (and possibly only one) is $\epsilon$ close to uniform [3]. Thus, a somewhere random extractor is a weakening of the extractor notion.

DEFINITION 5.0.1 (SOMEWHERE RANDOM SOURCE). *$B = (B_1, \ldots, B_b)$ is a $b$–block $(m,\epsilon)$ somewhere random source if each $B_i$ is a random variable over $\{0,1\}^m$ and there is a random variable $Y$ over $[0..b]$ such that:*

- *For every $i \in [1..b]$: $\Pr(Y=i) > 0 \implies |(B_i|Y=i) - U_m| \le \epsilon$.*
- *$\Pr(Y=0) \le \epsilon$.*

DEFINITION 5.0.2 (SOMEWHERE RANDOM EXTRACTOR). *Let $S : \{0,1\}^n \times \{0,1\}^t \to (\{0,1\}^m)^b$ be a function. Given a distribution $X$ on $\{0,1\}^n$ the distribution $S(X, U_t) = B_1 \circ \ldots \circ B_b$ is obtained by picking $x \in X, y \in U_t$ and computing $S(x,y)$. We say $S$ is a $(k,\epsilon)$* somewhere random extractor *if for any distribution $X$ with $H_\infty(X) \ge k$, $(B_1, \ldots, B_b)$ is a $b$-block $(m,\epsilon)$ somewhere random source.*

Nisan and Ta-Shma proved:

THEOREM 8 ([13]). *Suppose $E_1 : \{0,1\}^n \times \{0,1\}^{t_1} \to \{0,1\}^{t_2}$ is a $(k_1,\epsilon_1)$ extractor and $E_2 : \{0,1\}^n \times \{0,1\}^{t_2} \to \{0,1\}^m$ is a $(k_2,\epsilon_2)$ extractor. Then for any $s > 0$ there is an explicit function $E : \{0,1\}^n \times \{0,1\}^{t_1} \to (\{0,1\}^m)^n$ that is a $(k_1+k_2+s, \epsilon_1+\epsilon_2+8n2^{-s/3})$ somewhere random extractor.*

Using our extractors we show

THEOREM 9. *For every $n, k$ and constant $\epsilon$ there is a $(k, 4\epsilon)$ somewhere random extractor $S : \{0,1\}^n \times \{0,1\}^t \to (\{0,1\}^m)^n$ with $t = O(\log n)$ and $m = k + t - (3\log n + O(1))$.*

PROOF. We will take both $E_1$ and $E_2$ in Theorem 8 to be the second extractor listed in Table 1 but with different parameters. Let $t(n,k) = O(\log n + \log^2 k (\log\log k)^2)$ denote the number of random bits used by this extractor for error $\epsilon/4$. Set

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

Here $\Delta = O(1)$ is the entropy loss. Let $E_1 : \{0,1\}^n \times \{0,1\}^{t_1} \to \{0,1\}^{t_2}$ be a $(k_1, \epsilon/4)$-extractor, and let $E_2 : \{0,1\}^n \times \{0,1\}^{t_2} \to \{0,1\}^{k_2+t_2-\Delta}$ be a $(k_2,\epsilon/4)$-extractor.

[3]The formal definition is slightly different.

| Degree | Reference |
|---|---|
| $O(\frac{N}{a} \cdot 2^{\mathrm{poly}(\log\log N)})$ | [21] |
| $O(\frac{N}{a} \cdot \mathrm{polylog} N)$ | This paper |
| $\frac{N}{a} - 1$ | Lower bound |

**Table 4: Explicit $a$ expanding graphs with $N$ vertices**

By Theorem 8, we obtain a $(k,\epsilon)$ somewhere random extractor:

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

as required. □

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

LEMMA 5.0.4. *[13] Let $\epsilon < 1$. Let $S : \{0,1\}^n \times \{0,1\}^t \to (\{0,1\}^m)^b$ be a $(k,\epsilon)$ somewhere random extractor. Suppose $S(x,y) = S_1(x,y) \circ \ldots \circ S_b(x,y)$. Define the bipartite graph $G = (V_1 = \{0,1\}^n, V_2 = \{0,1\}^m, E)$ where $(v_1,v_2) \in E$ iff there is some $1 \le i \le b$ and some $z \in \{0,1\}^t$ such that $v_2 = S_i(v_1,z)$. Then $G$ is a $(K = 2^k, \epsilon)$ disperser.*

Plugging the somewhere random extractor of Theorem 9 into Lemma 5.0.4 we get Theorem 2.

# 6. APPLICATIONS

We now describe applications in which our constructions lead to concrete improvements.

## 6.1 $a$–expanding graphs

Wigderson and Zuckerman [31] showed that an explicit construction of a disperser with small entropy loss can be used for explicitly constructing $a$–expanding graphs and depth two super-concentrators.

DEFINITION 6.1.1 ([16]). *An undirected graph is $a$–expanding if every two disjoint sets of vertices of size at least $a$ are joined by an edge.*

Plugging our result into [31] yields almost optimal explicit $a$–expanding graphs. We summarize this in Table 4.

Explicit $a$–expanding graphs have been used to construct depth two super-concentrators, non-blocking networks, and algorithms for sorting and selecting in rounds. See Section 1.4 for more on the application to super-concentrators; for the other applications see [31].

## 6.2 Hardness of approximation.

Using explicit dispersers, Umans [29] showed that the following $\Sigma_2^p$ optimization problem and some related problems are $\Sigma_2^p$-hard to approximate to within an $n^{1/5-\epsilon}$ factor, for any constant $\epsilon > 0$.

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

The proof requires $(K = 2^k, \epsilon)$ dispersers with $k = O(\log n)$, constant $\epsilon$, $m = \Omega(k)$, and $d = O(\log n)$. However, in order to achieve an inapproximability factor of $n^{1-\epsilon}$ (which is optimal up to lower order terms), it is necessary for $d$ to have a *sublinear* dependence on $k$. Our extractors (see Table 1) are the first to simultaneously extract a constant fraction of the min-entropy and have $d = O(\log n)$ with a sublinear dependence on $k$. Plugging the construction in this paper into [29], we achieve inapproximability of $n^{1-\epsilon}$ for the problem above and several other optimization problems from [29], which were previously only achievable via quasi-polynomial reductions.

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. 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 helpful comments.

## 7. 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 *19th STOC*, 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 nonblocking 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 *32nd STOC*, pages 449–458, 2000.

[6] P. Feldman, J. Friedman, and N. Pippenger. Wide-sense nonblocking networks. *SIAM Journal on Discrete Mathematics*, 1:158–173, 1988.

[7] O. Gabber and Z. Galil. Explicit construction of linear sized superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[8] O. Goldreich, R. Impagliazzo, L. Levin, R. Venkatesan, and D. Zuckerman. Security preserving amplification of hardness. In *31st FOCS*, pages 318–326, 1990.

[9] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Near-optimal conversion of hardness into pseudo-randomness. In *40th FOCS*, pages 181–190, 1999.

[10] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Extractors and pseudo-random generators with optimal seed length. In *32nd STOC*, pages 1–10, 2000.

[11] N. Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42:1091–1106, 1995.

[12] G.A. Margulis. Explicit construction of concentrators. *Problems of Inform. Transmission*, 9:325–332, 1973.

[13] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58:148–173, 1999.

[14] N. Nisan and A. Wigderson. Hardness vs. randomness. *Journal of Computer and System Sciences*, 49:149–167, 1994.

[15] N. Nisan and D. Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, 1996.

[16] N. Pippenger. Sorting and selecting in rounds. *SIAM Journal on Computing*, 16(6):1032–1038, December 1987.

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

[18] R. Raz and O. Reingold. On recycling the randomness of states in space bounded computation. In *31st STOC*, pages 159–168, 1999.

[19] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *31st STOC*, pages 149–158, 1999.

[20] O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. In *41st FOCS*, pages 22–31, 2000.

[21] M. Saks, A. Srinivasan, and S. Zhou. Explicit OR-dispersers with polylog degree. *Journal of the ACM*, 45:123–154, 1998.

[22] M. Santha. On using deterministic functions in probabilistic algorithms. *Information and Computation*, 74(3):241–249, 1987.

[23] M. Sipser. Expanders, randomness, or time vs. space. *Journal of Computer and System Sciences*, 36:379–383, 1988.

[24] M. Sipser and D. A. Spielman. Expander codes. *IEEE Transactions on Information Theory*, 42(6):1710–1722, 1996.

[25] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SIAM Journal on Computing*, 28:1433–1459, 1999.

[26] A. Ta-Shma. On extracting randomness from weak random sources. In *28th STOC*, pages 276–285, 1996.

[27] A. Ta-Shma. Almost optimal dispersers. In *30th STOC*, pages 196–202, 1998.

[28] L. Trevisan. Construction of extractors using pseudo-random generators. In *31st STOC*, pages 141–148, 1999.

[29] C. Umans. Hardness of approximating $\Sigma_2^p$ minimization problems. In *40th FOCS*, pages 465–474, 1999.

[30] L.G. Valiant. Graph theoretic properties in computational complexity. *Journal of Computer and System Sciences*, 13:278–285, 1976.

[31] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.

[32] D. Zuckerman. General weak random sources. In *31st FOCS*, pages 534–543, 1990.

[33] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16:367–391, 1996.

[34] D. Zuckerman. Randomness-optimal oblivious sampling. *Random Structures and Algorithms*, 11:345–367, 1997.