Error Reduction for Extractors

Ran Raz* Omer Reingold† Salil Vadhan‡

Abstract

We present a general method to reduce the error of any extractor. Our method works particularly well in the case that the original extractor extracts up to a constant fraction of the source min-entropy and achieves a polynomially small error. In that case, we are able to reduce the error to (almost) any ε, using only $O(\log(1/\varepsilon))$ additional truly random bits (while keeping the other parameters of the original extractor more or less the same). In other cases (e.g. when the original extractor extracts all the min-entropy or achieves only a constant error) our method is not optimal but it is still quite efficient and leads to improved constructions of extractors.

Using our method, we are able to improve almost all known extractors in the case where the error required is relatively small (e.g. less than polynomially small error). In particular, we apply our method to the new extractors of [Tre99, RRV99] to get improved constructions in almost all cases. Specifically, we obtain extractors that work for sources of any min-entropy on strings of length n which: (a) extract any $1/n^\gamma$ fraction of the min-entropy using $O(\log n + \log(1/\varepsilon))$ truly random bits (for any $\gamma > 0$), (b) extract any constant fraction of the min-entropy using $O(\log^2 n + \log(1/\varepsilon))$ truly random bits, and (c) extract all the min-entropy using $O(\log^3 n + \log n \cdot \log(1/\varepsilon))$ truly random bits.

*Department of Applied Mathematics and Computer Science, Weizmann Institute, Rehovot, 76100 Israel. E-mail: ranraz@wisdom.weizmann.ac.il Work supported by an American-Israeli BSF grant 95-00238.

†Department of Applied Mathematics and Computer Science, Weizmann Institute, Rehovot, 76100 Israel. E-mail: reingold@wisdom.weizmann.ac.il Research supported by an Eshkol Fellowship of the Israeli Ministry of Science and by ESPRIT working group RAND2.

‡MIT Laboratory for Computer Science. 545 Technology Square. Cambridge, MA 02139. E-mail: salil@theory.lcs.mit.edu. URL: http://theory.lcs.mit.edu/~salil. Supported by a DOD/NDSEG fellowship and partially by DARPA grant DABT63-96-C-0018.

1 Introduction

Roughly speaking, an extractor is a function which extracts (almost) truly random bits from a weak random source, using a small number of additional random bits as a catalyst. More formally, a random variable (or a distribution) X on $\{0,1\}^n$ is said to have *min-entropy* k if for all $x \in \{0,1\}^n$, $\Pr[X = x] \le 2^{-k}$; k is a measure of how many "bits of randomness" the source contains. A function

$$\mathrm{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

is called a (k,ε)-*extractor* if for every distribution X on $\{0,1\}^n$ of min-entropy k, the induced distribution $\mathrm{EXT}(X, U_d)$ on $\{0,1\}^m$ has statistical difference at most ε from uniform (where U_d is the uniform distribution on $\{0,1\}^d$).

In other words, EXT "extracts" m (almost) truly random bits from a source with k bits of hidden randomness, using d additional random bits as a catalyst. The random variable X is usually referred to as the *source*. The d additional random bits are sometimes called the *seed* of the extractor. The statistical difference ε, between $\mathrm{EXT}(X, U_d)$ and the uniform distribution, is also called the *error* of the extractor.

Extractors were first defined in [NZ96]. A large body of work has focused on giving explicit constructions of extractors, as such constructions have a wide variety of applications. The goal is to explicitly construct extractors which minimize d, while m is as close to k as possible. Non-explicit constructions of extractors are able to extract all of the source min-entropy (i.e. $m = k$), using only $d = O(\log n + \log(1/\varepsilon))$ additional random bits. It can be proved that this number of additional random bits is optimal [NZ96, RT97].

Current explicit constructions, however, fail to achieve this optimal bound, though there has been steady progress towards this goal. Hence, constructing explicit extractors that achieve the optimal bound $d = O(\log n + \log(1/\varepsilon))$ for all settings parameters is still a major open problem. For more details about some previous work on extractors and their applications see the survey in [NT99].

Early works on extractors concentrated mainly on the case of relatively large error ε. From a theoretical point of view, however, the case of small error seems to be as interesting. In applications, the low-error case is particularly interesting when one wants to apply a sequential process, where an extractor is applied a large number of times. In such cases, if the error is not small enough it may accumulate and destroy the entire process. One example for such a situation is the recent paper [RR99], where extractors with exponentially small error are used (and indeed our work implies an improvement of the results in [RR99]).

In this paper, we concentrate on the dependency of the seed length d on the error ε. Our main goal is to construct efficient extractors for relatively small ε. Ideally, ε should add to d only an additive term of $O(\log(1/\varepsilon))$. Such a dependency was previously obtained only in certain cases, when there are restrictions on the relationship between the min-entropy k and the length n of the string coming from the source. Specifically, Zuckerman [Zuc97] has constructed extractors which use $O(\log n + \log(1/\varepsilon))$ truly random bits when k is at least a constant fraction of n. Extractors using $O(k + \log n + \log(1/\varepsilon))$ truly random bits were constructed by Srinivasan and Zuckerman [SZ98] and Goldreich and Wigderson [GW94], but this bound is good only when the min-entropy k is relatively small. Finally, extractors using $O((n-k) + \log(1/\varepsilon))$ truly random bits were constructed by Goldreich and Wigderson[GW94], but this bound is good only when the min-entropy k is very close to n.

In contrast to these previous results, the extractors constructed in this paper perform well for sources of *any* min-entropy, while maintaining an optimal dependence on the error ε. For sources of any min-entropy, Ta-Shma [NT99] has previously constructed extractors using $O(\mathrm{polylog}\, n \cdot \log(1/\varepsilon))$ truly random bits (with the degree of the polylog later improved in [RRV99]), but here we aim to obtain a *constant* multiple of $\log(1/\varepsilon)$.

1.1 Main Results

Our main result is an efficient method to reduce the error of an extractor from ε to any $\varepsilon' < \varepsilon$ without damaging its other parameters by much. The exact statement of the result is given in Section 2. Roughly speaking, given an arbitrary extractor $\textsc{Ext}$ that extracts m bits with error ε, we construct a new extractor $\textsc{Ext}'$ that extracts $(1-\alpha)\cdot m$ bits with error ε', (where $\alpha > 0$ is any constant). The number of truly random bits (i.e. the length of the seed) for the new extractor $\textsc{Ext}'$ is the same as the one for $\textsc{Ext}$ plus

- $O(\log(1/\varepsilon'))$ bits, if the original error ε is polynomially small (e.g. $\varepsilon = 1/m$), or
- $O(\log(1/\varepsilon') + \log m \cdot \mathrm{polyloglog}\, m)$ bits, if the original error ε is constant.

The method is general, and can be applied to any extractor. We can apply it to all previous constructions of extractors and get improved results, except in the few cases where optimal results were already achieved [Zuc97, GW94, SZ98]. In particular, applying our new method to the extractors of [Tre99, RRV99] directly gives the following results (in all the following $\alpha > 0$ is an arbitrary small constant):

1. **Extracting $m = k^{1-\alpha}$ random bits:**
   In this case we achieve
   $$d = O\left(\frac{\log^2 n}{\log k} + \log(1/\varepsilon)\right).$$
   This is obtained by using the result of [Tre99] with polynomially small error and further reducing the error to ε using our new method. The best previous results were $d = O(\log^2(n/\varepsilon)/\log k)$ and $d = O(\log^2 n \cdot \log(1/\varepsilon)/\log k)$, proved in [Tre99, RRV99].

2. **Extracting $m = (1-\alpha)\cdot k$ random bits:**
   In this case we achieve
   $$d = O(\log^2 n + \log(1/\varepsilon)).$$
   This is obtained by using the equivalent result in [RRV99] with polynomially small error and further reducing the error to ε using our new method. The best previous results were $d = O(\log^2(n/\varepsilon))$ and $d = O(\log^2 n \cdot \log(1/\varepsilon))$, proved in [RRV99].

3. **Extracting all k random bits (i.e. $m = k$):**
   In this case we achieve
   $$d = O((\log^2 n + \log(1/\varepsilon)) \cdot \log k).$$
   This is obtained by iterative application of the previous result $O(\log k)$ times (as in [WZ95]). The best previous results were $d = O(\log^2(n/\varepsilon)\cdot \log k)$ and $d = O(\log^2 n \cdot \log(1/\varepsilon) \cdot \log k)$, proved in [RRV99].

Strong Extractors. The original definition of extractors [NZ96] is somewhat stronger than the definition given above (which is due to [NT99]). Such a *strong* extractor (as named by Zuckerman [Zuc97]) $\textsc{Ext}$ has the property that for every source X with sufficient min-entropy, almost every seed r is "good" (i.e. $\textsc{Ext}(X, r)$ is close to uniform). Formally, a function

$$\textsc{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$$

is called a *strong* (k,ε)*-extractor* if for every distribution X on $\{0,1\}^n$ of min-entropy k, the induced distribution $\langle U_d, \textsc{Ext}(X, U_d)\rangle$ on $\{0,1\}^{d+m}$ has statistical difference at most ε from uniform (where the two occurrences of U_d represent the same variable).

Though for most applications of extractors "standard" extractors are sufficing, constructing strong extractors is still of interest (see, e.g., [Zuc97]). In fact, many of the constructions of extractors actually give *strong* extractors. In Section 6, we show that our method of reducing the error in extractors also applies to strong extractors: If the original extractor $\textsc{Ext}$ is a strong extractor, then the new extractor $\textsc{Ext}'$ is also strong. Since the constructions in [Tre99, RRV99] can be shown to give strong extractors, it follows that our concrete constructions of extractors (obtained by applying our new method to the extractors of [Tre99, RRV99]) also give strong extractors.

1.2 Techniques and Other Results

Our main lemma shows how to reduce the error from ε to $O(\varepsilon^2)$. The exact statement of the lemma is given in Section 2. Roughly speaking, given an extractor $\textsc{Ext}$ that extracts m bits with error ε, we construct a new extractor $\textsc{Ext}'$ that extracts $(1-\alpha)\cdot m$ bits with error $O(\varepsilon^2)$. The number of truly random bits for $\textsc{Ext}'$ is the same as the one for $\textsc{Ext}$ plus $O(\log(m/\varepsilon))$ additional random bits (more precisely, the number of additional random bits is $\mathrm{poly}(1/\alpha)\cdot\log(m/\varepsilon)$). Our main result (i.e. reducing ε to ε') is then obtained by iterative application of the main lemma (with different parameters α) $O(\log\log(1/\varepsilon')-\log\log(1/\varepsilon))$ times.

The most interesting part of this paper is probably the proof of the main lemma, as it uses several techniques that (as far as we know) were not used before. In short, the construction given in the proof is the following: The original (k,ε)-extractor $\textsc{Ext}\colon\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^m$ is applied *twice* to the string x coming from the source. The two applications of $\textsc{Ext}$ on x are done with two different (but not independent as random variables) seeds $r,r'\in\{0,1\}^d$. We denote the outputs by $y\in\{0,1\}^m$ and $y'\in\{0,1\}^m$, respectively, and we prove that the distribution of $(y,y')\in\{0,1\}^{2m}$ is of statistical difference $O(\varepsilon^2)$ from some distribution with min-entropy $\approx m$. We then apply to (y,y') the extractor constructed by Zuckerman in [Zuc97], with error ε^2.

Thus, the proof uses composition of extractors. Composition of extractors was used before (e.g. to extract more randomness and to deal with smaller min-entropy), but not as a technique to reduce the error. Even more interesting is the way we generate the two seeds r and r'. The first seed r is truly random. The second seed r', however, is not independent of r. It is generated by applying to r another extractor, constructed by Goldreich and Wigderson [GW94]. In other words, our construction uses two levels of extractors. $\textsc{Ext}$ is applied on the source, but the seed for $\textsc{Ext}$ is also recycled (using a different extractor).

In order to prove that this construction works, we prove a technical lemma (Lemma 17) that analyzes the source of the error in an extractor. Roughly speaking, the lemma shows that the main source of error in extractors is a small set of bad seeds for each value of x. This analysis may be interesting in its own right.

As mentioned above, our construction uses two previous constructions, the one of [Zuc97] and the one of [GW94]. The common feature of both of these constructions is that they both achieve optimal dependency on the error ε. More intuition for the proof of the main lemma are given in Section 3.

2 Formal Statement of Results

In this section, we give the exact statements of our results about reducing the error. We first give simplified statements of our results with parameters restricted to what we feel to be the most interesting ranges, and later we state the results in their full generality.

Reducing the Error. Our first theorem reduces the error of an extractor from $1/m$ (where m is the output length of the extractor) to an (almost) arbitrary $\varepsilon>0$ using $O(\log(1/\varepsilon))$ additional truly random bits.

Theorem 1 *Let $\alpha>0$ be an arbitrary constant. Suppose that there is an explicit*[1] *$(k,1/m)$-extractor $\textsc{Ext}\colon\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^m$. Then, for every $\varepsilon'>\exp(-n/(\log^* n)^{O(\log^* n)})$ there is an explicit (k',ε')-extractor $\textsc{Ext}'\colon\{0,1\}^n\times\{0,1\}^{d'}\to\{0,1\}^{m'}$, with*

$$\begin{aligned} k' &= k+O(\log(1/\varepsilon')) \\ m' &= (1-\alpha)\cdot m \\ d' &= d+O(\log(1/\varepsilon')) \end{aligned}$$

Our second theorem deals with the case that the initial error is a constant instead of an inverse polynomial. It reduces the error to $1/m$ using an almost-logarithmic number of truly random bits, so that Theorem 1 can then be applied.

Theorem 2 *Let $1>\varepsilon>0$ and $\alpha>0$ be arbitrary constants. Suppose that there is an explicit (k,ε)-extractor $\textsc{Ext}\colon\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^m$. Then, there is an explicit $(k',1/m)$-extractor $\textsc{Ext}'\colon\{0,1\}^n\times\{0,1\}^{d'}\to\{0,1\}^{m'}$, with*

$$\begin{aligned} k' &= k+O(\log m) \\ m' &= (1-\alpha)\cdot m \\ d' &= d+O(\log m\cdot\mathrm{polyloglog}\, m) \end{aligned}$$

[1] Of course, it is meaningless to speak of an individual extractor being "explicit," but we state our theorems in terms of individual extractors for readability. The theorems actually refer to a *family* of (k,ε)-extractors $\textsc{Ext}\colon\{0,1\}^n\times\{0,1\}^d\to\{0,1\}^m$, indexed by a the parameters n, k, m, d, and ε (with restrictions on their relative values), where such a family is called *explicit* if $\textsc{Ext}(x,y)$ can be evaluated in time $\mathrm{poly}(n,d)$ given x, y, n, k, m, d, and ε as input.

As mentioned earlier, both of our theorems are based on a construction which reduces the error from ε to $O(\varepsilon^2)$. The quality of this basic construction is given by the following lemma.

Lemma 3 *Let the parameters α and ε satisfy $1 > \alpha \geq 2n^{-1/(2\log^* n)}$ and $\varepsilon \geq \exp\left(-\alpha^{2\log^* n} n\right)$. Let $\mathrm{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ε)-extractor. Then, there exists a (k',ε')-extractor $\mathrm{EXT}'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$, with*

$$\begin{aligned} k' &= k + O(\log(1/\varepsilon)) \\ \varepsilon' &= O(\varepsilon^2) \\ m' &= (1-\alpha)\cdot m \\ d' &= d + O\left(\frac{\log(m/\varepsilon)}{\alpha^2}\right), \end{aligned}$$

and such that EXT' is computable in time $\mathrm{poly}(n,d')$ with two oracle queries to EXT.

In the above lemma, and throughout the paper, we use the notation $\exp(x)$ as shorthand for $2^{O(x)}$. We note that the hidden constant in the $O(\varepsilon^2)$ can be made arbitrarily close to 1 at the price of increasing the other hidden constants.

Improved Extractors. By applying Theorem 1 to the extractors of [Tre99, RRV99], we obtain the following improved constructions of extractors:

Theorem 4 *For every n, k, m, and ε such that $m \leq k \leq n$ and $\varepsilon > \exp\left(-n/(\log^* n)^{O(\log^* n)}\right)$, there are explicit (k,ε)-extractors $\mathrm{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with*

1. $d = O\left(\frac{\log^2 n}{\log(k/m)} + \log(1/\varepsilon)\right)$, or

2. $d = O(\log(1/\gamma)\cdot(\log^2 n + \log(1/\varepsilon)))$, where $1+\gamma = k/(m-1)$ and $1/m \leq \gamma < 1/2$.

In particular, using the first extractor with $m = k^{1-\gamma}$ for any constant $\gamma > 0$, we have $d = O((\log^2 n)/(\log k) + \log(1/\varepsilon))$. Using the first extractor with k/m constant, we can extract any constant fraction of the min-entropy using $d = O(\log^2 n + \log(1/\varepsilon))$. And, using the second extractor with $k = m$, we can extract all of the source min-entropy using $d = O((\log k)\cdot(\log^2 n + \log(1/\varepsilon)))$ truly random bits. Actually, using a technique from [RRV99], the output length in this last case can be increased to $m = k + d - 2\log(1/\varepsilon) - O(1)$ while only increasing d by a constant factor. This "entropy loss" of $2\log(1/\varepsilon) + O(1)$ is optimal up to an additive constant [RT97].

Generalizations. By allowing parameters to vary more freely in the proofs of Theorems 1 and 2, we can obtain results for reducing any initial error ε to (almost) any final error $\varepsilon' < \varepsilon$ while preserving the output length up to any $1-\alpha$ factor. These generalized theorems are given below.

Theorem 5 (Thm. 1, generalized) *Let $\mathrm{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ε)-extractor, with $\varepsilon \leq 1/m$. Then, for any $1 > \alpha \geq n^{-1/(3\log^* n)}$ and any $\varepsilon' > \exp\left(-n\cdot(\alpha/\log^* n)^{O(\log^* n)}\right)$, there exists a (k',ε')-extractor $\mathrm{EXT}'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$, with*

$$\begin{aligned} k' &= k + O(\log(1/\varepsilon')) \\ m' &= (1-\alpha)\cdot m \\ d' &= d + O\left(\frac{\log(1/\varepsilon')}{\alpha^2}\right), \end{aligned}$$

and such that EXT' is computable in time $\mathrm{poly}(n,d')$, making $O(\log(1/\varepsilon')/\log(1/\varepsilon))$ oracle queries to EXT.

Theorem 6 (Thm. 2, generalized) *Let $\beta > 0$ be any constant. Let $\mathrm{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ε)-extractor, with $\varepsilon > 1/m$. Then, for any $1 > \alpha \geq n^{-1/(3\log^* n)}$ and any $1-\beta > \varepsilon > \varepsilon' \geq 1/m$, there exists a (k',ε')-extractor $\mathrm{EXT}'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$, with*

$$\begin{aligned} k' &= k + O(\log(1/\varepsilon')) \\ m' &= (1-\alpha)\cdot m \\ d' &= d + O\left(\log m \cdot \frac{[\log\log(1/\varepsilon') - \log\log(1/\varepsilon)]^3}{\alpha^2}\right), \end{aligned}$$

and such that EXT' is computable in time $\mathrm{poly}(n,d')$ with $O(\log(1/\varepsilon')/\log(1/\varepsilon))$ oracle queries to EXT.

3 Overview of the Construction

In order to motivate our construction, we first discuss the possible sources of error in extractors. Consider a (k,ε)-extractor $\mathrm{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and source X of min-entropy $k' \geq k$ such that $\mathrm{EXT}(X,U_d)$ has statistical difference ε from the uniform distribution. This means that some strings in $\{0,1\}^m$ receive noticeably more probability mass under $\mathrm{EXT}(X,U_d)$ than they should; call these strings "heavy." Where can this error come from? Intuitively, we must be in one of the following two situations:

1. The error comes from the source: Some of the x's coming from X are "bad," in the sense that $\mathrm{EXT}(x,U_d)$ is a heavy output with probability much more than ε.

2. The error comes from the seeds: For most x's, roughly an ε fraction of the r's coming from U_d are "bad," in the sense that $\textsc{Ext}(x, r)$ is one of the heavy outputs.

The first possibility is easily dealt with by requiring k', the min-entropy of X, to be slightly higher than k. Intuitively, there can be at most 2^k bad x's, for otherwise the uniform distribution on those x's would result in a source of min-entropy k which is at distance much more than ε from uniform. So, if we require X' to be of min-entropy $k' = k + t$, a bad x will occur with probability at most 2^{-t}.

So, we need only deal with the second case, where the error comes from bad seeds, rather than bad outputs from the source. The second case says that if we throw away the bad r's for each x, then heavy strings will occur with very low probability. In other words, the output will be very close to a distribution with very high min-entropy (e.g. $m - 1$). More precisely, we will show that for every source X of min-entropy $k + t$ and every x coming from X, we can define a set $G_x \subset \{0, 1\}^d$ of "good" r's such that G_x is of density $1-O(\varepsilon)$ and $\textsc{Ext}(X, G_X)$ is at distance at most 2^{-t} from having min-entropy $m - 1$ (where $\textsc{Ext}(X, G_X)$ denotes the distribution obtained by sampling x according to X, choosing r uniformly in G_x, and outputting $\textsc{Ext}(x, r)$).

How does this help? It turns out that it is relatively easy to extract randomness from distributions of very high min-entropy, like min-entropy $m - 1$ over $\{0, 1\}^m$; Goldreich and Wigderson [GW94] give "optimal" extractors for this setting. So our task is reduced to obtaining a seed in G_x with probability better than $1-\varepsilon$. One way to do this is to try two independent seeds. Namely, consider $\textsc{Ext}'\colon \{0, 1\}^n \times \{0, 1\}^{2d} \to \{0, 1\}^n$, defined by

$$\textsc{Ext}'(x, (r_1, r_2)) = \textsc{Ext}(x, r_1)\,\textsc{Ext}(x, r_2).$$

This accomplishes what we want — at least one of the r_i's will land in G_x with probability at least $1 - O(\varepsilon^2)$, and hence one can argue that the output of $\textsc{Ext}'$ is at distance $O(\varepsilon^2)$ from min-entropy $m - 1$. But the output is now of length $2m$, so the result does not have min-entropy very close to its length, and we cannot use the Goldreich–Wigderson extractor. However, the min-entropy of the output is still a constant fraction of its length, and fortunately, Zuckerman [Zuc97] has constructed nearly optimal extractors for this setting. Thus, we consider the function

$$\textsc{Ext}''(x, (r_1, r_2, r_3)) = \textsc{Zuck}(\textsc{Ext}(x, r_1)\,\textsc{Ext}(x, r_2), r_3),$$

where $\textsc{Zuck}$ is the extractor of Zuckerman. $\textsc{Ext}''$ thus gives an output that is at distance $O(\varepsilon^2)$ from uniform, using $2d + O(\log m/\varepsilon) = O(d)$ truly random bits (where $O(\log m/\varepsilon)$ is the seed length for Zuckerman's extractor). So, we have roughly squared the error at the price of increasing the seed length by a constant factor. To reduce the error arbitrarily, one can now recurse. But the constant factor in seed length at each stage is too costly to obtain our desired result.

In order to improve upon this, we observe that it is not necessary that r_1 and r_2 be independent; we only need that one of the two will hit G_x with probability $1 - O(\varepsilon^2)$. One can generate a pair (r_1, r_2) satisfying this property using $d + O(\log(1/\varepsilon))$ truly random bits; for example, let r_2 be obtained by taking a random walk on a constructive expander graph starting at r_1, or let r_2 be obtained by applying the Goldreich–Wigderson extractor to r_1.[2] This modification allows us to obtain error $O(\varepsilon^2)$ at an *additive* cost of $O(\log(1/\varepsilon))$ truly random bits (assuming, for simplicity that $\varepsilon < 1/m$). Now if we recurse, these additive terms turn out to be a geometric series, and the total cost to reduce the error to ε' is $O(\log(1/\varepsilon'))$ truly random bits.

There is only one small difficulty left: Zuckerman's extractor is only optimal when extracting a constant fraction of the min-entropy. If we lose a constant fraction of the min-entropy at each stage of recursion, the final extractor will extract much less randomness than the original extractor. However, Zuckerman's extractor can extract more than a constant fraction of the min-entropy at a slight cost. Specifically, to extract a fraction $1 - \alpha$ of the min-entropy, the number of truly random bits used increases by a $\text{poly}(1/\alpha)$ factor. With appropriate choices of α during the recursion (ending with a constant α), we can ensure that the final extractor extracts a constant fraction of the randomness extracted by the original extractor, while using only $O(\log(1/\varepsilon'))$ additional truly random bits.

4 The Basic Step — Squaring the Error

As described in Section 3, the basic step of our construction is a general method for reducing the error of extractors from ε to $O(\varepsilon^2)$. The properties of this transformation are given in Lemma 3. In this section we formalize the description (given in Section 3) of this basic step and prove Lemma 3. In Section 5, we show how recursive applications of this step can further reduce the error to an (almost) arbitrarily small value.

4.1 Tools

To prove Lemma 3 we use two previous constructions of extractors. One construction was given by Zuckerman in [Zuc97] and the other was given by Goldreich and Wigderson in [GW94]. We apply both constructions in the setting of parameters where their seed-length is optimal: the extractor of [Zuc97] is used for sources of constant entropy

[2] These two methods are essentially equivalent, as the Goldreich–Wigderson extractor roughly amounts to taking a random walk on an expander from its input.

rate (i.e. of min-entropy $k = \Omega(n)$) whereas the extractor of [GW94] is used for sources of very high min-entropy (i.e. of min-entropy $k = n - O(\log(1/\varepsilon))$). We now give the formal statement of the constructions used in this paper:

Theorem 7 *([Zuc97]) Fix any constant $\delta > 0$. For any parameters n, k, ε, and α satisfying $k \geq \delta n$, $m \leq (1-\alpha)k$, $1 > \alpha \geq n^{-1/(2\log^* n)}$, and $\varepsilon \geq \exp\left(-\alpha^{2\log^* n} \cdot n\right)$, there exists an explicit (k,ε)-extractor* $\text{ZUCK}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$*, with*

$$d = O\left(\frac{\log(m/\varepsilon)}{\alpha^2}\right).$$

Remark 8 *In fact, the seed of this extractor is slightly shorter: Theorem 7 holds for $d = O\left(\frac{\log(m/\varepsilon)}{\alpha/\log(\alpha^{-1})}\right)$. Nevertheless, we replace the term $\alpha/\log(\alpha^{-1})$ with α^2 to simplify the exposition.*

Theorem 9 *([GW94]) For any $\varepsilon > 0$ and $0 < k \leq n$ there exists an explicit (k,ε)-extractor* $\text{GW}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^k$ *with*

$$d = O\left(n - k + \log(1/\varepsilon)\right).$$

Furthermore, $\text{GW}(U_n, U_d)$ *is uniformly distributed on* $\{0,1\}^k$.[3]

4.2 Extracting a Source of Constant Entropy-Rate

Let EXT be any (k,ε)-extractor with output length m. To prove Lemma 3, one has to show how to construct from EXT a comparable extractor EXT' that has error $O(\varepsilon^2)$. The main part of our construction is a method of using EXT to transform a source X of n-bit strings that has min-entropy roughly k to another source Y of $O(m)$-bit strings that is $O(\varepsilon^2)$ close to having min-entropy roughly m. In other words, we use EXT to obtain a source that is close to having a constant entropy-rate. Lemma 3 is then obtained by applying the extractor of [Zuc97] (that works well for such sources) on Y. An overview of this method was given in Section 3 and we now formalize its properties:

Lemma 10 *Let* $\text{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a (k,ε)-extractor. Then, there exists a function* $\text{EXT}^{rate}\colon \{0,1\}^n \times \{0,1\}^{\tilde{d}} \to \{0,1\}^{2m}$ *such that for any distribution X with min-entropy $\tilde{k}$ the induced distribution* $\text{EXT}^{rate}(X, U_{\tilde{d}})$ *is of statistical difference at most $\tilde{\varepsilon}$ from a source that has min-entropy $m - O(1)$, where*

$$\begin{aligned} \tilde{k} &= k + O(\log(1/\varepsilon)) \\ \tilde{\varepsilon} &= O(\varepsilon^2) \\ \tilde{d} &= d + O(\log(1/\varepsilon)). \end{aligned}$$

Moreover, EXT^{rate} *is computable in time* $\text{poly}(n, \tilde{d})$ *with two oracle queries to* EXT.

Proof of Lemma 10: Let $t = 2\log(1/\varepsilon)$ and assume wlog that t is an integer. Let

$$\text{GW}\colon \{0,1\}^{d+t} \times \{0,1\}^{\hat{d}} \to \{0,1\}^d$$

be the (d,ε)-extractor guaranteed by Theorem 9. Define $\tilde{d} = d + t + \hat{d}$. Note that indeed $\tilde{d} = d + O(\log(1/\varepsilon))$ (since by Theorem 9, $\hat{d} = O(\log(1/\varepsilon))$). Finally for any $x \in \{0,1\}^n$, $r \in \{0,1\}^{d+t}$ and $s \in \{0,1\}^{\hat{d}}$ define[4]

$$\text{EXT}^{rate}(x,(r,s)) \stackrel{\text{def}}{=} (\text{EXT}(x,r)\,\text{EXT}(x,\text{GW}(r,s))).$$

It is clear that EXT^{rate} is indeed computable in time $\text{poly}(n,\tilde{d})$ with two oracle queries to EXT. Set $\tilde{\varepsilon} = 7\varepsilon^2$ and $\tilde{k} = k + t$. Fix any source X of n-bit strings with min-entropy $\tilde{k}$. Let R be uniformly distributed on $\{0,1\}^{d+t}$ and let S be uniformly distributed on $\{0,1\}^{\hat{d}}$. We will prove that the induced distribution $\text{EXT}^{rate}(X,(R,S))$ is of statistical difference at most $\tilde{\varepsilon}$ from a source that has min-entropy $m-1$. To do so we first identify the set, $\mathbf{B}$, of "heavy" output strings (those whose probability mass under $\text{EXT}(X,U_d)$ is at least twice their probability mass under the uniform distribution). We then show that the probability that both $\text{EXT}(X,R)$ and $\text{EXT}(X,\text{GW}(R,S))$ are in $\mathbf{B}$, is at most $\tilde{\varepsilon}$.

Define $\mathbf{B} \stackrel{\text{def}}{=} \{z \in \{0,1\}^m \mid \Pr[\text{EXT}(X,U_d) = z] > 2^{-(m-1)}\}$. For any integer ℓ and any set $A \subseteq \{0,1\}^\ell$ define $\rho(A)$ to be the density of A in $\{0,1\}^\ell$ (i.e. the cardinality of A divided by 2^ℓ).

Claim 11 $\rho(\mathbf{B}) < \varepsilon$

Proof: By the definition of $\mathbf{B}$ we have that $\Pr[\text{EXT}(X,U_d) \in \mathbf{B}] > 2\rho(\mathbf{B})$. However, it is clear that $\Pr[U_m \in \mathbf{B}] = \rho(\mathbf{B})$. Since EXT is a (k,ε)-extractor and X has min-entropy $\tilde{k} > k$ we can conclude:

$$\rho(\mathbf{B}) < \Pr[\text{EXT}(X,U_d) \in \mathbf{B}] - \Pr[U_m \in \mathbf{B}] < \varepsilon$$

□

For every $x \in \{0,1\}^n$, the set $\mathbf{B}$ induces a set, $\mathbf{B}_x$, of "bad" seeds for x:

$$\mathbf{B}_x \stackrel{\text{def}}{=} \{r \in \{0,1\}^d \mid \text{EXT}(x,r) \in \mathbf{B}\}.$$

We show that for almost all x's there is only a 2ε-fraction of bad seeds.

[3] This property is not explicit in [GW94] but it immediately follows from the construction of their extractor.

[4] The expression $\text{EXT}(x,r)$ is a slight abuse of notation since r is longer than the seed length of EXT. We assume that EXT ignores these t extra bits of r.

Claim 12 $\Pr_X[\rho(\mathbf{B}_X) \geq 2\varepsilon] < 2^{-t} = \varepsilon^2$

Proof: Define X' to be the random variable X conditioned on the event $\rho(\mathbf{B}_X) \geq 2\varepsilon$. Suppose the claim is false and $\Pr_X[\rho(\mathbf{B}_X) \geq 2\varepsilon] \geq 2^{-t}$. This implies that X' has min-entropy k (recall that $\tilde{k} = k + t$). Therefore (since EXT is a (k, ε)-extractor) we have that $\Pr[\textsc{Ext}(X', U_d) \in \mathbf{B}] - \Pr[U_m \in \mathbf{B}] \leq \varepsilon$. On the other hand, by definition, $\Pr[\textsc{Ext}(X', U_d) \in \mathbf{B}] \geq 2\varepsilon$ whereas $\Pr[U_m \in \mathbf{B}] = \rho(\mathbf{B}) < \varepsilon$. This forms a contradiction and completes the proof of the claim. $\square$

We define the set of bad output strings of $\textsc{Ext}^{rate}$ (with respect to X) to be $\mathbf{B}' \stackrel{\text{def}}{=} \mathbf{B} \times \mathbf{B}$ (the set of strings in $\{0,1\}^m \times \{0,1\}^m$ such that both of their parts are in $\mathbf{B}$). For every $x \in \{0,1\}^n$, this induces a set of bad seeds for x: $\mathbf{B}'_x \stackrel{\text{def}}{=} \mathbf{B}_x \times \mathbf{B}_x$. We now show that: (1) The probability mass under $\textsc{Ext}^{rate}(X, (R, S))$ of any individual string $(u, v) \notin \mathbf{B}'$ is at most $2^{-(m-1)}$. (2) The probability mass under $\textsc{Ext}^{rate}(X, (R, S))$ of $\mathbf{B}'$ is at most $\tilde{\varepsilon}$. This will complete the proof of the lemma.

Claim 13 *For any* $(u, v) \notin \mathbf{B}'$,

$$\Pr\left[\textsc{Ext}^{rate}(X, (R, S)) = (u, v)\right] \leq 2^{-(m-1)}.$$

Proof: If $(u, v) \notin \mathbf{B}'$ then either $u \notin \mathbf{B}$ or $v \notin \mathbf{B}$. Since both R and $\mathrm{GW}(R, S)$ are uniformly distributed (though not independent) we get by the definition of $\mathbf{B}$ that:

1. If $u \notin \mathbf{B}$, then $\Pr\left[\textsc{Ext}^{rate}(X, (R, S)) = (u, v)\right] \leq \Pr[\textsc{Ext}(X, R) = u] \leq 2^{-(m-1)}$.
2. If $v \notin \mathbf{B}$, then $\Pr\left[\textsc{Ext}^{rate}(X, (R, S)) = (u, v)\right] \leq \Pr[\textsc{Ext}(X, \mathrm{GW}(R, S)) = v] \leq 2^{-(m-1)}$.

$\square$

Claim 14 *For every* $x \in \{0,1\}^n$, *if* $\rho(\mathbf{B}_x) < 2\varepsilon$ *then* $\Pr\left[(R, \mathrm{GW}(R, S)) \in \mathbf{B}'_x\right] < 6\varepsilon^2$

Proof: $\Pr\left[(R, \mathrm{GW}(R, S)) \in \mathbf{B}'_x\right] = \Pr[R \in \mathbf{B}_x] \cdot \Pr[\mathrm{GW}(R, S) \in \mathbf{B}_x \mid R \in \mathbf{B}_x]$. Therefore, if $\Pr[R \in \mathbf{B}_x] = \rho(\mathbf{B}_x) < 6\varepsilon^2$ we are done. Assume that $\rho(\mathbf{B}_x) \geq 6\varepsilon^2$. In this case, the distribution of R conditioned on the event $R \in \mathbf{B}_x$ still has min-entropy at least $d + t - \log(1/(6\varepsilon^2)) > d$. Therefore, by the definition of GW the distribution of $\mathrm{GW}(R, S)$ conditioned on the event $R \in \mathbf{B}_x$ is $\varepsilon - close$ to uniform. We can conclude that if $\rho(\mathbf{B}_x) \geq 6\varepsilon^2$ then $\Pr[\mathrm{GW}(R, S) \in \mathbf{B}_x \mid R \in \mathbf{B}_x] \leq \rho(\mathbf{B}_x) + \varepsilon < 3\varepsilon$ which completes the proof of the claim. $\square$

Claim 15 $\Pr\left[\textsc{Ext}^{rate}(X, (R, S)) \in \mathbf{B}'\right] < \tilde{\varepsilon}$

Proof: By the definition of $\mathbf{B}'$ and from Claims 12 and 14 we get that

$$\begin{aligned}
&\Pr\left[\textsc{Ext}^{rate}(X, (R, S)) \in \mathbf{B}'\right] \\
&= \Pr\left[(R, \mathrm{GW}(R, S)) \in \mathbf{B}'_X\right] \\
&\leq \Pr[\rho(\mathbf{B}_X) \geq 2\varepsilon] + \Pr\left[(R, \mathrm{GW}(R, S)) \in \mathbf{B}'_X \mid \rho(\mathbf{B}_X) < 2\varepsilon\right] \\
&< \varepsilon^2 + 6\varepsilon^2 = \tilde{\varepsilon}
\end{aligned}$$

$\square$

Define

$$G \stackrel{\text{def}}{=} \{z \in \{0,1\}^{2m} \mid \Pr\left[\textsc{Ext}^{rate}(X, (R, S)) = z\right] < 2^{-m}\}$$

(by definition, G contains almost all $2m$-bit strings). Let C be a random variable which is identically distributed to $\textsc{Ext}^{rate}(X, (R, S))$ in the event that $\textsc{Ext}^{rate}(X, (R, S)) \notin \mathbf{B}'$ and uniformly distributed over G in the event $\textsc{Ext}^{rate}(X, (R, S)) \in \mathbf{B}'$. By Claim 13 and the definition of G, C has min-entropy $m - 1$. By Claim 15, $\textsc{Ext}^{rate}(X, (R, S))$ is $\tilde{\varepsilon}$-close to C. This completes the proof of Lemma 10. $\square$

Remark 16 *We prove that Lemma 10 holds with* $\tilde{\varepsilon} = 7\varepsilon^2$. *However, the lemma also holds for* $\tilde{\varepsilon} = (1 + \gamma)\varepsilon^2$ *where* $\gamma > 0$ *is an arbitrarily small constant. Showing this requires two changes in the definition of* $\textsc{Ext}^{rate}$ *(for an appropriate constant* c_γ*): (1) Set* $t = c_\gamma \log(1/\varepsilon)$ *(2) Take* GW *to be a* $(d, \varepsilon/c_\gamma)$*-extractor. Repeating the original proof (with the required adjustments), it can now be shown that* $\textsc{Ext}^{rate}(X, (R, S))$ *is sufficiently close to having min-entropy* $m - c_\gamma$.

In Section 3, we discussed the possible sources of error in extractors. Lemma 17 below (which is implicit in the proof of Lemma 10) formalizes that discussion.

Lemma 17 *Let* $\textsc{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, ε)*-extractor, where* $\varepsilon < 1/4$. *Let* X *be any source of min-entropy* $k + t$. *Then there exist sets* $\{\mathbf{G}_x\}_{x \in \{0,1\}^n}$ *such that*

1. *For every* x, $\mathbf{G}_x \subset \{0,1\}^d$ *is of density* $1 - O(\varepsilon)$.
2. $\textsc{Ext}(X, \mathbf{G}_X)$ *is at distance at most* 2^{-t} *from having min-entropy* $m - O(1)$ *(where* $\textsc{Ext}(X, \mathbf{G}_X)$ *denotes the distribution obtained by sampling* x *according to* X*, choosing* r *uniformly in* $\mathbf{G}_x$*, and outputting* $\textsc{Ext}(x, r)$*).*

Remark 18 *The assumption that* $\varepsilon < 1/4$ *simplifies the proof and for any constant* $\beta > 0$ *it can be relaxed to* $\varepsilon < 1 - \beta$. *However, when* ε *is a constant, claiming that there is only an* $O(\varepsilon)$ *fraction of "bad" seeds is not very interesting.*

Proof sketch: Consider the sets $\mathbf{B}_x$ in the proof of Lemma 10. For every x such that $\rho(\mathbf{B}_x) \leq 2\varepsilon$, define $\mathbf{G}_x \stackrel{\text{def}}{=} \{0,1\}^d \setminus \mathbf{B}_x$. Otherwise, define $\mathbf{G}_x \stackrel{\text{def}}{=} \{0,1\}^d$. By the assumption that $\varepsilon < 1/4$, we have that for every $z \in \{0,1\}^m$, $\Pr[\textsc{Ext}(X, \mathbf{G}_X) = z] < 2\Pr[\textsc{Ext}(X, U_d) = z]$. From the definition of the sets $\mathbf{B}_x$ and Claim 12, it follows that the sets $\{\mathbf{G}_x\}_{x \in \{0,1\}^n}$ satisfy the conditions of Lemma 17. □

4.3 Using Zuckerman's Extractors

Lemma 10 gives us a simple way to use any extractor Ext with output length m in order to produce an $O(m)$-bit string y that is $O(\varepsilon^2)$ close to having min-entropy roughly m. Lemma 3 can now be easily obtained by applying the extractor of Theorem 7 on y to extract $(1-\alpha)\cdot m$ bits that are $O(\varepsilon^2)$-close to uniform. However, using this extractor imposes some limitations on ε and α (i.e. on the error and the number of bits that can be extracted). These limitations are stated in Theorem 7 and are carried on to Lemma 3 and to Theorems 1, 4 and 5. As discussed in Section 7, an improved construction of extractors for the case of constant entropy-rate (or even improved mergers [NT99]) may also improve our construction.

Proof of Lemma 3: Let the parameters α and ε satisfy $1 > \alpha \geq 2n^{-1/(2\log^* n)}$ and $\varepsilon \geq \exp\left(-\alpha^{\log^* n} n\right)$. Let $\textsc{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a (k,ε)-extractor. We will define a (k',ε')-extractor $\textsc{Ext}'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$, with the properties stated by the lemma.

If $\alpha = O(1/m)$ then extractors with output-length m and seed-length $O(1/\alpha + \log(n/\varepsilon))$ were already given in [SZ98, GW94]. Therefore we can assume that $\alpha < \gamma m$, for an arbitrarily small constant γ. Let

$$\textsc{Ext}^{rate}\colon \{0,1\}^n \times \{0,1\}^{\tilde{d}} \to \{0,1\}^{2m}$$

be the function guaranteed to exist by Lemma 10 such that for any distribution X with min-entropy $\tilde{k}$ the induced distribution $\textsc{Ext}^{rate}(X, U_{\tilde{d}})$ is of statistical difference at most $\tilde{\varepsilon}$ from a source that has min-entropy $m-c$. Define $m' = (1-\alpha/2)(m-c)$ (which indeed implies that $m' > (1-\alpha)\cdot m$ for a sufficiently small choice of γ). Let

$$\textsc{Zuck}\colon \{0,1\}^{2m} \times \{0,1\}^{\hat{d}} \to \{0,1\}^{m'}$$

be the $((m-c),\varepsilon^2)$-extractor guaranteed to exist by Theorem 7 (since $1 > \alpha/2 \geq n^{-1/(2\log^* n)}$ and $\varepsilon^2 \geq \exp\left(-\alpha^{2\log^* n}\cdot n\right)$).

Define $k' = \tilde{k}$ (which indeed implies that $k' = k + O(\log(1/\varepsilon))$), $\varepsilon' = \tilde{\varepsilon} + \varepsilon^2$ (which indeed implies that $\varepsilon' = O(\varepsilon^2)$) and $d' = \tilde{d} + \hat{d}$ (which indeed implies that $d' = d + O\left(\frac{\log(m/\varepsilon)}{\alpha^2}\right)$). Finally, define $\textsc{Ext}'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$ such that for every $x \in \{0,1\}^n$, $r \in \{0,1\}^{\tilde{d}}$ and $r' \in \{0,1\}^{\hat{d}}$

$$\textsc{Ext}'(x,(r,r')) \stackrel{\text{def}}{=} \textsc{Zuck}(\textsc{Ext}^{rate}(x,r),r')$$

It is clear that $\textsc{Ext}'$ is computable in time $\mathrm{poly}(n,d')$ with two oracle queries to Ext (given the properties of $\textsc{Ext}^{rate}$ and Zuck). It remains to show that $\textsc{Ext}'$ is a (k',ε')-extractor. Let X be any source with min-entropy k'. By the properties of $\textsc{Ext}^{rate}$ we have that $\textsc{Ext}^{rate}(X, U_{\tilde{d}})$ is $\tilde{\varepsilon}$-close to a source that has min-entropy $m-c$. Therefore, since Zuck is a $((m-c),\varepsilon^2)$-extractor, $\textsc{Ext}'(X,(U_{\tilde{d}},U_{\hat{d}}))$ is $\tilde{\varepsilon} + \varepsilon^2 = \varepsilon'$-close to the uniform distribution. □

Remark 19 *Following Remark 16, we note that Lemma 3 holds with $\varepsilon' = (1+\gamma)\varepsilon^2$ where $\gamma > 0$ is an arbitrarily small constant (at the price of increasing the other hidden constants of the lemma).*

5 Using Recursion to Reduce the Error

In this section we show how recursive applications of our basic step (i.e. of Lemma 3) can reduce the error of any extractor to an almost arbitrarily small ε. The only limitation on ε is the one imposed by the extractors of [Zuc97] (see Theorem 7). We prove the quality of our reduction in the two special cases we consider the most interesting: (1) Reducing the error from $1/m$ to an (almost) arbitrarily small $\varepsilon > 0$. (2) Reducing a constant error to error $1/m$. In the first case the reduction is optimal in that the seed of the extractor increases by only $O(\log(1/\varepsilon))$ additional bits. In the second case the increase in the seed-length is slightly super-logarithmic. The quality of these reductions is formalized in Theorems 1 and 2 which we prove in this section. The proof of the more general versions (i.e. Theorems 5 and 6) is more or less the same. However, we chose to prove the special cases for the sake of readability.

Reducing error $1/m$ to smaller error ε. Starting with a $(k,1/m)$-extractor, Ext, one can obtain a (k',ε)-extractor $\textsc{Ext}'$ by $O(\log\log(1/\varepsilon))$ applications of Lemma 3. However, in each one of these applications the new extractor has an output-length which is shorter by some α'-fraction than that of the old extractor. It turns out that one cannot keep α' constant in all these applications without either paying too much in the the seed-length or loosing too much in the output-length. We therefore use in our proof different α_i's for the different applications (in earlier applications ε is larger and we can therefore afford a smaller α_i without paying too much in the seed length).

Proof of Theorem 1: Let $\textsc{Ext}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be an explicit $(k,1/m)$-extractor. We define a sequence of (k_i,ε_i)-extractors $\{\textsc{Ext}_i\colon \{0,1\}^n \times \{0,1\}^{d_i} \to$

$\{0,1\}^{m_i}\}_{i=0}^{t}$ where $\textsc{Ext}_0 = \textsc{Ext}$, $\varepsilon_i = \varepsilon_0^{(1.9)^i}$, $t = O(\log\log(1/\varepsilon') - \log\log(m))$ (such that $\varepsilon_t = \varepsilon'$) and $\textsc{Ext}_t = \textsc{Ext}'$ satisfies the requirements of the theorem. $\textsc{Ext}_{i+1}$ is obtained from $\textsc{Ext}_i$ by applying Lemma 3 with $\alpha_i = \frac{\alpha}{c\cdot(t-i+1)^2}$ for some constant c that will be determined within the proof.

By Lemma 3, we can set $\varepsilon_{i+1} = O(\varepsilon_i^2)$. As long as $\varepsilon_i \le 1/m$ (which always holds since the sequence $\{\varepsilon_i\}$ is decreasing), we can set $\varepsilon_i = \varepsilon_0^{(1.9)^i}$ as stated above. Equivalently, $\varepsilon_{t-j} = \varepsilon_t^{(1.9)^{-j}} = (\varepsilon')^{(1.9)^{-j}}$. In order for Lemma 3 to apply, we need to verify that $\varepsilon_{t-j} > \exp(-\alpha_{t-j}^{2\log^* n}\cdot n)$, i.e. $\log(1/\varepsilon_{t-j})/(\alpha_{t-j}^{2\log^* n}) = O(n)$.

$$\begin{aligned}
&\log(1/\varepsilon_{t-j})\cdot\alpha_{t-j}^{-2\log^* n}\\
&= \left((1.9)^{-j}\cdot\log(1/\varepsilon')\right)\cdot\left(\frac{\alpha}{c\cdot(j+1)^2}\right)^{-2\log^* n}\\
&= \log(1/\varepsilon')\cdot\alpha^{-2\log^* n}\cdot\left(\frac{(j+1)^{O(\log^* n)}}{1.9^j}\right)\\
&\le \log(1/\varepsilon')\cdot\alpha^{-2\log^* n}\cdot(\log^* n)^{O(\log^* n)} \qquad (1)\\
&\le O(n). \qquad (2)
\end{aligned}$$

Inequality (2) follows from our requirement that $\varepsilon > \exp\left(-n/(\log^* n)^{O(\log^* n)}\right)$. Inequality (1) is obtained by a case analysis on the value of j. When $j \le (\log^* n)^2$, then $(j+1)^{O(\log^* n)} = (\log^* n)^{O(\log^* n)}$, and when $j > (\log^* n)^2$, $(j+1)^{O(\log^* n)}/1.9^j$ is bounded above by a constant independent of n.

By Lemma 3, $k_{i+1} = k_i + O(\log(1/\varepsilon_i))$. Therefore,

$$\begin{aligned}
k' \stackrel{\text{def}}{=} k_t &= k_0 + O\left(\sum_{i=0}^{t}\log(1/\varepsilon_i)\right)\\
&= k + O\left(\log(1/\varepsilon')\sum_{i=0}^{t}(1.9)^{-i}\right)\\
&= k + O\left(\log(1/\varepsilon'\right).
\end{aligned}$$

By Lemma 3, $m_{i+1} = (1-\alpha_i)m_i > m_i - \alpha_i m$. Therefore, for some choice of the constant c,

$$\begin{aligned}
m' \stackrel{\text{def}}{=} m_t &> m_0 - \left(\sum_{i=0}^{t}\alpha_i m\right)\\
&= m - \left(\sum_{i=0}^{t}\frac{1}{c\cdot(t-i+1)^2}\right)\alpha m\\
&> (1-\alpha)m
\end{aligned}$$

It remains to bound $d' \stackrel{\text{def}}{=} d_t$. Since

$$\begin{aligned}
d_{i+1} &= d_i + O\left(\frac{\log(m/\varepsilon_i)}{\alpha_i^2}\right)\\
&= d_i + O\left(\frac{\log(1/\varepsilon')}{\alpha^2}\frac{(t-i+1)^2}{2^{t-i}}\right),
\end{aligned}$$

we get that

$$\begin{aligned}
d' &= d_0 + \frac{\log(1/\varepsilon')}{\alpha^2}\cdot O\left(\sum_{j=0}^{t}\frac{(j+1)^2}{2^j}\right)\\
&= d + O\left(\frac{\log(1/\varepsilon')}{\alpha^2}\right)
\end{aligned}$$

Finally, since the depth of the recursion is $O(\log\log(1/\varepsilon'))$ and computing $\textsc{Ext}_{i+1}$ only requires two queries to $\textsc{Ext}_i$ and an additional $\mathrm{poly}(n, d_{i+1})$ time we can deduce that $\textsc{Ext}'$ is indeed computable in $\mathrm{poly}(n, d')$ time. □

Reducing a constant error to error $1/m$. Starting with a (k,ε)-extractor for some constant ϵ one can obtain a $(k, 1/m)$-extractor, $\textsc{Ext}'$, by $O(\log\log m)$ applications of Lemma 3. Therefore, the proof of Theorem 2 can be obtained in a very similar way to the proof of Theorem 1. However, in this case there is no gain in taking different values α_i's for the different applications of Lemma 3. The reason is that the seed length will now grow by $O(\log m/\alpha^2)$ at each application of Lemma 3 *regardless of the current error*. Therefore, there is no way to balance α_i with the ε_i's as done in the proof of Theorem 1.

Proof of Theorem 2: Let $\textsc{Ext}\colon\{0,1\}^n\times\{0,1\}^d \to \{0,1\}^m$ be an explicit (k,ϵ)-extractor for some constant ε. We define a sequence of explicit (k_i,ε_i)-extractors $\{\textsc{Ext}_i\colon\{0,1\}^n\times\{0,1\}^{d_i}\to\{0,1\}^{m_i}\}_{i=0}^{t}$ where $\textsc{Ext}_0 = \textsc{Ext}$, $\varepsilon_i = \varepsilon_0^{(1.9)^i}$, $t = O(\log\log m)$ (such that $\varepsilon_t = 1/m$) and $\textsc{Ext}_t = \textsc{Ext}'$ satisfies the requirements of the theorem. $\textsc{Ext}_{i+1}$ is obtained from $\textsc{Ext}_i$ by applying Lemma 3 with $\alpha' = \frac{\alpha}{t}$.

As noted in Remark 19, Lemma 3 holds with $\varepsilon' = (1+\gamma)\varepsilon$ where $\gamma > 0$ is an arbitrary constant. Therefore, by setting γ to be small enough we can indeed obtain the relation $\varepsilon_i = \varepsilon_0^{(1.9)^i}$ as desired. It is now easy to verify that:

$$\begin{aligned}
k_t &= k_0 + O\left(\sum_{i=0}^{t}\log(1/\varepsilon_i)\right) = k + O(\log m)\\
m_t &> (1-t\alpha')\cdot m = (1-\alpha)\cdot m\\
d_t &= d_0 + O\left(\sum_{i=0}^{t}\frac{\log(m/\varepsilon_i)}{(\alpha')^2}\right)\\
&= d + O(\log m\cdot t^3)\\
&= d + O(\log m\cdot \mathrm{polyloglog}\, m)
\end{aligned}$$

Finally, it is easy to verify that $\textsc{Ext}'$ is indeed computable in $\mathrm{poly}(n, d')$ time. □

6 Strong Extractors

As mentioned in the Introduction, all the results of this paper can be extended to *strong* extractors. Specifically, each of our transformations of an extractor EXT with error ε to an extractor EXT' with error ε' have the property that if EXT is a strong extractor then so is EXT'. This is significant because many of the known constructions of extractors actually give strong extractors. In particular, since the constructions in [Tre99, RRV99] can be shown to give strong extractors, our concrete constructions of extractors (Theorem 4) also give strong extractors.

Our method of reducing the error in extractors consists of recursive applications of the basic step: a transformation of an extractor EXT with error ε to an extractor EXT' with error $O(\varepsilon^2)$. Therefore, to show that this method applies to strong extractors, it is sufficient that the basic step applies in this case. We now state an analogous to Lemma 3 for the case of strong extractors:

Lemma 20 *Let the parameters α and ε satisfy $1 > \alpha \geq 2n^{-1/(2\log^* n)}$ and $\varepsilon \geq \exp\left(-\alpha^{2\log^* n} n\right)$. Let $\text{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a* strong *(k, ε)-extractor. Then, there exists a* strong *(k', ε')-extractor $\text{EXT}'\colon \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{m'}$, with*

$$\begin{aligned} k' &= k + O(\log(1/\varepsilon)) \\ \varepsilon' &= O(\varepsilon^2) \\ m' &= (1-\alpha)\cdot m \\ d' &= d + O\left(\frac{\log(m/\varepsilon)}{\alpha^2}\right), \end{aligned}$$

and such that EXT' is computable in time $\text{poly}(n, d')$ with two oracle queries to EXT.

Recall that reducing the error from ε to $O(\varepsilon^2)$ is done in two stages: (1) Using EXT to transform a source X of n-bit strings that has min-entropy roughly k to another source Y of $O(m)$-bit strings that is $O(\varepsilon^2)$-close to having min-entropy roughly m. (2) Applying the extractor of Zuckerman [Zuc97] on Y to obtain $(1-\alpha)\cdot m$ bits that are $O(\varepsilon^2)$-close to uniform. Since the Zuckerman's extractor is in itself a strong extractor it is sufficient to show that the first stage works in the case of strong extractors. We now state the properties of the first stage for this case (in analogy to Lemma 10):

Lemma 21 *Let $\text{EXT}\colon \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ be a strong (k, ε)-extractor. Then, for*

$$\begin{aligned} \tilde{k} &= k + O(\log(1/\varepsilon)) \\ \tilde{\varepsilon} &= O(\varepsilon^2) \\ \tilde{d} &= d + O(\log(1/\varepsilon)), \end{aligned}$$

there exists a function $\text{EXT}^{rate}\colon \{0,1\}^n \times \{0,1\}^{\tilde{d}} \to \{0,1\}^{2m}$ such that for any distribution X with min-entropy $\tilde{k}$ the induced distribution $\langle U_{\tilde{d}}, \text{EXT}^{rate}(X, U_{\tilde{d}})\rangle$ is of statistical difference at most $\tilde{\varepsilon}$ from a distribution $\langle \tilde{R}, C\rangle$ (where $\tilde{R}$ is distributed on $\{0,1\}^{\tilde{d}}$ and C is distributed on $\{0,1\}^{2m}$) with the following property: For any value $\tilde{r} \in \{0,1\}^{\tilde{d}}$ the distribution of C conditioned on the event $\tilde{R} = \tilde{r}$ has min-entropy $m - O(1)$. Moreover, EXT^{rate} is computable in time $\text{poly}(n, \tilde{d})$ with two oracle queries to EXT.

Proof sketch: The proof of Lemma 21 is very similar to the proof of Lemma 10 (in some sense even simpler). The main difference between the two proofs is in the definition of the sets $\mathbf{B}$, $\mathbf{B}_x$, $\mathbf{B}'$ and $\mathbf{B}'_x$. We therefore focus on these changes.

Define EXT^{rate} as in the proof of Lemma 10. Fix any source X of n-bit strings with min-entropy $\tilde{k}$. Let R be uniformly distributed on $\{0,1\}^{d+t}$ and let S be uniformly distributed on $\{0,1\}^{\hat{d}}$. We will prove that the induced distribution $\langle (R,S), \text{EXT}^{rate}(X,(R,S))\rangle$ has statistical difference at most $\tilde{\varepsilon}$ from a distribution $\langle \tilde{R}, C\rangle$ as in the statement of the lemma.

$\mathbf{B}$ is defined as in the proof of Lemma 10 *when we replace the extractor* EXT *with the extractor* $\widetilde{\text{EXT}}$ that is defined by $\widetilde{\text{EXT}}(x,r) = \langle r, \text{EXT}(x,r)\rangle$ ($\widetilde{\text{EXT}}$ is an extractor since EXT is a strong extractor). Therefore, we let $\mathbf{B}$ be the set of "heavy" *seed-output pairs* of EXT (instead of just "heavy" output strings). More precisely, define $\mathbf{B} \stackrel{\text{def}}{=} \{\langle r, z\rangle \in \{0,1\}^{d+m} \mid \Pr[\text{EXT}(X,r) = z] > 2^{-(m-1)}\}$. For every $x \in \{0,1\}^n$, the set $\mathbf{B}$ induces a set, $\mathbf{B}_x$, of "bad" seeds for x:

$$\mathbf{B}_x \stackrel{\text{def}}{=} \{r \in \{0,1\}^d \,|\, \langle r, \text{EXT}(x,r)\rangle \in \mathbf{B}\}.$$

Since $\widetilde{\text{EXT}}$ is an extractor we have (in exactly the same way as in the proof of Lemma 10) the following two claims:

Claim 22 $\rho(\mathbf{B}) < \varepsilon$

Claim 23 $\Pr_X[\rho(\mathbf{B}_X) \geq 2\varepsilon] < 2^{-t} = \varepsilon^2$

We define the set of bad output strings of EXT^{rate} (with respect to X) to be

$$\mathbf{B}' \stackrel{\text{def}}{=} \{\langle (r,s),(u,v)\rangle \mid \langle r,u\rangle \in \mathbf{B} \text{ and } \langle \text{GW}(r,s), v\rangle \in \mathbf{B}\}.$$

For every $x \in \{0,1\}^n$, this induces a set of bad seeds for x:

$$\mathbf{B}'_x \stackrel{\text{def}}{=} \{(r,s) \mid \langle (r,s), \text{EXT}^{rate}(x,(r,s))\rangle \in \mathbf{B}'\}.$$

By the definition of $\mathbf{B}'$ we have that

Claim 24 *For any $\langle (r,s),(u,v)\rangle \notin \mathbf{B}'$,*

$$\Pr\left[\text{EXT}^{rate}(X,(r,s)) = (u,v)\right] \leq 2^{-(m-1)}.$$

In exactly the same way as in the proof of Lemma 10 we have that

Claim 25 $\Pr\left[\langle (R,S), \text{EXT}^{rate}(X,(R,S))\rangle \in \mathbf{B}'\right] < \tilde{\varepsilon}$

For every possible seed $\tilde{r}$, define $G_{\tilde{r}} \stackrel{\text{def}}{=} \{z \in \{0,1\}^{2m} \mid \Pr\left[\text{EXT}^{rate}(X,\tilde{r}) = z\right] < 2^{-m}\}$ (by definition, $G_{\tilde{r}}$ contains almost all $2m$-bit strings). Let $\tilde{R}$ be the random variable (R,S). Let C be a random variable which is identically distributed to $\text{EXT}^{rate}(X,\tilde{R})$ in the event that $\langle \tilde{R}, \text{EXT}^{rate}(X,\tilde{R})\rangle \notin \mathbf{B}'$ and uniformly distributed over $G_{\tilde{R}}$ in the event $\langle \tilde{R}, \text{EXT}^{rate}(X,\tilde{R})\rangle \in \mathbf{B}'$. By Claim 24 and the definition of $G_{\tilde{r}}$, for any value $\tilde{r} \in \{0,1\}^{\tilde{d}}$ the distribution of C conditioned on the event $\tilde{R} = \tilde{r}$ has min-entropy $m-1$. By Claim 25, $\langle \tilde{R}, \text{EXT}^{rate}(X,\tilde{R})\rangle$ is $\tilde{\varepsilon}$-close to $\langle \tilde{R}, C\rangle$. This completes the proof of the lemma. $\square$

7 Discussion

Ideally, we would like to have a method to reduce the error of an extractor from constant to any ε, using only $O(\log(1/\varepsilon))$ additional random bits (and without changing any other parameters by much). This would imply that in order to come up with optimal extractors one only has to deal with the constant error case. Our method comes close to that goal, but it falls short in two points.

First, our method is only optimal when the original error is $\leq 1/m$. Indeed, if the error is $\leq 1/m$ we are able to reduce the error to any ε, using only $O(\log(1/\varepsilon))$ additional random bits. However, to reduce the error from constant to $1/m$ we need $O(\log m \cdot \text{polyloglog}(m))$ random bits, which is not optimal. Is there an improved method to reduce the error from constant to $1/m$ using only $O(\log m)$ random bits ?

The second problem with our construction is the entropy loss. Since we use Zuckerman's extractor, we are only able to extract $(1-\alpha)\cdot m$ bits of the source min-entropy, where m is the number of bits extracted by the original extractor. In particular, this is significant when the original extractor extracts all of the source min-entropy. Is it possible to improve the entropy loss of our construction? Our entropy loss is the same as the one in Zuckerman's construction. However, we use Zuckerman's extractor only as a "merger" in the sense of [NT99]. That is, we use it to combine two (dependent) distributions, one of which contains all the randomness we want to extract. Thus, we do not necessarily need its full power as an extractor. Can one replace Zuckerman's extractor in our construction by a different "merger" with a smaller entropy loss?

Acknowledgments

We thank Oded Goldreich and the anonymous reviewers for their comments.

References

[GW94] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Electronic Colloquium on Computational Complexity*, Technical Report TR94-002, 1994. Revised December 1996. `http://www.eccc.uni-trier.de/eccc`.

[NT99] Noam Nisan and Amnon Ta-Shma. Extracting randomness: A survey and new constructions. *Journal of Computer and System Sciences*, 58(1):148–173, 1999.

[NZ96] Noam Nisan and David Zuckerman. Randomness is linear in space. *Journal of Computer and System Sciences*, 52(1):43–52, February 1996.

[RR99] Ran Raz and Omer Reingold. On recycling the randomness of the states in space bounded computation. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, Atlanta, GA, May 1999.

[RRV99] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, Atlanta, GA, 1999.

[RT97] Jaikumar Radhakrishnan and Amnon Ta-Shma. Tight bounds for depth-two superconcentrators. In *38th Annual Symposium on Foundations of Computer Science*, pages 585–594, Miami Beach, Florida, 20–22 October 1997. IEEE.

[SZ98] Aravind Srinivasan and David Zuckerman. Computing with very weak random sources. To appear in *SIAM Journal on Computing*, 1998. Preliminary version in *FOCS '94*.

[Tre99] Luca Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, Atlanta, GA, May 1999. See also ECCC TR98-55.

[WZ95] Avi Wigderson and David Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. Technical Report CS-TR-95-21, University of Texas Department of Computer Sciences, 1995. To appear in *Combinatorica*.

[Zuc97] David Zuckerman. Randomness-optimal oblivious sampling. *Random Structures & Algorithms*, 11(4):345–367, 1997.