# Extracting Randomness via Repeated Condensing*

Omer Reingold† Ronen Shaltiel‡ Avi Wigderson§

**Abstract**

Extractors (defined by Nisan and Zuckerman) are procedures that use a *small* number of truly random bits (called the seed) to extract *many* (almost) truly random bits from arbitrary distributions as long as they have sufficient (min)-entropy. A natural weakening of an extractor is a *condenser*, whose output distribution has a higher entropy rate than the input distribution (without losing much of the initial entropy). An extractor can be viewed as an ultimate condenser as it outputs a distribution with the maximal entropy rate.

In this paper we construct explicit condensers with short seed length. The condenser constructions combine (variants or more efficient versions of) ideas from several works, including the block extraction scheme of [NZ96], the observation made in [SZ98, NTS99] that a failure of the block extraction scheme is also useful, the recursive "win-win" case analysis of [ISW99, ISW00], and the error correction of random sources used in [Tre01].

As a byproduct, (via repeated iterating of condensers), we obtain new extractor constructions. The new extractors give significant qualitative improvements over previous ones for sources of arbitrary min-entropy; they are nearly optimal *simultaneously* in the main two parameters - seed length and output length. Specifically, our extractors can make any of these two parameters optimal (up to a constant factor), only at a *poly-logarithmic* loss in the other. Previous constructions require *polynomial* loss in both cases for general sources.

We also give a simple reduction converting "standard" extractors (which are good for an average seed) to "strong" ones (which are good for most seeds), with essentially the same parameters. With it, all the above improvements apply to strong extractors as well.

# 1 Introduction

## 1.1 Extractors

One of the most successful ideas in modern computer science is that of probabilistic algorithms and protocols. These are procedures that use random coin tosses when performing computational tasks. A natural problem is how can computers obtain truly random bits.

A line of research (initiated by [vN51, Blu86, SV86]) is motivated by the question of availability of truly random bits. The idea is to make truly random bits available by refining the (imperfect)

---

*A preliminary version of this paper appeared in [RSW00].

†Incumbent of the Walter and Elise Haas Career Development Chair, Department of Computer Science Weizmann Institute of Science, Rehovot, Israel. omer.reingold@weizmann.ac.il. Most of this research was performed while at AT&T Labs - Research. Florham Park, NJ, and while visiting the Institute for Advanced Study, Princeton, NJ. Research was supported in part by US-Israel Binational Science Foundation Grant 2002246.

‡Department of Computer Science, University of Haifa, Haifa 31905, Israel. ronen@cs.haifa.ac.il. Part of this research was performed while staying at the Institute for Advanced Study, Princeton, NJ. This research was also supported in part by the Borland Scholar program

§Institute for advanced study, Princeton, NJ. avi@ias.edu. This research was supported by USA-Israel BSF Grant 97-00188.

randomness found in some natural physical processes. The goal is to design procedures called "randomness extractors" that given a sample from an arbitrary source of randomness produces truly random bits. It was shown by [SV86] that this task cannot be performed by deterministic algorithms, even for some randomness sources that have a simple and "nice" structure. In light of this, the goal of this line of research became "spending" as few as possible truly random bits in order to extract as many as possible (almost) truly random bits from arbitrary imperfect random sources which contain sufficient randomness.

The most general definition of weak random sources and the formal definition of extractors emerged from the works of [CG89, Zuc90, Zuc96, NZ96]. The definition of extractors [NZ96] requires quantifying two notions: The first is the amount of randomness in probability distributions, which is measured using a variant of the entropy function called *min-entropy*.

**Definition 1.1** *A distribution $X$ is called a $k$-*source *if the probability it assigns to every element in its range is bounded above by $2^{-k}$. The* min-entropy *of $X$, (denoted by $H_\infty(X)$) is the maximal $k$ such that $X$ is a $k$-source.*

The second notion is the quality of the extracted bits, which is measured using the statistical distance between the extracted bits and truly uniform ones.

**Definition 1.2** *Two distributions $P, Q$, (over the same domain $\Omega$) are $\epsilon$-*close *if they have statistical distance of at most $\epsilon$. (For every event $A \subseteq \Omega$, the probability that the two distributions assign to $A$ differ by at most $\epsilon$).*

Extractors are functions that use *few* truly random bits to extract *many* (almost) truly random bits from arbitrary distributions which "contain" sufficient randomness. A formal definition follows. We use $U_m$ to denote the uniform distribution on $m$ bit strings.

**Definition 1.3** *[NZ96] A function $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $(k, \epsilon)$-*extractor *if for every $k$-source $X$, the distribution $Ext(X, U_d)$, (obtained by running the extractor on an element sampled from $X$ and a uniformly chosen $d$ bit string, which we call the seed) is $\epsilon$-close to $U_m$. The entropy-loss of an extractor is defined to be $k + d - m$.*

Informally, we will say that an extractor uses a seed of length $d$ in order to extract $m$ bits from distributions on $n$ bits which contain $k$ random bits. We refer to the ratio between $m$ and $k$ as the fraction of the randomness which the extractor extracts, and to the ratio between $k$ and $n$ as the entropy rate of the source.

Apart from their original application of obtaining random bits from natural sources, extractors turned out to be useful in many areas in complexity theory and combinatorics, with examples being pseudo-random generators for space bounded computation, deterministic amplification, oblivious sampling, constructive leader election and explicit constructions of expander graphs, super-concentrators and sorting networks. The reader is referred to the excellent survey articles [Nis96, NTS99]. (A more recent survey article that complements the aforementioned articles can be found in [Sha02]).

## 1.2 Extractor constructions: goals and previous work

We now survey some of the goals in extractor constructions and previous research towards achieving these goals. Extractor constructions are measured by viewing $d$ and $m$ as functions of the source parameters ($n$ and $k$) and the required error $\epsilon$. A recent result of [RRV99] enables us to rid ourselves of $\epsilon$, and concentrate on the case that $\epsilon$ is some fixed small constant[1]. We maintain this convention throughout the introduction.

When constructing extractors there are two possible objectives: minimizing the seed length $d$ and maximizing the output size $m$. It should be noted that the existence of an optimal extractor (which optimizes both parameters simultaneously, and matches the known lower bounds due to [RTS00]) can be easily proven using the probabilistic method. Thus, the goal is to match this performance with explicit constructions. A (family of) extractors is *explicit* if it can be computed in polynomial time.[2]

In the remainder of this sub-section we survey the currently known explicit extractors constructions known for the two objectives. (The reader is also referred to [Sha02] for a more recent survey article that covers some subsequent work [TSUZ01, TSZS01, SU01, LRVW03].) Tables 1,2 contain some extractor constructions, but are far from covering the mass of work done in this area. In the following paragraphs we focus on extractors which work for arbitrary min-entropy threshold $k$.

**1. Minimizing the seed length:** A lower bound of $\Omega(\log n)$ on the seed length was given in [NZ96, RTS00]. In contrast to the (non-explicit) optimal extractor which uses a seed of length $O(\log n)$ to extract *all* the randomness from the source, explicit constructions of extractors can optimize the seed length only at the cost of extracting a small fraction of the randomness. For large $k$ ($k = \Omega(n)$) the extractor of [Zuc97] uses a seed of length $O(\log n)$ to extract a constant fraction of the initial randomness. However, for smaller $k$, explicit constructions can only extract a polynomial fraction of the initial randomness, ($m = k^{1-\alpha}$ for an arbitrary constant $\alpha$). This result was first achieved in [Tre01] for $k = n^{\Omega(1)}$, and later extended for any $k$ in [ISW00].

**2. Maximizing the output size:** Current constructions of explicit extractors can extract large fractions of the randomness only at the cost of enlarging the seed length. A general method to increase the fraction extracted at the cost of enlarging the seed length was given in [WZ93]. The best extractors which extract *all* the randomness out of random sources are constructed this way from extractors which extract a constant fraction of the randomness. In light of this, we focus our attention to extractors which extract a constant fraction. The best such explicit extractor is by [RRV02], which uses a seed of length $O(\log^2 n)$.

Concluding this presentation, we stress that while there are explicit constructions which are optimal in any of these two parameters (for arbitrary $k$), the cost is a polynomial loss in the other.

### 1.3 New Results

We give two constructions, each optimal in one of the parameters and losing only a *poly-logarithmic* factor in the other. Thus, both come closer to simultaneously optimizing both parameters. (We remark that in a subsequent work, [LRVW03] give constructions that lose only a *constant* factor in the other parameter.) The results are stated for constant $\epsilon$, (see section 7 for the exact dependence on $\epsilon$). In the first construction we extract any constant fraction of the initial randomness using a seed of length $O(\log n \cdot (\log\log n)^2)$. This improves the best previous such result by [RRV02] which uses a seed of length $O(\log^2 n)$.

[1] [RRV99] gives a general explicit transformation which transforms an extractor with constant error into an extractor with arbitrary small error while harming the other parameters only slightly more than they need to be harmed. The exact dependence of our results on $\epsilon$ is presented in section 7.

[2] More formally, a family $E = \{E_n\}$ of extractors is defined given polynomially computable integer functions $d(n), m(n), k(n), \epsilon(n)$ such that for every $n$, $E_n : \{0,1\}^n \times \{0,1\}^{d(n)} \to \{0,1\}^{m(n)}$ is a $(k(n), \epsilon(n))$-extractor. The family is explicit in the sense that $E_n$ can be computed in time polynomial in $n + d(n)$.

Table 1: Extracting a constant fraction: $m = (1-\alpha)k$ for arbitrary $\alpha > 0$

| reference | min-entropy $k$ | seed length $d$ |
|---|---|---|
| [Zuc97] | $k = \Omega(n)$ | $O(\log n)$ |
| [TS96] | any $k$ | $O(\log^9 n)$ |
| [ISW00] | $k = 2^{O(\sqrt{\log n})}$ | $O(\log n \cdot \log\log\log n)$ |
| [RRV02] | any $k$ | $O(\log^2 n)$ |
| **Thm. 1.4** | any $k$ | $O(\log n \cdot (\log\log n)^2)$ |
| optimal | any $k$ | $O(\log n)$ |

Table 2: Optimizing the seed length: $d = O(\log n)$

| reference | min-entropy $k$ | output length $m$ |
|---|---|---|
| [Zuc97] | $k = \Omega(n)$ | $(1-\alpha)k$ |
| [Tre01] | $k = n^{\Omega(1)}$ | $k^{1-\alpha}$ |
| [ISW00] | $k = 2^{O(\sqrt{\log n})}$ | $\Omega(\frac{k}{\log\log\log n})$ |
| [ISW00] | any $k$ | $k^{1-\alpha}$ |
| **Thm. 1.6** | any $k$ | $\Omega(\frac{k}{\log n})$ |
| optimal | any $k$ | $k$ |

All the results are stated for constant $\epsilon$. $\alpha$ is an arbitrary small constant.

**Theorem 1.4** *For every $n, k$ and constant $\epsilon$, there are explicit $(k,\epsilon)$-extractors $Ext : \{0,1\}^n \times \{0,1\}^{O(\log n \cdot (\log\log n)^2)} \to \{0,1\}^{(1-\alpha)k}$, where $\alpha > 0$ is an arbitrary constant.*[3]

Using [WZ93], we get the following corollary[4], which also improves the previous best construction which extract all the randomness by [RRV02].

**Corollary 1.5** *For every $n, k$ and constant $\epsilon$, there are explicit $(k,\epsilon)$-extractors $Ext : \{0,1\}^n \times \{0,1\}^{O(\log n \cdot (\log\log n)^2 \cdot \log k)} \to \{0,1\}^k$.*

Our second construction uses the optimal seed length (that is $O(\log n)$) to extract $m = \Omega(k/\log n)$ bits, this improves the best previous result by [ISW00] which could only extract $m = k^{1-\alpha}$ bits.

**Theorem 1.6** *For every $n, k$ and constant $\epsilon$, there are explicit $(k,\epsilon)$-extractors $Ext : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Omega(k/\log n)}$.*

Using [ISW00], we get the following corollary, (in which the "loss" depends only on $k$)[5].

**Corollary 1.7** *For every $n, k$ and constant $\epsilon$, there are explicit $(k,\epsilon)$-extractors $Ext : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Omega(k/(\log k \cdot \log\log k))}$.*

[3] We remark that our construction requires $k \geq 8\log^5 n$. Nevertheless, Theorem 1.4 follows as stated because the case of $k < 8\log^5 n$ was already covered in [ISW00]. This is also the case in the next Theorems.

[4] In fact, the version of Theorem 1.4 stated above does not suffice to conclude the corollary. To use the method of [WZ93] we need a version where the error is $1/polylogn$ which follows just the same from our analysis, see section 7.

[5] [ISW00] show that an extractor $Ext : \{0,1\}^{k^{O(1)}} \times \{0,1\}^d \to \{0,1\}^m$ can be used to construct an extractor $Ext : \{0,1\}^n \times \{0,1\}^{d+O(\log n)} \to \{0,1\}^{\Omega(m/\log\log k)}$ for any $n$.

## 1.4 Condensers

The main technical contribution of this paper are constructions of various "condensers".[6] A condenser is a generalization of an extractor.

**Definition 1.8** *A $(k, k', \epsilon)$-condenser is a function $Con : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{n'}$, such that for every $k$-source $X$ of length $n$, the distribution $Con(X, U_d)$ is $\epsilon$-close to a $k'$-source.*

Note that an extractor is a special case of a condenser, when $n' = k'$. Condensers are most interesting when $k'/n' > k/n$ (that is the entropy rate of the output distribution is larger than that of the initial distribution). Such condensers can be used to construct extractors via repeatedly condensing the source until an entropy rate 1 is achieved. In this paper we give a new construction of condensers. One possible setting of the parameters gives that for constant $\epsilon > 0$ and every $n$ and $k$ with $k \geq 8 \log^5 n$ there is an explicit $(k, \Omega(k), \epsilon)$-condenser $Con : \{0,1\}^n \times \{0,1\}^{O(\log n \cdot \log \log n)} \to \{0,1\}^{k \log n}$. The exact parameters can be found in Section 5.

Somewhat surprisingly, an important component in the construction of the condenser above is a condenser that does not condense at all! That is one in which the entropy rate of the output distribution is *smaller* than that of the initial distribution. The usefulness of such objects in constructing extractors is demonstrated in [NZ96] (see also [SZ98, Zuc97]). We refer to such condensers as "block extraction schemes" and elaborate on their role in extractors (and condensers) constructions in Section 1.6. In this paper we give a construction of a block extraction scheme that uses a seed of length $O(\log \log n)$. The exact parameters can be found in Section 3.

## 1.5 Transforming general extractors into strong extractors

Speaking informally, a strong extractor is an extractor in which the output distribution is independent of the seed[7]. In some applications of extractors it is beneficial to have the strong version. Most extractor constructions naturally lead to strong extractors, yet some (with examples being [TS96, ISW00] and the constructions of this paper), are not strong or difficult to analyze as strong. We solve this difficulty by giving a general explicit transformation which transforms any extractor into a strong extractor with essentially the same parameters. Exact details are given in section 8.

## 1.6 Technique

In this section we attempt to explain the main ideas in this paper without delving into technical details.

### High level overview

In contrast to the latest papers on extractors [Tre01, RRV02, ISW00] we do not use Trevisan's paradigm. Instead, we revisit [NZ96] and attempt to construct block sources, (A special kind of sources which allow very efficient extraction). Following [SZ98, NTS99], we observe that when failing to produce a block source the method of [NZ96] "condenses" the source. This enables us to use a "win-win" case analysis as in [ISW99, ISW00] which eventually results in a construction of a

[6] The notion of condensers was also used in [RR99]. While similar in spirit, that paper deals with a completely different set of parameters and uses different techniques. The notion of condensers also comes up in subsequent work [TSUZ01, CRVW02, LRVW03] as a tool for constructing extractors and expander graphs.

[7] In the original paper [NZ96] strong versions of extractors were defined and constructed, and the notion of a "non-strong" extractor was later given by Ta-Shma, [TS96].

condenser. Our extractors are then constructed by repeated condensing. A critical component in obtaining improved parameters is derandomizing the construction of block-sources of [NZ96].

### Block sources

We begin by describing a special kind of sources called block-sources (a precise definition appears in Definition 2.3) that allow extractors with very efficient parameters. Consider a source distribution $X$ that is composed of two independent concatenated distributions $X = (X_1, X_2)$, where $X_1$ is a $k_1$-source and $X_2$ is a $k_2$-source. Extractors for this special scenario (which are called block-source extractors) can be constructed by composing two extractors: An extractor with optimal seed length can be used to extract random bits from $X_2$, and these bits, (being independent of $X_1$), can be used as seed to extract all the randomness from $X_1$ using an extractor with large output. (Note, that with today's best extractors this argument uses the optimal seed length to extract all the randomness from $X_1$, as long as $k_2$ is at least $polylog(n)$). The requirement that $X_1$ and $X_2$ be independent could be relaxed in the following way, (that was suggested in [CG89]): Intuitively, it is sufficient that $X_1$ "contains" $k_1$ random bits, and $X_2$ "contains" $k_2$ random bits even conditioned on $X_1$. Such sources are called block-sources[8]. Thus, extracting randomness from general sources can be achieved by giving a construction which uses few random bits to transform a general source into a block source, and using a block-source extractor.

This approach was suggested by Nisan and Zuckerman in [NZ96]. They constructed a "block extraction scheme". That is a condenser that given an arbitrary source $X$, uses few random bits to produce a new source $B$ (called a block) which is shorter than the initial source, and contains a large fraction of the initial randomness. This means that the distribution $(B, X)$ meets the first requirement of block sources: The first block contains randomness. Intuitively, to meet the second requirement one should give an upper bound on the amount of randomness contained in $B$, and conclude that there is some randomness in $X$ which is not contained in $B$. However, in the construction of Nisan and Zuckerman such an upper bound can be achieved only "artificially" by setting the parameters so that the length of $B$ is smaller than $k$. This indeed gives that $B$ did not "steal all the entropy" in the source. However, this approach has a costly effect. In the analysis of Nisan and Zuckerman the amount of randomness that is guaranteed to be in $B$ is proportional to its length. Thus, decreasing the length of $B$ reduces the amount of entropy that can be guaranteed. In particular, when $k < \sqrt{n}$ choosing the length of $B$ this way, it may be the case that $B$ contains no randomness. As a result, the extractors of [NZ96] do not work when $k < \sqrt{n}$.

### Condensing by a "win-win" analysis

A way to go around this problem was suggested in [SZ98, NTS99]. We now explain a variant of the argument in [NTS99] that we use in our construction. Recall that we want to obtain a large block $B$ that does not "steal all the entropy". An important observation is that the case in which the block-extraction scheme fails to produce such a block is also good in the sense that it means that $B$ "stole all the entropy from the source". As $B$ is shorter than the initial source, it follows that it is more condensed. We now explain this approach in more detail.

Suppose we use the block extraction scheme to produce a large block $B$ (say of length $n/2$) and consider the distribution $(B, X)$. Recall that for our purposes to get a useful block-source, it suffices that $X$ contains very few random bits that are not contained in $B$. An important observation is

[8]More precisely, a $(k_1, k_2)$-block source is a distribution $X = (X_1, X_2)$ such that $X_1$ is a $k_1$-source, and for every possible value $x_1$ of $X_1$, the distribution of $X_2$, conditioned on the event that $X_1 = x_1$, is a $k_2$-source.

that when the procedure above fails to construct a block source, this happens because (almost) all the randomness "landed" in $B$. In this case, we obtained a block that is more condensed than the initial source $X$. (It has roughly the same amount of randomness and half the length).

Using this idea repeatedly, at each step either we construct a block source, (from which we can extract randomness), or we condense the source. There is a limit on how much we can condense the source. Certainly, when the length reduces below the original entropy $k$, no further condensing is possible. This means that running this procedure repeatedly enough times we eventually obtain a block source.

The outcome of the procedure above are several "candidate" distributions, where one of them is a block source. Not knowing which is the "right one", we run block source extractors on all of them, (using the same seed). We know that one of the distributions we obtain is close to uniform. It turns out that the number of candidate distributions is relatively small (about $\log(n/k)$. Consider the distribution obtained by concatenating the output distributions of the block-source extractors. This distribution contains a portion which is (close to) uniformly distributed on roughly $k$ bits and thus has entropy close to that of the initial source. Moreover, the distribution is on strings of length not not much larger than $k$ (the length is roughly $k \log(n/k)$). We conclude that the aforementioned distribution is more condensed than the initial source, and that the procedure described above is a condenser!

We can now obtain an extractor construction by repeatedly condensing the source (using fresh randomness in each iteration) until it becomes close to uniform. However, it turns out that the parameters of the constructed condenser are not good enough to yield a good extractor. Our actual construction of condensers is achieved by using the procedure above with an improved version of the block extraction scheme of Nisan and Zuckerman.

**Improved block extraction**

Let us delve into the parameters. The block extraction scheme of Nisan and Zuckerman spends $O(\log n)$ random bits when producing a block $B$ of length $n/2$, and can guarantee that $B$ is an $\Omega(k/\log(n/k))$-source. This turns out to be too costly to run our condenser construction.

One problem is that the number of random bits used by the block extraction scheme is too large for our purposes. Note that the block extraction scheme already spends $O(\log n)$ random bits. Using the strategy described above we will need to run it roughly $\log(n/k)$ times, resulting in a large seed length ($\log n \cdot \log(n/k)$). We overcome this problem by derandomizing the construction of Nisan and Zuckerman. We reduce the number of random bits used from $\log n$ to $\log\log n$, allowing us to run it a larger number of times while still obtaining short seed length.

A second problem is that we want the block $B$ to contain a constant fraction of the initial randomness $k$. The analysis of Nisan and Zuckerman only guarantees that the block $B$ contains $\Omega(k/\log(n/k))$ random bits. Note that while this quantity is smaller than we want, it does achieve the goal when $k$ is a constant fraction of $n$. We introduce another modification to the construction of Nisan and Zuckerman in order to increase the randomness in $B$. The idea is to reduce the case of $k = o(n)$ into the case of $k = \Omega(n)$. This is done by error correcting the source prior to using the block extraction scheme. We give a non-constructive argument to show that every error corrected random source has a "piece" of length $k$ which is an $\Omega(k)$-source. When analyzing the scheme we use the analysis of Nisan and Zuckerman on this piece. Intuitively, this enables the analysis of the block extraction scheme to be carried out on the condensed piece, where it performs best.

### 1.7 Organization of the paper

Section 2 includes some preliminaries. In section 3 we construct a block extraction scheme. In section 4 we use the method of [NTS99] to show that when using the block extraction scheme either we get a block source or we condense the source. In section 5 we run the block extraction scheme recursively and obtain condensers. In section 6 we use the condensers to construct extractors. Section 7 gives the exact dependence of our extractors on the error parameter $\epsilon$. In section 8 we show how to transform arbitrary extractors into strong extractors.

## 2 Preliminaries

### 2.1 Probability distributions

Given a probability distribution $P$ we use the notation $\Pr_P[\cdot]$ to denote the probability of the relevant event according to the distribution $P$. We sometimes fix some probability space (namely a set $\Omega$ and a distribution over $\Omega$) and then use the notation $\Pr[\cdot]$ to denote the probability of events in this probability space. We use $\Pr[E_1, E_2]$ to denote $\Pr[E_1 \cap E_2]$.
We need the following notion of "collision probability".

**Definition 2.1** *For a distribution $P$ over $\Omega$, define $C(P) = \sum_{\omega\in\Omega} P(\omega)^2$. In words, $C(P)$ is the probability that two independent samples from $P$ gave the same outcome. We refer to $C(P)$ as the collision probability of $P$.*

We need the following useful Lemma.

**Lemma 2.2** *Let $V$ be $\rho$-close to a $k$-source. Define $L = \{v | \Pr[V = v] > 2^{-(k-1)}\}$. It follows that $\Pr(V \in L) < 2\rho$.*

**Proof:** Let $V'$ be a $k$-source such that $V$ and $V'$ are $\rho$-close. We have that $|\Pr_V(L) - \Pr_{V'}(L)| < \rho$. However, $V'$ assigns small probability to all elements in $L$, whereas $V$ assigns large probability to these elements. This gives that $\Pr_V(L) - \Pr_{V'}(L) > \Pr_{V'}(L)$, Which means that $\Pr_{V'}(L) < \rho$. Using the inequality above we get that $\Pr_V(L) < 2\rho$. □

### 2.2 Block sources

Block sources are random sources which have a special structure. The notion of block sources was defined in [CG89].

**Definition 2.3** *[CG89] Two random variables $(X_1, X_2)$ form a $(k_1, k_2)$-block source if $X_1$ is a $k_1$-source, and for every possible value $x_1$ of $X_1$ the distribution of $X_2$, given that $X_1 = x_1$, is a $k_2$-source.*

Block source extractors are extractors which work on block sources.

**Definition 2.4** *A $(k, t, \epsilon)$-block source extractor is a function $Ext : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^d \to \{0,1\}^m$, such that for every $(k, t)$-block source $(X_1, X_2)$, (where $X_1, X_2$ are of length $n_1$,$n_2$ respectively), the distribution $Ext(X_1, X_2, U_d)$ is $\epsilon$-close to $U_m$.*

Block sources allow the following composition of extractors.

**Lemma 2.5** *(implicit in [NZ96] and appears in [SZ98]) If there exist an explicit $(k, \epsilon_1)$-extractor $Ext_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^m$, and an explicit $(t, \epsilon_2)$-extractor $Ext_2 : \{0,1\}^{n_2} \times \{0,1\}^{d_2} \to \{0,1\}^{d_1}$, then there exists an explicit $(k, t, \epsilon_1 + \epsilon_2)$-block-source extractor $Ext : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^{d_2} \to \{0,1\}^m$, where $Ext(x_1, x_2; y) = Ext_1(x_1, Ext_2(x_2, y))$.*

Following [NZ96, SZ98, Zuc97] we can use the above Theorem to compose two extractors: one which optimizes the seed length and another which optimizes the output length. The resulting block-source extractor will "inherit" the nice properties of both its component extractors. Particularly, taking $Ext_1$ to be the extractor of [RRV02] and $Ext_2$ to be the extractor of [ISW00], we get the following block-source extractor:

**Corollary 2.6** *For every integers $n_1 \leq n_2$, $k$ and $t \geq \log^4 n_1$ there is an explicit $(k, t, \frac{1}{n_1} + \frac{1}{n_2})$-block source extractor $BE : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \times \{0,1\}^{O(\log n_2)} \to \{0,1\}^k$.*

Thus, to construct extractors which achieve short seed length and large output simultaneously, it suffices to use few random bits, and convert any $k$-source into a $(k', \log^4 n)$-block source such that $k'$ is not much smaller than $k$.

This turns out to be a tricky problem. No such (efficient in terms of random bits spent) scheme is known when $k < \sqrt{n}$.

## 2.3 Error correcting codes

Our construction uses error correcting codes.

**Definition 2.7** *An error-correcting code with distance $d$ is a function $EC : \{0,1\}^n \to \{0,1\}^{\bar{n}}$ such that for every $x_1, x_2 \in \{0,1\}^n$ such that $x_1 \neq x_2$, $d_{Hamming}(EC(x_1), EC(x_2)) \geq d$. (Here $d_{Hamming}(z_1, z_2)$ denotes the Hamming distance between $z_1, z_2$). An error-correcting code is explicit if $EC$ can be computed in polynomial time.*

We use the following construction of error correcting codes.

**Theorem 2.8** *[Jus72] There exist constants $0 < b < a$ and an explicit error correcting code $EC : \{0,1\}^n \to \{0,1\}^{an}$ with distance $bn$.*

## 2.4 Almost $l$-wise independent distributions

We use the following notion of efficiently constructible distributions.

**Definition 2.9** *Call a distribution $P$ on $n$ bits, polynomially constructible using $u(n)$ bits*[9]*, if there exists an algorithm $A : \{0,1\}^{u(n)} \to \{0,1\}^n$ which runs in time polynomial in $n$, such that the distribution $A(Y)$ where $Y$ is chosen uniformly from $\{0,1\}^{u(n)}$ is identical to $P$.*

Naor and Naor [NN93] defined "almost $l$-wise independent distribution".

**Definition 2.10** *([NN93]) A distribution $(P_1, \cdots, P_n)$ over $\{0,1\}^n$ is said to be $(\epsilon, l)$-wise dependent with mean $p$ if for every subset $\{i_1, \cdots, i_l\}$ of $[n]$, the distribution $(P_{i_1}, \cdots, P_{i_l})$ is $\epsilon$-close to the distribution over $l$ bit strings where all bits are independent and each of them takes the value 1 with probability $p$.*

[9]Naturally, one should speak about a sequence $P = \{P_n\}$ of distributions for this to make sense.

Naor and Naor showed that almost $l$-wise independent distributions can be constructed using very few random bits.

**Theorem 2.11** *[NN93] For every $n, l$ and $\epsilon$, an $(\epsilon, l)$-wise dependent distribution with mean $1/2$ is polynomially constructible using $O(\log\log n + l + \log(1/\epsilon))$ bits.*

We require distributions that are almost $l$-wise independent with mean different than $1/2$. Nevertheless, it is very easy to construct such distributions from Theorem 2.11.

**Corollary 2.12** *For every $n, \epsilon$ and $q$, an $(\epsilon, 2)$-wise dependent distribution with mean $2^{-q}$ is polynomially constructible using $O(\log\log n + q + \log(1/\epsilon))$ bits.*

**Proof:** We use Theorem 2.11 to construct a distribution that is $(\epsilon, 2q)$-wise dependent with mean $1/2$ over $qn$ bits. Note that this costs $O(\log\log(qn) + q + \log(1/\epsilon)) = O(\log\log n + q + \log(1/\epsilon))$ bits. We denote these bits by $P_1, \ldots, P_{qn}$. We divide them into $n$ blocks of length $q$ and define $n$ bits $P'_1, \ldots, p'_n$ as follows: $P'_i$ is set to 'one' if and only if all the bits in the $i$'th block are 'one'. In particular each $P'_i$ is a function of the bits in the $i$'th block. It follows that the distribution $P'_1, \ldots, P'_n$ is $(\epsilon, 2)$-wise dependent. $\square$

Given $(X_1, \cdots, X_n)$ that form a $(0, 2)$-wise dependent distribution, Chebyshev's inequality gives that for every $0 < \lambda < 1$

$$\Pr(|\sum_{1 \le i \le n} X_i - pn| > \lambda pn) < \frac{1}{\lambda^2 pn}$$

The same argument can be applied to $(\epsilon, 2)$-wise dependent distributions and gives that:

**Lemma 2.13** *[NN93] If $(X_1, \cdots, X_n)$ is a $(\epsilon, 2)$-wise dependent distribution with mean $p$, then for every $0 < \lambda < 1$*

$$\Pr(|\sum_{1 \le i \le n} X_i - pn| > \lambda pn) < O(\frac{1}{\lambda^2 pn} + \sqrt{\epsilon})$$

*as long as $\epsilon < \frac{\lambda^4 p^4}{25}$.*

# 3 Improved block extraction

Some constructions of block sources from general sources [NZ96, SZ98, Zuc97] rely on a building block called a "block extraction scheme". In our terminology a block extraction scheme is a condenser. Nevertheless, we choose to redefine this notion as it is more convenient to present the parameters in a different way.

**Definition 3.1** *Let $n, k$ and $r$ be integers and $\rho, \gamma$ be numbers. A function $B : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{n/r}$ is a $(k, \rho, \gamma)$-block-extraction scheme if it is a $(k, \gamma \cdot \left(\frac{k}{r}\right), \rho)$-condenser.*

In words, a block extraction scheme takes a $k$-source of length $n$ and uses a seed to produce a distribution (which we call a block) of length $n/r$. The parameter $\gamma$ measures the entropy rate of the new distribution in terms of the entropy rate of the initial distribution. For example when $\gamma = 1$ this means that the two distributions have the same rate and the block-extraction scheme "preserves" the entropy rate in the source. In this section, we discuss constructions in which $\gamma < 1$ meaning that the entropy rate in the output distribution is *smaller* than that of the initial distribution.

Using this notation, Nisan and Zuckerman proved the following Theorem:

**Lemma 3.2** *[NZ96] There exists a constant $c > 0$ such that for every $n, k, r$ and $\rho \geq 2^{-ck/r}$ there is an explicit $(k, \rho, \Omega(\frac{1}{\log \frac{n}{k}}))$-block extraction scheme $B : \{0,1\}^n \times \{0,1\}^{O(\log n \log \frac{1}{\rho})} \to \{0,1\}^{n/r}$.*

We prove the following Lemma that improves Lemma 3.2 for some choice of parameters, namely for $1 < r \leq \log n$ and $\rho = 1/\log^{O(1)} n$.

**Lemma 3.3** *There exists a constant $c > 0$ such that for every $n, k, r$ and $\rho \geq c\sqrt{\frac{r}{k}}$ there is an explicit $(k, \rho, \Omega(1))$-block extraction scheme $B : \{0,1\}^n \times \{0,1\}^{O(\log \log n + \log(1/\rho) + \log r)} \to \{0,1\}^{n/r}$.*

Lemma 3.3 improves Lemma 3.2 in two respects (as long as one settles for small $r \leq \log n$ and large $\rho > 1/\log^{O(1)} n$).

1. We reduce the number of random bits spent by the block extraction scheme. In [NZ96] the number of random bits is logarithmic in $n$, whereas in Lemma 3.3 the number of random bits is double logarithmic in $n$.

   This is achieved by derandomizing the proof of Nisan and Zuckerman using almost $l$-wise independent distributions. In section 3.1 we describe the property that a distribution needs to have in order to allow the Nisan-Zuckerman analysis, and construct a small sample space with this property.

2. We increase the amount of randomness guaranteed in the output block. In [NZ96] the amount of randomness guaranteed in the output block $B$ is $\Omega(\frac{k}{r \log(n/k)})$. Lemma 3.3 guarantees that $B$ contains $\Omega(\frac{k}{r})$ random bits.

   Note that the two quantities are the same when $k = \Omega(n)$. Indeed, our improvement is achieved by reducing the case of $k = o(n)$ to that of $k = \Omega(n)$. We start from a $k$-source with $k = o(n)$. In section 3.3 we show that once a random source is error corrected, there are some $k$ indices, (to the error corrected source) which induce an $\Omega(k)$-source. This induced source has a constant entropy rate. When applying the argument of Nisan and Zuckerman our analysis concentrates on this source which allows us to guarantee that the block contains more randomness. The exact analysis is given in section 3.4.

The remainder of this section is devoted to proving Lemma 3.3.

## 3.1 The analysis of Nisan and Zuckerman

The block extraction scheme of Nisan and Zuckerman is obtained by restricting the source to some subset of the indices which is selected using few random bits. More precisely, they construct a small sample space of subsets of $[n]$ (having a property that we immediately describe) and prove that the distribution obtained by sampling an element from a $k$-source and restricting it to the indices in a random set from the sample space contains a large fraction of the initial randomness. In this section we construct a smaller such sample space which enables us to spend less random bits to construct a block extraction scheme. We now describe the property used by Nisan and Zuckerman. Intuitively, a $k$-source has $k$ random bits "hidden" somewhere.

**Definition 3.4** *Let $n, k, r$ and $\delta$ be parameters. A distribution $S$ over subsets of $[n]$ is called $r$-intersecting for sets of size $k$ with error probability $\delta$ if for every $G \subseteq [n]$ with $|G| \geq k$, $\Pr_S(|S \cap G| < \frac{k}{8r}) < \delta$.*

The following is implicit in [NZ96].

**Lemma 3.5** *[NZ96] There exists some constant $c > 0$ such that if $X$ is a $k$-source on $\{0,1\}^n$ and $S$ is a distribution over subsets of $[n]$ which is $r$-intersecting for sets of size $\frac{ck}{\log(n/k)}$ with error probability $\delta$ then the distribution $X|_S$ (obtained by sampling $x$ from $X$ and $s$ from $S$ and computing $x|_s$) is $(4\sqrt{\delta} + 2^{-\Omega(k)})$-close to a $\Omega(\frac{k}{r\log(n/k)})$-source.*

Nisan and Zuckerman use a construction based on $O(\log(1/\delta))$-wise independence to prove the following Lemma.

**Lemma 3.6** *[NZ96] For every $n, k, r$ and $\delta > 2^{-O(k/r)}$ there is a distribution over subsets of $[n]$ that are of size $n/r$ and this distribution is $r$-intersecting for sets of size $k$ with error probability $\delta$. Furthermore, the distribution is polynomially constructible using $O(\log n \cdot \log(1/\delta))$ bits.*

Using Lemma 3.5 this immediately implies the block extraction scheme of Lemma 3.2. We will be mostly interested in the case when $r$ is small, (say $r \le \log n$) and $\delta$ is large, (say $\delta \ge \log^{-O(1)} n$). For this setup we can save random bits and make the dependence on $n$ double logarithmic.

**Lemma 3.7** *There exists a constant $c > 0$ such that for every $n, k, r$ and $\delta > cr/k$, there is a distribution over subsets of $[n]$ that are of size $n/r$ and this distribution is $r$-intersecting for sets of size $k$ with error probability $\delta > 0$. Furthermore, the distribution is polynomially constructible using $O(\log\log n + \log r + \log(1/\delta))$ bits.*

We prove this Lemma in the following subsection.

## 3.2 A small sample space for intersecting large sets

We now prove Lemma 3.7. We view distributions over $n$ bit strings as distributions over subsets of $[n]$. More specifically, we identify the $n$ bit values $(W_1, \cdots, W_n)$ with the set $\{i | W_i = 1\}$. We construct random variables $(W_1, \cdots, W_n)$ with the following properties:

- For every $1 \le i \le n$, $\Pr(W_i = 1) \approx 1/2r$.
- For every set $G \subseteq [n]$ with $|G| \ge k$, the probability that the sum of the $W_i$'s for $i \in G$ is far from the expected $|G|/2r$ is small. (It is important to note that we allow the "small probability" to be quite large, since we are shooting for large $\delta$).

Note that the second condition gives both the intersecting property and the fact that the selected sets are rarely of size larger than $n/r$ (by considering $G = [n]$). We are interested in constructing such a distribution using as few as possible random bits. A pairwise independent distribution has these properties but takes $\log n$ random bits to construct. We can do better by using the "almost $l$-wise independent" distributions of [NN93].

**Construction 3.8** *Let $q$ be an integer such that $1/4r < 2^{-q} \le 1/2r$, and $\epsilon = min(c\delta^2, c/r^4)$ where $c > 0$ is a constant that will be determined later. Let $W = (W_1, \cdots, W_n)$ be the $(\epsilon, 2)$-wise dependent distribution with mean $2^{-q}$ guaranteed in corollary 2.12. Note that this requires $O(\log\log n + \log r + \log(1/\epsilon))$ random bits.*

The next Lemma follows:

**Lemma 3.9** *There exist constants $c, c' > 0$ such that when using construction 3.8 with the constant $c$, then for every $\delta > c' \cdot r/k$ the distribution $W$ has the following properties:*

1. *For every set $G \subseteq [n]$ such that $|G| \geq k$, $\Pr(\sum_{i \in G} W_i < \frac{k}{8r}) < \delta/3$.*
2. $\Pr(\sum_{1 \leq i \leq n} W_i > \frac{n}{r}) < \delta/3$.

**Proof:** We use Lemma 2.13 to deduce both parts of the Lemma and we use the fact that the $W_i$'s are $(\epsilon, 2)$ dependent with mean $p = 1/2^q$ where $1/4r < p \leq 1/2r$. We start by proving the first part. For this part, we set $\lambda = 1/2$ and assume without loss of generality that $|G| = k$. Note that:

$$\{\sum_{i \in G} W_i < \frac{k}{8r}\} \subseteq \{\sum_{i \in G} W_i < \frac{k}{2 \cdot 2^q}\} \subseteq \{|\sum_{i \in G} W_i - \frac{k}{2^q}| > \frac{\lambda k}{2^q}\}$$

Thus, it suffices to bound the probability of the latter event. To meet the condition in Lemma 2.13 we need to make sure that $\epsilon < \frac{\lambda^4 p^4}{25} = \Theta(1/r^4)$. The requirement that $\epsilon < c/r^4$ takes care of this condition by choosing a sufficiently small small constant $c > 0$. Applying Lemma 2.13 we get that the probability of deviation from the mean is bounded by $O(r/k + \delta\sqrt{c})$. We have that $r/k \leq \delta/c'$. We can set $c'$ to be large enough so that $O(r/k + \delta\sqrt{c}) \leq \delta/6 + O(\delta\sqrt{c})$. This is bounded from above by $\delta/3$ for small enough $c > 0$.

The proof of the second item is similar. We choose $\lambda = 1$ and note that:

$$\{\sum_{1 \leq i \leq n} W_i > \frac{n}{r}\} \subseteq \sum_{1 \leq i \leq n} W_i > \frac{2n}{2^q}\} \subseteq \{|\sum_{1 \leq i \leq n} W_i - \frac{n}{2^q}| > \frac{\lambda n}{2^q}\}$$

Using the fact that $n \geq k$ we can repeat the computations above and conclude that the probability of this event is also bounded by $O(\delta\sqrt{c})$. The Lemma follows by choosing a sufficiently small $c > 0$. □

We are ready to prove Lemma 3.7

**Proof:** (of Lemma 3.7) The first item of Lemma 3.9 shows that $W$ is a distribution over subsets of $[n]$ that is $r$-intersecting for sets of size $k$ with error probability $\delta/3$. The second item shows that $W$ could be transformed into a distribution over subsets of size exactly $n/r$ without changing it by much. This change is done by adding arbitrary indices to the set if its size is smaller than $n/r$ and deleting arbitrary indices if its size is larger than $n/r$. Adding indices will not spoil the intersecting property, and the probability that we need to delete indices is bounded by $\delta/3$. The Lemma follows. □

### 3.3 Error corrected random sources

In this subsection we develop another tool required for the proof of Lemma 3.3 and show that if we apply an error correcting code to an arbitrary $k$-source, we obtain a $k$-source which has $k$ indices which induce an $\Omega(k)$-source.

In the remainder of this section we fix $a, b$ and $EC$ to be as in Theorem 2.8. For a vector $x \in \{0,1\}^n$ and a set $T \subseteq [n]$ we use $x|_T$ to denote the restriction of $x$ to $T$.

**Lemma 3.10** *Let $X$ be a $k$-source on $\{0,1\}^n$. There exists a set $T \subseteq [an]$ of size $k$, such that $EC(X)|_T$ is an $\Omega(k)$-source.*

Lemma 3.10 is an immediate corollary of lemma 3.11 which was mentioned to us by Russell Impagliazzo. A very similar argument also appears in [SZ98].

**Lemma 3.11** *[Imp99] Let $X$ be a $k$-source on $\{0,1\}^n$. For every $v$, there exists a set $T \subseteq [an]$ of size $v$, such that $EC(X)|_T$ is a $\frac{1}{2} \cdot \log 1/(2^{-k} + (1-\frac{b}{a})^v)$-source.*

The following fact states that if a distribution has low collision probability then it has high min-entropy. This follows because a distribution with low min-entropy has an element which gets large probability. This element has a large chance of appearing in two consecutive independent samples.

**Fact 3.12** *If $C(P) \leq 2^{-k}$ then $P$ is a $k/2$-source.*

Our goal is showing that there exists a subset of $[an]$ on which the error corrected source has low collision probability. We will show that a random (multi)-set has this property.
**Proof:** (of Lemma 3.11) Consider the following probability space: $X_1, X_2$ are independently chosen from the distribution $X$, and $T = (I_1, .., I_v)$ is chosen independently where each $I_j$ is uniformly distributed in $[an]$. Consider the following event: $B = \{EC(X_1)|_T = EC(X_2)_T\}$. We first show that the probability of $B$ is small.

$$\Pr(B) = \Pr(B|X_1 = X_2)\Pr(X_1 = X_2) + \sum_{a_1 \neq a_2} \Pr(B|X_1 = a_1, X_2 = a_2)\Pr(X_1 = a_1, X_2 = a_2) \quad (1)$$

$X$ is a $k$-source, and therefore $\Pr(X_1 = X_2) \leq 2^{-k}$. For given $a_1 \neq a_2$, we know that the distance between $EC(a_1)$ and $EC(a_2)$ is at least $bn$. Thus, any of the $I_j$'s has a chance of $\frac{b}{a}$ to "hit" a coordinate where $EC(a_1)$ and $EC(a_2)$ disagree. Having chosen $v$ such coordinates the probability that none of them differentiated between $EC(a_1)$ and $EC(a_2)$ is bounded by $(1-\frac{b}{a})^v$. Plugging this in 1 we get that

$$\Pr(B) \leq 2^{-k} + (1-\frac{b}{a})^v$$

In the sample space we considered, $T$ was chosen at random. Still, by averaging there is a fixing $T'$ of the random variable $T$ for which the above inequality holds. For this $T'$ we have that the probability that independently chosen $X_1$ and $X_2$ have $EC(X_1)|_{T'} = EC(X_2)|_{T'}$ is small. In other words we have that

$$C(EC(X)|_{T'}) \leq 2^{-k} + (1-\frac{b}{a})^v$$

The Lemma immediately follows from fact 3.12. □

## 3.4 Construction of block extraction scheme

In this subsection we put everything together and prove Lemma 3.3. We are ready to define our block extraction scheme. Recall that $EC$ is the error correcting code from Theorem 2.8 and $a$ and $b$ are the constants which existence is guaranteed by that Theorem.

**Construction 3.13 (block extraction scheme)** *Given $n, k, r, \rho$ and a constant $e$ (which will be fixed later) let $d = O(\log\log n + \log r + \log(1/\rho))$ be the number of bits used by Lemma 3.7 to construct a distribution over subsets of $[an]$ that is $ar$-intersecting for sets of size $ek$ with error probability $(\frac{\rho}{4})^2$. For $y \in \{0,1\}^u$, let $S_y$ be the set defined by $y$ in the intersecting distribution. We now define:*

$$B(x, y) = EC(x)|_{S_y}$$

We are finally ready to prove Lemma 3.3.

**Proof:** (of Lemma 3.3) Let $V$ denote the distribution $EC(X)$. Lemma 3.10 implies that there exists a set $T \subseteq [an]$ of size $k$ such that $V|_T$ is a $\eta k$-source, (for some constant $\eta > 0$). Consider the distribution $S \cap T$, (the restriction of the intersecting distribution to the coordinates of $T$). Note that the distribution $S \cap T$ is a distribution over subsets of $T$. We claim that it has the same intersecting properties of $S$. Namely, that $S \cap T$ is $ar$-intersecting for sets of size $ek$ with error probability $(\frac{\rho}{4})^2$). (This follows from the definition as every subset $G \subseteq T$ is in particular a subset of $[n]$). We now use Lemma 3.5 on the source $V|_T$ using the intersecting distribution $S \cap T$. Let us first check that the conditions of Lemma 3.5 are met. We fix the constant $e$ of construction 3.13, setting $e = (c\eta)/(-\log \eta)$, where $c$ is the constant from Lemma 3.5. The conditions of Lemma 3.5 are met since $V|_T$ is a $\eta k$-source of length $k$ and we have a distribution which is intersecting sets of size $ek = \frac{c(\eta k)}{\log(k/(\eta k))}$. We conclude from the lemma that $V|_{S \cap T}$ is $\rho$-close to an $\Omega(k/r)$-source. We now claim that $V|_S$ is $\rho$-close to an $\Omega(k/r)$-source. This is because adding more coordinates cannot reduce the entropy. The lemma follows as we have shown that $B(X, U_d) = V|_S$ is $\rho$-close to an $\Omega(k/r)$-source. □

# 4 Partitioning to two "good" cases

Let $B$ be the block extraction scheme constructed in the previous section and let $X$ be a $k$-source. We consider the distribution $(B(X,Y), X)$ (where $Y$ is a random seed that is independent of $X$). Following the intuition explained in Section 1.6 we'd like to argue that for every $k$-source $X$ at least one of the following good cases happen:

- $(B(X,Y), X)$ is (close to) a block source.
- $B(X,Y)$ is (close to) having higher entropy rate than $X$. That is $B(X,Y)$ is more condensed than $X$.

In this section we show that although the statement above does not hold as stated, we can prove a more technical result with the same flavor.

**Remark 4.1** *Here is a counterexample for the statement above assuming $k \le n/2$. To make the example more simple we assume that the block extraction scheme does not error correct the source prior to choosing a subset. Consider a source $X$ which tosses a random bit $b$ and depending on the outcome decides whether to sample from a distribution $X_1$ in which the first $k$ bits are random and the remaining $n-k$ bits are fixed, or from a distribution $X_2$ that is $k$-wise independent. $X_1$ corresponds to the first good case (and yields a block source) as $B$ is expected to hit about $k/2$ random bits. $X_2$ corresponds to the second (and yields a condensed block) as the block that $B$ outputs contains $n/2$ bits and thus "steals all the randomness". However, the "combination distribution" $X$ doesn't satisfy any of the two good cases.*

A way of going around this problem was devised in [NTS99]. The idea is to argue that the example in the remark above is the "worst possible" and show that any source $X$ can be partitioned into two sources where for each one of them one of the good cases holds. To make this formal, we introduce the following notation:

**Definition 4.2 (conditioning random variables)** *Fix some probability space. Let $X$ be a random variable and $E$ be an event with positive probability. We define the probability distribution of $X$ conditioned on $E$ which we denote by $P_{(X|E)}$ as follows: For every possible value $x$ in the range of $X$,*

$$P_{(X|E)}(x) = \begin{cases} \Pr(X = x|E) & \Pr(X = x, E) > 0 \\ 0 & \Pr(X = x, E) = 0 \end{cases}$$

- *For a random variable $X$ and an event $E$ we say that $X$ is a $k$-source in $E$ if $P_{(X|E)}$ is a $k$-source.*
- *For two random variables $A, B$ and an event $E$ we say that the pair $(A, B)$ is a $(k_1, k_2)$-block source in $E$ if $P_{((A,B)|E)}$ is a $(k_1, k_2)$-source.*

*We use the convention that if $E$ has zero probability then any random variable is a $k$-source (or $(k_1, k_2)$-block source) in $E$.*

With this notation, the precise statement is that given some $k$-source $X$ and uniformly distributed seed $Y$ for the block extraction scheme we can partition this probability space into three sets: The first has negligible weight and can be ignored. On the second, the block extraction scheme produces a block source, and on the third, the block extraction scheme condenses the source. We now state this precisely.
For the remainder of this section we fix the following parameters:

- A $k$-source $X$ over $n$ bit strings.
- An $(k, \rho, \gamma)$ block extraction scheme $B : \{0,1\}^n \times \{0,1\}^u \to \{0,1\}^{n/r}$ for $r \geq 2$.
- A parameter $t$. (Intuitively, $t$ measures how much randomness we want the second block of a block-source to contain).

We now fix the following probability space that will be used in the remainder of this section. The probability space is over the set $\Omega = \{0,1\}^n \times \{0,1\}^u$ and consists of two independent random variables $X$ (the given $k$-source) and $Y$ that is uniformly distributed over $\{0,1\}^u$.

**Lemma 4.3** *There exist a partition of $\{0,1\}^n \times \{0,1\}^u$ into three sets $BAD$, $BLK$, $CON$ with the following properties:*

1. $\Pr(BAD) \leq 2(\rho + 2^{-t})$
2. *$(B(X,Y), X)$ is a $(\frac{\gamma k}{r} - t, t)$-block source in $BLK$.*
3. *$B(X,Y)$ is a $(k - 2t)$-source in $CON$.*

In the remainder of this section we use ideas similar to that in [NTS99] to prove Lemma 4.3. The idea is to partition the elements in the probability space into three sets according to their "weight": The "small weight" elements form the set $CON$. Intuitively the small weight elements induce a source of high min-entropy. The "medium-weight" elements form the set $BLK$. Intuitively the medium weight elements induce a source of medium min-entropy. Thus, they contain some (but not all!) of the min-entropy of the initial source. The fraction of "large weight" elements is bounded by $\rho$, (the error parameter of the block extraction scheme). These elements form the set $BAD$ and can be ignored because of their small fraction.

The following definition is motivated by the intuition above. (The partition of Lemma 4.3 will be based on the following partition).

**Definition 4.4** *We partition $\Omega = \{0,1\}^n \times \{0,1\}^u$ according to the "weight" of the elements.*

$$\begin{array}{lclllll} LRG & = & \{(x,y) \in \Omega & | & 2^{-(\frac{\gamma k}{r}-1)} < & \Pr(B(X,Y) = B(x,y)) & & \} \\ MED & = & \{(x,y) \in \Omega & | & 2^{-(k-t)} < & \Pr(B(X,Y) = B(x,y)) & \leq 2^{-(\frac{\gamma k}{r}-1)} & \} \\ SML & = & \{(x,y) \in \Omega & | & & \Pr(B(X,Y) = B(x,y)) & \leq 2^{-(k-t)} & \} \end{array}$$

We prove the following Lemma.

**Lemma 4.5** *The sets $LRG, MED$ and $SML$ have the following properties:*

1. $\Pr(LRG) \leq 2\rho$
2. $(B(X,Y),X)$ *is a* $(\frac{\gamma k}{r} - \log \frac{1}{\Pr(MED)} - 1, t)$*-block source in* $MED$.
3. $B(X,Y)$ *is a* $(k - (t + \log \frac{1}{\Pr(SML)}))$*-source in* $SML$.

**Proof:** (of lemma 4.5)

**Proof of first item.** We apply Lemma 2.2 choosing $V = B(X,Y)$. Note that $V$ is guaranteed to be $\rho$-close to a $\gamma k/r$-source and therefore by the Lemma $\Pr(LRG) \leq 2\rho$.

**Proof of second item.** Note that we need to prove the following:

- For every $(x,y) \in MED$, $\Pr(B(X,Y) = B(x,y)|MED) \leq 2^{-(\frac{\gamma k}{r} - \log \frac{1}{\Pr(MED)} - 1)}$.
- For every $(x,y) \in MED$, $\Pr(X = x|B(X,Y) = B(x,y), MED) \leq 2^{-t}$.

For the first statement we use the following inequality: For every two events $E_1, E_2$:

$$\Pr(E_1|E_2) = \frac{\Pr(E_1 \cap E_2)}{\Pr(E_2)} \leq \frac{\Pr(E_1)}{\Pr(E_2)} \tag{2}$$

Applying this rule on the first statement gives

$$\Pr(B(X,Y) = B(x,y)|MED) \leq \frac{\Pr(B(X,Y) = B(x,y))}{\Pr(MED)} \leq \frac{2^{-(\frac{\gamma k}{r}-1)}}{\Pr(MED)} \leq 2^{-(\frac{\gamma k}{r} - \log \frac{1}{\Pr(MED)} - 1)}$$

here we used the definition of $MED$.
We now prove the second statement:

$$\Pr(X = x|B(X,Y) = B(x,y), MED) = \frac{\Pr(X = x, B(X,Y) = B(x,y), MED)}{\Pr(B(X,Y) = B(x,y), MED)}$$

Note that whether a given pair $(x,y)$ is in the set $MED$ depends only on the value $B(x,y)$. Thus, the event $\{B(X,Y) = B(x,y), MED\} = \{B(X,Y) = B(x,y)\}$ because when $B(X,Y) = B(x,y)$ for $(x,y) \in MED$ we already know that $(X,Y) \in MED$. Thus,

$$= \frac{\Pr(X = x, B(X,Y) = B(x,y))}{\Pr(B(X,Y) = B(x,y))} \leq \frac{\Pr(X = x)}{\Pr(B(X,Y) = B(x,y))} \leq \frac{2^{-k}}{2^{-(k-t)}} = 2^{-t}$$

where the last inequality follows from the fact that $X$ is a $k$-source and from the definition of $MED$.

**Proof of the third item.** Note that we need to prove that for $(x,y) \in SML$:

$$\Pr(B(X,Y) = B(x,y)|SML) \leq 2^{-(k-(t+\log \frac{1}{\Pr(SML)}))}$$

The proof is similar to the proof of the first part in the second item. More precisely, we use the rule (2) above. We argue that:

$$\Pr(B(X,Y) = B(x,y)|SML) \leq \frac{\Pr(B(X,Y) = B(x,y)}{\Pr(SML)} \leq \frac{2^{-(k-t)}}{\Pr(SML)} \leq 2^{-(k-(t+\log \frac{1}{\Pr(SML)}))}$$

□

We are now ready to prove Lemma 4.3.
**Proof:** (of Lemma 4.3) We need to slightly change the partition above. The sets $LRG, MED$ and $SML$ are almost the partition we want. We only need to avoid a setup in which the sets $MED$ or $SML$ are too small, since in this case the effect of conditioning is too costly. Still, if one of the sets is very small we can safely add it to the "bad" elements and ignore it. This is the intuition behind the following partition, which partitions $\{0,1\}^n \times \{0,1\}^u$ into three sets:

1. The set $BAD$ will contain all $(x,y) \in LRG$. It will also contain all $(x,y) \in SML$ if $\Pr(SML) < 2^{-t}$, and all $(x,y) \in MED$ if $\Pr(MED) < 2^{-t}$.
2. The set $BLK$, (which corresponds to the set $MED$) contains all $(x,y) \in MED$ if $\Pr(MED) \geq 2^{-t}$. (Thus, $BLK$ is empty if $\Pr(MED) < 2^{-t}$).
3. The set $CON$, (which corresponds to the set $SML$) contains all $(x,y) \in SML$ if $\Pr(SML) \geq 2^{-t}$. (Thus, $BLK$ is empty if $\Pr(SML) < 2^{-t}$).

Lemma 4.3 follows from Lemma 4.5. □

# 5 Constructing condensers

In this section we use a win-win analysis as outlined in Section 1.6 to construct a condenser. The main Theorem of this Section is the following:

**Theorem 5.1** *For every $n$ and $k$ such that $k \geq 8\log^5 n$ and $2 \leq r \leq \log^2 n$ there is an explicit $(k, \Omega(k/r), 1/\log^2 n)$-condenser $Con : \{0,1\}^n \times \{0,1\}^{O(\frac{\log(n/k)\cdot\log\log n}{\log r}+\log n)} \to \{0,1\}^{\frac{k\log(n/k)}{r\log r}}$*

It is helpful to consider two particular corollaries: For the first one we choose $r = 2$. This gives that the condenser maintains a constant fraction of the initial randomness.

**Corollary 5.2** *For every $n$ and $k$ such that $k \geq 8\log^5 n$, there is an explicit $(k, \Omega(k), 1/\log^2 n)$-condenser $Con : \{0,1\}^n \times \{0,1\}^{O(\log\frac{n}{k}\log\log n+\log n)} \to \{0,1\}^{O(k\log\frac{n}{k})}$.*

For the second condenser we choose $r = \Theta(\log n)$. This gives a condenser with seed $O(\log n)$ that maintains a $(1/logn)$-fraction of the initial randomness.

**Corollary 5.3** *For every $n$ and $k$ such that $k \geq 8\log^5 n$, there is an explicit $(k, \Omega(\frac{k}{\log n}), 1/\log^2 n)$-condenser $Con : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\frac{k}{2}}$.*

In the remainder of this section we prove Theorem 5.1.

### 5.1 Getting a block-source

We now implement the idea presented in the introduction. Namely, that running the block extraction scheme recursively, eventually produces a block source. In the next definition we recursively run the block extraction scheme. That is given an $n$ bit string $x$ we use a fresh random seed $y$ of length $O(\log\log n)$ to obtain $x' = B(x, y)$ and continue this process recursively on $x'$.

**Definition 5.4** *Let $n, k$ and $r$ be parameters such that $k \geq 8\log^5 n$ and $1 < r \leq \log^2 n$. Let $l$ be a number that will be determined later. Let $t = \log^4 n$ and $\rho = 1/\log^4 n$*

*We first define sequences of numbers $n_0, \ldots, n_l$ and $k_0, \ldots, k_l$ as follows: $n_i = n/r^i$ and $k_i = k - 2ti$. Let $l$ be the smallest integer such that $n_i < k_i$. We soon show that such an $l$ exists and $l = O(\log_r(n/k))$.*

*By lemma 3.3 there exists a universal constant $\gamma > 0$ such that for every $i$ there is a $(k_i, \rho, \gamma)$-block extraction scheme $B^i : \{0,1\}^{n_i} \times \{0,1\}^{u_i} \to \{0,1\}^{n_i/r}$. Furthermore, note that $u_0 \geq u_i$ for every $1 \leq i \leq l$. Let $u = u_0$, and observe that for this choice of parameters $u = O(\log\log n)$.*

*For every $0 \leq i \leq l$ we define a function $D_i : \{0,1\}^n \times (\{0,1\}^u)^l \to \{0,1\}^{\frac{n}{r^i}}$, in the following manner:*

- *$D_0(x; y_1 \cdots y_l) = x$.*
- *For $i > 0$, $D_i(x; y_1, \cdots, y_l) = B^i(D_{i-1}(x; y_1, \cdots, y_l), y_i)$.*

*It is easy to see that $D_i$ does not depend on $y_{i+1}, \cdots, y_l$, and that for each $i$, computing $D_i$ takes polynomial time.*

Let $X$ be some $k$-source over $n$ bit strings. Consider the following probability space over $\Omega = \{0,1\}^n \times (\{0,1\}^u)^l$. It consists of the random variable $X$ and an independent random variable $Y = (Y_1, \ldots, Y_l)$ that is uniformly distributed over $(\{0,1\}^u)^l$. We also define random variables $B_0, \ldots, B_l$ by $B_i = D_i(X, Y)$. Following the intuition in Section 1.6 we want to argue that there exists a small $l$ and an $1 \leq i \leq l$ such that $(B_i, B_{i-1})$ is a block source. This does not hold. However, we can use the machinery developed in Section 4 to show a result with the same flavor.

**Lemma 5.5** *Let $t = \log^4 n$. If $k \geq 8\log^5 n$ then there exists a partition of $\{0,1\}^n \times (\{0,1\}^u)^l$ into $l + 1$ sets: $BLK_1, \cdots, BLK_l$ and $BAD$ with the following property:*

1. $\Pr(BAD) \leq 2l(\rho + 2^{-t})$
2. *$(B_i, B_{i-1})$ is a $(k', t)$-block source in $BLK_i$, (where $k' \geq \frac{\gamma(k-2lt)}{r}$).*
3. *$l = O(\log_r(n/k))$*

The remainder of this section is devoted to proving Lemma 5.5. The proof is just a recursive application of of Lemma 4.3 and the reader is encouraged to skip it on a first reading.

**Proof:**(of Lemma 5.5) For $0 \leq i \leq l$, we recursively define sets $BAD_i, BLK_i, CON_i \subseteq \{0,1\}^n \times (\{0,1\}^u)^l$ and distributions $X_i$ that is over $n_i$ bits. We define $BAD_0 = BLK_0 = \emptyset$, $CON_0 = \{0,1\}^n \times (\{0,1\}^u)^l$ and $X_0 = X$. For $i > 0$, suppose that sets $BAD_{i-1}, BLK_{i-1}, CON_{i-1}$ has already been defined, and that the distribution of $X_{i-1}$ is $P_{(B_{i-1}|CON_{i-1})}$ and that $X_{i-1}$ is a $k_{i-1}$-source. (Note that this holds for $i = 1$). We now recursively define sets $BAD_i, BLK_i, CON_i$ that are a partition of $CON_{i-1}$ and a distribution $X_i$.

We first apply Lemma 4.3 on the $i$'th application of the block-extraction scheme $B^i$ on $X_{i-1}$ and $Y_i$. It follows that $\{0,1\}^{n_{i-1}} \times \{0,1\}^u$ can be partitioned into three sets $BAD, BLK, CON$ as in the lemma.

We "pull these events back to the original probability space". That is we want to see these sets as a partition of $CON_{i-1}$. More precisely, we define:

$$BAD_i = \{(x, y_1, \ldots, y_l) \in CON_{i-1} : D_{i-1}(x, y_1, \ldots, y_l) \in BAD\}$$
$$BLK_i = \{(x, y_1, \ldots, y_l) \in CON_{i-1} : D_{i-1}(x, y_1, \ldots, y_l) \in BLK\}$$
$$CON_i = \{(x, y_1, \ldots, y_l) \in CON_{i-1} : D_{i-1}(x, y_1, \ldots, y_l) \in CON\}$$

Note that this is a partition of $CON_{i-1}$. Recall that $B_i = D_i(X,Y) = B^i(D_{i-1}(X,Y), Y_i)$. Thus, the distribution $P_{(B_i|CON_i)}$ is exactly the same as $P_{(B^i(X_{i-1},Y_i)|CON)}$. Similarly $P_{(B_i|BLK_i)}$ is exactly the same as $P_{(B^i(X_{i-1},Y_i)|BLK)}$. We conclude that the guarantees of Lemma 4.3 give the following:

1. $\Pr(BAD_i) \leq 2(\rho + 2^{-t})$
2. $(B_i, B_{i-1})$ is a $(\frac{\gamma k_{i-1}}{r} - t, t)$-block source in $BLK_i$.
3. $B_i$ is a $k_i$-source in $CON_i$.

We now define $X_i$ to be the distribution $P_{(B_i|CON_i)}$ that is over $n_i$ bits. Indeed, we have that $X_i$ is a $k_i$-source. Thus, we can successfully define sets $BAD_i, BLK_i, CON_i$ such that for each $i > 0$, these sets are a partition of $CON_{i-1}$ and the three properties above hold.

We now show that at some step $i$, $CON_i = \emptyset$. After $i$ steps, the length of the $i$'th block is $n_i = n/r^i$ and $k_i = k - 2it$. Thus, after $l = \log_r(4n/k)$ steps we have that the $i$'th block is of length at most $k/4$. At this point $k_l = k - 2lt \geq k - 2\log^5 n \geq k/2$. It follows that $n_l < k_l$ and that there is some $i \leq l$ for which $CON_i = \emptyset$ as otherwise the third item above cannot hold (simply because it is impossible to have a distribution with more entropy than length).

We define: $BAD = \cup_{1 \leq i \leq l} BAD_i$. It follows that $BLK_1, \ldots, BLK_l$ and $BAD$ are a partition of $\Omega = \{0,1\}^n \times (\{0,1\}^l)^u$ and the lemma follows. □

### 5.2 Getting a condenser

In the previous section we showed how to get $\ell = O(\log_r(n/k)$ pairs of distributions such that (at least in some sense) one of them is a block-source. Had we been able to construct a single block source, we could have used the block source extractor of corollary 2.6 to get an extractor. At this point however, we have many candidates (pairs $B_i, B_{i-1}$). We now run block source extractors on all pairs (using the same seed). It follows that one of the outputs is close to uniform and therefore the concatenation of the outputs gives a condenser. We now formalize this intuition.

**Construction 5.6** *We use the parameters of Definition 5.4, namely: Let $n, k$ and $r$ be parameters such that $k \geq 8\log^5 n$ and $1 < r \leq \log^2 n$. Let $l = \log_r(4n/k)$, $t = \log^4 n$ and $\rho = 1/\log^4 n$. Let $u = O(\log\log n)$ be the seed length for the block extraction scheme as determined in Definition 5.4. Let $k' = \frac{\gamma(k-2lt)}{r}$*

*We define a function $Con : \{0,1\}^n \times \{0,1\}^{ul+O(\log n)} \to \{0,1\}^{n'}$ where $n'$ is determined later. Given inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^{ul+O(\log n)}$, $Con$ interprets its second argument as $l$ strings $y_1, \cdots, y_l \in \{0,1\}^u$ and an additional string $s$ of length $O(\log n)$. For $0 \leq i \leq l$ it computes $b_i = D_i(x; y_1, \cdots, y_l)$, (where $D_i$ is taken from definition 5.4), and $o_i = BE(b_i, b_{i-1}, s)$, (where $BE$ is the block source extractor of corollary 2.6 using output length $k'$). The final output is $(o_1, \cdots, o_l)$, (which makes $n' = lk'$).*

We now prove Theorem 5.1.

**Proof:** (of Theorem 5.1) Let $X$ be a $k$-source. For this proof we fix a probability space consisting of independent random variables $X$, $Y$ and $Z$ where $Y=(Y_1,\ldots,Y_l)$ is uniformly distributed over $(\{0,1\}^u)^l$ and $Z=(Z_1,\ldots,Z_l)$ is uniformly distributed over $(\{0,1\}^{k'})^l$. We now define more random variables as a function of the initial random variables. We define random variables $B_1,\ldots,B_l$ as before by $B_i=D_i(X,Y)$. We also define random variables $O_1,\ldots,O_l$ by $O_i=BE(B_{i-1},B_i)$. Note that $CON(X,Y)=(O_1,\ldots,O_l)$. We also define random variables $(R_1,\ldots,R_l)$ over $(\{0,1\}^{k'})^l$ as follows: Let $BLK_1,\cdots,BLK_l$ and $BAD$ be the sets of Lemma 5.5. If $(X;Y_1,\cdots,Y_l)\in BAD$ we set $R=Z$. Otherwise, $(X;Y_1,\cdots,Y_l)$ belong to a unique $BLK_i$. In this case we set $R_i=Z_i$ and $R_j=O_j$ for $j\neq i$. Note that $R$ is a $k'$-source. To complete the proof we now show that $(R_1,\cdots,R_l)$ is $(2l(\rho+2^{-t})+2/\log^4 n)$-close to $(O_1,\cdots,O_l)$. This suffices as $2l(\rho+2^{-t})+2/\log^4 n\leq 1/\log^2 n$.

By Lemma 5.5 we have that $(B_i,B_{i-1})$ is close to a block source in $BLK_i$. The block source extractor $BE$ has error $\epsilon'=1/|B_2|+1/|B_1|$. Recall that the length of all blocks $B_i$ is at least $k'\geq\log^4 n$. It follows that $\epsilon'<2/\log^4 n$ and that $O_i$ is $\epsilon'$-close to uniform in $BLK_i$. We conclude that for every $i$, $(R_1,\cdots,R_l)$ is $\epsilon'$-close to $(O_1,\cdots,O_l)$ in $BLK_i$. This gives that $(R_1,\cdots,R_l)$ is $\epsilon'$-close to $(O_1,\cdots,O_l)$ in the complement of $BAD$. By Lemma 5.5 the probability of $BAD$ is at most $2l(\rho+2^{-t})$. Thus, $(R_1,\cdots,R_l)$ is $2l(\rho+2^{-t})+\epsilon'$ close to $(O_1,\cdots,O_l)$. □

Let us compare the entropy rates of the new source and the initial source. The new source has min-entropy $k'$ which is approximately $k$ and length approximately $k\cdot\log\frac{n}{k}$, whereas the initial source had length $n=k\cdot\frac{n}{k}$. Note that $\log(n/k)<n/k$ and thus $Con$ indeed improves the entropy rate and is a $(k,k',2l(\rho+2^{-t})+\epsilon')$-condenser.

**Remark 5.7** *Actually, the distribution $(O_1,\cdots,O_l)$ is a source of a special kind called a "somewhere random source" by Ta-Shma in [TS96]. In [TS96] it was shown that extracting randomness from such sources is easier using special extractors which are called "somewhere random mergers". At this point we could have used Ta-Shma's "somewhere random mergers", to extract the randomness from $(o_1,\cdots,o_l)$. Instead, we use different methods which exploit the fact that $l$ is relatively small to obtain better results.*

# 6 Constructing extractors

In this section we use the condensers constructed in the previous section to prove the two main Theorems, (Theorems 1.4,1.6).

For Theorem 1.4 we use the condenser of Corollary 5.2 repeatedly (with fresh seeds) to condense the source until we achieve constant entropy rate. (This is guaranteed to happen after no more than $\log^* n$ iterations). For constant entropy rate, Zuckerman's extractor ([Zuc97] see table 2) uses the optimal seed length to extract a constant fraction. This procedure loses some randomness in the iteration process, and results in an extractor which extracts a sub-constant fraction of the initial randomness. We then use [WZ93] to increase this fraction to an arbitrary constant. This informal argument is made precise in the following proof:

**Proof:** (of Theorem 1.4) Without loss of generality we assume that $k\geq 8\log^5 n$ as the extractor of [ISW00] achieves the required result for $k<8\log^5 n$. It is easy to check that given a $(k,k',\epsilon)$-condenser $Con_1:\{0,1\}^n\times\{0,1\}^d\rightarrow\{0,1\}^{n'}$ and a $(k',k'',\epsilon')$-condenser $Con_2:\{0,1\}^{n'}\times\{0,1\}^{d'}\rightarrow\{0,1\}^{n''}$, composing the condensers produces a $(k,k'',\epsilon+\epsilon')$-condenser $Con:\{0,1\}^n\times\{0,1\}^{d+d'}\rightarrow\{0,1\}^{n''}$.

Let us denote the entropy rate of a source by $R(X)=k/n$ and let $R'(X)=n/k=1/R(X)$. The condenser of Corollary 5.2 produces a source $X'$ that is close to a $\Omega(k)$ source over $k\log(n/k)$ bits.

Thus, $R(X') = \Theta(1/\log(1/R(X)))$ or in other words, we have that $R'(X') = \Theta(\log(R'(X))$. We now apply the condenser recursively on $X'$ using a fresh seed. After $\log^* R'(X) \le \log^* n$ iterations the entropy rate becomes constant. Once the ratio is constant Zuckerman's extractor ([Zuc97], see table 1), can be used to extract a constant fraction (say half) of the randomness using a fresh seed of length $O(\log n)$ and error $1/\log^2 n$. Overall, we've used at most $\log^* n$ iterations where in each of them we required a seed of length at most $O(\log n \cdot \log\log n)$ and the final application of Zuckerman's extractor requires an additional $O(\log n)$ bits. Thus, the strategy described above gives an extractor that uses seed length $O(\log n \cdot \log\log n \cdot \log^* n)$ bits. Recall that our condenser loses a constant fraction of the randomness in every iteration. Thus, after $\log^* n$ iterations we extract only $k/2^{O(\log^* n)}$ random bits from the source, and produce an extractor which extracts a $1/2^{O(\log^* n)}$ fraction of the initial randomness. To get to a constant fraction we use the method of Wigderson and Zuckerman, [WZ93].[10] Implementing the technique of Wigderson and Zuckerman multiplies the seed and error by $2^{O(\log^* n)}$. Thus, the total number of random bits is $\log n \cdot \log\log n \cdot \log^* n \cdot 2^{O(\log^* n)} = O(\log n \cdot (\log\log n)^2)$ as required. Furthermore the final extractor has error smaller than $1/\log n$. □

In the case of Theorem 1.6 we are shooting for the optimal seed length and cannot afford the condenser of Corollary 5.2 or repeated condensing. Instead we use the condenser of Corollary 5.3 interpreting it as a block extraction scheme. Viewed this way the condenser extracts a block $B$ of length $k/2$, therefore the distribution $(B(X,Y),X)$ forms a block source, since $B$ is too short to "steal" all the randomness from $X$. (This intuition is formalized in the next Lemma). All that is left is to use the block source extractor of corollary 2.6. The precise details follow.

**Lemma 6.1** *Let $Con$ be the condenser of Corollary 5.3. If $X$ is a $k$-source for $k \ge 8\log^5 n$ then the distribution $(Con(X, U_{O(\log n)}), X)$ is $O(1/\log^2 n)$-close to an $(\Omega(k/\log n), \log^4 n)$-block source.*

**Proof:** Fix some $n$ and $k \ge 8\log^5 n$, and let $Con : \{0,1\}^n \times \{0,1\}^{u=O(\log n)} \to \{0,1\}^{k/2}$ be the $(k, \Omega(k/\log n), 1/\log^2 n)$-condenser of Corollary 5.3. For this proof we view $Con$ as a block-extraction scheme $B : \{0,1\}^n \times \{0,1\}^u \to \{0,1\}^{n/r}$ for $r = 2n/k$. It follows that $B$ is a $(k,\rho,\gamma)$-block extraction scheme for $\rho = 1/\log^2 n$ and $\gamma = \Omega(r/\log n)$. We remark that in particular $\gamma \gg 1$.

We now consider the probability space of Section 4. The probability space is over the set $\Omega = \{0,1\}^n \times \{0,1\}^u$ and consists of two independent random variables $X$ (the given $k$-source) and $Y$ that is uniformly distributed over $\{0,1\}^u$. We set $t = \log^4 n$ and apply Lemma 4.3 and let $BAD, BLK, CON$ be the sets guaranteed by the lemma. We claim that $CON = \emptyset$.

This is because the Lemma guarantees that $(B(X,Y)$ is a $(k-2t)$-source in $CON$. Note that the output length of $B$ is $k/2$ whereas $k - 2t > k/2$ because $k \ge 8\log^5 n$. Thus, the Lemma says that in $CON$, there is a random variable which has min-entropy larger than it length. This statement can only be true if $CON = \emptyset$.

Lemma 4.3 also gives that:

- $\Pr(BAD) \le 2(\rho + 2^{-t})$
- $(B(X,Y),X)$ is a $(\frac{\gamma k}{r} - t, t)$-block source in $BLK$.

[10] Wigderson and Zuckerman suggested to repeatedly extract randomness from the source, (using fresh seeds), until one extracts the desired fraction. This gives that if $m = k/p$ then $m$ could be increased to $(1-\alpha)k$, (where $\alpha$ is an arbitrary constant), at the cost of multiplying $d$ by $O(p)$. (An exact formulation of the Wigderson and Zuckerman technique can be found for example in [Nis96, NTS99]).

Thus, $(B(X,Y),X)$ is $O(\rho+2^{-t})$-close to a $(\frac{\gamma k}{r}-t,t)$-block source. Using again that $k \geq 8\log^5 n$, we conclude that the distribution $(Con(X,U_{O(\log n)}),X)$ is $O(1/\log^2 n)$-close to an $(\Omega(k/\log n),\log^4 n)$-block source as required. □

We now prove Theorem 1.6.
**Proof:** (of Theorem 1.6) As in the proof of Theorem 1.4 we can without loss of generality assume that $k \geq 8\log^5 n$ because the extractor of [ISW00] achieves the required result for $k < 8\log^5 n$. Given a $k$-source, we use Lemma 6.1 to get a distribution that is close to a block-source and use the block-source extractor of corollary 2.6. □

# 7 Achieving small error

The statement of Theorems 1.4,1.6 is for constant error $\epsilon$. The analysis provided in this paper gives a slightly better result and allows to replace the requirement that $\epsilon$ be a constant with $\epsilon = 1/(logn)^c$ for any constant $c$. Still, our technique does not give good dependence of the seed length on the error. We get better dependence on $\epsilon$ using the error reduction transformation of [RRV99], which transforms an extractor with large, (say constant) error into an extractor with arbitrary small error, while losing only a little bit in the other parameters. More precisely, after undergoing the transformation, a factor of $O(\log m(\log\log m)^{O(1)} + \log(1/\epsilon))$ is added to $d$, and the fraction extracted decreases by a constant. The latter loss makes no difference from our point of view since we are only able to extract constant fractions. The first loss isn't significant in the case of Theorem 1.4, since the seed size is already larger than the optimal one by a multiplicative *polyloglog*$(n)$ factor. However, it completely spoils Theorem 1.6 and makes it inferior to Theorem 1.4. Here is Theorem 1.4 rephrased using the error reduction transformation of [RRV99]:

**Theorem 7.1** *(Theorem 1.4 rephrased for non-constant $\epsilon$) For every $n,k$ and $\epsilon > exp(\frac{-n}{(\log^* n)^{O(\log^* n)}})$, there are explicit $(k,\epsilon)$-extractors $Ext : \{0,1\}^n \times \{0,1\}^{O(\log n\cdot(\log\log n)^{O(1)}+\log(1/\epsilon))} \to \{0,1\}^{(1-\alpha)k}$, where $\alpha > 0$ is an arbitrary constant.*

# 8 Transforming arbitrary extractors into strong extractors

It is sometimes helpful to have a stronger variant of extractors, called a *strong* extractor. A strong extractor is required to extract randomness "only from the source" and not "from the seed".

**Definition 8.1** *A $(k,\epsilon)$-extractor $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is* strong *if for every $k$-source $X$, the distribution $(Ext(X,U_d) \circ U_d)$ (obtained by concatenating the seed to the output of the extractor) is $\epsilon$-close to a $U_{m+d}$.*

Intuitively, this is helpful since a strong extractor has the property that for any source a $1-\epsilon$ fraction of the seeds extract randomness from that source. It is interesting to note that the concept of strong-extractors preceded that of non-strong extractors, and the strong version was the one which was defined in the seminal paper of [NZ96]. Several extractors constructions, (with examples being [TS96, ISW00] and the constructions of this paper) are non-strong or difficult to analyze as strong.

The following Theorem shows that every non-strong extractor can be transformed into a strong one with essentially the same parameters.

**Theorem 8.2** *Any explicit $(k,\epsilon)$-extractor $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ can be transformed into an explicit strong $(k, O(\sqrt{\epsilon}))$-extractor $Ext' : \{0,1\}^n \times \{0,1\}^{d+d'} \to \{0,1\}^{m-(d+L+1)}$ for $d' = polylog(d/\epsilon)$ and $L = 2\log(1/\epsilon) + O(1)$.*

Let us consider the parameters of $Ext'$ compared to that of $Ext$. The seed length of $Ext'$ is longer than that of $Ext$ by a factor that is only polylogarithmic (for large $\epsilon$). The output length of $Ext'$ is shorter than that of $Ext$ by $d+L+1$. The loss of $d$ bits is unavoidable as the output of $Ext$ may contain $d$ bits of randomness from the seed. The additional loss of $L+1$ bits can sometimes be recovered (at the cost of increasing the seed length). Exact details are given in Remark 8.3.

**Remark 8.3** *In [RRV02] it was shown that any strong extractor which has seed length $d$ and entropy-loss $\Delta$ can be transformed into a strong extractor with seed length $d+O(\Delta)$ and an optimal entropy loss of $2\log(1/\epsilon)+O(1)$. Thus, if the initial extractor $Ext$ had an entropy loss of $\Delta$, we can use our construction to get an extractor $Ext'$ with the parameters mentioned above, and then use [RRV02] to construct a strong extractor $Ext''$ with seed length $d'' = d + d' + O(\Delta)$ and optimal entropy loss. This addition is affordable if $\Delta$ is small.*

The intuition above also gives a hint for the construction. The output of $Ext$ may contain $d$ bits of randomness which "belong" to the seed. Still, it contains roughly $m-d$ bits which *do not* depend on the seed. Thus, fixing the seed, the output of $Ext$ is a random source of length $m$ which contains roughly $m-d$ random bits. We can now use another extractor to extract this randomness and "dismantle" the correlation between the seed and output. The extractor we need is one that works well when the source lacks only a very small amount of randomness. Such a construction was given by [GW97] and improved by [RVW00].

**Theorem 8.4** *[RVW00] There are explicit strong $(k,\epsilon)$-extractors $RVW : \{0,1\}^n \times \{0,1\}^{d'} \to \{0,1\}^{k-L}$ For $d' = polylog((n-k)/\epsilon)$ and $L = 2\log(1/\epsilon) + O(1)$.*

**Construction 8.5** *Given a $(k,\epsilon)$-extractor $Ext : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ we construct $Ext'$ as follows. Let $RVW : \{0,1\}^m \times \{0,1\}^{d'=polylog(d/\epsilon)} \to \{0,1\}^{m-d-1-L}$ be an $(m-d-1,\epsilon)$-extractor guaranteed by Theorem 8.4. Then,*

$$Ext'(x;(y,z)) = RVW(Ext(x,y), z)$$

The actual proof that $Ext'$ has the desired properties is slightly more complicated than the above presentation. This is mainly because the above presentation ignores the error of $Ext$. We now give the formal proof.

**Proof:** (of Theorem 8.2) We now describe a probability space for this proof. It consists of three independent random variables: an arbitrary $k$-source $X$ over $n$ bit strings, a uniformly chosen string $Y$ of length $d$ and a uniformly chosen string $Z$ of length $d'$.

Given strings $x \in \{0,1\}^n$ and $y \in \{0,1\}^d$ we define the *weight* of $(x,y)$, denoted $w(x,y)$ in the following way:

$$w(x,y) = \Pr(Ext(X,Y) = Ext(x,y))$$

That is the weight of the string $Ext(x,y)$ according to the distribution $Ext(X,Y)$. We say that a pair $(x,y)$ is heavy if $w(x,y) > 2^{-(m-1)}$. We first claim that only few pairs are heavy.

**Claim 1** $\Pr((X,Y)$ *are heavy*$) < 2\epsilon$.

**Proof:** Let $V$ denote the distribution $Ext(X, Y)$. We now use Lemma 2.2. We have that $V$ is $\epsilon$-close to an $m$-source (the uniform distribution). Therefore the probability under $V$ of hitting an element $v$ such that $\Pr_V(V = v) > 2^{-(m-1)}$ is bounded by $2\epsilon$ and the claim follows. $\square$

Call a seed $y \in \{0,1\}^d$ *bad* if $\Pr((X, y) \text{ is heavy}) > \sqrt{2\epsilon}$. That is if for many choices of $x$, the output element is heavy. We now claim that there are few bad seeds.

**Claim 2** *The fraction of bad seeds is at most $\sqrt{2\epsilon}$.*

**Proof:** The proof is a Markov argument. If the fraction of bad seeds were more than $\sqrt{2\epsilon}$ than $\Pr((X, Y) \text{ is heavy}) > 2\epsilon$. $\square$

The following claim shows that running the extractor with a good seed produces a source which lacks very few random bits.

**Claim 3** *For a good seed $y$, $Ext(X, y)$ is $\sqrt{2\epsilon}$-close to an $(m - d - 1)$-source.*

**Proof:** For a good $y$ we know that $\Pr((X, y) \text{ is heavy}) < \sqrt{2\epsilon}$. That is at least a $1 - \sqrt{2\epsilon}$ fraction of the $x$'s have $w(x, y) \leq 2^{-(m-1)})$. For such an $x$,

$$\Pr(Ext(X, y) = Ext(x, y)) = \Pr(Ext(X, Y) = Ext(x, y)|Y = y) \leq \frac{w(x, y)}{2^{-d}} \leq 2^{-(m-d-1)}$$

$\square$

We have that $Ext'$ runs $RVW$ on the source $Ext(X, Y)$ using a fresh seed $Z$. Using the fact that for a good seed $y$, $Ext(X, y)$ is close to a high entropy source we derive the following claim.

**Claim 4** *Given $y \in \{0,1\}^d$, let $D_y$ denote $(Ext'(X; (y, Z)) \circ Z)$. For every good $y$, $D_y$ is $(2\sqrt{\epsilon} + \epsilon)$-close to uniform.*

**Proof:** Note that $Ext'(X; (y, Z)) = RVW(Ext(X, y), Z)$. The claim follows immediately from claim 3 and the fact that $RVW$ is a strong extractor. $\square$

We now complete the proof of the theorem. Let $D$ denote $(EXT'(X; (Y, Z)) \circ Z)$. We need to show that $(D \circ Y)$ is $O(\sqrt{\epsilon})$-close to uniform. Note that $D = D_Y$ (that is $D$ is a convex combination of the distributions $D_y$). As the fraction of bad seeds is at most $O(\sqrt{\epsilon})$ it is sufficient to show that that for any good seed $y$, $(D_y \circ Y)$ is $O(\sqrt{\epsilon})$-close to uniform. Note that as $Y$ is independent of $D_y$ it is sufficient to show that $D_y$ is $O(\sqrt{\epsilon})$-close to uniform which follows from Claim 4. $\square$

# 9 Discussion

In a subsequent work [LRVW03] achieve extractors with better parameters than those constructed here. Namely, for constant error $\epsilon > 0$ they achieve a $(k, \epsilon)$-extractor $E : \{0,1\}^n \times \{0,1\}^{O(\log n)} \to \{0,1\}^{\Omega(k)}$ for every choice of $k$. Their construction uses some of the ideas of this paper (condensers, win-win analysis) as well as additional ideas.

The next milestone in extractor constructions seems to be achieving seed length $d = O(\log n)$ and output length $m = k + d - O(1)$. (We remark that the difference between output length $\Omega(k)$

and $k$ is significant in some applications of extractors). This has already been achieved by [TSUZ01] for small values of $k$, ($k = 2^{\log n/\log\log n}$) in a subsequent work.

Another important goal is to achieve the "correct dependance" of the seed length on $\epsilon$ for non-constant $\epsilon$. Namely, to achieve an extractor with seed length $d = O(\log(n/\epsilon))$ and output length (say) $m = \Omega(k)$. We remark that both our approach and the approach of [LRVW03] do not give this dependance.

## 10 Acknowledgments

We thank Russel Impagliazzo and Amnon Ta-Shma for helpful discussions, and particularly for bringing to our attention their insights on error correcting random sources. We are also grateful to anonymous referees for helpful comments.

## References

[Blu86] M. Blum. Independent unbiased coin flips from a correlated biased source–A finite state markov chain. *COMBINAT: Combinatorica*, 6, 1986.

[CG89] B. Chor and O. Goldreich. On the power of two-point based sampling. *Journal of Complexity*, 5(1):96–106, March 1989.

[CRVW02] M. R. Capalbo, O. Reingold, S. P. Vadhan, and A. Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 659–668, 2002.

[GW97] O. Goldreich and A. Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *RSA: Random Structures and Algorithms*, 11, 1997.

[Imp99] R. Impagliazzo. private communication, 1999.

[ISW99] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Near-optimal conversion of hardness into pseudo-randomness. In *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 181–190, 1999.

[ISW00] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the thirty second annual ACM Symposium on Theory of Computing: Portland, Oregon, May 21–23*, pages 1–10, 2000.

[Jus72] J. Justesen. A class of constructive asymptotically good algebraic codes. *IEEE Trans. Info. Theory*, 18:652–656, 1972.

[LRVW03] C. J. Lu, Omer Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, New York, NY, USA, 2003. ACM Press.

[Nis96] N. Nisan. Extracting randomness: How and why: A survey. In *Proceedings, Eleventh Annual IEEE Conference on Computational Complexity*, pages 44–58, Philadelphia, Pennsylvania, 24–27 May 1996. IEEE Computer Society Press.

[NN93] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM Journal on Computing*, 22(4):838–856, August 1993.

[NTS99] N. Nisan and A. Ta-Shma. Extracting randomness: A survey and new constructions. *JCSS: Journal of Computer and System Sciences*, 58, 1999.

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

[RR99] R. Raz and O. Reingold. On recycling the randomness of states in space bounded computation. In *STOC: ACM Symposium on Theory of Computing (STOC)*, 1999.

[RRV99] R. Raz, O. Reingold, and S. Vadhan. Error reduction for extractors. In *40th Annual Symposium on Foundations of Computer Science: October 17–19, 1999, New York City, New York,*, pages 191–201, 1999.

[RRV02] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *JCSS: Journal of Computer and System Sciences*, 65, 2002.

[RSW00] O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, 2000.

[RTS00] J. Radhakrishnan and A. Ta-Shma. Bounds for dispersers, extractors, and depth-two superconcentrators. *SIAM Journal on Discrete Mathematics*, 13(1):2–24, February 2000.

[RVW00] O. Reingold, S. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In IEEE, editor, *41st Annual Symposium on Foundations of Computer Science: proceedings: 12–14 November, 2000, Redondo Beach, California*, pages 3–13, 2000.

[Sha02] Ronen Shaltiel. Recent developments in extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77, 2002.

[SU01] R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudo-random generator. In *42nd IEEE Symposium on Foundations of Computer Science: proceedings: October 14–17, 2001, Las Vegas, Nevada, USA*, pages 648–657, 2001.

[SV86] M. Santha and U. V. Vazirani. Generating quasi-random sequences from semi-random sources. *Journal of Computer and System Sciences*, 33(1):75–87, April 1986.

[SZ98] A. Srinivasan and D. Zuckerman. Computing with very weak random sources. *SICOMP: SIAM Journal on Computing*, 28, 1998.

[Tre01] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, 48(4):860–879, July 2001.

[TS96] A. Ta-Shma. On extracting randomness from weak random sources (extended abstract). In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing*, pages 276–285, Philadelphia, Pennsylvania, 22–24 May 1996.

[TSUZ01] Amnon Ta-Shma, Christopher Umans, and David Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing: Hersonissos, Crete, Greece, July 6–8, 2001*, pages 143–152, 2001.

[TSZS01] A. Ta-Shma, D. Zuckerman, and S. Safra. Extractors from Reed-Muller codes. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science*, pages 181–190, 1109 Spring Street, Suite 300, Silver Spring, MD 20910, USA, 2001. IEEE Computer Society Press.

[vN51] J. von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12:36–38, 1951.

[WZ93] A. Wigderson and D. Zuckerman. Expanders that beat the eigenvalue bound: Explicit construction and applications. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on the Theory of Computing*, pages 245–251, San Diego, California, 16–18 May 1993.

[Zuc90] D. Zuckerman. General weak random sources. In *31st Annual Symposium on Foundations of Computer Science*, volume II, pages 534–543, St. Louis, Missouri, 22–24 October 1990. IEEE.

[Zuc96] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4/5):367–391, October/November 1996.

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