Extractors for a Constant Number of Polynomially Small Min-Entropy Independent Sources

Anup Rao*
Department of Computer Science,
University of Texas at Austin
arao@cs.utexas.edu

March 22, 2006

Abstract

We consider the problem of randomness extraction from independent sources. We construct an extractor that can extract from a constant number of independent sources of length n, each of which have min-entropy n^γ for an arbitrarily small constant $\gamma > 0$. Our extractor is obtained by composing seeded extractors in simple ways. We introduce a new technique to *condense* independent somewhere-random sources which looks like a useful way to manipulate independent sources. Our techniques are different from those used in recent work [BIW04, BKS+05, Raz05, Bou05] for this problem in the sense that they do not rely on any results from additive number theory.

Using Bourgain's extractor [Bou05] as a black box, we obtain a new extractor for 2 independent block-sources with few blocks, even when the min-entropy is as small as polylog(n). We also show how to modify the 2 source disperser for linear min-entropy of Barak et al. [BKS+05] and the 3 source extractor of Raz [Raz05] to get dispersers/extractors with exponentially small error and linear output length where previously both were constant.

In terms of Ramsey Hypergraphs, for every constant $1 > \gamma > 0$ our construction gives a family of explicit $O(1/\gamma)$-uniform hypergraphs on N vertices that avoid cliques and independent sets of size $2^{(\log N)^\gamma}$.

Keywords: Extractor, Independent Sources, Ramsey Graphs

*Supported in part by an MCD fellowship from UT Austin and NSF Grant CCR-0310960.

1 Introduction

The use of randomness is widespread in computer science. Many of the best performing algorithms and protocols in many different areas of computer science are randomized. To guarantee their performance these algorithms usually rely on a perfect source of uncorrelated uniformly random bits, yet such a source may not be easy to obtain. We might instead have access to an imperfect random source where the bits are correlated and not uniformly random.

This motivates the study of objects called *extractors*. Informally, an extractor is an explicit efficiently computable function $\mathsf{Ext} : \{0,1\}^n \to \{0,1\}^m$ that takes as input bits from an imperfect random source and produces bits that are close to uniformly random (the distance of the output distribution from the uniform distribution is called the error of the extractor). If we had access to such a function, we could use it to extract truly random bits from an imperfect random source. We would then use the extracted bits in our application. Thus we could achieve performance guarantees even with imperfect sources of randomness. Extractors were first considered and studied with exactly this goal in mind. A long sequence of works involving many researchers in the 80's and 90's has developed extractor constructions and applications. Extractors are now known to have applications in a wide range of problems and are interesting objects in their own right. For surveys of the origins, applications and constructions we refer the interested reader to [Nis96, NT99, Sha02].

1.1 Modeling the source

To formalize the problem of randomness extraction, we must decide on a model for the types of imperfect sources that the extractors can handle. If we intend to extract m random bits, information theoretic considerations show that the imperfect source must contain at least m bits of entropy. The goal is to construct extractors which output the most number of random bits for a source with given entropy, have small error and work for very general models. The most general model that has been considered to date is what we will call a *weak source* [CG88]. The only constraint on a weak source that supplies n total bits is that the probability of getting any particular string from the source is at most 2^{-k}, where k is called the *min-entropy* of the source. Such a source is called an (n, k)-source. Unfortunately it can be shown that there is no deterministic extractor that can extract from general weak sources.

One way to get around this problem is to restrict the source so that it consists of a sample from a weak source and an additional much shorter independent *seed* of truly uniformly random bits. In this paper we will call an extractor for such sources a *seeded extractor*. These kinds of extractors were first considered by Nisan and Zuckerman [NZ93]. For any $n, k \in \mathbb{N}$ we now know how to construct extractors that can extract a constant fraction of k bits which are almost uniformly random using a very short (only a constant multiple of $\log n$) length seed from any (n, k)-source [LRVW03]. This is sufficient to simulate any algorithm that relies on truly uniformly random bits for its efficiency with the aid of a weak source in polynomial time. The assumption that we have access to a truly uniformly random seed is restrictive. These extractors are not appropriate for many applications where randomness is needed for more than improving efficiency. For instance, many cryptographic applications need to pick a key uniformly at random. Such an operation cannot be simulated using a seeded extractor and a weak source without access to an independent uniformly random seed.

Several other models for sources have been considered [vN51, Blu84, Vaz85, SV86, CFG+85, CW89, TV00, MU02, KZ03, GRS04, GR05]. In this paper, we assume we have access to a few independent sources, each of which have high enough min-entropy. The probabilistic method shows that extractors for this model exist even when given access to just 2 independent $(n, \mathrm{polylog}(n))$-sources. The challenge is to construct a polynomial time computable function that is a good extractor for such sources.

One important reason why this model is interesting is its connection to explicit constructions of Ramsey Graphs. Every function with two inputs can be viewed as a coloring of the corresponding complete bipartite graph. When the function is an extractor for 2 independent sources, the extractor property guarantees that this coloring gives a bipartite Ramsey Graph, i.e. a two colored complete bipartite graph with no large monochromatic bipartite clique. It is easy to convert any bipartite Ramsey Graph into a regular Ramsey Graph, so this immediately gives explicit constructions of Ramsey Graphs. When the extractor requires a few (say constant u) number of sources, the corresesonding coloring can be used to efficiently construct a u-uniform Ramsey Hypergraph.

1.2 Previous Results and Overview of New Results

The problem of extracting from several independent sources was first considered by Chor and Goldreich [CG88][1]. They demonstrated extractors for 2 independent $(n, (1/2 + \alpha)n)$-sources, for all constant $\alpha \in (0, 1/2]$.

Since then there had not been much success in improving the entropy requirements until Barak, Impagliazzo and Wigderson [BIW04] showed how to extract from a constant number of independent $(n, \delta n)$-sources, where δ (the *min-entropy rate* of the source) is allowed to be any arbitrarily small constant. The number of sources used depends on δ. Subsequently Barak et al. [BKS$^+$05] showed how to extract a constant number of bits with constant error from 3 $(n, \delta n)$-sources, where δ is an arbitrarily small constant. In this work they also present 2-source *dispersers* (a disperser is an object similar to but somewhat weaker than an extractor) that output a constant number of bits with constant error and work for min-entropy rate δ where δ is an arbitrarily small constant.

Raz [Raz05] gave an extractor for 2 independent sources where one source needs to have min-entropy rate greater than and bounded away from 1/2 and the other source may have polylogarithmically small min-entropy. In this case his extractor can extract a linear fraction of the min-entropy with exponentially small error. Improving the 3 source extractor of Barak et al., he constructed an extractor for 3 independent sources where one source must have constant min-entropy rate and the other two need polylogarithmic min-entropy. In this case his extractor can extract a constant number of bits with constant error.

Bourgain [Bou05] gave another extractor for 2 independent sources. His extractor can extract from 2 $(n, (1/2 - \alpha_0)n)$-sources, where α_0 is some small universal constant. This is the first extractor to break the 1/2 min-entropy rate barrier for 2 sources. His extractor outputs a linear fraction of the min-entropy, with exponentially small error.

Other than Raz's extractor for 2 sources, all of these recent results were made possible by new breakthroughs on the sum-product estimate for finite fields [BKT04, Kon03], a result from additive number theory. A common feature of the work of Raz (in the case of 3 sources) [Raz05] and Barak et al. [BKS$^+$05] is that they reduce the general problem of extracting from independent sources to the problem of extracting from independent sources that come from a much more restricted class, called *somewhere-random* sources. They then build extractors for these sources. A key step in our construction is building much better extractors for independent *somewhere-random* sources.

[1] Santha and Vazirani [SV86] also considered extracting from independent sources, but the sources had additional restrictions placed on them.

Construction	Min-Entropy	Output	Error	Ref
$O(\text{poly}(1/\delta))$-source extractor	δn	$\Theta(n)$	$2^{-\Omega(n)}$	[BIW04]
3-source extractor	δn, any constant δ	$\Theta(1)$	$O(1)$	[BKS+05]
3-source extractor	One source: δn, any constant δ. Other sources may have $k = \text{polylog}(n)$ min-entropy.	$\Theta(1)$	$O(1)$	[Raz05]
2-source extractor	One source: $(0.5+\alpha)n$, $\alpha > 0$. Other source may have $k = \text{polylog}(n)$ min-entropy.	$\Theta(k)$	$2^{-\Omega(k)}$	[Raz05]
2-source extractor	$(0.5-\alpha_0)n$ for some universal constant $\alpha_0 > 0$	$\Theta(n)$	$2^{-\Omega(n)}$	[Bou05]
2-source disperser	δn, constant δ	$\Theta(1)$	0	[BKS+05]
$O(1/\delta)$-source extractor	$k = n^{\delta}$	$\Theta(k)$	$k^{-\Omega(1)}$	This work.
3-source extractor	One source: δn, any constant δ. Other sources may have $k = \text{polylog}(n)$ min-entropy.	$\Theta(k)$	$2^{-k^{\Omega(1)}}$	This work.
2-source disperser	δn, constant δ	$\Theta(n)$	$2^{-n^{\Omega(1)}}$	This work.
2-source disperser	$n^{o(1)}$	1	0	[BRSW06]

Table 1: Performance of recent extractors and dispersers for independent sources

1.2.1 New Results

- The main result of this paper is a polynomial time computable extractor that extracts k random bits from $O(\frac{\log n}{\log k})$ independent (n,k)-sources with error $1/n^c$ for any $k(n) > \log^4 n$ and any constant $c > 1$. An interesting setting of parameters is when $k = n^{\gamma}$ for some $0 < \gamma < 1$. In this case we get an extractor for a constant number of sources that extracts a constant fraction of the total min-entropy with polynomially small error. Formally, the theorem we will prove is the following:

  Theorem 1.1 (Main Theorem, Section 3). *For every constant $c > 0$ there exists a constant c' such that for every n, k with $k = k(n) = \Omega(\log^4 n)$ there exists a polynomial time computable function $\mathsf{IExt} : (\{0,1\}^n)^u \to \{0,1\}^k$ with $u \leq c' \frac{\log n}{\log k}$ s.t. if $X^1, X^2, \ldots, X^u$ are independent (n,k) sources then*

  $$|\mathsf{IExt}(X^1, \ldots, X^u) - U_k| < \epsilon$$

  with $\epsilon = 1/n^c$.

  Our construction is the first to extract from a constant number of sources that have polynomially small min-entropy. In fact we are not aware of previous constructions of explicit dispersers (or even explicit Ramsey Hypergraphs) of this type for such low min-entropy. Our extractor for a constant number of independent sources has the additional feature that it does not rely on results from additive number theory. It is built by composing existing strong seeded extractors in simple ways.

- It turns out that with hardly any extra effort, it is easy to use the ideas that go into the main theorem above to prove something stronger. We can even construct an extractor for just two independent sources, as long as the sources are guaranteed to be *block sources*. Block sources have been involved in many earlier works in extractors. Informally, a source is a block source if it can broken up into several blocks, such that every block has high enough min-entropy even conditioned on the event that all the previous blocks in the source are fixed to some value in their support. Two blocks in a block source aren't completely independent, but they do satisfy the property that the second block is hard to predict even if we know what the value of the first block is. The concatenation of several independent sources is of course a block source. Thus block-sources are a strictly more general class of sources than independent sources. It can be shown that there is no deterministic extractor for a single block source.

  In this paper, we give a polynomial time computable extractor that extracts $\Omega(k)$ random bits from 2 independent block-sources when each block source is required to have at most $O(\frac{\log n}{\log k})$ blocks of length n, where each block has min-entropy $k = k(n) > \log^4 n$ conditioned on the previous blocks. The error for the extractor is $1/n^{\Omega(1)}$. Formally:

  Theorem 1.2 (2 Block-Source Extractor, Section 4)**.** *For every n, k with $k > \log^4 n$, there exists a polynomial time computable function $\overline{\mathsf{IExt}} : \{0,1\}^{un} \times \{0,1\}^{un} \rightarrow \{0,1\}^k$ with $u = O(\frac{\log n}{\log k})$ s.t. , for constant $\gamma < 1$ if $X = X^1 \circ \cdots \circ X^u$ and $Y = Y^1 \circ \cdots \circ Y^u$ are independent $(k, \ldots, k)$ block-sources,*

  $$|\overline{\mathsf{IExt}}(X, Y) - U_k| < \epsilon$$

  where $\epsilon = 1/n^{\Omega(1)}$.

- Several constructions of Barak et al. [BKS+05] and Raz [Raz05] worked by first reducing the extractor/disperser problem they were solving to the case of extracting from independent somewhere random sources. Then they used brute force search to construct an extractor for this case. In this paper we obtain much better explicit extractors for independent somewhere random sources. This allows us to improve some of the results from earlier works.

  - We obtain a polynomial time computable 2-source disperser that outputs a linear number of bits with exponentially small error when the min-entropy rate of the source is an arbitrarily small constant, by modifying the construction of Barak et al. [BKS+05], which had constant output length.
  - We construct a polynomial time computable extractor for 3 sources that extracts a linear number of bits with exponentially small error when one source has min-entropy rate that is an arbitrarily small constant and the other two may have min-entropy that is polylogarithmically small, by modifying the construction of Raz [Raz05], which had constant error and constant output length.

Remark 1.3. Recently Ronen Shaltiel showed how to improve the output length of all of the above extractors [Sha05]. His techniques show how to get extractors which output $k - o(k)$ output bits, where now k is the *total* entropy in all sources, by paying a small price in the error of the extractors.

1.2.2 Techniques

Many extractor constructions in the past have been based on the paradigm of iterative condensing [RSW00, TSUZ01, CRVW02, LRVW03, BIW04]. The idea is to start with some distribution that has low min-entropy and apply a function (called a *condenser*) whose output has a better min-entropy rate. Repeating this process, we eventually obtain a distribution which has very high min-entropy rate. Then we can apply some other extractor which works for such a high min-entropy rate to obtain random bits. The extractor in this paper can also be viewed as an example of this paradigm, with a slight twist.

We make progress by considering a more restricted model for sources called *somewhere random* sources (SR-sources for short). SR-sources were first introduced by by Ta-Shma [TS96]. They have been used in several earlier works on extractors. A source is a $(t \times r)$ SR-source if the bits that it gives can be divided into t rows, each of length r, such that at least one row contains uniformly random bits. The other rows may depend on the uniform row in arbitrary ways. SR-sources are general enough so that there is no deterministic extractor for SR-sources. An important concept that we introduce is that of *aligned* SR-sources. Two SR-sources with the same number of rows are said to be *aligned* if there is an i such that the i'th row of both sources are distributed uniformly.

We will think of the number of rows of an SR-sources as a measure of the quality of the source. The fewer the number of rows, the better the quality is. Our construction will manipulate SR-sources. We will iteratively improve the quality (reduce the number of rows) of the SR-sources that we are working with until extracting randomness from them becomes easy.

Our construction will use *strong seeded extractors* as a basic tool. A strong seeded extractor can be viewed as a small family of deterministic functions (each function in the family indexed by a unique seed), such that for any fixed adversarially chosen source of randomness, almost all functions from the family are good extractors for that source. Several constructions of strong seeded extractors with seed length $O(\log n)$ (giving a family of polynomially many functions) are known (e.g. [LRVW03, Tre01]).

Now we describe some observations that go into the construction. We will then show how to put these together to get the high level view of our extractor construction (Figure 1).

Idea 1: General Sources can be turned into aligned SR-sources. A strong seeded extractor can be used to convert any general weak source into an SR-source. Given a sample from the weak source, we simply evaluate the extractor on the sample with all possible seeds, getting one row of the output for each fixed seed. For any fixed weak source, the strong extractor property guarantees that most seeds will give a distribution that is statistically close to uniform. As long as the seed length required by the extractor is $O(\log n)$, we get a polynomial time algorithm to convert any weak source to a distribution that is statistically close to an SR-source with $\mathrm{poly}(n)$ rows. A simple union bound argument can be used to show that if we convert a constant number of independent sources to independent SR-sources in this way, the SR-sources we obtain are also aligned.

Idea 2: Extraction is easy from *high quality* independent aligned SR-sources. It is easy to extract from independent SR-sources when each source has very few rows relative to the length of each of the rows. In the extreme case, when an SR-source has just one row, it is a uniformly random string. A slightly more non-trivial example is when we have two independent aligned SR-sources, each with two rows. In this case it is easy to see that if we output the bitwise XOR of the first row of the first source with the second row of the second source, we get uniformly random bits. Building on these simple ideas, we will show how to build extractors for just 2 aligned SR-sources even when the number of rows is superconstant. We will be able to extract from such sources as long as the number of rows is significantly less than the length of each row.

Idea 3: Condensers for *low quality* SR-sources can be obtained via Idea 2. We build condensers for SR-sources in the following sense: given a few input independent aligned SR-sources, our

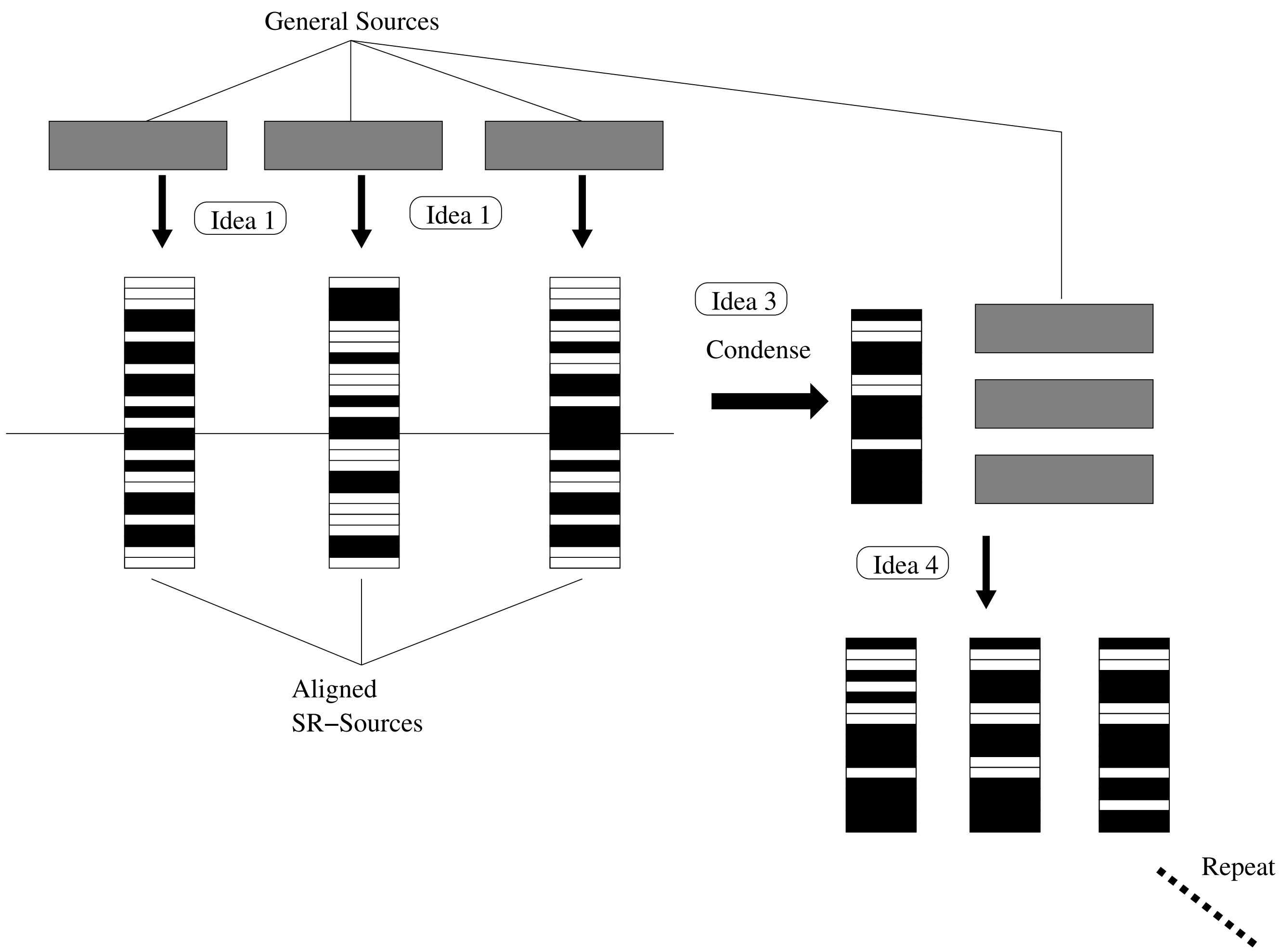

Figure 1: High level picture of the extractor

condenser's output is essentially the distribution of independent aligned SR-sources with fewer rows. Suppose we have a construction of an extractor for c aligned SR-sources with t' rows. Suppose we are given c aligned SR-sources, each with $t > t'$ rows. We can run our extractor with the first t' rows of all of the SR-sources to get a single output. Then we can repeat this with the next t' rows of each of the SR-sources. In this way we obtain t/t' outputs, one of which is guaranteed to be uniformly random, i.e. we obtain a new SR-source with t/t' rows. In this way, we obtain a condenser which given c independent SR-sources, outputs one SR-source with fewer rows.

Idea 4: The quality of SR-sources can be transferred. A single SR-source S with t rows can be used to convert many other independent sources into SR-sources with t rows. Simply use the t rows of S as seeds with a strong seeded extractor to extract from each of the other independent sources. With high probability, the random row of S is a good seed to extract from all the other independent sources simultaneously. It turns out that the output we obtain in this way is close to a convex combination of independent aligned SR-sources, each with t rows. This observation can be interpretted as a way to *transfer* quality from a single SR-source to many other independent sources.

Given these observations, the high level informal view of our extractor construction is the following:

1. Use **Idea 1** to convert a constant number of independent sources into SR-sources with $t = \mathrm{poly}(n)$ rows each.

2. Use **Idea 3** to condense these sources to get 1 SR-source S with much fewer rows t'. If $t' = 1$, stop and output the random row, else continue.

3. Using **Idea 4**, take a constant number of input independent sources and transfer the quality of S to these sources, to get a constant number of independent SR-sources with t' rows each.

4. Go to step 2.

The number of sources required depends on how quickly the number of rows in the SR-sources we are working with drop down to 1. We will give two condenser constructions. The first one is simpler (essentially based on the XOR extractor discussed in **Idea 2**), but only gives an extractor for $\log n$ sources. The second one is more involved, but gives an extractor for a constant number of sources when the min-entropy is polynomially small.

2 Preliminaries

2.1 Notation

Throughout this paper, we will use capital letters to denote distributions and sets. When it isn't ambiguous we will sometimes use the same capital letter to denote a distribution and its support. We will usually use the same small letter to denote an instantiation of the capital letter, for e.g. for a set X, we would use x to denote an element in X. If R is a random variable, we will write $r \in R$ when we really mean $r \in \mathsf{supp}(R)$.

Given a distribution X over $\{0,1\}^n$ and a set $S \subseteq [n]$, we will use X_S to denote the restriction of X onto the indices in S.

We use the convention that $N = 2^n$, $M = 2^m$ and $K = 2^k$.

All logarithms are meant to be base 2.

We will use U_m to denote the uniformly random distribution on $\{0,1\}^m$.

We will use the symbol $\circ$ to denote concatenation.

We will construct and compose several explicit extractors and condensers in this paper. We will usually use long names to make clear what kinds of sources the functions are meant to manipulate. When a function f can be applied to a very restricted family of sources (the restrictions may not be apparent from the naming), we will usually name it $\overline{f}$ to indicate that its use is restricted.

2.2 Min-Entropy and Special Sources

We will be concerned with the treatment of various kinds of distributions that are *nice* in that they contain a lot of usable randomness. Here we discuss some ways to measure this niceness:

Definition 2.1. The *min-entropy* of a random variable R is defined to be: $H_\infty(R) = -\log(\max_{x \in R}(R(x)))$. The *min-entropy rate* of a distribution R on $\{0,1\}^n$ is $H_\infty(R)/n$.

Definition 2.2. An (n,k)-*source* denotes some random variable X over $\{0,1\}^n$ with $H_\infty(X) \geq k$.

Definition 2.3. A distribution $X^1 \circ X^2 \circ \cdots \circ X^u$ is called a $(k_1, k_2, \ldots, k_u)$*-block-source* if for all $i = 1, \ldots, u$, we have that for all $x_1 \in X^1, \ldots, x_{i-1} \in X^{i-1}$, $H_\infty(X^i | X^1 = x_1, \ldots, X^{i-1} = x_{i-1}) \geq k_i$, i.e., each block has high min-entropy even conditioned on the previous blocks.

Proposition 2.4. *Let $X^1, \ldots, X^u$ be independent sources with $H_\infty(X^i) = k_i$ for $i = 1, \ldots, u$. Then $X^1 \circ \cdots \circ X^u$ is a $(k_1, k_2, \ldots, k_u)$-block-source.*

In some situations we will be interested in the min-entropy of a random variable when it is conditioned on *typical* instantiations. We will need the following proposition:

Proposition 2.5. *Let X be a random variable with $H_\infty(X) = k$. Let A be any event in the same probability space. Then*

$$H_\infty(X|A) < k' \Rightarrow \Pr[A] < 2^{k'-k}$$

Definition 2.6. A source X is $(t \times r)$ *somewhere-random*[2] (*SR-source* for short) if it is a random variable on t rows of $\{0,1\}^r$ s.t. X is distributed uniformly randomly over one of the rows. Every other row may depend on the random row in arbitrary ways.

Our constructions are obtained by manipulating somewhere-random sources in various ways. Often we will need to consider only a small subset of the bits of a somewhere random source.

Definition 2.7. Given a $(t \times r)$ somewhere random source X, for $w \leq r$, a *slice of width w* of X is the $(t \times w)$ somewhere random source obtained from X by restricting X to the first w bits in each of its rows.

Definition 2.8. We will say that a $(t \times r)$ source (i.e., a distribution on t rows of $\{0,1\}^r$) X has *somewhere-min-entropy* k, if X has min-entropy k in one of its rows.

2.3 Statistical Distance

Sometimes the distributions we get are not exactly the distributions we want, but they may be *close* enough. The measure of *closeness* we will use is this one:

Definition 2.9. Let D and F be two distributions on a set S. Their *statistical distance* is

$$|D - F| = \max_{T \subseteq S}(|D(T) - F(T)|) = \frac{1}{2} \sum_{s \in S} |D(s) - F(s)|$$

If $|D - F| \leq \epsilon$ we shall say that D is ϵ*-close* to F.

Proposition 2.10. *Let D and F be any two distributions over a set S s.t. $|D - F| \leq \epsilon$. Let g be any function on S. Then $|g(D) - g(F)| \leq \epsilon$.*

In a few of our proofs we will need to change a distribution that we are working with to a statistically close distribution while maintaining its independence from various other distributions.

[2] This definition is slightly different from the original one used by Ta-Shma [TS96]. The original definition considered the closure under convex combinations of the class defined here (i.e. convex combinations of sources which have one random row). We use this definition because we can do so without loss of generality and it considerably simplifies the presentation.

Proposition 2.11. *If $X^1, \ldots, X^l$ are independent random variables with $|X^i - Y^i| < \epsilon$, then $X^1 \circ X^2 \circ \cdots \circ X^l$ is $l\epsilon$-close to $Y^1 \circ Y^2 \circ \cdots \circ Y^l$ where the random variables Y^i are independent of each other.*

We will need the following lemma to reduce the error in the constructions.

Lemma 2.12. *[BIW04] Let $Z^1, \ldots, Z^v$ be independent distributions over $\{0,1\}^k$ with $|Z^i - U_k| < \epsilon$ for every $i = 1, \ldots, v$. Then*

$$|Z^1 \oplus Z^2 \oplus \cdots \oplus Z^v - U_k| < \epsilon^v$$

2.4 Convex Combinations

Definition 2.13. Let $\mathcal{P}$ be a property of sources. Let X be some random variable over some universe. We will say that X is a *convex combination* of sources with property $\mathcal{P}$ if there exists some random variable I over an arbitrary universe s.t. for all $i \in \mathsf{supp}(I)$, $X|I = i$ has property $\mathcal{P}$.

A key observation that is essential to our results is that random variables that are convex combinations of sources with some good property are usually good themselves. This is captured in the following easy propositions:

Proposition 2.14. *Let X, Z be random variables s.t. X is a convex combination of sources which are ϵ-close to Z. Then X is ϵ-close to Z.*

Proposition 2.15. *Let X, I be random variables s.t. X is a convex combination of random variables $\{X_i\}_{i \in I}$. Let f be some function s.t. for all $i \in I$, $f(X_i)$ is a convex combination of sources that have some property $\mathcal{P}$. Then $f(X)$ is a convex combination of sources that have property $\mathcal{P}$.*

2.5 Extractors and Dispersers

Definition 2.16. A function $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ is a *strong (k, ϵ) seeded extractor* if for any (n, k) source X and for Y chosen uniformly at random from $\{0,1\}^t$, we have

$$|Y \circ \mathsf{Ext}(X, Y) - Y \circ U_m| < \epsilon$$

where U_m is independent of Y.

Definition 2.17. A function $\mathsf{IExt} : (\{0,1\}^n)^u \to \{0,1\}^m$ is a (k, ϵ) *extractor for u independent sources* if for any independent (n, k) sources $X^1, \ldots, X^u$ we have

$$|\mathsf{IExt}(X^1, \ldots, X^u) - U_m| < \epsilon$$

Definition 2.18. A function $\mathsf{IExt} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ is a (k, ϵ) *2-source extractor* if for any independent (n, k) sources X, Y we have

$$|\mathsf{IExt}(X, Y) - U_m| < \epsilon$$

We will say that IExt is a *strong 2-source extractor* if

$$|Y \circ \mathsf{IExt}(X, Y) - Y \circ U_m| < \epsilon$$

where U_m is independent of Y.

Definition 2.19. A function $\mathsf{IDisp} : (\{0,1\}^n)^u \to \{0,1\}^m$ is an (k,ϵ) *u-source disperser* if for all sets $A_1, A_2, \ldots, A_u \subseteq \{0,1\}^n$, with $|A_1|, |A_2|, \ldots, |A_u| \geq 2^k$, $|\mathsf{IDisp}(A_1, \ldots, A_u)| \geq (1-\epsilon)2^m$.

Many of our extractors use previous extractor constructions as black boxes. The parameters of our results depend on the parameters of the extractors used as black boxes. Here we list the previous constructions that we will need to achieve the parameters claimed.

2.6 Seeded Extractors

Theorem 2.20. *[LRVW03] For any constant $\alpha \in (0,1)$, every $n \in \mathbb{N}$ and $k \leq n$ and every $\epsilon \in (0,1)$ where $\epsilon > exp(-\sqrt{k})$, there is an explicit (k,ϵ) seeded extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^{O(\log n + \log(n/k)\log(1/\epsilon))} \to \{0,1\}^{(1-\alpha)k}$.

Theorem 2.21. *[Tre01, RRV02] For every $n, k, m \in \mathbb{N}$ and $\epsilon > 0$, such that $m \leq k \leq n$, there is an explicit (k,ϵ)-strong seeded extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *with* $d = O\Big(\frac{\log^2(n/\epsilon)}{\log(k/m)}\Big)$.

We shall be interested in the following two instantiations of this theorem, obtained by setting the parameters appropriately:

Corollary 2.22. *[Tre01, RRV02] For every $n \in \mathbb{N}$, constants $r > 0, \gamma < 1$, there is an explicit (n^γ, n^{-r})-strong seeded extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{n^{\gamma'}}$ *with* $d = O(\log(n))$.

Corollary 2.23. *[Tre01, RRV02] For every $n, k \in \mathbb{N}$, there is an explicit (k,ϵ)-strong seeded extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{\Omega(k)}$ *with* $d = O(\log^2(n/\epsilon))$.

The first instantiation will be used when we need an extractor that has a good seed length. The second will be used when we need an extractor that has good output length.

If we need to get almost all of the randomness in the source out, the following corollary is available:

Corollary 2.24. *[Tre01, RRV02] For every $n, k \in \mathbb{N}$, $\epsilon > 0$, there is an explicit (k,ϵ)-strong extractor* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{k - O(\log^3(n/\epsilon))}$ *with* $d = O(\log^3(n/\epsilon))$.

2.7 Few Source Extractors

Ran Raz constructed an extractor that can extract when only one source is required to have min-entropy rate greater than half.

Theorem 2.25. *[Raz05] For any n_1, n_2, k_1, k_2, m and any $0 < \delta < 1/2$ s.t. ,*

- $n_1 \geq 6 \log n_1 + 2 \log n_2$
- $k_1 \geq (0.5 + \delta) n_1 + 3 \log n_1 + \log n_2$
- $m \leq \delta \min[n_1/8, b_2/40] - 1$

There is a polynomial time computable function $\mathsf{Raz} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to \{0,1\}^m$ *s.t. if X is an (n_1, k_1) source and Y is an independent (n_2, k_2) source,*

$$|X \circ \mathsf{Raz}(X,Y) - X \circ U_m| < \epsilon$$

and

$$|Y \circ \mathsf{Raz}(X,Y) - Y \circ U_m| < \epsilon$$

where $\epsilon = 2^{-1.5m}$.

Recently Jean Bourgain constructed a strong 2-source extractor for min-entropy rate slightly less than half.

Theorem 2.26. *[Bou05] There exists a polynomial time computable function* $\mathsf{Bou} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^m$ *and a universal constant* α_0 *s.t. if* X, Y *are independent* $(n, (1/2 - \alpha_0)n)$ *sources,*

$$|Y \circ \mathsf{Bou}(X, Y) - Y \circ U_m| \leq 2^{-\Omega(n)}$$

where $m = \Omega(n)$ *and* U_m *is independent of* Y*.*

2.8 Multisource Dispersers vs Ramsey Hypergraphs

Here we outline how to convert any efficiently computable multisource disperser into an explicit Ramsey Hypergraph.

Proposition 2.27. *Let* $\mathsf{IDisp} : (\{0,1\}^n)^u \to \{0,1\}$ *be a* (k, ϵ) *u-source disperser. Then* IDisp *can be used to give an explicit u-uniform Ramsey Hypergraph on* 2^n *vertices that avoids monochromatic cliques of size* $u2^k$*.*

Proof Sketch: Consider the u-uniform hypergraph defined as follows: given any potential edge $\{a_1, a_2, \ldots, a_u\}$ of the graph, first sort the vertices according to some predetermined total order to ensure that $a_1 \geq a_2 \geq \cdots \geq a_u$ in this order. Then color the edge red if $\mathsf{IDisp}(a_1, a_2, \ldots, a_u) = 0$, else color it blue.

Now let S be any subset of the vertices of this graph of size $u2^k$. Then we can use the total order to partition the vertices of S into u sets $S_1, \ldots, S_u$ of size 2^k by taking the highest 2^k vertices in the total order as S_1, then the next 2^k vertices as S_2 and so on. By the disperser property of IDisp, $\mathsf{IDisp}(S_1, \ldots, S_u) = \{0,1\}$. Thus S contains hyperedges of both colors.

□

3 Extracting from Independent Sources by Condensing SR-Sources

Our main result is a new deterministic extractor that can extract from a constant number of independent sources which have min-entropy that is polynomially small in their length. It turns out that the extractor we build can actually handle a slightly more general class of sources. We can extract from just 2 independent sources, where each source is a block-source with a constant number of blocks. To simplify the presentation, we will first present the extractor assuming that we have access to truly independent sources. In the next section we will show how to prove that the extractor succeeds even when given just two independent block-sources.

Our algorithms will repeatedly condense SR-sources. Starting with a number of independent SR-sources, we will iteratively reduce the number of rows in each of the sources, until the number of rows is so small that extracting randomness becomes easy.

In this section we will prove the main theorem of this paper, which we restate here.

Theorem 3.1 (Main Theorem)**.** *For every constant* $c > 0$ *there exists a constant* c' *such that for every* n, k *with* $k = k(n) = \Omega(\log^4 n)$ *there exists a polynomial time computable function* $\mathsf{IExt} : (\{0,1\}^n)^u \to \{0,1\}^k$ *with* $u \leq c' \frac{\log n}{\log k}$ *s.t. if* $X^1, X^2, \ldots, X^u$ *are independent* (n, k) *sources then*

$$|\mathsf{IExt}(X^1, \ldots, X^u) - U_k| < \epsilon$$

with $\epsilon = 1/n^c$*.*

Setting the parameters appropriately gives the following corollary:

Corollary 3.2. *For every constant $c > 0$ and $\gamma < 1$ there exists a polynomial time computable function* $\mathsf{IExt} : (\{0,1\}^n)^u \to \{0,1\}^{n^\gamma}$ *with u some large constant s.t. if $X^1, X^2, \dots, X^u$ are independent (n, n^γ) sources then*

$$|\mathsf{IExt}(X^1, \dots, X^u) - U_{n^\gamma}| < \epsilon$$

with $\epsilon = 1/n^c$.

Our first step will be to convert each of the sources to an SR-source. The following proposition, which we state without proof, shows how to do such a conversion.

Proposition 3.3. *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded (k, ϵ) strong extractor. Let X be any (n, k) source. Let $\{0,1\}^d = \{s_1, s_2, \dots, s_{2^d}\}$. Then* $\mathsf{Ext}(X, s_1) \circ \mathsf{Ext}(X, s_2) \circ \dots \circ \mathsf{Ext}(X, s_{2^d})$ *is ϵ-close to a $(2^d \times m)$ SR-source.*

Using any good seeded strong extractor with seed length $O(\log n)$, we can do the conversion in polynomial time.

Definition 3.4. We will say that a collection of SR-sources $X^1, \dots, X^u$ is *aligned* if there is some i for which the i'th row of every SR-source in the collection is uniformly distributed.

If the strong extractor that we used to convert the input general sources to SR-sources has error ϵ, at most $\sqrt{\epsilon}$ fraction of the rows in each source are not $\sqrt{\epsilon}$-close to uniform. Thus, if we are given u sources, as long as $u\sqrt{\epsilon} < 1$, we will have one aligned row in *every* source which is $\sqrt{\epsilon}$-close to uniform. Using Proposition 2.11 these sources are $u\sqrt{\epsilon}$-close to being the distribution of independent aligned SR-sources.

Proposition 3.5. *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be a seeded (k, ϵ) strong extractor. Let $X^1, \dots, X^u$ be independent (n, k) sources, with $u\sqrt{\epsilon} < 1$. Let $\{0,1\}^d = \{s_1, s_2, \dots, s_{2^d}\}$. Let Z^i denote* $\mathsf{Ext}(X^i, s_1) \circ \mathsf{Ext}(X^i, s_2) \circ \dots \circ \mathsf{Ext}(X^i, s_{2^d})$. *Then $Z^1 \circ \dots \circ Z^u$ is $u\sqrt{\epsilon}$-close to the distribution of u independent aligned $(2^d \times m)$ SR-sources.*

If Ext is a strong seeded extractor with seed length $O(\log n)$ and output length m, we can use Proposition 3.5 to reduce the problem of extracting from independent sources to the problem of extracting from aligned independent $(\mathrm{poly}(n) \times m)$ SR-sources. It turns out that it is easy to extract from independent aligned SR-sources when each SR-source contains very few rows.

The rest of this section is organized in the following way:

1. We will describe a couple of ways to extract from a few independent aligned $(c \times n)$ SR-sources when c is a constant.

2. We will show how to use the extractors from the previous step to build condensers for independent aligned SR-sources. We will use the condensers to give a basic extractor that can extract from $O(\log n)$ independent aligned SR-sources which have $\mathrm{poly}(n)$ rows. Using Proposition 3.5, this will give an extractor for $O(\log n)$ general independent sources.

3. We will add a few more tricks to bring down the number of sources required to $O(\log n / \log k)$.

3.1 Strong Extractors for independent aligned $(2 \times n)$ SR-sources

If we were given just 2 independent aligned $(2 \times n)$ SR-sources X^1 and X^2, it is easy to see that

$$|X^1_{\{1\}\times[n]} \oplus X^2_{\{2\}\times[n]} - U_n| = 0$$

i.e. to get random bits we just have to XOR the first row from the first source with the second row from the second source.

We will actually need something stronger: a strong extractor for such sources. We can get such a strong extractor for 3 $(2 \times n)$ aligned independent SR-sources by composing the XOR function with a strong seeded extractor. The following theorem is easy to see. We state it without proof.

Theorem 3.6. *Let* $\mathsf{Ext} : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *be an* (k, ϵ) *strong seeded extractor. Let* $\overline{\mathsf{BasicSRExt}} : \{0,1\}^n \times \{0,1\}^{2d} \times \{0,1\}^{2d} \to \{0,1\}^m$ *be defined as* $\overline{\mathsf{BasicSRExt}}(x, y^1, y^2) = \mathsf{Ext}(x, y^1_{\{1\}\times[d]} \oplus y^2_{\{2\}\times[d]})$*. Then if* X, Y^1, Y^2 *are independent sources, with* X *an* (n, k) *source and* Y^1, Y^2 *aligned* $(2 \times d)$ *SR-sources,*

$$|\overrightarrow{Y} \circ \overline{\mathsf{BasicSRExt}}(X, \overrightarrow{Y}) - \overrightarrow{Y} \circ U_m| < \epsilon$$

where $\overrightarrow{Y}$ *denotes* $Y^1 \circ Y^2$ *and* U_m *is independent of* $\overrightarrow{Y}$*.*

Claim 3.7. *Let* X *and* Y *be as in Theorem 3.6. Then*

$$\Pr_{\vec{y} \leftarrow_R \overrightarrow{Y}}[|\overline{\mathsf{BasicSRExt}}(X, \vec{y}) - U_m| \geq \sqrt{\epsilon}] < \sqrt{\epsilon}$$

To give an example of the kinds of parameters that can be achieved, if we start with 3 independent aligned $(2 \times n)$ SR-sources and use the extractor promised by Corollary 2.24, we can get $n - o(n)$ random bits out, with error that is exponentially small in n. Setting parameters appropriately, we can get the following corollaries.

Corollary 3.8. *Let* Ext *be the extractor from Corollary 2.24. For all* n, k, d *with* $d > \log^4(n/\epsilon)$*,* $\overline{\mathsf{BasicSRExt}}$ *can be set up to output* $m = k - O(\log^3(n/\epsilon))$ *random bits with error* ϵ*.*

Corollary 3.9. *Let* Ext *be the extractor from Corollary 2.24. For all* n, k, d *with* $d > \log^7(n)$*,* $\overline{\mathsf{BasicSRExt}}$ *can be set up to output* $m = k - O(\log^6 n)$ *random bits with error* $\epsilon < 2^{-\log^2 n}$*.*

Corollary 3.10. *Let* Ext *be the extractor from Corollary 2.24. For all* n, k, d*, there is a constant* $\gamma < 1$ *s.t. as long as* $d = n^\gamma$*,* $\overline{\mathsf{BasicSRExt}}$ *can be set up to output* $m = k - \sqrt{n}$ *random bits with error* $\epsilon < 2^{-n^{\Omega(1)}}$*.*

Another way to get a strong extractor from just two SR-sources of this type is to use Bourgain's recent extractor Theorem 2.26. This already gives a strong extractor for two SR-sources with min-entropy rate half. However Bourgain's extractor extracts only a constant fraction of the randomness. We can remedy this by composing it with a strong seeded extractor to get almost all the randomness out. We state the following theorem without proof.

Theorem 3.11. *Let* $\mathsf{Ext} : \{0,1\}^{2n} \times \{0,1\}^d \to \{0,1\}^m$ *be a* (k, ϵ) *strong seeded extractor. Let* Bou *and* α_0 *be as in Theorem 2.26. Let* n, k' *be such that* $n - 100k' > k$ *and* d *be the output length of* Bou *when applied to two independent* $(2k', k')$ *sources. Let* $\overline{\mathsf{Basic2SRExt}} : \{0,1\}^{2n} \times \{0,1\}^{2n} \to \{0,1\}^m$ *be defined as* $\overline{\mathsf{Basic2SRExt}}(X, Y) = \mathsf{Ext}(X_{\{1,2\}\times[n]}, \mathsf{Bou}(Y_{\{1,2\}\times[k']}, X_{\{1,2\}\times[k']}))$*. Then if* X, Y *are independent aligned* $(2 \times n)$ *SR-sources,*

$$|Y \circ \overline{\mathsf{Basic2SRExt}}(X,Y) - Y \circ U_m| < \epsilon + 2^{-\Omega(k')}$$

where U_m is independent of Y.

As with the previous construction, depending on the parameters of the strong seeded extractor used, we can get various tradeoffs between the error and the output length.

Claim 3.12. *Let X and Y be as in Theorem 3.11. Then*

$$\Pr_{y \leftarrow_R Y}[|\overline{\mathsf{Basic2SRExt}}(X,y) - U_m| \geq \sqrt{\epsilon + 2^{-\Omega(k')}}] < \sqrt{\epsilon + 2^{-\Omega(k')}}$$

3.2 Warm-up: Extracting randomness from $O(\log n)$ independent sources

To illustrate some of the ideas behind the final few source extractor, we will first describe how to extract from $O(\log n)$ independent sources. The final extractor construction will be slightly more involved but will use the major ideas that we develop here.

In this section, we will prove the following theorem:

Theorem 3.13. *For every n, k with $k = k(n) = \Omega(\log^4 n)$, there exists a polynomial time computable function $\mathsf{IExt}' : (\{0,1\}^n)^u \to \{0,1\}^m$ s.t. $m = \Omega(k)$, $u = O(\log(n))$ and if $X^1, X^2, \dots, X^u$ are independent (n,k) sources then*

$$|\mathsf{IExt}'(X^1, \dots, X^u) - U_m| < \epsilon$$

where $\epsilon = 1/n^{\Omega(1)}$.

As discussed, we will start by converting the independent (n,k) sources to distributions which are statistically close to being independent aligned $(n^{O(1)} \times m)$ SR-sources using Proposition 3.5. Here m is the output length of the strong seeded extractor used in the conversion [3].

Our extractor will then be obtained by iteratively condensing the original distribution. We will start with $O(\log n)$ independent aligned $(n \times k)$ SR-sources. In each step we will consume 2 SR-sources, but will reduce the number of rows in each of the other SR-sources by a *factor* of 2 [4]. After $O(\log n)$ steps, we will have reduced the number of rows to 2. We can then use the XOR function to extract random bits. Now we describe one condensing step in detail.

Let X_j denote the j'th pair of rows of the SR-source X.

Construction: $\mathsf{ICond}(x^1, \dots, x^u)$
Input: $x^1, \dots, x^u$, a sample from u $(t \times r)$ SR-sources.
Output: $z^1, \dots, z^{u-2}$.
Let $\overline{\mathsf{BasicSRExt}}$ be as in Theorem 3.6.

1. For $1 \leq i \leq u-2$, $1 \leq j \leq t/2$ let $z^i_j = \overline{\mathsf{BasicSRExt}}(x^i_j, x^{u-1}_j, x^u_j)$.

2. For $1 \leq i \leq u-2$, let z^i be the SR-source whose rows are $z^i_1, z^i_2, \dots, z^i_{t/2}$.

[3] There is a tradeoff between m and the error incurred in the conversion. This tradeoff in general depends on the relationship between n and k. To give a feel for the parameters, if $k = n^\gamma$ for some $\gamma \in (0,1)$, we can do the conversion with $m = k - o(k)$ and error that is polynomially small.

[4] Actually, in each step we will convert SR-sources with t rows into SR-sources with $\lceil t/2 \rceil$ rows. To simplify the presentation we will assume that t is always even, this does not really affect any of the claims made

Lemma 3.14. *If $X^1, \dots, X^u$ are independent aligned $(t \times r)$ SR-sources, $Z^1, \dots, Z^{u-2}$ are $2u\sqrt{\epsilon}$-close to being a convex combination of independent aligned $(t/2 \times m)$ SR-sources, where m and ϵ are the output length and error of $\overline{\mathsf{BasicSRExt}}$.*

Assuming this lemma, we can prove the theorem for this section.

Proof of Theorem 3.13. If we use the strong SR-source extractor promised by Corollary 3.10, after applying the condenser $O(\log n)$ times, we will have reduced the number of rows in each of the sources to 2 and the length of each of the sources will still be $k - O(\sqrt{k} \log n)$. We can then use the XOR function to get a distribution which is statistically close to uniform. The error adds in each step, but since the error of $\overline{\mathsf{BasicSRExt}}$ can be made as small as $2^{-k^{\Omega(1)}}$, for large k this does not affect the final error. The dominant error comes in the first step, when we convert the general sources to SR-sources. This concludes the proof of Theorem 3.13. ■

Proof of Lemma 3.14. Let $\overrightarrow{Y}$ denote the concatenation of X^{u-1}, X^u. Let h be s.t. the h'th pair of rows X^i_h in each of the sources X^i contains the truly random row. Let $\overrightarrow{Y_h}$ denote the concatenation of X^{u-1}_h, X^u_h. Let m be the output length of $\overline{\mathsf{BasicSRExt}}$. We will prove the lemma by partitioning the support of $\overrightarrow{Y}$ into a good set and a bad set s.t.

Claim 3.15. *For good $\vec{y}$, the distribution $Z^1|\overrightarrow{Y} = \vec{y} \circ \cdots \circ Z^{u-2}|\overrightarrow{Y} = \vec{y}$ is $u\sqrt{\epsilon}$-close to being a collection of independent aligned $(\lceil t/2 \rceil \times m)$ SR-sources.*

Claim 3.16. $\Pr[\overrightarrow{Y}$ *is not good*$] < u\sqrt{\epsilon}$

We will call $\vec{y}$ good for X^i if

$$|\overline{\mathsf{BasicSRExt}}(X^i_h, \vec{y}_h) - U_m| < \sqrt{\epsilon}$$

We will call $\vec{y}$ good if it is good for all $1 \le i \le u-2$.

Since $\overline{\mathsf{BasicSRExt}}$ is a strong extractor, for any i, at most a $\sqrt{\epsilon}$ fraction of the seeds are bad for X^i by Claim 3.7. The second claim then follows by the union bound.

For any fixed $\vec{y}$, $Z^1|\overrightarrow{Y} = \vec{y}, \dots, Z^u|\overrightarrow{Y} = \vec{y}$ are independent. When $\vec{y}$ is good, each of the Z^i's is $\sqrt{\epsilon}$-close to being a $(t/2 \times m)$ SR-source, with the hth row being the random one. By the union bound and Proposition 2.11 we get the first claim and the lemma follows.

□

3.3 Extracting from fewer sources

In this section we will prove Theorem 3.1. We will obtain the final extractor in the following steps.

1. We will show how to extract from 3 independent aligned $(n^\gamma \times n)$ SR-sources for any constant $\gamma < 1$.
2. We will show how to use the extractor from the previous step to extract from $O(\frac{\log n}{\log k})$ independent (n, k) sources when $k > \log^4 n$.

3.3.1 Extracting from 3 independent aligned $(n^\gamma \times n)$ SR-sources, for $\gamma < 1$

Theorem 3.17. *For every constant $\gamma < 1$ there exists a polynomial time computable function $\overline{\mathsf{3SRExt}} : (\{0,1\}^{n^{\gamma+1}})^3 \to \{0,1\}^m$ s.t. for if X^1, X^2, X^3 are independent aligned $(n^\gamma \times n)$ SR-sources,*

$$|\overline{\mathsf{3SRExt}}(X^1, X^2, X^3) - U_m| < 2^{-n^{\Omega(1)}}$$

where $m = n - O(n^\beta)$ for some $\beta \in (0,1)$.

Our starting point is the extractor that can extract from $O(\log n)$ sources. We'd like to do more or less the same thing in this situation, but somehow *reuse* the sources. As in that situation, our extractor will work by repeated condensing, but this time we will not discard any sources. Starting with 3 independent aligned SR-sources, in each step we will output a distribution that is a convex combination of 3 independent aligned SR-sources. The number of rows in each of the sources will be reduced by a factor of 2 and the length of each row will be reduced by a little bit.

Intuitively what we will try and do is: to condense the i'th source, we will get the other two sources to conspire against it. We will use the strong independent aligned $(2 \times n)$ SR-sources extractor of Theorem 3.6 on the i'th source, with small slices of the other sources as 'seed'. Conditioned on all the small sections of the sources that we've used as seed, we show that the condensing succeeds, we obtain 3 new SR-sources which have half the number of rows as the original sources. Since we're conditioning on the only part that's involved in the interactions between the sources, after conditioning, the output of the condensing step is a collection of independent sources. Iterating this condensing process, we will eventually obtain a single string that is statistically close to uniformly distributed.

Now we describe one condensing step in detail. As in the previous section, we will assume that t is even.

We are given: X^1, X^2, X^3, independent aligned $(t \times r)$ SR-sources.

Let w and l be parameters that we will choose later (we will have to set w to roughly polylog(r) and l to roughly r^μ for some constant $\mu < 1$). Let $S_1 = \{1,2\}\times[w], S_2 = \{3,4\}\times[w], \ldots, S_{t/2} = \{t-1,t\}\times[w]$.

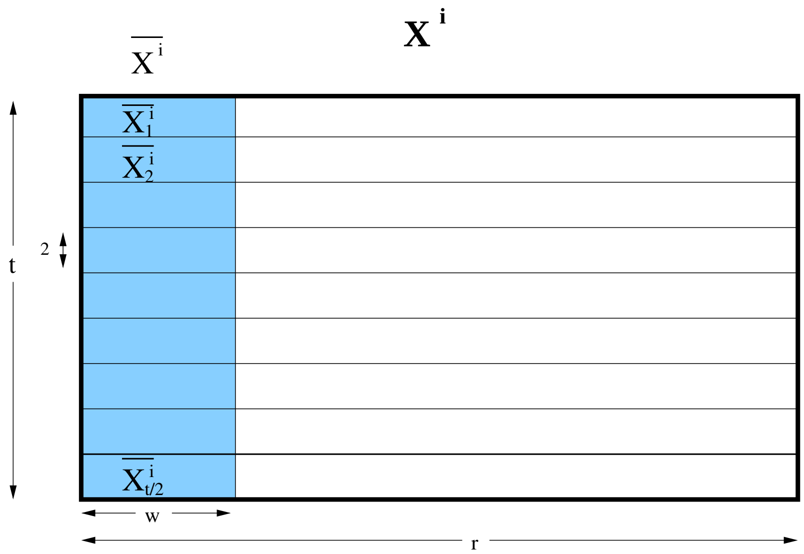

Figure 2: Notation in one source

We will adhere to the following notational guidelines:

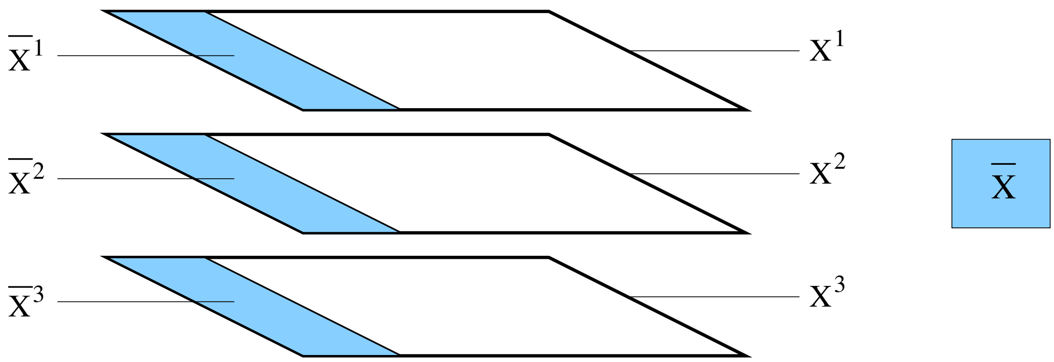

Figure 3: The region $\overline{X}$

- The superscript of an expression (if any) indicates which one of the independent sources we are referring to.
- The subscript of an expression (if any) indicates which of the rows we are referring to.
- An over-line (for example: $\overline{X}$) indicates whether we are referring to the entire row or just a small section of the rows.

For all i, j, we introduce the following notation:

- $\overline{X_j^i} = X_{S_j}^i$, a small slice of the jth pair of rows in X^i.
- $\overline{X^i} = X_{[t]\times[w]}^i = \overline{X_1^i} \circ \cdots \circ \overline{X_{t/2}^i}$, a small slice of X^i.
- $\overline{X} = \overline{X^1} \circ \overline{X^2} \circ \overline{X^3}$.
- $\overline{X_j} = \overline{X_j^1} \circ \overline{X_j^2} \circ \overline{X_j^3}$.
- $\overline{X_j^{\neq i}}$ denotes the concatenation of $\overline{X_j^v}$ for all $v \neq i$.

Construction: $\overline{\mathsf{ICond}}(x^1, x^2, x^3)$
Input: x^1, x^2, x^3, a sample from independent aligned $(t \times r)$ SR-sources.
Output: z^1, z^2, z^3.
Let $\overline{\mathsf{BasicSRExt}}$ be the extractor that can extract from 3 independent sources A, B, C when A is a $(tr, r - l)$ source and B, C are independent aligned $(2 \times w)$ sources promised by Theorem 3.6.

1. For all i, j let $z_j^i = \overline{\mathsf{BasicSRExt}}(x^i, \overline{x_j^{\neq i}})$.
2. For all i, let x^i be the SR-source whose rows are $z_1^i, z_2^i, \ldots, z_{t/2}^i$.
3. For all i, output z^i.

Lemma 3.18. *Let $\overline{\mathsf{ICond}}$ be as above. If X^1, X^2, X^3 are independent aligned $(t \times r)$ SR-sources, Z^1, Z^2, Z^3 obtained by $\overline{\mathsf{ICond}}$ are $3(2\sqrt{\epsilon} + 2^{-(l-tw)})$-close to being a convex combination of independent aligned $(t/2 \times m)$ SR-sources, where m and ϵ are the output length and error of $\overline{\mathsf{BasicSRExt}}$.*

Proof. We will roughly follow the proof for the situation in which we had $O(\log n)$ sources. There are a few additional complications that we have to deal with now.

Let h be s.t. S_h contains the truly random row. We will prove the lemma by partitioning the support of $\overline{X}$ into a good set and a bad set s.t.

Claim 3.19. *For good* $\overline{x}$*, the distribution* $(Z^1|\overline{X}=\overline{x}) \circ (Z^2|\overline{X}=\overline{x}) \circ (Z^3|\overline{X}=\overline{x})$ *is* $3\sqrt{\epsilon}$*-close to being a collection of independent aligned* $(t/2 \times m)$ *SR-sources.*

Claim 3.20. $\Pr[\overline{X}$ *is not good*$] < 3(\sqrt{\epsilon} + 2^{-(l-tw)})$.

Here we will need a more involved notion of good. For any fixed i, we will say $\overline{x}$ is good for i if

$$|\overline{\mathsf{BasicSRExt}}(X^i|\overline{X^i}=\overline{x^i}, \overline{x_h^{\neq i}}) - U_m| < \sqrt{\epsilon}$$

We will call $\overline{x}$ good if it is good for all i.

For any fixed $\overline{x}$, $Z^1|\overline{X}=\overline{x}, Z^2|\overline{X}=\overline{x}, Z^3|\overline{X}=\overline{x}$ are independent. When $\overline{x}$ is good, we have that for any i,

$$\begin{aligned}(Z_h^i|\overline{X}=\overline{x}) &= \overline{\mathsf{BasicSRExt}}(X^i|\overline{X}=\overline{x}, \overline{x_h^{\neq i}}) \\ &= \overline{\mathsf{BasicSRExt}}(X^i|\overline{X^i}=\overline{x^i}, \overline{x_h^{\neq i}}) \qquad \text{since } X^i \text{ is independent of } X^j \text{ for } j \neq i\end{aligned}$$

which is $\sqrt{\epsilon}$-close to the uniform distribution by our notion of *good*. Thus we get that Z^i is $\sqrt{\epsilon}$-close to being a $(t/2 \times m)$ SR-source, with the hth row being the random one. This proves the first claim.

Now we prove the second claim.

Proof of Claim 3.20. We will first bound the probability that $\overline{X}$ is bad for a fixed i and then use the union bound to bound the probability that it is bad for all i.

Intuitively there are two ways in which $\overline{X}$ can be bad for i. Either the bits in $\overline{X}$ that came from X^i stole too much entropy from X^i, or the bits of $\overline{X}$ that are not from X^i failed to produce a good seed to extract from $X^i|\overline{X}^i=\overline{x}^i$. Both of these events happen with extremely small probability, so we can use the union bound to say that with high probability neither occurs.

In the following statements, we will explicitly state what the probabilities are over to avoid confusion. The probability we are trying to bound is this one:

$$\begin{aligned}&\Pr_{\overline{x} \leftarrow_{\mathrm{R}} \overline{X}}[\overline{x} \text{ is bad for } i] \\ &= \Pr_{\overline{x} \leftarrow_{\mathrm{R}} \overline{X}}[|\overline{\mathsf{BasicSRExt}}((X^i|\overline{X^i}=\overline{x^i}), \overline{x_h^{\neq i}}) - U_m| \geq \sqrt{\epsilon}]\end{aligned}$$

We will rewrite this probability in this way:

$$\begin{aligned}&\Pr_{\overline{x} \leftarrow_{\mathrm{R}} \overline{X}}[\overline{x} \text{ is bad for } i] \\ &= \sum_{p \in \mathsf{supp}(\overline{X^i})} \Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X}|\overline{X}^i=p)}[\overline{x} \text{ is bad for } i] \Pr_{v \leftarrow_{\mathrm{R}} \overline{X}^i}[v = p]\end{aligned}$$

We'd like to argue that for every term in this sum, either the first probability is small, or the second probability is small. To this end, we partition the support of $\overline{X}^i$ into two sets. Recall that $H_\infty(X^i) \geq r$. We will say p is *atypical* if $H_\infty(X^i | \overline{X^i} = p) < r - l$. Otherwise we will say that p is *typical*. By Proposition 2.5,

Claim 3.21. *For* atypical *p,* $\Pr_{v \leftarrow_R \overline{X}^i}[v = p] < 2^{-l}$.

On the other hand, when p is *typical*,

$$\begin{aligned}
&\Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X} | \overline{X^i} = p)} [\overline{x} \text{ is bad for } i] \\
&= \Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X} | \overline{X^i} = p)} [|\overline{\mathsf{BasicSRExt}}(X^i | \overline{X^i} = p, \overline{x_h^{\neq i}}) - U_m| \geq \sqrt{\epsilon}] \\
&= \Pr_{\overline{x'} \leftarrow_{\mathrm{R}} (\overline{X_h^{\neq i}} | \overline{X^i} = p)} [|\overline{\mathsf{BasicSRExt}}(X^i | \overline{X^i} = p, \overline{x'}) - U_m| \geq \sqrt{\epsilon}]
\end{aligned}$$

Now observe that

$$\begin{aligned}
&\overline{X_h^{\neq 1}} | \overline{X^1} = p \\
&= \overline{X_h^2} | \overline{X^1} = p \circ \overline{X_h^3} | \overline{X^1} = p \\
&= \overline{X_h^2} \circ \overline{X_h^3}
\end{aligned}$$

since $\overline{X_h^2}$ and $\overline{X_h^3}$ are independent of $\overline{X^1}$. We get similar statements for $\overline{X_h^{\neq 2}} | \overline{X^2} = p$ and $\overline{X_h^{\neq 3}} | \overline{X^3} = p$. Therefore, for p that is *typical*, using Claim 3.7 we have,

$$\begin{aligned}
&\Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X} | \overline{X^i} = p)} [\overline{x} \text{ is bad for } i] \\
&= \Pr_{\overline{x'} \leftarrow_{\mathrm{R}} \overline{X_h^{\neq i}}} [|\overline{\mathsf{BasicSRExt}}(X^i | \overline{X^i} = p, \overline{x'}) - U_m| \geq \sqrt{\epsilon}] \\
&< \sqrt{\epsilon}
\end{aligned}$$

Thus,

Claim 3.22. *For* typical *p,* $\Pr_{\overline{x} \leftarrow_R (\overline{X} | \overline{X^i} = p)}[\overline{x}$ *is bad for* $i] < \sqrt{\epsilon}$.

Going back to the quantity we were trying to bound,

$$\begin{aligned}
&\Pr_{\overline{x} \leftarrow_{\mathrm{R}} \overline{X}} [\overline{x} \text{ is bad for } i] \\
&= \sum_{p \in \mathsf{supp}(\overline{X^i})} \Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X} | \overline{X^i} = p)} [\overline{x} \text{ is bad for } i] \Pr_{v \leftarrow_{\mathrm{R}} \overline{X^i}} [v = p] \\
&= \sum_{p \text{ is typical}} \Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X} | \overline{X^i} = p)} [\overline{x} \text{ is bad for } i] \Pr_{v \leftarrow_{\mathrm{R}} \overline{X^i}} [v = p] + \sum_{p \text{ is atypical}} \Pr_{\overline{x} \leftarrow_{\mathrm{R}} (\overline{X} | \overline{X^i} = p)} [\overline{x} \text{ is bad for } i] \Pr_{v \leftarrow_{\mathrm{R}} \overline{X^i}} [v = p] \\
&\leq \sqrt{\epsilon} + 2^{-l} 2^{tw}
\end{aligned}$$

The first sum was bounded using the fact that $\sum_p \Pr_{v \leftarrow_{\mathrm{R}} \overline{X^i}}[v = p] \le 1$ and Claim 3.22. The second sum was bounded using the fact that the total number of possible *atypical* p's is at most 2^{tw} and Claim 3.21. Using the union bound over all i, Claim 3.20 follows.

$\square$

This concludes the proof of Lemma 3.18. $\square$

Proof of Theorem 3.17. Let $\overline{\mathsf{3SRExt}}$ be the following function.

Construction: $\overline{\mathsf{3SRExt}}(x^1, x^2, x^3)$
Input: x^1, x^2, x^3, independent aligned $(n^\gamma \times n)$ SR-sources, $\gamma \in (0,1)$.
Output: z.

1. Repeatedly condense the sources using $\overline{\mathsf{ICond}}$ from Lemma 3.18 until each source has just one row.
2. Output z, the row from the first source.

Since we need to repeat the condensation step at most $\lceil \log n \rceil$ times, by Lemma 3.18 the final error is $O((\sqrt{\epsilon} + 2^{-(l-tw)}) \log n)$. If we use $\overline{\mathsf{BasicSRExt}}$ as in Corollary 3.8, the final output length is at least $n - O(l \log^4(n/\epsilon))$.

Setting $l = 2n^{(1+\gamma)/2}$, $w = l/(2t)$ and $\epsilon = 2^{-n^{\Omega(1)}}$, we get a total error of $2^{-n^{\Omega(1)}}$ with final output length at least $n - n^\beta$ for some $\beta \in (0,1)$.

$\blacksquare$

Replacing the extractor from Theorem 3.6 with Bourgain's extractor from Theorem 3.11, we can extract from just 2 independent aligned $(n^\gamma \times n)$ SR-sources for any $\gamma \in (0,1)$. In addition, we can actually show that the extractor is strong. To summarize, we obtain the following theorem, which we state without proof:

Theorem 3.23. *For every constant $\gamma < 1$ and n, n', t with $t = t(n, n')$ s.t. $t < n^\gamma$ and $t < n'^\gamma$ there exists a constant $\alpha < 1$ and a polynomial time computable function $\overline{\mathsf{2SRExt}} : \{0,1\}^{tn} \times \{0,1\}^{tn'} \to \{0,1\}^m$ s.t. if X is a $(t \times n)$ SR-source and Y is an independent aligned $(t \times n')$ SR-source,*

$$|Y \circ \overline{\mathsf{2SRExt}}(X, Y) - Y \circ U_m| < \epsilon$$

and

$$|X \circ \overline{\mathsf{2SRExt}}(X, Y) - X \circ U_m| < \epsilon$$

where U_m is independent of X, Y, $m = \min[n, n'] - \min[n, n']^\alpha$ and $\epsilon = 2^{-\min[n,n']^{\Omega(1)}}$.

Remark 3.24. Using Bourgain's extractor here seems like overkill. Bourgain's extractor can extract from *any* 2 sources with min-entropy rate slightly less than half, where as our sources have a lot of structure. It would be interesting to find a simple construction like that in Theorem 3.6 which is a strong extractor for 2 independent aligned $(2 \times n)$ SR-sources.

3.3.2 Extracting from $O(\frac{\log n}{\log k})$ independent (n, k) SR-sources

In this section we describe the final trick needed to build the independent sources extractor. To do so we will use the extractor $\overline{\mathsf{3SRExt}}$ from Theorem 3.17 as a black box to build a condenser. For a parameter r, we will set up $\overline{\mathsf{3SRExt}}$ to extract from three independent $(\sqrt{r} \times r)$ SR-sources. The extractor is obtained by repeatedly using the construction from the previous section to obtain SR sources with few and fewer rows.

Let $S_1 = \{1, \ldots, \sqrt{r}\} \times [r], S_2 = \{\sqrt{r} + 1, \ldots, 2\sqrt{r}\} \times [r], \ldots, S_{t/\sqrt{r}} = \{t - \sqrt{r} + 1, \ldots, t\} \times [r]$.

Construction: $\mathsf{ICond}(x, y^1, y^2, y^3)$

Input: x, a sample from an (n, k) source and y^1, y^2, y^3, a sample from independent $(t \times r)$ SR-sources.

Output: z.

Let Ext be a (k, ϵ) extractor with output length m and seed length r.

1. For all $1 \leq j \leq t/\sqrt{r}$, let $z_j = \mathsf{Ext}(x, \overline{\mathsf{3SRExt}}(y^1_{S_j}, y^2_{S_j}, y^3_{S_j}))$.
2. Let $z = z_1 \circ \cdots \circ z_{t/\sqrt{r}}$.

The following lemma is easy to see given our previous work:

Lemma 3.25. *If X, Y^1, Y^2, Y^3 are independent sources, with X an (n, k) source and Y^1, Y^2, Y^3 aligned $(t \times r)$ SR-sources, then Z is ϵ close to a $(t/\sqrt{r} \times m)$ SR-sources.*

Here the error can be made exponentially small in r. In addition, notice that if X^1, X^2, X^3 are independent (n, k) sources and Y^1, Y^2, Y^3 are as before, $\mathsf{ICond}(X^1, Y^1, Y^2, Y^3) \circ \mathsf{ICond}(X^2, Y^1, Y^2, Y^3) \circ \mathsf{ICond}(X^3, Y^1, Y^2, Y^3)$ is $\sqrt{\epsilon}$ close to being a convex combination of independent aligned $(t/\sqrt{r} \times m)$ SR-sources as long as $3\sqrt{\epsilon} < 1$. Using this basic tool, we can now prove the main theorem.

Proof of Theorem 3.1. First consider the following function to extract from independent (n, k) sources. Here h is a constant that depends on the seed length of the strong seeded extractor used and the output length of the strong extractor.

Construction: $\mathsf{IExt}(X^1, \ldots, X^{3h\frac{\log n}{\log k}+3})$

1. First use a strong seeded extractor to convert three of the sources to aligned independent SR-sources.
2. Iteratively run ICond on three general input sources using the three SR-sources as seed. In each step we obtain three SR-sources with fewer rows. After $h \log(n)/\log(k)$ iterations, we will have brought the number of rows down to small enough.
3. Finally apply $\overline{\mathsf{3SRExt}}$ to get random bits.

The error adds in each step, but as long as $k > \log^4 n$, the dominant error comes from the conversion of the general source to an SR-source (where we incur error of $1/\mathrm{poly}(n)$).

In this way we get an extractor for $O(\frac{\log n}{\log k})$ independent (n, k) sources with output length $k - o(k)$ and error of $1/\mathrm{poly}(n)$. To get k bits, we can simply run the extractor twice on two disjoint sets of independent sources to double the output length. To reduce the error, we will use Lemma 2.12. Using the lemma, we can increase the number of sources used by a constant factor to reduce the error to less than $1/n^c$ for any constant c.

■

4 Extracting from 2 Independent Block-Sources

In this section we show how to use essentially the same construction from the previous section to obtain an extractor for 2 independent block-sources with very few blocks. This is analogous to an idea from [BKS+05], where they show how to use a 4-source extractor to get an extractor for 2 block-sources with just 2 blocks each. The main theorem of this section is the following:

Theorem 4.1 (2 Block-Source Extractor)**.** *For every n, k with $k > \log^4 n$, there exists a polynomial time computable function $\overline{\mathsf{IExt}} : \{0,1\}^{un} \times \{0,1\}^{un} \to \{0,1\}^k$ with $u = O(\frac{\log n}{\log k})$ s.t. , for constant $\gamma < 1$ if $X = X^1 \circ \cdots \circ X^u$ and $Y = Y^1 \circ \cdots \circ Y^u$ are independent $(k, \dots, k)$ block-sources,*

$$|\overline{\mathsf{IExt}}(X, Y) - U_k| < \epsilon$$

where $\epsilon = 1/n^{\Omega(1)}$.

The extractor here is essentially the same one as the one we constructed in the previous section for a few truly independent sources, if we use $\overline{\mathsf{2SRExt}}$ from Theorem 3.23 at the lowest level instead of $\overline{\mathsf{3SRExt}}$.

Let $S_1 = \{1, \dots, \sqrt{r}\} \times [r], S_2 = \{\sqrt{r}+1, \dots, 2\sqrt{r}\} \times [r], \dots, S_{t/\sqrt{r}} = \{t - \sqrt{r} + 1, \dots, t\} \times [r]$.

Construction: $\mathsf{ICond}(x^1 \circ \cdots \circ x^u, y^1 \circ \cdots \circ y^u)$

Input: $x = x^1 \circ \cdots \circ x^u$ and $y = y^1 \circ \cdots \circ y^u$, samples from independent block-sources with x^1 and y^1 independent aligned $(t \times r)$ SR-sources.

Output: $a = a^1 \circ \cdots \circ a^{u-1}$ and $b = b^1 \circ \cdots \circ b^{u-1}$.

Let Ext be a (k, ϵ) extractor with output length m and seed length r.

1. For all $1 \le j \le t/\sqrt{r}$, let $a^1_j = \mathsf{Ext}(X^2, \overline{\mathsf{2SRExt}}(Y^1_{S_j}, X^1_{S_j}))$.
2. For all $1 \le j \le t/\sqrt{r}$, let $b^1_j = \mathsf{Ext}(Y^2, \overline{\mathsf{2SRExt}}(X^1_{S_j}, Y^1_{S_j}))$.
3. Let $a^1 = a^1_1 \circ \cdots \circ a^1_{t/\sqrt{r}}$.
4. Let $b^1 = b^1_1 \circ \cdots \circ b^1_{t/\sqrt{r}}$.
5. For all $2 \le i \le u-1$, let $a^i = X^{i+1}$, $b^i = Y^{i+1}$.

The following lemma can be obtained by applying the techniques from the previous section. We state it without proof.

Lemma 4.2. *If $X = X^1 \circ \cdots \circ X^u$ and $Y = Y^1 \circ \cdots \circ Y^u$ are independent $(k, \dots, k)$-block-sources with each block except the first one of length n, and X^1 and Y^1 independent aligned $(t \times r)$ SR-sources, then $A = A^1 \circ \cdots \circ A^{u-1}$ and $B = B^1 \circ \cdots \circ B^{u-1}$ are statistically close to being a convex combination of independent $(k, \dots, k)$-block-sources with each block except the first one of length n with A^1 and B^1 independent aligned $(t/\sqrt{r} \times m)$ SR-sources.*

Here the error can be made exponentially small in r. Iteratively applying this condenser, we obtain the extractor for Theorem 4.1.

Remark 4.3. It can be shown that the extractor from Theorem 4.1 is strong. The extractor can also be made to work when the two sources and all blocks are of different lengths with different min-entropies, as long as the parameters are all polynomially related.

4.1 A new 2-source somewhere-extractor

One of the main results in the work of Barak et al. [BKS+05] was a new explicit bipartite Ramsey Graph (a 2-source disperser). A basic tool used as a black box in their disperser is a *somewhere-extractor*:

Definition 4.4. A function $\mathsf{SExt} : \{0,1\}^{n_1} \times \{0,1\}^{n_2} \to (\{0,1\}^m)^u$ is a (k_1, k_2, ϵ) *2-source somewhere-extractor* if for all X, Y independent (n_1, k_1) and (n_2, k_2) sources respectively, SExt is ϵ close to a $(u \times m)$ somewhere-random source.

To build the 2-source disperser, the techniques of [BKS+05] require that the somewhere-extractor used outputs a constant number of rows with each row of length some arbitrarily large constant and error that is some arbitrarily small constant when given two independent constant min-entropy rate sources. Using their constant seed condenser, they construct a 2-source somewhere-extractor that outputs a constant number of rows of linear length, with exponentially small error. Here we outline how to use our previous construction to give an alternate somewhere-extractor for two constant min-entropy rate sources that outputs a constant number of rows of linear length with polynomially small error. This is good enough for the application of building a 2-source disperser.

Construction: $\mathsf{SExt}(x, y)$
Input: x, y, a sample from two independent $(n_1, \delta_1 n_1)$, $(n_2, \delta_2 n_2)$ sources.
Output: $z = z^1 \circ \cdots \circ z^v$
Let IExt be as in Theorem 4.1.

1. Let u be the number of blocks required by IExt in each source when the two sources have min-entropy rate $\min(\delta_1/2, \delta_2/2)$.
2. Let γ be s.t. $u\gamma \ll \min(\delta_1/100, \delta_2/100)$.
3. Partition the bits of x and y into $1/\gamma$ blocks, each of equal length. $x = x^1, \dots, x^{1/\gamma}$, $y = y^1, \dots, y^{1/\gamma}$.
4. For $i = 1, 2, \dots, \binom{1/\gamma}{u} u!$, let $\pi_i(x)$ denote the string obtained by choosing u blocks from $x^1, \dots, x^u$ and permuting them in the i'th way. Similarly define $\pi_i(y)$.
5. For all pairs $(i, j) \in [\binom{1/\gamma}{u} u!] \times [\binom{1/\gamma}{u} u!]$, let the (i, j)'th block of z be $\mathsf{IExt}(\pi_i(x), \pi_j(y))$.
6. Output z.

Lemma 4.5. *If X is an $(n_1, \delta_1 n_1)$ source and Y is an independent $(n_2, \delta_2 n_2)$ source, $\mathsf{SExt}(X, Y)$ is $1/n^{\Omega(1)}$ close to being somewhere-random.*

Proof Sketch: It can be shown that every $(n_1, \delta_1 n_1)$ source X is statistically close to a convex combination of sources $\{X^l\}_{l \in I}$ s.t. for each X^l, there exists an i for which $\pi_i(X^l)$ is a block-source. A similar statement is true for every source Y that is a $(n_2, \delta_2 n_2)$ source. Thus we get that $X \circ Y$ is statistically close to a convex combination of sources s.t. for every source in the combination there exists some (i, j) s.t. $\pi_i(X), \pi_j(Y)$ are independent block-sources. The extractor succeeds in extracting random bits for that choice of (i, j). Further, the number of possible choices for (i, j) is just a constant.

□

Remark 4.6. Applying Raz's merger [Raz05] to the sources before we partition them, we can actually ensure that almost all rows in the output are statistically close to uniformly random.

5 Improving the constructions of Barak et al. [BKS+05] and Raz [Raz05]

The constructions of Barak et al. [BKS+05] and Raz [Raz05] work by first converting the input sources to a convex combination of two or more independent somewhere-random sources, where each somewhere-random source has a constant number of rows. At this point they take a constant sized section of each of the sources and do a brute force search for an optimal independent sources extractor. In this way they obtain a constant number (you can actually get say $\log\log\log n$) random bits with large error (say $1/\log\log n$).

Instead of doing brute force search in these construction, we can apply our techniques to get almost all the random bits out of the somewhere random sources, with exponentially small error.

Here we list the new theorems we can obtain by composing our SR-source extractor with the techniques of these papers:

Theorem 5.1 (3 source extractor, enhancing [BKS+05])**.** *For every n and constant $\delta > 0$ there exists a polynomial time computable function $\mathsf{IExt} : (\{0,1\}^n)^3 \rightarrow \{0,1\}^m$ s.t. if X^1, X^2, X^3 are independent $(n, \delta n)$ sources,*

$$|\mathsf{IExt}(X^1, X^2, X^3) - U_m| < \epsilon$$

where $m = \Omega(n)$ and $\epsilon = 2^{-\Omega(n)}$

Theorem 5.2 (2 source disperser, enhancing [BKS+05])**.** *For every n and constant $\delta > 0$ there exists a polynomial time computable function $\mathsf{IDisp} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^m$ s.t. if $X, Y \subseteq \{0,1\}^n$ are sets s.t. $|X|, |Y| \geq 2^{\delta n}$, $2^m - |\mathsf{IDisp}(X, Y)| < \epsilon 2^m$, where $m = \Omega(n)$ and $\epsilon = 2^{-\Omega(n)}$.*

Theorem 5.3 (3 source weak seed extractor, enhancing [Raz05])**.** *For every $n, k = k(n) = \log^4(n)$ and constant δ there exists a polynomial time computable function $\mathsf{IExt} : (\{0,1\}^n)^3 \rightarrow \{0,1\}^m$ s.t. for all independent sources X^1, X^2, Y with X^1, X^2 (n, k) sources and Y a $(n, \delta n)$ source,*

$$|\mathsf{IExt}(X^1, X^2, Y) - U_m| < \epsilon$$

where $m = \Omega(k)$ and $\epsilon = 2^{-\Omega(k)}$.

As we have discussed, all of these theorems are obtained by modifying the corresponding constructions from [BKS+05, Raz05]. We defer the discussion of the details of the modifications to the full version of this paper, but it is not hard to obtain the constructions for these theorems given those works and the techniques in this paper. These previous works simply reduce the problem to that of extracting from independent SR-sources which have a constant number of rows. Given the work in the earlier sections, this is a scenario that we can easily handle. In fact we can even get the extremely low error bounds that we have claimed in the theorems above.

Another way to compose our techniques with previous work to get something new was noticed by Avi Wigderson. The rest of the results in this section are due to him. He observed that recent constructions of randomness efficient condensers [BKS+05, Raz05] immediately imply the following theorem:

Theorem 5.4. *For every sufficiently small constant $\gamma > 0$ there exist constants $\alpha = \alpha(\gamma) > 0$, $\beta(\gamma) > 2\gamma$ and a polynomial time computable function $\mathsf{Cond} : \{0,1\}^n \rightarrow (\{0,1\}^{n^\beta})^{n^\gamma}$ s.t. for any $(n, n^{1-\alpha})$ source X, $\mathsf{Cond}(X)$ is $2^{-n^{\Omega(1)}}$-close to a source with somewhere min-entropy rate 0.9.*

Once we have this condenser, we can compose it with Theorem 2.25 to get the following theorem:

Theorem 5.5. *There exists a polynomial time computable function* $\mathsf{Ext} : (\{0,1\}^n)^3 \to \{0,1\}^{\Omega(n^\delta)}$ *s.t. for any sufficiently small constant* $\delta > 0$ *there exists a constant* $\alpha = \alpha(\delta) > 0$ *so that if* X^1 *is an* $(n, n^{1-\alpha})$ *source and* X^2, X^3 *are* (n, n^δ) *sources, with all sources independent of each other,* $\mathsf{Ext}(X^1, X^2, X^3)$ *is* ϵ*-close to the uniform distribution with with* $\epsilon < 2^{-n^{\Omega(1)}}$.

Proof Sketch: Set $\gamma = \delta/2$. Let $\alpha(\gamma)$ be as in Theorem 5.4. We first apply the function Cond promised by Theorem 5.4 to convert the first source to a source with n^γ rows, so that the source has somewhere-min-entropy rate 0.9. We now interpret this source as n^γ candidate 0.9-min-entropy rate seeds. We use these seeds with Raz's strong extractor from Theorem 2.25 and the other two sources to obtain two sources which, conditioned on the seeds, are statistically close to independent aligned $(n^\gamma \times n^\delta)$ somewhere random sources. Since $\delta > \gamma = \delta/2$, we can then use our extractor from Theorem 3.23 to get $\Omega(n^\delta)$ bits which are exponentially close to uniformly distributed. ■

In this way we obtain an extractor that can extract from just 3 sources which need have only polynomial min-entropy (the polynomial cannot be arbitrarily small).

6 Subsequent Work

Recently Ronen Shaltiel came up with a generic way to improve the output length of certain kinds of extractors [Sha05]. It turns out that our extractor is of the kind that can be improved using his techniques. As a consequence, we can improve the output length of all of our extractors to output $k - o(k)$ bits, where k is the *total* entropy in all of the input sources.

7 Acknowledgments

I would like to thank Amnon Ta-Shma for pointing out that the independent sources extractor from $O(\log n)$ sources works even when the min-entropy of the sources is as small as polylog(n) and Avi Wigderson for his observations on obtaining a 3 source extractor: Theorem 5.5. Thanks to David Zuckerman, Adam Klivans, Salil Vadhan, Noam Livne and Vladimir Trifonov for useful comments and discussions. Thanks also to the anonymous referees for many useful comments that helped to improve the presentation.

References

[BIW04] B. Barak, R. Impagliazzo, and A. Wigderson. Extracting randomness using few independent sources. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 384–393, 2004.

[BKS+05] B. Barak, G. Kindler, R. Shaltiel, B. Sudakov, and A. Wigderson. Simulating independence: New constructions of condensers, Ramsey graphs, dispersers, and extractors. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 1–10, 2005.

[BRSW06] B. Barak, A. Rao, R. Shaltiel, and A. Wigderson. 2 source dispersers for $n^{o(1)}$ entropy and ramsey graphs beating the frankl-wilson construction. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, 2006.

[Blu84] M. Blum. Independent unbiased coin flips from a correlated biased source: a finite state Markov chain. In *Proceedings of the 25th Annual IEEE Symposium on Foundations of Computer Science*, pages 425–433. IEEE, 1984.

[Bou05] J. Bourgain. More on the sum-product phenomenon in prime fields and its applications. *International Journal of Number Theory*, 1:1–32, 2005.

[BKT04] J. Bourgain, N. Katz, and T. Tao. A sum-product estimate in finite fields, and applications. *Geometric and Functional Analysis*, 14:27–57, 2004.

[CRVW02] M. Capalbo, O. Reingold, S. 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.

[CFG+85] B. Chor, J. Friedman, O. Goldreich, J. Håstad, S. Rudich, and R. Smolensky. The bit extraction problem or t–resilient functions. In *Proceedings of the 26th Annual IEEE Symposium on Foundations of Computer Science*, pages 396–407, 1985.

[CG88] B. Chor and O. Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988.

[CW89] A. Cohen and A. Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 14–19, 1989.

[GR05] A. Gabizon and R. Raz. Deterministic extractors for affine sources over large fields. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, 2005.

[GRS04] A. Gabizon, R. Raz, and R. Shaltiel. Deterministic extractors for bit-fixing sources by obtaining an independent seed. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.

[KZ03] J. Kamp and D. Zuckerman. Deterministic extractors for bit-fixing sources and exposure-resilient cryptography. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science*, pages 92–101, 2003.

[Kon03] S. Konyagin. A sum-product estimate in fields of prime order. Technical report, Arxiv, 2003. `http://arxiv.org/abs/math.NT/0304217`.

[LRVW03] C. J. Lu, O. Reingold, S. Vadhan, and A. Wigderson. Extractors: Optimal up to constant factors. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing*, pages 602–611, 2003.

[MU02] E. Mossel and C. Umans. On the complexity of approximating the VC dimension. *Journal of Computer and System Sciences*, 65:660–671, 2002.

[Nis96] N. Nisan. Extracting randomness: How and why – a survey. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 44–58, 1996.

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

[NZ93] N. Nisan and D. Zuckerman. More deterministic simulation in logspace. In *Proceedings of the 25th Annual ACM Symposium on Theory of Computing*, pages 235–244, 1993.

[Raz05] R. Raz. Extractors with weak random seeds. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 11–20, 2005.

[RRV02] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in trevisan's extractors. *J. Comput. Syst. Sci*, 65(1):97–128, 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*, pages 22–31, 2000.

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

[Sha02] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77:67–95, 2002.

[Sha05] R. Shaltiel. How to get more mileage from randomness extractors. Technical Report TR05-145, ECCC: Electronic Colloquium on Computational Complexity, 2005. Submitted for publication.

[TS96] A. Ta-Shma. Refining randomness. In *ECCCTH: Electronic Colloquium on Computational Complexity, theses*, 1996.

[TSUZ01] A. Ta-Shma, C. Umans, and D. Zuckerman. Loss-less condensers, unbalanced expanders, and extractors. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 143–152, 2001.

[Tre01] L. Trevisan. Extractors and pseudorandom generators. *Journal of the ACM*, pages 860–879, 2001.

[TV00] L. Trevisan and S. Vadhan. Extracting randomness from samplable distributions. In *Proceedings of the 41st Annual IEEE Symposium on Foundations of Computer Science*, pages 32–42, 2000.

[Vaz85] U. Vazirani. Towards a strong communication complexity theory or generating quasi-random sequences from two communicating slightly-random sources (extended abstract). In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing*, pages 366–378, 1985.

[vN51] J. von Neumann. Various techniques used in connection with random digits. *Applied Math Series*, 12:36–38, 1951. Notes by G.E. Forsythe, National Bureau of Standards. Reprinted in *Von Neumann's Collected Works*, 5:768-770, 1963.