Unbalanced Expanders and Randomness Extractors from Parvaresh–Vardy Codes*

VENKATESAN GURUSWAMI†
Dept. of Computer Science & Engineering
University of Washington
Seattle, WA 98195
venkat@cs.washington.edu

CHRISTOPHER UMANS‡
Computer Science Department
California Institute of Technology
Pasadena, CA 91125
umans@cs.caltech.edu

SALIL VADHAN§
School of Engineering and Applied Sciences
Harvard University
Cambridge, MA 02138
salil@eecs.harvard.edu

June 13, 2008

Abstract

We give an improved explicit construction of highly unbalanced bipartite expander graphs with expansion arbitrarily close to the degree (which is polylogarithmic in the number of vertices). Both the degree and the number of right-hand vertices are polynomially close to optimal, whereas the previous constructions of Ta-Shma, Umans, and Zuckerman (STOC '01) required at least one of these to be quasipolynomial in the optimal. Our expanders have a short and self-contained description and analysis, based on the ideas underlying the recent list-decodable error-correcting codes of Parvaresh and Vardy (FOCS '05).

Our expanders can be interpreted as near-optimal "randomness condensers," that reduce the task of extracting randomness from sources of arbitrary min-entropy rate to extracting randomness from sources of min-entropy rate arbitrarily close to 1, which is a much easier task. Using this connection, we obtain a new, self-contained construction of randomness extractors that is optimal up to constant factors, while being much simpler than the previous construction of Lu et al. (STOC '03) and improving upon it when the error parameter is small (e.g. 1/poly(n)).

Keywords: expander graphs, randomness extractors, error-correcting codes, list decoding, condensers.

*Preliminary versions of this paper appeared on *ECCC* [GUV1] and in *CCC '07* [GUV2].

†Supported by NSF CCF-0343672, a Sloan Research Fellowship, and a David and Lucile Packard Foundation Fellowship.

‡Supported by NSF CCF-0346991, BSF 2004329, a Sloan Research Fellowship, and an Okawa Foundation research grant.

§Supported by NSF CCF-0133096, ONR N00014-04-1-0478, BSF 2002246, a Guggenheim Fellowship, and the Miller Institute for Basic Research in Science.

1 Introduction

One of the exciting developments in the theory of pseudorandomness has been the discovery of intimate connections between a number of fundamental and widely studied objects — expander graphs, randomness extractors, list-decodable error-correcting codes, pseudorandom generators, and randomness-efficient samplers. Indeed, substantial advances have been made in our understanding of each of these objects by translating intuitions and techniques from the study of one to the study of another. In this work, we continue in this tradition. Specifically, we use ideas from recent breakthrough constructions of list-decodable codes, due to Parvaresh and Vardy [PV], to give improved and simplified constructions of both unbalanced bipartite expander graphs and randomness extractors.

1.1 Unbalanced expander graphs

Expanders are graphs that are sparse yet very highly connected. They have a wide variety of applications in theoretical computer science, and there is a rich body of work on constructions and properties of expanders. (See the survey [HLW]). The classic measure of the connectivity of an expander is *vertex expansion*, which asks that every set S of vertices that is not too large have significantly more than $|S|$ neighbors. This property is formalized for bipartite graphs through the following definitions.

Definition 1.1. *A* bipartite (multi)graph *with N left-vertices, M right-vertices, and left-degree D is specified by a function $\Gamma : [N] \times [D] \to [M]$, where $\Gamma(x, y)$ denotes the y'th neighbor of x. For a set $S \subseteq [N]$, we write $\Gamma(S)$ to denote its set of neighbors $\{\Gamma(x, y) : x \in S, y \in [D]\}$.*

Definition 1.2. *A bipartite graph $\Gamma : [N] \times [D] \to [M]$ is a (K, A)* expander *if for every set $S \subseteq [N]$ of size K, we have $|\Gamma(S)| \geqslant A \cdot K$. It is a $(\leqslant K_{max}, A)$* expander *if it is a (K, A) expander for all $K \leqslant K_{max}$.*

The typical goals in constructing expanders are to maximize the expansion factor A and minimize the degree D. In this work, we are also interested minimizing the the size M of the right-hand side, so $M \ll N$ and the graph is highly unbalanced. Intuitively, this makes expansion harder to achieve because there is less room in which to expand. Using the probabilistic method, it can be shown that very good expanders exist — with expansion $A = (1 - \varepsilon) \cdot D$, degree $D = O(\log(N/M)/\varepsilon)$, and $M = O(K_{max} D/\varepsilon) = O(K_{max} A/\varepsilon)$ right vertices. Thus, if $M \leqslant N^c$ for some constant $c < 1$, then the degree is logarithmic in N, and logarithmic degree is in fact necessary if $M = O(K_{max} A)$.[1] However, applications of expanders require *explicit constructions* — ones where the neighbor function Γ is computable in polynomial time (in its input length, $\log N + \log D$) — and the best known explicit constructions still do not match the ones given by the probabilistic method.

Most classic constructions of expanders, such as [Mar1, GG, LPS, Mar2], focus on the balanced (or non-bipartite) case (i.e. $M = N$), and thus are able to achieve constant degree $D = O(1)$. The expansion properties of these constructions are typically proven by bounding the second-largest eigenvalue of the adjacency matrix of the graph. While such 'spectral' expansion implies various combinatorial forms of expansion (e.g., vertex expansion) and many other useful properties, it seems insufficient for deducing vertex expansion beyond $D/2$ [Kah] or for obtaining highly imbalanced expanders with polylogarithmic degree [WZ]. This is unfortunate, because some applications of expanders require these properties. A

[1] More generally, the degree must be at least $\Omega(\log(N/K_{max})/\log(M/(K_{max} A)))$, as follows from the lower bounds on the degree of dispersers [RT].

beautiful example of such an application was given by Buhrman et. al. [BMRV]. They showed that a $(\leqslant K_{max}, A)$ expander with N left-vertices, M right-vertices, and expansion $A = (1-\varepsilon)D$ yields a method for storing any set $S \subseteq [N]$ of size at most $K_{max}/2$ in an M-bit data structure so that membership in S can be probabilistically tested by reading only *one bit* of the data structure. An optimal expander would give $M = O(K_{max} \log N)$, only a constant factor more than what is needed to represent an arbitrary set of size $K_{max}/2$ (even without supporting efficient membership queries).

Explicit constructions of expanders with expansion $A = (1-\varepsilon)D$ were obtained by Ta-Shma, Umans, and Zuckerman [TUZ] for the highly imbalanced (and nonconstant-degree) case and Capalbo et al. [CRVW] for the balanced (and constant-degree) case. The constructions of Ta-Shma et al. [TUZ] can make either one of the degree or right-hand side polynomially larger than the nonconstructive bounds mentioned above, at the price of making the other quasipolynomially larger. That is, one of their constructions gives $D = \mathrm{poly}(\log N)$ and $M = \mathrm{quasipoly}(K_{max}D) \stackrel{\mathrm{def}}{=} \exp(\mathrm{poly}(\log(K_{max}D)))$, whereas the other gives $D = \mathrm{quasipoly}(\log N)$ and $M = \mathrm{poly}(K_{max}D)$. The quasipolynomial bounds were improved recently in [TU], but remained superpolynomial.

We are able to simultaneously achieve $D = \mathrm{poly}(\log N)$ and $M = \mathrm{poly}(KD)$, in fact with a good tradeoff between the degrees of these two polynomials.

Theorem 1.3. *For all constants $\alpha > 0$: for every $N \in \mathbb{N}$, $K_{max} \leqslant N$, and $\varepsilon > 0$, there is an explicit $(\leqslant K_{max}, (1-\varepsilon)D)$ expander $\Gamma : [N] \times [D] \to [M]$ with degree $D = O((\log N)(\log K_{max})/\varepsilon)^{1+1/\alpha}$ and $M \leqslant D^2 \cdot K_{max}^{1+\alpha}$.*

The construction of our expanders is based on the recent list-decodable codes of Parvaresh and Vardy [PV], and can be described quite simply. The proof of the expansion property is inspired by the list-decoding algorithm for the PV codes, and is short and self-contained. An overview of this 'list-decoding approach' to proving expansion is provided in Section 2.1.

1.2 Randomness extractors

One of the main motivations and applications of our expander construction is the construction of *randomness extractors*. These are functions that convert weak random sources, which may have biases and correlations, into almost-perfect random sources. For general models of weak random sources, this is impossible, so the extractor is also provided with a short 'seed' of truly random bits to help with the extraction [NZ]. This seed can be so short (e.g. of logarithmic length) that one can often eliminate the need for any truly random bits by enumerating all choices for the seed. For example, this allows extractors to be used for efficiently simulating randomized algorithms using only a weak random source [Zuc1, NZ]. Extractors have also found a wide variety of other applications in theoretical computer science beyond their original motivating application, and thus a long body of work has been devoted to providing efficient constructions of extractors. (See the survey of Shaltiel [Sha].)

To formalize the notion of an extractor, we need a few definitions. Following [CG, Zuc1], the randomness in a source is measured by *min-entropy*: a random variable $\mathbf{X}$ has min-entropy at least k iff $\Pr[\mathbf{X} = x] \leqslant 2^{-k}$ for all x. Sometimes we refer to such a random variable as a *k-source*. A random variable $\mathbf{Z}$ is *ε-close* to a distribution D if for all events A, $\Pr[\mathbf{Z} \in A]$ differs from the probability of A under the distribution D by at most ε. Then an extractor is defined as follows:

Definition 1.4 ([NZ]). *A function $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k, ε)* extractor *if for every $\mathbf{X}$*

with min-entropy at least k, $E(\mathbf{X}, \mathbf{Y})$ is ε-close to uniform, when $\mathbf{Y}$ is uniformly distributed on $\{0,1\}^d$. An extractor is explicit *if it is computable in polynomial time.*

The competing goals when constructing extractors are to obtain a short seed length and to obtain a long output length. Nonconstructively, it is possible to simultaneously have a seed length $d = \log n + 2\log(1/\varepsilon) + O(1)$ and an output length of $m = k + d - 2\log(1/\varepsilon) - O(1)$, and both of these bounds are optimal up to additive constants (for $k \leqslant n/2$) [RT]. It remains open to match these parameters with an explicit construction.

Building on a long line of work, Lu et al. [LRVW] achieved seed length and output length that are within constant factors of optimal, provided that the error parameter ε is not too small. More precisely, they achieve seed length $d = O(\log n)$ and output length $m = (1-\alpha)k$ for $\varepsilon \geqslant n^{-1/\log^{(c)} n}$, where α and c are any two positive constants. For general ε, they pay with either a larger seed length of $d = O((\log^* n)^2 \log n + \log(1/\varepsilon))$, or a smaller output length of $m = k/\log^{(c)} n$ for any constant c.

In this work, we also achieve extractors that are optimal up to constant factors, but are able to handle the full range of error parameters ε.

Theorem 1.5. *For every constant $\alpha > 0$, and all positive integers n, k and all $\varepsilon > 0$, there is an explicit construction of a (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \log(1/\varepsilon))$ and $m \geqslant (1-\alpha)k$.*

Our extractor is also substantially simpler than that of [LRVW], which is a complex recursive construction involving many tools. The key component in our construction is the interpretation of our expander graph as a *randomness condenser*:

Definition 1.6. *A function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is an $k \to_\varepsilon k'$* condenser *if for every $\mathbf{X}$ with min-entropy at least k, $C(\mathbf{X}, \mathbf{Y})$ is ε-*close *to a distribution with min-entropy k', when $\mathbf{Y}$ is uniformly distributed on $\{0,1\}^d$. A condenser is* explicit *if it is computable in polynomial time. A condenser is called* lossless *if $k' = k + d$.*

Observe that a $k \to_\varepsilon k'$ condenser with output length $m = k'$ is an extractor, because the unique distribution on $\{0,1\}^m$ with min-entropy m is the uniform distribution. Condensers are a natural stepping-stone to constructing extractors, as they can be used to increase the *entropy rate* (the ratio of the min-entropy in a random variable to the length of the strings over which it is distributed), and it is often easier to construct extractors when the entropy rate is high. Condensers have also been used extensively in less obvious ways to build extractors, often as part of complex recursive constructions (e.g., [ISW, RSW, LRVW]). Nonconstructively, there exist *lossless* condensers with seed length $d = \log n + \log(1/\varepsilon) + O(1)$, and output length $m = k + d + \log(1/\varepsilon) + O(1)$.

As shown by [TUZ], lossless condensers are equivalent to bipartite expanders with expansion close to the degree. Applying this connection to Theorem 1.3, we obtain the following condenser:

Theorem 1.7. *For all constants $\alpha \in (0,1)$: for every $n \in \mathbb{N}$, $k \leqslant n$, and $\varepsilon > 0$, there is an explicit $k \to_\varepsilon k + d$ (lossless) condenser $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = (1 + 1/\alpha) \cdot (\log n + \log k + \log(1/\varepsilon)) + O(1)$ and $m \leqslant 2d + (1+\alpha)k$.*

Consider the case that α is a constant close to 0. Then the condenser has seed length $O(\log(n/\varepsilon))$ and output min-entropy rate roughly $1/(1+\alpha)$. Thus, the task of constructing extractors for arbitrary

seed length d	output length	Thm.
$\log n + O(\log(k/\varepsilon))$	$(1-\gamma)k$	4.19
$\log n + O(\log k \cdot \log(k/\varepsilon))$	$k + d - 2\log(1/\varepsilon) - O(1)$	4.21

Figure 1: Extractors in this paper for min-entropy k and error ε. Above, $\gamma \in (0,1)$ is an arbitrary constant.

seed length d	output length	output entropy	Thm.
$(1+\gamma)\log(nk/\varepsilon) + O(1)$	$(1+1/\gamma)k + 2d$	$k+d$ (lossless)	4.3
$\log(nk/\varepsilon) + O(1)$	$d \cdot (k+2)$	$k+d$ (lossless)	4.4

Figure 2: Condensers in this paper for min-entropy k and error ε. Above, $\gamma > 0$ is an arbitrary constant.

min-entropy is reduced to that of constructing extractors for min-entropy rate close to 1, which is a much easier task. Indeed, when ε is constant, we can use a well-known and simple extractor based on expander walks. When ε is sub-constant, we can use Zuckerman's extractor for constant entropy rate [Zuc2] to obtain the proper dependence on ε as long as $\varepsilon > \exp(-k/2^{O(\log^* k)})$. Moreover, by combining our condenser with ideas from the early constructions of extractors (the Leftover Hash Lemma, block-source extraction, and simple compositions), we are able to give a completely self-contained proof of Theorem 1.5 with no constraint on the error parameter ε at all.

Our main extractors and condensers are summarized in Figures 1 and 2.

1.3 Organization and pointers to main results

We begin with a high level overview of our construction and proof method in Section 2. We describe and analyze our expander construction in Section 3 (our main Theorem 1.3 concerning expanders is proved as Theorem 3.5). We then interpret our expander as a lossless condenser and use it to obtain our extractors in a self-contained way in Section 4 (our main Theorem 1.5 concerning extractors is proved as Theorem 4.19).

In Section 6, we analyze a variant of our main condenser that has a simpler description in terms of just Reed-Solomon codes and is a univariate analogue of [SU], and whose analysis is based on [GR]. We give two variants of such condensers, both of which have parameters slightly worse than our main condenser. Specifically, one is lossless but limited to achieving entropy rate $1/2$, and the other can achieve entropy rate close to 1 but loses a constant fraction of the source min-entropy. The latter is analyzed using a list-decoding view of lossy condensers that we describe in Section 5. In Section 7, we describe an application of our lossless expanders to dictionary data structures for answering set membership queries in the bitprobe model, following [BMRV] who first made this beautiful connection. Finally we conclude in Section 8 with some open problems.

1.4 Notation

Throughout this paper, we use boldface capital letters for random variables (e.g., "$\mathbf{X}$"), capital letters for indeterminates, and lower case letters for elements of a set. Also throughout the paper, $\mathbf{U_t}$ is the random variable uniformly distributed on $\{0,1\}^t$. The *support* of a random variable $\mathbf{X}$ is $\mathrm{supp}(\mathbf{X}) \stackrel{\text{def}}{=}$

$\{x : \Pr[\mathbf{X} = x] > 0\}$. The *statistical distance* between random variables (or distributions) $\mathbf{X}$ and $\mathbf{Y}$ is $\max_T |\Pr[\mathbf{X} \in T] - \Pr[\mathbf{Y} \in T]|$. We say $\mathbf{X}$ and $\mathbf{Y}$ are ε-*close* if their statistical distance is at most ε. All logs are base 2.

2 Overview of our approach

In this section we give a high level overview of our construction and the proof technique.

2.1 Expansion via list-decoding

Before explaining our approach, we briefly review the basics of list-decodable codes. A *code* is mapping $\mathrm{C} : [N] \to [M]^D$, encoding messages of bit-length $n = \log_2 N$ to D symbols over the alphabet $[M]$. (Contrary to the usual convention in coding theory, we use different alphabets for the message and the encoding.) The *rate* of such a code is $\rho = n/(D \log_2 M)$. We say that C is (ε, K) *list-decodable* if for every $r \in [M]^D$, the set $\mathrm{LIST}(r, \varepsilon) \stackrel{\text{def}}{=} \{x : \Pr_y[\mathrm{C}(x)_y = r_y] \geqslant \varepsilon\}$ is of size at most K. We think of r as a *received word* obtained by corrupting all but an ε fraction of symbols in some codeword. The list-decodability property says that there are not too many messages x that could have led to the received word r. The goal in constructing list-decodable codes is to optimize the tradeoff between the agreement ε and the rate ρ, which are typically constants independent of the message length n. Both the alphabet size M and the list-size K should be relatively small (e.g. constant or $\mathrm{poly}(n)$). Computationally, we would like efficient algorithms both for computing $\mathrm{C}(x)$ given x and for enumerating the messages in $\mathrm{LIST}(r, \varepsilon)$ given a received word r.

The classic Reed-Solomon codes were shown to achieve these properties with polynomial-time list-decoding in the seminal work of Sudan [Sud]. The tradeoff between ε and ρ was improved by Guruswami and Sudan [GS], and no better result was known for a number of years. Indeed, their result remains the best known for decoding Reed-Solomon codes. Recently, Parvaresh and Vardy [PV] gave an ingenious variant of Reed-Solomon codes for which the agreement-rate tradeoff is even better, leading finally to the *optimal* tradeoff (namely, $\rho = \varepsilon - o(1)$) achieved by Guruswami and Rudra [GR] using "folded" Reed-Solomon codes.

Our expanders are based on the Parvaresh-Vardy codes. Specifically, for a left-vertex $x \in [N]$ and $y \in [D]$, we define the y'th neighbor of x to be $\Gamma(x, y) = (y, \mathrm{C}(x)_y)$, where $\mathrm{C} : [N] \to [M]^D$ is a Parvaresh-Vardy code with a somewhat unusual setting of parameters. (Note that here we take the right-hand vertex set to be $[D] \times [M]$.) To prove that this graph is an expander, we adopt a 'list-decoding' view of expanders. Specifically, for a right-set $T \subseteq [D] \times [M]$, we define

$$\mathrm{LIST}(T) \stackrel{\text{def}}{=} \{x \in [N] : \Gamma(x) \subseteq T\}.$$

Then the property of Γ being a (K, A) expander can be reformulated as follows:

$$\text{for all right-sets } T \text{ of size less than } AK, \text{ we have } |\mathrm{LIST}(T)| < K.$$

We note that a similar formulation of expansion appears in [GT] (where it is restricted to sets T of the form $\Gamma(S)$ for sets $S \subseteq [N]$ of size at most K).

Let us compare this to the standard list-decodability property for error-correcting codes. Notice that for a received word $r \in [M]^D$,

$$\begin{aligned} \mathrm{LIST}(r,\varepsilon) &= \{x : \Pr_y[\mathrm{C}(x)_y = r_y] \geqslant \varepsilon\} \\ &= \{x : \Pr_y[\Gamma(x,y) \in T_r] \geqslant \varepsilon\}, \end{aligned}$$

where $T_r = \{(y, r_y) : y \in [D]\}$. Thus, the two list-decoding problems are related, but have the following key differences:

- In the coding setting, we only need to consider sets T of the form T_r. In particular, these sets are all very small — containing only D of the possible DM right vertices.
- In the expander setting, we only need to bound the number of left-vertices whose neighborhood is entirely contained in T, whereas in the coding setting we need to consider left-vertices for which even an ε fraction of neighbors are in T_r.
- In the coding setting, it is desirable for the alphabet size M to be small (constant or $\mathrm{poly}(n)$), whereas our expanders are most interesting and useful when M is in the range between, say, $n^{\omega(1)}$ and $2^{n/2}$.
- In the coding setting, the exact size of $\mathrm{LIST}(r,\varepsilon)$ is not important, and generally any $\mathrm{poly}(n/\varepsilon)$ bound is considered sufficient. In the expander setting, however, the relation between the list size and the size of T is crucial. A factor of 2 increase in the list size (for T of the same size) would change our expansion factor A from $(1-\varepsilon)D$ to $(1-\varepsilon)D/2$.

For these reasons, we cannot use the analysis of Parvaresh and Vardy [PV] as a black box. Indeed, in light of the last item, it is somewhat of a surprise that we can optimize the bound on list size to yield such a tight relationship between $|T|$ and $|\mathrm{LIST}(T)|$ and thereby provide near-optimal expansion.

This list-decoding view of expanders is related to the list-decoding view of randomness extractors that was implicit in Trevisan's breakthrough extractor construction [Tre] and was crystallized by Ta-Shma and Zuckerman [TZ]. There one considers *all* sets $T \subseteq [D] \times [M]$ (not just ones of bounded size) and bounds the size of $\mathrm{LIST}(T, \mu(T)+\varepsilon) = \{x : \Pr_y[\Gamma(x,y) \in T] \geqslant \mu(T)+\varepsilon\}$, where $\mu(T) \stackrel{\mathrm{def}}{=} |T|/(DM)$ is the density of T. Indeed, our work began by observing a strong similarity between a natural 'univariate' analog of the Shaltiel–Umans extractor [SU] and the Guruswami–Rudra codes [GR], and by hoping that the list-decoding algorithm for the Guruswami–Rudra codes could be used to prove that the univariate analog of the Shaltiel–Umans construction is indeed a good extractor (as conjectured in [KU]). However, we were only able to bound $|\mathrm{LIST}(T,\varepsilon)|$ for "small" sets T, which led to constructions of *lossy* condensers, as in the preliminary version of our paper [GUV1]. In the present version, we instead bound the size of $\mathrm{LIST}(T) = \mathrm{LIST}(T,1)$, and this bound is strong enough to yield expanders with expansion $(1-\varepsilon)\cdot D$ and thus directly implies lossless condensers, as discussed above. (We still consider lossy condensers in Section 5 of this paper for the purpose of analyzing a variant of our main construction.)

It is also interesting to compare our construction and analysis to recent constructions of extractors based on algebraic error-correcting codes, namely those of Ta-Shma, Zuckerman, and Safra [TZS] and Shaltiel and Umans [SU]. Both of those constructions use multivariate polynomials (Reed–Muller codes) as a starting point, and rely on the fact that these codes are *locally decodable*, in the sense that any bit of the message can be recovered by reading only a small portion of the received word. While the advantage of local decodability is clear in the computational setting (i.e., constructions of pseudorandom generators [STV, SU, Uma]),

where it enables efficient reductions, it is less clear why it is needed in the information-theoretic setting of extractors, where the 'decoding' only occurs in the analysis. Indeed, Trevisan's extractor [Tre] corresponds to the pseudorandom generator construction of [STV], but with the locally list-decodable code replaced by a standard list-decodable code. However, the extractor analyses of [TZS] and [SU] seem to rely essentially on multivariate polynomials and local (list-)decodability. Our construction works with univariate polynomials and the analysis does not require any local decoding – indeed, univariate polynomial (Reed-Solomon) codes are not locally decodable.

2.2 Parvaresh-Vardy codes and the proof technique

Our constructions are based on Parvaresh-Vardy codes [PV], which in turn are based on Reed-Solomon codes. A Reed-Solomon codeword is a univariate degree $n-1$ polynomial $f \in \mathbb{F}_q[Y]$, evaluated at all points in the field. A Parvaresh-Vardy codeword is a bundle of several related degree $n-1$ polynomials $f_0, f_1, f_2, \ldots, f_{m-1}$, each evaluated at all points in the field. The evaluations of the various f_i at a given field element are packaged into a symbol from the larger alphabet $\mathbb{F}_{q^m}$. The purpose of this extra redundancy is to enable a better list-decoding algorithm than is possible for Reed-Solomon codes.

The main idea in [PV] is to view degree $n-1$ polynomials as elements of the extension field $\mathbb{F} = \mathbb{F}_q[Y]/E(Y)$, where E is some irreducible polynomial of degree n. The f_i (now viewed as elements of $\mathbb{F}$) are chosen so that $f_i = f_0^{h^i}$ for $i \geqslant 1$, and a positive integer parameter h. As explained in Section 2.1, our expander is constructed directly from Parvaresh-Vardy codes as follows:

$$\Gamma(f_0, y) = [y, f_0(y), f_1(y), \ldots, f_{m-1}(y)].$$

In the analysis, our task is to show that for any set T of size L, the set $\mathrm{LIST}(T) = \{f_0 : \Gamma(f_0) \subseteq T\}$ is small. To do this we follow the list-decoding analysis of [PV], which in turn has the same general structure as the list-decoding algorithms for Reed–Solomon codes [Sud, GS]. We first produce a non-zero polynomial $Q : \mathbb{F}_q^{1+m} \to \mathbb{F}_q$ that vanishes on T. Now, for every $f_0 \in \mathrm{LIST}(T)$, we have

$$Q(y, f_0(y), \ldots, f_{m-1}(y)) = 0 \quad \forall y \in \mathbb{F}_q,$$

and by ensuring that Q has small degree (which is possible because T is not too large), we will be able to argue that the univariate polynomial $Q(Y, f_0(Y), \ldots, f_{m-1}(Y))$ is the zero polynomial. Recalling the definition of the f_i, and viewing the f_i as elements of the extension field $\mathbb{F} = \mathbb{F}_q[Y]/E(Y)$, we observe that f_0 is a *root* of the univariate polynomial

$$Q^*(Z) \stackrel{\text{def}}{=} Q(Y, Z, Z^h, Z^{h^2}, \ldots, Z^{h^{m-1}}) \bmod E(Y).$$

This is because when simplifying the formal polynomial $Q^*(f_0(Y)) \bmod E(Y)$, we can first take each $f_0(Y)^{h^i}$ term modulo $E(Y)$, resulting in $f_i(Y)$, and we have just argued that $Q(Y, f_0(Y), \ldots, f_{m-1}(Y))$ is the zero polynomial, so it is still the zero polynomial modulo $E(Y)$. This argument holds for every $f_0 \in \mathrm{LIST}(T)$, and so we can upper-bound $|\mathrm{LIST}(T)|$ by the degree of Q^*.

3 Expander Graphs

We first formally develop the list-decoding view of expanders described in Section 2.1.

Definition 3.1. *For a bipartite graph $\Gamma : [N] \times [D] \to [M]$ and a set $T \subseteq [M]$, define*

$$\mathrm{LIST}(T) = \{x \in [N] : \Gamma(x) \subseteq T\}.$$

The proof of the next lemma follows from the definitions:

Lemma 3.2. *A graph Γ is a (K, A) expander iff for every set T of size at most $AK - 1$, $\mathrm{LIST}(T)$ is of size at most $K - 1$.*

3.1 The construction

Fix the field $\mathbb{F}_q$ and let $E(Y)$ be an irreducible polynomial of degree n over $\mathbb{F}_q$. We identify elements of $\mathbb{F}_q^n$ with univariate polynomials over $\mathbb{F}_q$ with degree at most $n - 1$. Fix an integer parameter h.

Our expander is the bipartite graph $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined as:

$$\Gamma(f, y) \stackrel{\text{def}}{=} [y, f(y), (f^h \bmod E)(y), (f^{h^2} \bmod E)(y), \cdots, (f^{h^{m-1}} \bmod E)(y)]. \tag{1}$$

In other words, the bipartite graph has "message" polynomials $f(Y)$ on the left, and the y'th neighbor of $f(Y)$ is simply the y'th symbol of the Parvaresh-Vardy encoding of $f(Y)$. For ease of notation, we will refer to $(f^{h^i} \bmod E)$ as "f_i."

Theorem 3.3. *The graph $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined in (1) is a $(\leqslant K_{max}, A)$ expander for $K_{max} = h^m$ and $A = q - (n-1)(h-1)m$.*

Proof. Let K be any integer less than or equal to $K_{max} = h^m$, and let $A = q - (n-1)(h-1)m$. By Lemma 3.2, it suffices to show that for every set $T \subseteq \mathbb{F}_q^{m+1}$ of size at most $AK - 1$, we have $|\mathrm{LIST}(T)| \leqslant K - 1$. Fix such a set T.

Our first step is to find a nonzero "low-degree" polynomial $Q(Y, Y_1, \ldots, Y_m)$ that vanishes on T. Specifically, Q will only have nonzero coefficients on monomials of the form $Y^i M_j(Y_1, \ldots, Y_m)$ for $0 \leqslant i \leqslant A-1$ and $0 \leqslant j \leqslant K-1 \leqslant h^m - 1$, where $M_j(Y_1, \ldots, Y_m) = Y_1^{j_0} \cdots Y_m^{j_{m-1}}$ and $j = j_0 + j_1 h + \cdots + j_{m-1} h^{m-1}$ is the base-h representation of j. (For simplicity, one may think of $K = h^m$, in which case we are simply requiring that Q has degree at most $h - 1$ in each variable Y_i.) For each $z \in T$, requiring that $Q(z) = 0$ imposes a homogeneous linear constraint on the AK coefficients of Q. Since the number of constraints is smaller than the number of unknowns, this linear system has a nonzero solution. Moreover, we may assume that among all such solutions, Q is the one of smallest degree in the variable Y. This implies that if we write Q in the form

$$Q(Y, Y_1, \ldots, Y_m) = \sum_{j=0}^{K-1} p_j(Y) \cdot M_j(Y_1, \ldots, Y_m)$$

for univariate polynomials $p_0(Y), \ldots, p_{K-1}(Y)$, then at least one of the p_j's is not divisible by $E(Y)$. Otherwise $Q(Y, Y_1, \ldots, Y_m)/E(Y)$ would have smaller degree in Y and would still vanish on T (since E is irreducible and thus has no roots in $\mathbb{F}_q$).

Consider any polynomial $f(Y) \in \mathrm{LIST}(T)$. By the definition of $\mathrm{LIST}(T)$ and our choice of Q, it holds that

$$Q(y, f_0(y), f_1(y), \ldots, f_{m-1}(y)) = 0 \qquad \forall y \in \mathbb{F}_q.$$

That is, the univariate polynomial $R_f(Y) \stackrel{\text{def}}{=} Q(Y, f_0(Y), \ldots, f_{m-1}(Y))$ has q zeroes. Since the degree of $R_f(Y)$ is at most $(A-1)+(n-1)(h-1)m < q$, it must be identically zero. So

$$Q(Y, f_0(Y), \ldots, f_{m-1}(Y)) = 0$$

as a formal polynomial. Now recall that $f_i(Y) \equiv f(Y)^{h^i} \pmod{E(Y)}$. Thus,

$$\begin{aligned} &Q(Y, f(Y), f(Y)^h, \ldots, f(Y)^{h^{m-1}}) \\ &\equiv\ Q(Y, f_0(Y), \ldots, f_{m-1}(Y)) \ \equiv\ 0 \pmod{E(Y)}. \end{aligned}$$

So if we interpret $f(Y)$ as an element of the extension field $\mathbb{F} = \mathbb{F}_q[Y]/E(Y)$, then $f(Y)$ is a root of the univariate polynomial Q^* over $\mathbb{F}$ defined by

$$\begin{aligned} Q^*(Z) &\stackrel{\text{def}}{=} Q(Y, Z, Z^h, Z^{h^2}, \ldots, Z^{h^{m-1}}) \bmod E(Y) \\ &= \sum_{j=0}^{K-1} (p_j(Y) \bmod E(Y)) \cdot M_j(Z, Z^h, \ldots, Z^{h^{m-1}}) \\ &= \sum_{j=0}^{K-1} (p_j(Y) \bmod E(Y)) \cdot Z^j. \end{aligned}$$

Since this holds for every $f(Y) \in \mathrm{LIST}(T)$, we deduce that Q^* has at least $|\mathrm{LIST}(T)|$ roots in $\mathbb{F}$. On the other hand, Q^* is a non-zero polynomial, because at least one of the $p_j(Y)$'s is not divisible by $E(Y)$. Thus, $|\mathrm{LIST}(T)|$ is bounded by the degree of Q^*, which is at most $K-1$. □

Remark 3.4. Observe that for all $S \subseteq \mathbb{F}_q$, the subgraph of Γ that comes from taking only y-th edges for $y \in S$, is a $(\leqslant K_{max}, A)$ expander for $A = |S| - (n-1)(h-1)m$ by the same argument.

3.2 Setting parameters

The following theorem differs from Theorem 1.3 only by allowing α to be non-constant.

Theorem 3.5 (Thm. 1.3, generalized). *For all positive integers N, $K_{\max} \leqslant N$, all $\varepsilon > 0$, and all $\alpha \in (0, \log x/\log\log x)$ for $x = (\log N)(\log K_{max})/\varepsilon$, there is an explicit $(\leqslant K_{max}, (1-\varepsilon)D)$ expander $\Gamma : [N] \times [D] \to [M]$ with degree $D = O\Big(((\log N)(\log K_{max})/\varepsilon)^{1+1/\alpha}\Big)$ and $M \leqslant D^2 \cdot K_{max}^{1+\alpha}$. Moreover, D and M are powers of 2.*

Proof. Let $n = \log N$ and $k = \log K_{\max}$. Let $h_0 = (2nk/\varepsilon)^{1/\alpha}$, $h = \lceil h_0 \rceil$, and let q be the power of 2 in the interval $(h^{1+\alpha}/2, h^{1+\alpha}]$.

Set $m = \lceil (\log K_{\max})/(\log h) \rceil$, so that $h^{m-1} \leqslant K_{\max} \leqslant h^m$. Then, by Theorem 3.3, the graph $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined in (1) is a $(\leqslant h^m, A)$ expander for $A = q - (n-1)(h-1)m$. Since $K_{\max} \leqslant h^m$, it is also a $(\leqslant K_{\max}, A)$ expander.

Note that the number of left-vertices in Γ is $q^n \geqslant N$, and the number of right-vertices is

$$M = q^{m+1} \leqslant q^2 \cdot h^{(1+\alpha)\cdot(m-1)} \leqslant q^2 \cdot K_{\max}^{1+\alpha}.$$

The degree is

$$\begin{aligned} D &\stackrel{\text{def}}{=} q \leqslant h^{1+\alpha} \leqslant (h_0+1)^{1+\alpha} \\ &= O(h_0^{1+\alpha}) = O\Big(((\log N)(\log K_{\max})/\varepsilon)^{1+1/\alpha}\Big). \end{aligned}$$

where the second-to-last equality follows from the fact that $h_0 = (nk/\varepsilon)^{1/\alpha} \geqslant \alpha$ (due to the upper bound on α).

To see that the expansion factor $A = q - (n-1)(h-1)m \geqslant q - nhk$ is at least $(1-\varepsilon)D = (1-\varepsilon)q$, note that

$$nhk \leqslant \varepsilon \cdot h^{1+\alpha} \leqslant \varepsilon q,$$

where the first inequality holds because $h^\alpha \geqslant nk/\varepsilon$.

Finally, the construction is explicit because a representation of $\mathbb{F}_q$ for q a power of 2 (i.e. an irreducible polynomial of degree $\log q$ over $\mathbb{F}_2$) as well as an irreducible polynomial $E(Y)$ of degree n over $\mathbb{F}_q$ can be found in time $\mathrm{poly}(n, \log q) = \mathrm{poly}(\log N, \log D)$ [Sho]. □

Remark 3.6. In this proof we work in a field $\mathbb{F}_q$ of characteristic 2, which has the advantage of yielding a polynomial-time construction even when we need to take q to be superpolynomially large (which occurs when $\varepsilon(n) = n^{-\omega(1)}$). When $\varepsilon \geqslant 1/\mathrm{poly}(n)$, then we could use any prime power q instead, with some minor adjustments to the construction and the parameters claimed in the theorem.

In the above theorem, α is restricted to be slightly sublogarithmic in nk/ε. It will sometimes be useful to use the following variant, which corresponds to a logarithmic value of α and yields a degree with a linear dependence on $\log N$.

Theorem 3.7. *For all positive integers N, $K_{\max} \leqslant N$, and all $\varepsilon > 0$, there is an explicit $(\leqslant K_{max}, (1-\varepsilon)D)$ expander $\Gamma : [N] \times [D] \to [M]$ with degree $D \leqslant 2(\log N)(\log K_{max})/\varepsilon$ and $M \leqslant (4K_{max})^{\log D}$. Moreover, D and M are powers of 2.*

Proof. The proof is along the same lines as that of Theorem 3.5, except we take $h = 2$, $q \in (nk/\varepsilon, 2nk/\varepsilon]$, and $m = \lceil \log K_{max} \rceil$. Then we can bound the degree by $D = q \leqslant 2nk/\varepsilon$, the number of right-hand vertices by $M = q^{m+1} = (4 \cdot 2^{m-1})^{\log q} \leqslant (4K_{max})^{\log q}$, and the expansion by $A = q - (n-1)(h-1)m \geqslant q - nk \geqslant (1-\varepsilon)D$. □

4 Lossless condensers and extractors

In this section we prove our main extractor theorem.

4.1 Lossless condensers

We first interpret the expanders constructed in the previous section as lossless condensers (see Definition 1.6). This connection, due to Ta-Shma, Umans, and Zuckerman [TUZ], is based on viewing a function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ as the neighbor function of a bipartite graph with 2^n left-vertices, 2^m right-vertices, and left-degree 2^d. It turns out that this graph has expansion close to the degree if and only if C is a lossless condenser.

Lemma 4.1 ([TUZ]). *For $n, m, d \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $k \in [0,n]$ such that $2^k \in \mathbb{N}$, $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $k \to_\varepsilon k+d$ condenser iff the corresponding bipartite graph is a $(2^k, (1-\varepsilon)\cdot 2^d)$ expander.*

One minor technicality in the above connection is that it requires that 2^k be an integer, whereas the notion of condenser makes sense for all $k \in [0,n]$. However, this is easily handled by rounding, if we allow a tiny increase in the error parameter ε. Specifically, we have the following generalization of the "if" direction of Lemma 4.1:

Lemma 4.2. *For $n, m, d \in \mathbb{N}$, $\varepsilon \in (0,1)$, and $k \in [0,n]$, $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a $k \to_\varepsilon k+d$ condenser if the corresponding bipartite graph is a $(\lceil 2^k \rceil, (1-\varepsilon)\cdot 2^d)$ expander and a $(\lfloor 2^k \rfloor, (1-\varepsilon)\cdot 2^d)$ expander.*

Proof. Let $K = 2^k \notin \mathbb{N}$ and $L = \lfloor K \rfloor$. Every k-source is a convex combination of sources $\mathbf{X}$ in which some set S of L elements each have probability mass exactly $1/K$, and one element $x \notin S$ has probability $1 - L/K$; thus it suffices to prove the lemma for such sources $\mathbf{X}$. We can decompose $\mathbf{X} = p\mathbf{X_1} + (1-p)\mathbf{X_2}$ where $\mathbf{X_1}$ is uniform on S, $\mathbf{X_2}$ is uniform on $S \cup \{x\}$, and $p \in [0,1]$ satisfies $p/L + (1-p)/(L+1) = 1/K$ (so that all elements of S have probability exactly $1/K$).

By Lemma 4.1, $C(\mathbf{X_1}, \mathbf{U_d})$ is ε-close to a source $\mathbf{Z_1}$ of min-entropy $\log(LD)$, where $D = 2^d$, and $C(\mathbf{X_2}, \mathbf{U_d})$ is ε-close to a source $\mathbf{Z_2}$ of min-entropy $\log((L+1)D)$. Then $C(\mathbf{X}, \mathbf{U_d})$ is ε-close to $\mathbf{Z} = p\mathbf{Z_1} + (1-p)\mathbf{Z_2}$. We now claim that $\mathbf{Z}$ is a $(k+d)$-source. Indeed, for every z,

$$\Pr[\mathbf{Z} = z] \leqslant p \cdot \Pr[\mathbf{Z_1} = z] + (1-p)\Pr[\mathbf{Z_2} = z] \leqslant p \cdot \frac{1}{LD} + (1-p) \cdot \frac{1}{(L+1)D} = \frac{1}{KD}.$$

□

Using this lemma, the following are immediate consequences of Theorems 3.5 and 3.7.

Theorem 4.3 (Theorem 1.7, generalized). *For every $n \in \mathbb{N}$, $k_{max} \leqslant n$, $\varepsilon > 0$, and $\alpha \in (0, \log(nk_{max}/\varepsilon)/\log\log(nk_{max}/\varepsilon))$, there is an explicit function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = (1+1/\alpha)\cdot(\log n + \log k_{max} + \log(1/\varepsilon)) + O(1)$ and $m \leqslant 2d + (1+\alpha)k_{max}$ such that for all $k \leqslant k_{max}$, C is a $k \to_\varepsilon k+d$ (lossless) condenser.*

Theorem 4.4. *For every $n \in \mathbb{N}$, $k_{max} \leqslant n$, and $\alpha, \varepsilon > 0$, there is an explicit function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d \leqslant \log n + \log k_{max} + \log(1/\varepsilon) + 1$ and $m \leqslant d \cdot (k_{max} + 2)$ such that for all $k \leqslant k_{max}$, C is a $k \to_\varepsilon k+d$ (lossless) condenser.*

Once we have condensed almost all of the entropy into a source with high entropy rate (as in Theorem 4.3), extracting (most of) that entropy is not that difficult. All we need to do is to compose the condenser with an extractor that works for high entropy rates. The following standard fact makes the composition formal:

Proposition 4.5. *Suppose $C : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{n'}$ is an $k \to_{\varepsilon_1} k'$ condenser, and $E : \{0,1\}^{n'} \times \{0,1\}^{d_2} \to \{0,1\}^m$ is a (k', ε_2)-extractor, then $E \circ C : \{0,1\}^n \times \{0,1\}^{d_1+d_2} \to \{0,1\}^m$ defined by $(E \circ C)(x, y_1, y_2) \stackrel{\text{def}}{=} E(C(x, y_1), y_2)$ is a $(k, \varepsilon_1 + \varepsilon_2)$-extractor.*

In the next section, we will use this proposition to compose our condenser with a simple extractor for high entropy rates to obtain our main extractor theorem (Theorem 1.5) for the case of constant error ε. For subconstant error, we could compose with Zuckerman's extractor for constant entropy rate [Zuc2], which works provided $\varepsilon > \exp(-k/2^{O(\log^* k)})$. Instead, in Section 4 we combine our condenser with ideas from the early constructions of extractors (the Leftover Hash Lemma, block-source extraction, and simple compositions), to obtain a completely self-contained proof of Theorem 1.5 with no constraint on the error parameter ε at all.

4.2 Extractors for constant error

In this section, we prove Theorem 1.5 for the case of constant error ε (which suffices for many applications of extractors). It is obtained by composing our condenser with a extractor for min-entropy rate close to 1. A standard extractor construction for this setting is based on expander walks [Gil, Zuc2, Zuc3]. Specifically, such an extractor can be obtained by combining the equivalence between extractors and 'averaging samplers' [Zuc2], and the fact that expander walks are an averaging sampler, as established by the Chernoff bound for expander walks [Gil].[2]

Theorem 4.6. *For all constants $\alpha, \varepsilon > 0$, there is a constant $\delta < 1$ for which the following holds: for all positive integers n, there is an explicit construction of a $(k = \delta n, \varepsilon)$ extractor $E : \{0,1\}^n \times \{0,1\}^t \to \{0,1\}^m$ with $t \leqslant \log(\alpha n)$ and $m \geqslant (1-\alpha)n$.*

For completeness, we present the short proof:

Proof. Let $m = \lceil (1-\alpha)n \rceil$, and for some absolute constants $c > 1$ and $\lambda < 1$, let G be an explicit 2^c-regular expander on 2^m vertices (identified with $\{0,1\}^m$) and second eigenvalue $\lambda = \lambda(G) < 1$. Let L be the largest power of 2 at most $(n-m)/c$ (so $L > (n-m)/(2c)$), and let $t = \log L \leqslant \log(\alpha n)$. The extractor E is constructed as follows. Its first argument x is used to describe a walk $v_1, v_2, \ldots, v_L$ of length L in G by picking v_1 based on the first m bits of x, and each further step of the walk from the next c bits of x — so in all, L must satisfy $n = m + (L-1)c$. The seed y is used to pick one of the vertices of the walk at random. The output $E(x,y)$ of the extractor is the m-bit label of the chosen vertex.

Let $\mathbf{X}$ be a random variable with min-entropy $k = \delta n$. We wish to prove that for any $S \subseteq \{0,1\}^m$, the probability that $E(\mathbf{X}, \mathbf{U_t})$ is a vertex in S is in the range $\mu \pm \varepsilon$ where $\mu = |S|/2^m$. Fix any such subset S. Call an $x \in \{0,1\}^n$ "bad" if

$$\left| \Pr_y[E(x,y) \in S] - \mu \right| > \varepsilon/2.$$

The known Chernoff bounds for random walks on expanders [Gil] imply that the number of bad x's is at most

$$2^n \cdot e^{-\Omega(\varepsilon^2(1-\lambda)L)} = 2^n \cdot e^{-\Omega(\varepsilon^2(1-\lambda)\alpha n/c)} = 2^n \cdot 2^{-\Omega(\varepsilon^2 \alpha n)}$$

(since c, λ are absolute constants). Therefore the probability that $\mathbf{X}$ is bad is at most $2^{-\delta n} \cdot 2^n \cdot 2^{-\Omega(\varepsilon^2 \alpha n)}$, which is exponentially small for large enough $\delta < 1$. Therefore

$$|\Pr[E(\mathbf{X}, \mathbf{U_t}) \in S] - \mu| \leqslant \varepsilon/2 + 2^{-\Omega(n)} \leqslant \varepsilon,$$

implying that E is a (k, ε)-extractor. □

[2] The papers [IZ, CW] prove hitting properties of expander walks, and observe that these imply objects related to (but weaker than) extractors, known as dispersers.

Combining this with our condenser, we obtain the following extractor:

Theorem 4.7 (Thm. 1.5 for constant error). *For all constants $\alpha, \varepsilon > 0$: for all positive integers n, k, there is an explicit construction of a (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n)$ and $m \geqslant (1-\alpha)k$.*

Proof. Given constant $\alpha, \varepsilon > 0$, apply Theorem 4.6 to obtain a $\delta = 1 - \gamma$ for a constant $\gamma > 0$ and an explicit $(k, \varepsilon/2)$ extractor $E : \{0,1\}^a \times \{0,1\}^t \to \{0,1\}^m$ with $a = \lfloor k/(1-\gamma) \rfloor$, $t \leqslant \log a$, and $m \geqslant (1-\alpha)a \geqslant (1-\alpha)k$.

By Theorem 4.3, there is an explicit $k \to_{\varepsilon/2} k + d$ condenser $C : \{0,1\}^n \times \{0,1\}^u \to \{0,1\}^b$ with $u = O(\log n)$ and $b \leqslant (1 + \gamma/2) \cdot k + 2u \leqslant a$, where the latter inequality holds because we may assume $k \geqslant (4u+2)/\gamma$. (Otherwise a trivial extractor that outputs its seed will satisfy the theorem.)

By Proposition 4.5, we obtain a (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with seed length $d = t + u = O(\log n)$ and output length $m \geqslant (1-\alpha)k$. □

4.3 Extractors for arbitrary error

In this section, provide a self-contained construction of extractors that are optimal up to constant factors, with no constraint on the error parameter. It is obtained by combining our condenser with the ideas from the early constructions of extractors [Zuc1, NZ, SZ, Zuc2, GW]. Beyond our condenser, the only tools needed are the universal hashing and some simple (and standard) methods to compose extractors. In this section, we often use the term *k-source* to mean a random variable with min-entropy at least k.

4.3.1 The Leftover Hash Lemma

The Leftover Hash Lemma [ILL], which predates the general definition of extractors [NZ], shows that universal hash functions are randomness extractors, albeit with a large seed length:

Lemma 4.8 ([ILL]). *For all $n \in \mathbb{N}$, $k \leqslant n$, and $\varepsilon > 0$, there is an explicit (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = n$ and $m \geqslant k + d - 2\log(1/\varepsilon)$.*

Note that the output length is optimal, but the seed length is linear rather than logarithmic in n. Nevertheless, this extractor was a very useful component in early constructions of extractors with (poly)logarithmic seed length [Zuc1, NZ, Zuc2]. Indeed, it was dubbed the "Mother of all Extractors" by Nisan [NT].

Proof Sketch. We associate $\{0,1\}^n = \{0,1\}^d$ with the finite field $\mathbb{F}$ of size 2^n. Given $x, y \in \mathbb{F}$, we define $E(x,y) = (y, xy|_m)$, where $xy|_m$ is the first $m = \lceil k + d - 2\log(1/\varepsilon) \rceil$ bits of the product $xy \in \mathbb{F}$.

The fact that this is a (k, ε) extractor follows from the Leftover Hash Lemma [ILL] and the fact that the set of functions $h_y(x) = xy|_m$ is 2-universal. For completeness, we sketch the proof here. Let $\mathbf{X}$ be a k-source on $\{0,1\}^n$, and $\mathbf{Y}$ be uniform on $\{0,1\}^d$. Then, it can be shown that the *collision probability*[3] of $E(\mathbf{X}, \mathbf{Y}) = (\mathbf{Y}, \mathbf{X}\mathbf{Y}|_m)$ is at most $(1/D) \cdot (1/K + 1/M) \leqslant (1 + 2\varepsilon^2)/(DM)$. ($1/D$ is the collision probability of $\mathbf{Y}$, $1/K$ is the collision probability of $\mathbf{X}$, and $1/M$ is the probability that $x\mathbf{Y} = x'\mathbf{Y}$ for any two distinct $x \neq x'$.) This is equivalent to saying that the ℓ_2 distance of the distribution $E(\mathbf{X}, \mathbf{Y})$ from

[3] The *collision probability* of a random variable $\mathbf{Z}$ is $\sum_z \Pr[\mathbf{Z} = z]^2 = \Pr[\mathbf{Z} = \mathbf{Z}']$, where $\mathbf{Z}'$ is an iid copy of $\mathbf{Z}$.

uniform is at most $\sqrt{2\varepsilon^2/DM} \leqslant 2\varepsilon/\sqrt{DM}$. Then the statistical distance to uniform equals 1/2 the ℓ_1 distance, which in turn is at most a factor of $\sqrt{DM}$ larger than the ℓ_2 distance. □

We note that by composing our lossless condenser (Theorem 4.3) with this extractor via Proposition 4.5, we can reduce the seed length from n to $O(k + \log(n/\varepsilon))$, matching the low min-entropy extractors of [SZ] (which are based on generalization of the Leftover Hash Lemma to almost-universal hash functions):

Lemma 4.9. *For every constant $\alpha > 0$, for all $n \in \mathbb{N}$, $k \leqslant n$, and $\varepsilon > 0$, there is an explicit extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = (1+\alpha)k + O(\log(n/\varepsilon))$ and $m \geqslant k + d - 2\log(1/\varepsilon)$ (the constant in $O(\log(n/\varepsilon))$ depends on α).*

Remark 4.10. It was pointed out to us by Michael von Korff and Kai-Min Chung that the seed length can be reduced further to $\alpha k + O(\log(n/\varepsilon))$ for an arbitrarily small constant $\alpha > 0$ by condensing to length $n' = (1+\alpha)k + O(\log(n/\varepsilon))$, and then applying the "high min-entropy" extractor of [GW], which requires a seed of length $n' - k + O(\log(1/\varepsilon)) = \alpha k + O(\log(n/\varepsilon))$ and has optimal output length $m = k + d - 2\log(1/\varepsilon) - O(1)$ (if implemented using Ramanujan expander graphs). In the next section, we will see another way (Lemma 4.11) to achieve this constant-factor savings in seed length, which has the advantage of being self-contained (not relying on Ramanujan expanders) but has the disadvantage of only extracting a constant fraction of the min-entropy.

4.3.2 An extractor with seed much shorter than its output

Our goal in this subsection is to constructing the following extractor, which will be the main building block for our recursive construction:

Lemma 4.11. *For every* constant $t > 0$ *and all positive integers $n \geqslant k$ and all $\varepsilon > 0$, there is an explicit (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $m = \lceil k/2 \rceil$ and $d \leqslant k/t + O(\log(n/\varepsilon))$.*

The point is that this extractor has a seed length that is an arbitrarily large constant factor (namely $t/2$) smaller than its output length. This will be useful as a building block for our recursive construction of extractors optimal up to constant factors in Section 4.3.3. We now turn to defining block sources and collecting basic results about extracting randomness from them.

A *block source* is a useful model of a weak random source that has more structure than an arbitrary k-source:

Definition 4.12 ([CG]). $\mathbf{X} = (\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_t})$ *is a* $(k_1, k_2, \ldots, k_t)$ block source *if for every $x_1, \ldots, x_{i-1}$, $\mathbf{X_i}|_{\mathbf{X_1}=x_1,\ldots,\mathbf{X_{i-1}}=x_{i-1}}$ is a k_i-source. If $k_1 = k_2 = \cdots = k_t = k$, then we call X a $t \times k$* block source.

Note that a $(k_1, k_2, \ldots, k_t)$ block source is also a $(k_1 + \cdots + k_t)$-source, but it comes with additional structure — each block is guaranteed to contribute some min-entropy. Thus, extracting randomness from block sources is easier task than extracting from general sources. Indeed, we have the following standard lemma:

Lemma 4.13. *Let $E_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{m_1}$ be a (k_1, ε_1)-extractor, and $E_2 : \{0,1\}^{n_2} \times \{0,1\}^{d_2} \to \{0,1\}^{m_2}$ be a (k_2, ε_2)-extractor with $m_2 \geqslant d_1$. Define $E'((x_1, x_2), y_2) = (E_1(x_1, y_1), z_2)$, where (y_1, z_2) is obtained by partitioning $E_2(x_2, y_2)$ into a prefix y_1 of length d_1 and a suffix z_2 of length $m_2 - d_1$.*

Then for every (k_1, k_2) block source $\mathbf{X} = (\mathbf{X_1}, \mathbf{X_2})$ taking values in $\{0,1\}^{n_1} \times \{0,1\}^{n_2}$, it holds that $E'(\mathbf{X}, \mathbf{U_{d_2}})$ is $(\varepsilon_1 + \varepsilon_2)$-close to $\mathbf{U_{m_1}} \times \mathbf{U_{m_2 - d_1}}$.

Proof. $(\mathbf{X_1}, \mathbf{Y_1}, \mathbf{Z_2}) = (\mathbf{X_1}, E_2(\mathbf{X_2}, \mathbf{U_{d_2}}))$ is ε_2-close to $(\mathbf{X_1}, \mathbf{U_{m_2}}) = (\mathbf{X_1}, \mathbf{U_{d_1}}, \mathbf{U_{m_2 - d_1}})$.

Thus, $(E_1(\mathbf{X_1}, \mathbf{Y_1}), \mathbf{Z_2})$ is ε_2-close to $(E_1(\mathbf{X_1}, \mathbf{U_{d_1}}), \mathbf{U_{m_2 - d_1}})$, which is ε_1-close to $(\mathbf{U_{m_1}}, \mathbf{U_{m_2 - d_1}})$.

By the triangle inequality, $E'(X, \mathbf{U_{d_2}}) = (E_1(\mathbf{X_1}, \mathbf{Y_1}), \mathbf{Z_2})$ is $(\varepsilon_1 + \varepsilon_2)$-close to $(\mathbf{U_{m_1}}, \mathbf{U_{m_2 - d_1}})$. □

The benefit of this composition is that the seed length of E' equals that of only one of the extractors (namely E_2), rather than being the sum of the seed lengths. Thus, we get to extract from multiple blocks at the "price of one." Moreover, since we can take $d_1 = m_2$, which is typically larger than d_2, the seed length of E' can even be much smaller than that of E_1.

The lemma extends naturally to extracting from many blocks:

Lemma 4.14. *For $i = 1, \ldots, t$, let $E_i : \{0,1\}^{n_i} \times \{0,1\}^{d_i} \to \{0,1\}^{m_i}$ be a (k_i, ε_i)-extractor, and suppose that $m_i \geqslant d_{i-1}$ for every $i = 1, \ldots, t$, where we define $d_0 = 0$. Define $E'((x_1, \ldots, x_t), y_t) = (z_1, \ldots, z_t)$, where for $i = t, \ldots, 1$, we inductively define (y_{i-1}, z_i) to be a partition of $E_i(x_i, y_i)$ into a d_{i-1}-bit prefix and a $(m_i - d_{i-1})$-bit suffix.*

Then for every $(k_1, \ldots, k_t)$ block source $X = (\mathbf{X_1}, \ldots, \mathbf{X_t})$ taking values in $\{0,1\}^{n_1} \times \cdots \{0,1\}^{n_t}$, it holds that $E'(X, U_{d_t})$ is ε-close to U_m for $\varepsilon = \sum_{i=1}^{t} \varepsilon_i$ and $m = \sum_{i=1}^{t} (m_i - d_{i-1})$.

In light of this composition, many constructions of extractors work by first converting the source into a block source and then applying block-source extraction as above. Our construction will also use this approach (recursively). It is based on the observation that our condenser gives a very simple way to convert a general source into a block source. Indeed, every source of sufficiently high min-entropy is already a block source.

Lemma 4.15. *If X is a $(n - \Delta)$-source of length n, and $X = (\mathbf{X_1}, \mathbf{X_2})$ is a partition of X into blocks of lengths n_1 and n_2, then $(\mathbf{X_1}, \mathbf{X_2})$ is ε-close to some $(n_1 - \Delta, n_2 - \Delta - \log(1/\varepsilon))$ block source.*

The intuition behind the above lemma is that if X is missing only Δ bits of entropy, then no substring of it can be missing more than Δ bits of entropy (even conditioned on the others). The additional $\log(1/\varepsilon)$ bits of entropy loss in $\mathbf{X_2}$ is to ensure that the min-entropy of $\mathbf{X_2}$ is high conditioned on all but an ε fraction of values of $\mathbf{X_1}$.

Consider a k-source $\mathbf{X}$ of length $n = (4/3)k$, i.e. the source has min-entropy rate $3/4$, as can be achieved by applying our condenser. Then setting $\Delta = k/3$ and breaking $\mathbf{X}$ into two halves of length $n/2 = (2/3)k$, we have a block source in which each block has min-entropy roughly $k/3$. Then, by Lemma 4.13, if we want to extract $\Omega(k)$ bits using a seed of length $O(\log n)$, it suffices to have a $(k/3, \varepsilon)$ extractor E_1 with output length $m_1 = \Omega(k)$ and a $(k/3, \varepsilon)$ extractor E_2 with seed length $d_2 = O(\log n)$ such that the output length m_2 of E_2 is at least the seed length d_1 of E_1 (e.g. both can be $\mathrm{poly}(\log k)$). By now, there are many such pairs (E_1, E_2) in the literature, some of which are quite clean and direct. Still, we do not use that approach here, because it is not self-contained, and, more importantly, it does not yield extractors with arbitrarily small error ε.

By induction, we have the following:

Corollary 4.16. *If X is a $(n-\Delta)$-source of length n, and $X = (\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_t})$ is a partition of X into t blocks, each of length at least n', then $(\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_t})$ is $t\varepsilon$-close to some $t \times (n' - \Delta - \log(1/\varepsilon))$ block source.*

Returning to our goal of constructing the extractors of Lemma 4.11, here is our plan for the proof. To convert a general k-source $\mathbf{X}$ into a block source with $t = O(1)$ blocks, we can first use our condenser of Theorem 4.3 to obtain a k-source $\mathbf{X}'$ of length $(1+\alpha)k$ for a sufficiently small constant α, which we then break into t equal-sized blocks. By applying Corollary 4.16 with $\Delta = \alpha k$, the result will be close to a source with min-entropy at least $k/t - \alpha k = \Omega(k)$ per block, provided $\alpha < 1/t$. Applying block-source extraction with the extractor of Lemma 4.8, we obtain extractor promised in Lemma 4.11. The formal details follow.

Proof of Lemma 4.11: Round t up to an integer, and set $\varepsilon_0 = \varepsilon/(4t+1)$. Given a k-source $\mathbf{X}$, we apply the condenser of Theorem 4.3 with error ε_0 and parameter $\alpha = 1/(6t)$. With a seed of length $d' = O(\log(n/\varepsilon_0)) = O(\log(n/\varepsilon))$, this provides us with an $\mathbf{X}'$ of length at most $n' = (1+\alpha)k + O(\log(n/\varepsilon))$ that is ε_0-close to a k-source.

Next, we partition $\mathbf{X}'$ into $2t$ blocks, each of size $n'' = \lfloor n'/(2t) \rfloor$ or $n''+1$. By Corollary 4.16, the result is $(\varepsilon_0 + 2t\varepsilon_0)$-close to a $2t \times k''$ source, where

$$k'' = n'' - \alpha k - O(\log(n/\varepsilon)) \geqslant k/(2t) - \alpha k - O(\log(n/\varepsilon)) = k/(3t) - O(\log(n/\varepsilon)) \; .$$

Now we perform block-source extraction using the "Leftover Hash Lemma" extractor E'' of Lemma 4.8 with input length $n''+1$, min-entropy k'', and error ε_0 to extract from each block. The seed length for E'' is $d'' \leqslant n''+1 = k/t + O(\log(n/\varepsilon)$, and output length $m'' \geqslant \max\{d'', k'' + d'' - 2\log(1/\varepsilon_0)\}$. (Output length $m'' = d''$ is always achievable by simply having the extractor output its seed.)

Applying the block-source extractor of Lemma 4.14 with $E_i = E''$ for every i, the number of bits we extract is

$$m \geqslant 2t \cdot (m'' - d'') \geqslant 2t \cdot (k'' - 2\log(1/\varepsilon_0)) = 2k/3 - O(\log(n/\varepsilon)) \geqslant \lceil k/2 \rceil$$

(the last step follows since if $k \leqslant O(\log(n/\varepsilon))$ we can simply output the seed). The statistical distance increases by at most $2t \cdot \varepsilon_0$, for an output that has distance at most $(4t+1) \cdot \varepsilon_0 = \varepsilon$ from uniform. The total seed length needed for the block-source extraction is $d' + d'' = k/t + O(\log(n/\varepsilon))$. □

4.3.3 The recursion and extractors optimal up to constant factors

We now apply the above techniques recursively to construct an extractor that is optimal up to constant factors for all settings of parameters. This extractor outputs only half of the min-entropy from the source, but we will be able to easily boost this to an output length of $(1-\alpha)k$ for any desired constant $\alpha > 0$, using standard techniques (Theorem 4.19).

Theorem 4.17. *For all positive integers n, k and all $\varepsilon > 0$, there is an explicit construction of a (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = O(\log n + \log(1/\varepsilon))$ and $m \geqslant k/2$.*

Overview of the Construction. Note that for small min-entropies k, namely $k = O(\log(n/\varepsilon))$, this is already achieved by Lemma 4.11 with seed length d smaller than the output length m by any constant factor. (If we allow $d \geqslant m$, then extraction is trivial — just output the seed.) Thus, our goal will be

to recursively construct extractors for large min-entropies using extractors for smaller min-entropies. Of course, if $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ is a (k_0, ε) extractor, say with $m = k_0/2$, then it is also a (k, ε) extractor for every $k \geqslant k_0$. The problem is that the output length is only $k_0/2$ rather than $k/2$. Thus, we need to increase the output length. This can be achieved by simply applying extractors for smaller min-entropies several times.

Lemma 4.18 ([WZ, RRV]). *Suppose $E_1 : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{m_1}$ is a (k_1, ε_1) extractor and $E_2 : \{0,1\}^n \times \{0,1\}^{d_2} \to \{0,1\}^{m_2}$ is a (k_2, ε_2) extractor for $k_2 \leqslant k_1 - m_1 - s$. Then $E' : \{0,1\}^n \times \{0,1\}^{d_1+d_2} \to \{0,1\}^{m_1+m_2}$ defined by $E'(x,(y_1,y_2)) = E_1(x,y_1) \circ E_2(x,y_2)$ is a $(k_1, (1/(1-2^{-s})) \cdot \varepsilon_1 + \varepsilon_2)$ extractor.*

The intuition is that most outputs of E_1 have probability mass $\approx 2^{-m_1}$; thus after conditioning on the output of E_1, the source still has min-entropy $\approx k_1 - m_1$.

To see how we might apply this, consider setting $k_1 = .8k$ and $m_1 = k_1/2$, $\varepsilon_1 = \varepsilon_2 = \varepsilon$, $s = 1$, $k_2 = k_1 - m_1 - 1 \in [.3k, .4k]$, and $m_2 = k_2/2$. Then we obtain a $(k, 3\varepsilon)$ extractor E' with output length $m = m_1 + m_2 > k/2$ from two extactors for min-entropies k_1, k_2 that are smaller than k by a constant factor.

Now, however, the problem is that the seed length grows by a constant factor (e.g. if $d_1 = d_2$, we get seed length $2d$ rather than d). Fortunately, block source extraction (Lemma 4.13, with the extractor of Lemma 4.11 as E_2) gives us a method to reduce the seed length by a constant factor. (The seed length of the composed extractor E' will be the same of that as E_2, which will be a constant factor smaller than its output length m_2, which we can take to be equal to the seed length d_1 of E_1. Thus, the seed length of E' will be a constant factor smaller than that of E_1.) To apply this, we will convert our source to a block source by condensing it to high min-entropy rate and applying Corollary 4.16.

One remaining issue is that the error ε still grows by a constant factor. However, we can start with polynomially small error at the base of the recursion and there are only logarithmically many levels of recursion, so we can afford this blow-up.

We now proceed with the proof details. It will be notationally convenient to do the steps in the reverse order from the description above — first we will reduce the seed length by a constant factor, and then apply Lemma 4.18 to increase the output length.

Proof of Theorem 4.17. Fix $n \in \mathbb{N}$ and $\varepsilon_0 > 0$. Set $d = c \log(n/\varepsilon_0)$ for an error parameter ε_0 and a sufficiently large constant c to be determined in the proof below. (To avoid ambiguity, we will keep the dependence on c explicit throughout the proof, and all big-Oh notation hides universal constants independent of c.) For $k \in [0, n]$, let $i(k)$ be the smallest nonnegative integer i such that $k \leqslant 2^i \cdot 8d$. This will be the level of recursion in which we handle min-entropy k; note that $i(k) \leqslant \log k \leqslant \log n$.

For every $k \in [0, n]$, we will construct an explicit $E_k : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{\lceil k/2 \rceil}$ that is a $(k, \varepsilon_{i(k)})$ extractor, for an appropriate sequence $\varepsilon_0 \leqslant \varepsilon_1 \leqslant \varepsilon_2 \cdots$. Note that we require the seed length to remain d and the fraction of min-entropy extracted to remain 1/2 for all values of k. The construction will be by induction on $i(k)$.

Base Case: $i(k) = 0$, i.e. $k \leqslant 8d$. The construction of E follows from Lemma 4.11, setting $t = 9$ and taking c to be a sufficiently large constant.

Inductive Case: We construct E_k for $i(k) \geqslant 1$ from extractors $E_{k'}$ with $i(k') < i(k)$ as follows. Given a k-source $\mathbf{X}$ of length n, E_k works as follows.

1. We apply our condenser (Theorem 4.3) to convert $\mathbf{X}$ into a source $\mathbf{X}'$ that is ε_0-close to a k-source of length $(9/8)k + O(\log(n/\varepsilon_0))$. This requires a seed of length $O(\log(n/\varepsilon_0))$.

2. We divide $\mathbf{X}'$ into two equal-sized halves $(\mathbf{X_1}, \mathbf{X_2})$. By Corollary 4.16, $(\mathbf{X_1}, \mathbf{X_2})$ is $2\varepsilon_0$-close to a $2 \times k'$ block source for
$$k' = k/2 - k/8 - O(\log(n/\varepsilon_0)) \ .$$
Note that $i(k') < i(k)$. Since $i(k) \geqslant 1$, we also have $k' \geqslant 3d - O(\log(n/\varepsilon_0)) \geqslant 2d$, for a sufficiently large choice of the constant c.

3. Now we apply block-source extraction as in Lemma 4.13. We take E_2 to be a $(2d, \varepsilon_0)$ extractor from Lemma 4.11 with parameter $t = 16$, which will give us $m_2 = d$ output bits using a seed of length $d_2 = (2d)/16 + O(\log(n/\varepsilon_0))$. For E_1, we use our recursively constructed $E_{k'}$, which has seed length d, error $\varepsilon_{i(k')}$, and output length $\lceil k'/2 \rceil \geqslant k/6$ (where the latter inequality holds for a sufficiently large choice of the constant c, because $k > 8d > 8c\log(1/\varepsilon)$).

All in all, our extractor so far has seed length at most $d/8 + O(\log(n/\varepsilon_0))$, error at most $\varepsilon_{i(k)-1} + O(\varepsilon_0)$, and output length at least $k/6$. This would be sufficient for our induction except that the output length is only $k/6$ rather than $k/2$. We remedy this by applying Lemma 4.18.

With one application of the extractor above, we extract at least $m_1 = k/6$ bits of the source min-entropy. Then with another application of the extractor above for min-entropy threshold $k_2 = k - m_1 - 1 = 5k/6 - 1$, by Lemma 4.18, we extract another $(5k/6 - 1)/6$ bits and so on. After four applications, we have extracted all but $(5/6)^4 \cdot k + O(1) \leqslant k/2$ bits of the min-entropy. Our seed length is then $4 \cdot (d/8 + O(\log(n/\varepsilon_0))) \leqslant d$ and the total error is $\varepsilon_{i(k)} = O(\varepsilon_{i(k)-1})$.

Solving the recurrence for the error, we get $\varepsilon_i = 2^{O(i)} \cdot \varepsilon_0 \leqslant \mathrm{poly}(n) \cdot \varepsilon_0$, so we can obtain error ε by setting $\varepsilon_0 = \varepsilon/\mathrm{poly}(n)$. As far as explicitness, we note that computing E_k consists of four evaluations of our condenser from Theorem 4.3, four evaluations of $E_{k'}$ for values of k' such that $i(k') < (i(k) - 1)$, four evaluations of the explicit extractor from Lemma 4.11, and simple string manipulations that can be done in time $\mathrm{poly}(n, d)$. Thus, the total computation time is at most $4^{i(k)} \cdot \mathrm{poly}(n, d) = \mathrm{poly}(n, d)$. □

4.3.4 Main extractor theorem

The extractor of Theorem 4.17 extracts only half of the min-entropy from the source, but we can obtain extractors that obtain any constant fraction of the min-entropy or all the min-entropy by repeated application of Lemma 4.18.

Theorem 4.19 (main extractor result). *For every constant $\alpha > 0$: for all positive integers $n \geqslant k$ and all $\varepsilon > 0$, there is an explicit (k, ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $m = (1 - \alpha)k$ and $d = \log n + O(\log(k/\varepsilon))$.*

Proof. Achieving the parameters in the theorem, except with seed length $O(\log(n/\varepsilon))$ follows immediately by applying Lemma 4.18 $O(1/\alpha)$ times with both extractors being taken from Theorem 4.17. To achieve the promised seed length $\log n + O(\log(k/\varepsilon))$, we first apply our condenser from Theorem 4.4 to the

source. This requires a seed of length $d \leqslant \log n + \log k + \log(1/\varepsilon) + 1$ to condense the source to length $n' \leqslant d \cdot (k+2) = O(k \cdot \log(n/\varepsilon))$, while retaining all of the min-entropy (up to statistical distance ε). Then extracting a constant fraction of the min-entropy only requires an additional seed length $O(\log(n'/\varepsilon)) = O(\log k + \log\log n + \log(1/\varepsilon)) = O(\log(k/\varepsilon))$. (We assume $k \geqslant \log n$; otherwise we can use the trivial extractor that just outputs the seed.) □

Note that an additional improvement of Theorem 4.19 over Theorem 4.17 is that it achieves a constant of 1 in front of the $\log n$. Indeed, when $k = n^{o(1)}$ and $\varepsilon = 1/n^{o(1)}$, the seed length is within a $(1 + o(1))$ factor of the optimal bound $\log n + 2\log(1/\varepsilon) + O(1)$, improving over the extractors of Lu et al. [LRVW] in which the seed length is only optimal to within some large constant factor. (In the conference version of this paper [GUV2], we also showed how to use our techniques together with [Zuc3] to improve the seed length of Theorem 4.19 to $(1+\gamma)\log n + \log k + O(1)$ for arbitrarily small constants $\varepsilon, \gamma > 0$; we omit that result here because the improvement is only for a rather limited range of parameters.)

4.3.5 Extracting all the min-entropy

Next, we give an extractor that extracts all of the min-entropy. In order to also get the min-entropy of the seed, we will use the following variant of Lemma 4.18, where the second extractor is also applied to the seed of the first extractor.

Lemma 4.20 ([RRV]). *Suppose $E_1 : \{0,1\}^{n_1} \times \{0,1\}^{d_1} \to \{0,1\}^{m_1}$ is a (k_1, ε_1) extractor and $E_2 : \{0,1\}^{n_1+d_1} \times \{0,1\}^{d_2} \to \{0,1\}^{m_2}$ is a (k_2, ε_2) extractor for $k_2 \leqslant k_1 + d_1 - m_1 - s$. Then $E' : \{0,1\}^{n_1} \times \{0,1\}^{d_1+d_2} \to \{0,1\}^{m_1+m_2}$ defined by $E'(x,(y_1,y_2)) = E_1(x,y_1) \circ E_2((x,y_1),y_2)$ is a $(k_1, (1/(1-2^{-s})) \cdot \varepsilon_1 + \varepsilon_2)$ extractor.*

Theorem 4.21. *For all positive integers $n \geqslant k$ and all $\varepsilon > 0$, there is an explicit (k,ε) extractor $E : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $m = k + d - 2\log(1/\varepsilon) - O(1)$ and $d = \log n + O(\log k \cdot \log(k/\varepsilon))$.*

Proof. Similar to the proof of Theorem 4.19, we show how to get the larger seed length $O(\log k \cdot \log(n/\varepsilon))$ first; then the result follows by composing the extractor with our condenser from Theorem 4.4.

By applying Lemma 4.18 (with $s = 1$) to our extractors from Theorem 4.17 (with error $\varepsilon_0 = \varepsilon/6k$) $\log k$ times, we obtain a (k, ε_1) extractor $E_1 : \{0,1\}^n \times \{0,1\}^{d_1} \to \{0,1\}^{m_1}$ with seed length $d_1 = O(\log k \cdot \log(n/\varepsilon_0)) = O(\log k \cdot \log(n/\varepsilon))$, output length $m_1 = k$, and error $\varepsilon_1 \leqslant 2 \cdot 2^{\log k} \cdot \varepsilon_0 = \varepsilon/3$. (With $s = 1$, each application of Lemma 4.18 doubles the error and adds ε_0.) Now we use Lemma 4.20 to compose E_1 with the (k_2, ε_2) extractor $E_2 : \{0,1\}^{n+d_1} \times \{0,1\}^{d_2} \to \{0,1\}^{m_2}$ from Lemma 4.9, for min-entropy $k_2 = k + d_1 - m_1 - 1 = d_1 - 1$ and error $\varepsilon_2 = \varepsilon/3$. E_2 has seed length $d_2 = k_2 + O(\log((n+d_1)/\varepsilon_2)) = O(\log k \cdot \log(n/\varepsilon))$, and output length $m_2 = k_2 + d_2 - 2\log(1/\varepsilon_2) - O(1)$. The final extractor E' from Lemma 4.20 has seed length $d_1 + d_2 = O(\log k \cdot \log(n/\varepsilon))$ and output length $m_1 + m_2 = k + d_1 + d_2 - 2\log(1/\varepsilon) - O(1)$. □

Remark 4.22. In some applications of extractors, it is useful to have *strong extractors*, where the seed appears as a substring of the output in a fixed set of coordinates. All of our extractors (namely Theorem 4.17, Theorem 4.19, and Theorem 4.21) can be made to have this property (with no loss in the claimed parameters).[4] To achieve this, we first observe that our condenser (Theorem 4.3) is already strong. (Indeed, the

[4] Another common definition of strong extractor requires that the joint distribution of the seed and output is ε-close to uniform. A strong extractor with output length m in that definition is equivalent to a strong extractor with output length $m+d$ in our definition.

seed y is the first component of the output of $C = \Gamma$ in Equation (1).) Then the fact that C is a $k \to_\varepsilon k + d$ condenser implies that for every k-source $\mathbf{X}$, $C(\mathbf{X}, \mathbf{U_d})$ is ε-close to a joint distribution $(\mathbf{U_d}, \mathbf{Z})$ where for every $y \in \{0,1\}^d$, $\mathbf{Z}|_{\mathbf{U_d}=y}$ is a k-source. Thus, whenever we condense the source in our construction, we can simply save the seed for the output, and operate only on $\mathbf{Z}$ as our condensed source. All of the other compositions and transformations in our construction preserve this notion of strongness.

Remark 4.23. One of the major remaining open problems about extractors is to extract all of the min-entropy (as in Theorem 4.21) with a seed length of $O(\log(n/\varepsilon))$ (as in Theorem 4.19). To this end, it is worth pointing out where we lose entropy in the proof of Theorem 4.19. The first place is in Lemma 4.11, but as pointed out in Remark 4.10 this can be avoided by combining our condenser with extractors from Ramanujan expanders. The other place we lose entropy is in our (repeated) use of Lemma 4.15, where we view a high min-entropy source as a block source. Intuitively, the entropy loss comes because we do not know from which of the two blocks the entropy is missing, so we pessimistically assume it is missing from both. This entropy loss problem has arisen in previous work, and in fact the "zig-zag product" for extractors [RVW] solves it for the case of very high min-entropy $n - \Delta$ (where we can find optimal extractors for sources of length $O(\Delta)$ by exhaustive search). Needless to say, it would be very interesting to eliminate the entropy loss in our setting too.

5 List-decoding view of lossy condensers

In Section 6, we give a (arguably simpler) construction of condensers from Reed-Solomon codes instead of Parvaresh-Vardy codes. The price for this modification is that the resulting objects are no longer *lossless* condensers, but instead just ordinary (lossy) condensers.[5] In this section, we develop a list-decoding characterization of lossy condensers that will be used in the subsequent sections. For this we will need some lemmas about min-entropy.

Proposition 5.1. *A distribution D with min-entropy $\log(K - c)$ is c/K-close to some distribution with min-entropy $\log K$.*

Proof. The distance from D to the closest distribution with min-entropy $\log K$ is

$$\sum_{a:D(a)\geqslant 1/K} (D(a) - 1/K) \leqslant 1 - (K - c) \cdot 1/K = c/K.$$

□

The following lemma gives a useful sufficient condition for a distribution to be close to having large min-entropy:

Lemma 5.2. *Let $\mathbf{Z}$ be a random variable and K a positive integer.*

1. *Suppose that for all sets T of size K, $\Pr[\mathbf{Z} \in T] \leqslant \varepsilon$. Then $\mathbf{Z}$ is ε-close to having min-entropy at least $\log(K/\varepsilon)$.*
2. *Conversely, if $\mathbf{Z}$ is ε-close to having min-entropy at least $\log(K/\varepsilon)$, then $\Pr[\mathbf{Z} \in T] \leqslant 2\varepsilon$ for all sets T of size K.*

[5] We are able to get a lossless condenser from Reed-Solomon codes when the output entropy rate is less than $1/2$.

Proof. 1. Let T be a set of the K heaviest elements x (under the distribution of $\mathbf{Z}$). Let $2^{-\ell}$ be the average probability mass of the elements in T. Then $\varepsilon \geqslant \Pr[\mathbf{Z} \in T] = 2^{-\ell}K$, so $\ell \geqslant \log(K/\varepsilon)$. But every element outside T has weight at most $2^{-\ell}$, and with all but probability ε, $\mathbf{Z}$ hits elements outside T.

2. Suppose that $\mathbf{Z}'$ is the random variable of min-entropy at least $\log(K/\varepsilon)$ that is ε-close to $\mathbf{Z}$, and let T be a set of size K. Then $\Pr[\mathbf{Z} \in T] \leqslant \Pr[\mathbf{Z}' \in T] + \varepsilon \leqslant |T| \cdot (\varepsilon/K) + \varepsilon = 2\varepsilon$.

□

Now we can develop a "list-decoding" view of lossy condensers, analogous to the one we have used for expanders (Lemma 3.2) and the one known for extractors [TZ]. The following definition should be compared to Definition 3.1:

Definition 5.3. *For a function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and a set $T \subseteq \{0,1\}^m$, define*

$$\mathrm{LIST}(T, \varepsilon) \stackrel{\mathrm{def}}{=} \left\{ x : \Pr_y[C(x,y) \in T] > \varepsilon \right\}.$$

Similar to the situation with expanders, if we can bound the size of $\mathrm{LIST}(T, \varepsilon)$ for all sets T that are not too large, then we have a condenser:

Lemma 5.4. *Fix a function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ and positive integers H and L.*

1. *Suppose that every set $T \subseteq \{0,1\}^m$ of size at most L, we have $|\mathrm{LIST}(T, \varepsilon)| \leqslant H$. Then C is a*

$$\log(H/\varepsilon) \to_{2\varepsilon} \log(L/\varepsilon) - 1$$

condenser.

2. *Conversely, suppose that C is a*

$$\log H \to_{\varepsilon} \log(L/\varepsilon)$$

condenser. Then for every set $T \subseteq \{0,1\}^m$ of size at most L, we have $|\mathrm{LIST}(T, 2\varepsilon)| \leqslant H$.

Proof. 1. We have a random variable $\mathbf{X}$ with min-entropy $\log(H/\varepsilon)$. For a fixed T of size at most L, the probability that $\mathbf{X}$ is in $\mathrm{LIST}(T, \varepsilon)$ is at most ε; if that does not happen, then the probability $C(\mathbf{X}, \mathbf{U_t})$ lands in T is at most ε. Altogether the probability $C(\mathbf{X}, \mathbf{U_t})$ falls in T is at most 2ε. Now apply Lemma 5.2.

2. Suppose that there is a set $T \subseteq \{0,1\}^m$ of size at most L for which $|\mathrm{LIST}(T, 2\varepsilon)| > H$. Let $\mathbf{X}$ be a random variable uniformly distributed over $\mathrm{LIST}(T, 2\varepsilon)$; note that $\mathbf{X}$ has min-entropy greater than $\log H$. The probability that $C(\mathbf{X}, \mathbf{U_t})$ lands in T is greater than 2ε. By Lemma 5.2, $C(\mathbf{X}, \mathbf{U_t})$ is not ε-close to any random variable of min-entropy $\log(L/\varepsilon)$, contradicting the condenser property.

□

Thus, up to a constant factor in the error ε and $\log(1/\varepsilon)$ bits of source min-entropy, proving that a function is a condenser is equivalent to bounding the size of $|\mathrm{LIST}(T, \varepsilon)|$ for sets T of a some size L. In the conference version of this paper [GUV2], we used this list-decoding view of lossy condensers to show that we can eliminate the $\log k$ in the seed length of the condenser of Theorem 4.3 (for $k = k_{max}$), at the price of losing a constant fraction of the min-entropy. (The idea was to use the "multiple roots" trick of [GS] in the list-decoding analysis.) We omit that result in this version because the improvement is rather small, and instead use the lossy condenser framework to analyze a "Reed–Solomon" version of our construction.

6 Condensers from Reed-Solomon codes

We use one of the main ideas from the folded Reed-Solomon code construction of Guruswami and Rudra [GR] to argue that a small modification to our construction gives a good condenser from (folded) Reed-Solomon codes, answering a question raised in [KU]. There are two variants of the Reed-Solomon construction: the first is lossy (it loses a constant fraction of the source entropy), but it achieves entropy rate arbitrarily close to 1 (just like the main condenser of Theorem 4.3); the second (pointed out to us by Ariel Gabizon) is lossless, but it only achieves entropy rate 1/2.

6.1 Lossy Reed-Solomon condenser

Let q be an arbitrary prime power, and let $\zeta \in \mathbb{F}_q$ be a generator of the multiplicative group $\mathbb{F}_q^*$. Then the polynomial $E(Y) = Y^{q-1} - \zeta$ is irreducible over $\mathbb{F}_q$ [LN, Chap. 3, Sec. 5]. The following identity holds for all $f(Y) \in \mathbb{F}_q[Y]$:

$$f(Y)^q \equiv f(Y^q) \equiv f(Y^{q-1}Y) \equiv f(\zeta Y) \pmod{E(Y)}.$$

In this case, if we modify our basic function Γ (see (1)) slightly so that we raise f to successive powers of q rather than h, we obtain the function $C : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined by:

$$\begin{aligned} C(f, y) &\stackrel{\text{def}}{=} [y, f(y), (f^q \bmod E)(y), (f^{q^2} \bmod E)(y), \cdots, (f^{q^{m-1}} \bmod E)(y)] \\ &= [y, f(y), f(\zeta y), \cdots, f(\zeta^{m-1} y)]. \end{aligned} \tag{2}$$

In other words, our function interprets its first argument as describing a univariate polynomial over $\mathbb{F}_q$ of degree at most $n-1$ (i.e., a Reed-Solomon codeword), it uses the seed to select a random location in the codeword, and it outputs m successive symbols of the codeword, together with the seed. This is precisely the analogue of the Shaltiel-Umans q-ary extractor construction [SU], for univariate polynomials rather than multivariate polynomials. Alternatively (and following the correspondence with codes described in Section 2.1), $C(f, y)$ is the y'th symbol in an encoding of the "message" f in the "folded Reed–Solomon code" of Guruswami and Rudra [GR]. (Actually, the folded Reed-Solomon codes only take y's from a subset of $\mathbb{F}_q$ in order to save on the codeword length.)

With a minor modification to the proof of Theorem 3.3, we show that this is good condenser:

Theorem 6.1. *Define C as in (2) and $\mathrm{LIST}(T, \varepsilon)$ with respect to C as in Definition 5.3. Then for every $T \subseteq \mathbb{F}_q^{m+1}$ of size at most $L = Ah^m - 1$, we have*

$$|\mathrm{LIST}(T, \varepsilon)| \leqslant (h-1) \cdot \frac{q^m - 1}{q - 1},$$

where $A = \varepsilon q - (n-1)(h-1)m$.

Proof. Let $T \subseteq \mathbb{F}_q^{m+1}$ with $|T| \leqslant Ah^m - 1$. The proof follows along the lines of Theorem 3.3. We interpolate a nonzero polynomial $Q(Y, Y_1, Y_2, \ldots, Y_m)$ that vanishes on T, and and has degree at most $A-1$ in Y and at most $(h-1)$ in each Y_j. The number of coefficients of such a Q equals Ah^m which exceeds $|T|$, and therefore such a nonzero polynomial Q indeed exists. We can also ensure that $E(Y)$ does not divide Q. For every $f(Y) \in \mathrm{LIST}(T, \varepsilon)$, the polynomial $R_f(Y) \stackrel{\text{def}}{=} Q(Y, f(Y), f(\zeta Y), \ldots, f(\zeta^{m-1}Y))$ has more than εq roots, and degree at most $(A-1) + (n-1)(h-1)m$, and therefore must be the zero polynomial. We define Q^* slightly differently:

$$Q^*(Z) \stackrel{\text{def}}{=} Q(Y, Z, Z^q, Z^{q^2}, \ldots, Z^{q^{m-1}}) \bmod E(Y).$$

As before, Q^* is a nonzero polynomial over the extension field $\mathbb{F} = \mathbb{F}_q[Y]/(E(Y))$. Further, every $f(Y) \in \mathrm{LIST}(T, \varepsilon)$, viewed as an element of the extension field $\mathbb{F}$, is a root of Q^*. It follows that $|\mathrm{LIST}(T, \varepsilon)| \leqslant \deg(Q^*)$. The degree of Q^* is at most

$$(h-1)(1 + q + q^2 + \cdots + q^{m-1}) = (h-1) \cdot \frac{q^m - 1}{q - 1}\,,$$

and this proves the claimed bound. □

By picking parameters suitably in the above construction, we obtain the following condenser. Unlike our basic condenser (Theorem 4.3), this condenser is no longer lossless. Instead, the ratio of the input and output min-entropies is $\approx (1 + 1/\alpha)$, which means that we retain only a $\alpha/(1+\alpha)$ fraction of the min-entropy.

Theorem 6.2 (Reed-Solomon lossy condenser). *For every $n \in \mathbb{N}$, $\ell \leqslant n$ such that 2^ℓ is an integer, and $\alpha, \varepsilon > 0$, there is an explicit function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^{n'}$ defined in (2) that is a*

$$(1 + 1/\alpha)\ell t + \log(1/\varepsilon) \to_{3\varepsilon} \ell t + d - 2$$

condenser with $d \leqslant (1 + 1/\alpha)t$ and $n' \leqslant (1 + 1/\alpha)\ell t + d$, where $t = \lceil \alpha \log(4n\ell/\varepsilon) \rceil$, provided $\ell t \geqslant \log(1/\varepsilon)$.

Proof. Set $h = 2^t$ and note that $h^{1/\alpha} \geqslant 4n/\varepsilon$. Let q be the power of 2 in $(h^{1+1/\alpha}/2, h^{1+1/\alpha}]$. Set $m = \ell$. Note that

$$A \stackrel{\text{def}}{=} \varepsilon q - (n-1)(h-1)m \geqslant \varepsilon q - nhm \geqslant \varepsilon q/2,$$

because $q \geqslant h^{1+1/\alpha}/2 \geqslant 2nh\ell/\varepsilon$, and $m = \ell$.

Consider the function $C : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined in (2). By Theorem 6.1, for every $T \subseteq \mathbb{F}_q^{m+1}$ of size at most $L = Ah^m - 1$ we have $|\mathrm{LIST}(T, \varepsilon)| \leqslant q^m - 1$. Applying Lemma 5.4, we find that C is a

$$\log\left(\frac{q^m - 1}{\varepsilon}\right) \to_{2\varepsilon} \log\left(\frac{Ah^m - 1}{2\varepsilon}\right)$$

condenser. By Proposition 5.1, the output distribution of the condenser C is within statistical distance $\frac{1}{Ah^m} \leqslant 2^{-\ell t} \leqslant \varepsilon$ of a distribution with min-entropy at least

$$\log\left(\frac{Ah^m}{2\varepsilon}\right) \geqslant \log q + \ell t - 2 = \ell t + d - 2\,.$$

We can thus conclude that C is a

$$(1 + 1/\alpha)\ell t + \log(1/\varepsilon) \to_{3\varepsilon} \ell t + d - 2$$

condenser. This is the claimed condenser; the upper bounds on d and n' follow from the fact that $q = 2^d \leqslant 2^{(1+1/\alpha)t}$.

Finally, the construction is explicit because a representation of $\mathbb{F}_q$ for q a power of 2 as well as a generator of $\mathbb{F}_q^*$ can be found in time $\mathrm{poly}(\log q)$ [Sho]. □

6.2 Lossless Reed-Solomon condenser

The variant in this subsection is lossless, and so it is most convenient to describe it as an expander graph first and then apply Lemma 4.2. The construction is again obtained by a careful choice of h and the irreducible $E(Y)$. In this variant we require that the parameter h is a prime power greater than n, and that q is a power of h (so $\mathbb{F}_q$ contains a subfield $\mathbb{F}_h$). Let $\zeta \in \mathbb{F}_h$ be a generator of the multiplicative group $\mathbb{F}_h^*$ (compare with the previous section which selected a generator of $\mathbb{F}_q^*$), and define the polynomial $E(Y) = Y^{h-1} - \zeta$. The advantage of these choices for our construction was pointed out to us by Ariel Gabizon.

We identify elements of $\mathbb{F}_h^n$ with polynomials over $\mathbb{F}_h$ that have degree at most $n-1$ (compare with the previous section in which the polynomials were over $\mathbb{F}_q$). The following identity holds for all $f(Y) \in \mathbb{F}_h[Y]$ and $i \geqslant 0$:

$$f(Y)^{h^i} = f(Y^{h^i}) = f(Y^{(h-1)(h^{i-1}+h^{i-2}+\cdots+h+1)}Y) \equiv f(\zeta^i Y) \pmod{E(Y)}. \tag{3}$$

As usual, for ease of notation, we will refer to $(f^{h^i} \bmod E)$ as "f_i." Our expander is the bipartite graph $\Gamma_{\mathrm{RS}} : \mathbb{F}_h^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined as:

$$\begin{aligned} \Gamma_{\mathrm{RS}}(f, y) &\stackrel{\mathrm{def}}{=} [y, f_0(y), f_1(y), f_2(y), \cdots, f_{m-1}(y)] \\ &= [y, f(y), f(\zeta y), f(\zeta^2 y), \ldots, f(\zeta^{m-1} y)]. \end{aligned} \tag{4}$$

Analogous to Theorem 3.3, we have the following:

Theorem 6.3. *The graph $\Gamma_{\mathrm{RS}} : \mathbb{F}_h^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ defined in (4) is a $(\leqslant K_{max}, A)$ expander for $K_{max} = h^m$ and $A = q - (n-1)(h-1)m$, provided $\log_h q$ and $h - 1$ are relatively prime.*

Proof. The proof is exactly the same as the proof of Theorem 3.3, after noting two facts: first, by Eqn. (3) the degree of each of the f_i is at most $n-1$ (even if $h-1$ is larger than n); second, $E(Y)$ as defined in this section is irreducible over $\mathbb{F}_q$ [LN, Chap. 3, Sec. 5] (this is where the coprime requirement on $\log_h q$ and $h-1$ is used). □

Setting parameters we obtain (compare to Theorem 3.5):

Theorem 6.4 (Reed-Solomon expander). *For all positive integers N, $K_{\max} \leqslant N$, and all $1 \geqslant \varepsilon > 0$, there is an explicit $(\leqslant K_{max}, (1-\varepsilon)D)$ expander $\Gamma_{\mathrm{RS}} : [N] \times [D] \to [M]$ with degree $D = O((\log N)(\log K_{max})/\varepsilon)^2$ and $M \leqslant (DK_{max})^2$. Moreover, D and M are powers of 2.*

Proof. We set $n = \log N$, $k = \log K_{\max}$, and h to be the power of 2 in the range $((nk/\varepsilon), 2(nk/\varepsilon)]$. Set $q = h^2$. Observe that $h - 1$ and 2 are relatively prime, so Theorem 6.3 applies. The remainder of the proof proceeds exactly as the proof of Theorem 3.5 with $\alpha = 1$. □

Finally, applying Lemma 4.2, we immediately obtain the following lossless condenser based on Reed-Solomon codes:

Theorem 6.5 (Reed-Solomon lossless condenser). *For every $n \in \mathbb{N}$, $k_{max} \leqslant n$, and $\varepsilon > 0$, there is an explicit function $C : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ with $d = 2(\log n + \log k_{max} + \log(1/\varepsilon)) + O(1)$ and $m \leqslant 2(d + k_{max})$ such that for all $k \leqslant k_{max}$, C is a $k \to_\varepsilon k + d$ (lossless) condenser.*

6.3 Limitation of the Reed-Solomon condensers

For the Reed-Solomon-based construction, a relatively simple argument shows that the entropy rate must in general be a constant less than 1. The example below comes from [GHSZ, TZ] (it applies to the function Γ_{RS} as well as the function C from Eqn. (2), for which it is stated):

Lemma 6.6. *Define C as in Eqn. (2). For every positive integer $p < n$ such that $p|(q-1)$, there is a source $\mathbf{X}$ with min-entropy at least $\lfloor n/p \rfloor \cdot \log q$ for which the support of $C(\mathbf{X}, \mathbf{U}_{\log \mathbf{q}})$ is entirely contained within a set of size w^m, where $w = (q-1)/p + 1$.*

Proof. Take the source to be p-th powers of all polynomials over $\mathbb{F}_q$ of degree at most $\lfloor (n-1)/p \rfloor$. Every output symbol of C is an evaluation of such a polynomial, and therefore must be a p-th power or 0. There are thus only $w = (q-1)/p + 1$ possible output symbols, so the output is contained within a set of size w^m. □

For such a source $\mathbf{X}$, the output min-entropy of C is at most $m \log w$ and the output length is $m \log q$. Thus the output entropy rate is at most

$$\frac{\log w}{\log q} \approx 1 - \frac{\log p}{\log q}.$$

So for example, for a source obtained when $p \approx \sqrt{n}$, the Reed-Solomon condenser C yields constant entropy rate bounded away from 1 unless the seed length $\log q$ is $\omega(\log n)$.

This implies that the entropy rates obtained in Theorems 6.2 and 6.5 are not an artifacts of the analysis. That is, it is not possible to improve the entropy rates (e.g., to $1 - o(1)$) simply by giving a different, improved analysis.

7 Application to Storing Sets

Buhrman, Miltersen, Radhakrishnan, and Srinivasan [BMRV] showed that unbalanced expanders with expansion close to the degree can be used to construct the following kind of data structures for storing sets:

Definition 7.1. *A* randomized bitprobe data structure for set membership *consists of two algorithms:*

- *A (deterministic)* encoding algorithm *that takes a set $S \subseteq [N]$ of size L (specified as a list of elements), a parameter $\varepsilon > 0$, and outputs an encoding $X \in \{0,1\}^M$.*

- *A (randomized)* decoding algorithm *that is given the parameters N, L, ε, an element $x \in [N]$, and* oracle access *to the encoding X, and outputs a bit b.*

*We require that if X is the output of the encoding algorithm on set S, then for every x, the decoding algorithm's output will correctly indicate whether or not x is in S, with probability at least $1 - \varepsilon$ over the algorithm's coin tosses. A q-*query *scheme is one in which the decoding algorithm makes at most q queries to the encoding X. M is called the* length *of the data structure, and ε the error probability.*

We say the data structure is explicit *if the encoding can be computed in time polynomial in its input and output lengths, i.e. time $\mathrm{poly}(L, \log N, \log(1/\varepsilon), M)$ and the decoding can be computed in time polynomial in its input length, i.e. time $\mathrm{poly}(\log N, \log(1/\varepsilon))$.*

The construction of such data structures from expanders is given by the following theorem. As observed by Ta-Shma [Ta-], to have an explicit data structure, we need an expander that not only has an efficiently computable neighbor function but which can also be efficiently "list decoded."

Theorem 7.2 (implicit in [BMRV], explicit in [Ta-]). *If there is a $(\leqslant 2L, (1-\varepsilon)D)$ expander $\Gamma : [N] \times [D] \to [M]$, then there is a randomized one-query bitprobe data structure for subsets of $[N]$ of size at most L with length M and error probability at most 4ε.*

Moroever, if the expander is explicit and for every set $T \subseteq [M]$ of size at most LD, we can compute $\mathrm{LIST}(T, 4\varepsilon)$ in time $\mathrm{poly}(L, \log N, \log(1/\varepsilon), M)$, then the data structure is explicit.

With an optimal expander we have $M = O(LD) = O(L \cdot (\log N)/\varepsilon)$; therefore, the length of the data structure is only an $O(1/\varepsilon)$ factor larger than the $L \log N$ bits that are needed describe the set S without concern for efficient membership tests.

We now observe that our expanders have the list decoding property needed for Theorem 7.2:

Lemma 7.3. *Define $\Gamma : \mathbb{F}_q^n \times \mathbb{F}_q \to \mathbb{F}_q^{m+1}$ as in (1). Then given $T \subseteq \mathbb{F}_q^{m+1}$ and $\varepsilon > 0$, we can compute $\mathrm{LIST}(T, \varepsilon)$ in time $\mathrm{poly}(|T|, n, m, q, \log h)$ provided that $|T| \leqslant Ah^m - 1$, where $A = \varepsilon q - (n-1)(h-1)m$.*

Proof. The observation is that essentially the proof of Theorem 3.3 gives an *algorithm* for computing $\mathrm{LIST}(T, \varepsilon)$. (The proof of Theorem 3.3 corresponds to the case that $\varepsilon = 1$, but as seen in the proof of Theorem 6.1, it generalizes to arbitrary ε if we set $A = \varepsilon q - (n-1)(h-1)m$.) We go through the steps here:

- Set $H = \lceil (|T| + 1)/A \rceil$. Find a polynomial $Q(Y, Y_1, \ldots, Y_m)$ vanishing on T with nonzero coefficients on monomials of the for $Y^i M_j(Y_1, Y_2, \ldots, Y_m)$ for $0 \leqslant i \leqslant A - 1$ and $0 \leqslant j \leqslant H - 1$ (borrowing the notation from the proof of Theorem 3.3). This requires solving a linear system over $\mathbb{F}_q$ with $|T|$ equations and AH unknowns. To ensure Q is not divisibly by $E(Y)$, we repeatedly remove factors of $E(Y)$; there can by at most $A/(n-1)$ such factors.

- As in the proofs of Theorems 3.3 and 6.1, every $f(Y) \in \mathrm{LIST}(T, \varepsilon)$ is a root of the polynomial $Q^*(Z) = Q(Y, Z, Z^h, \ldots, Z^{h^{m-1}}) \bmod E(Y)$ over $\mathbb{F} = \mathbb{F}_q[Y]/E(Y)$. We construct Q^* by first substituting the Z variable and then reducing H different univariate polynomials $p_j(Y)$, each of degree at most $A - 1$, modulo $E(Y)$, which is of degree at most $n - 1$.

- Find the roots f of $Q^*(Z)$, which is a polynomial of degree at most $H - 1$ over the field $\mathbb{F}$, which is of size q^n.

- For each such root f, check whether it is an element of $\mathrm{LIST}(T, \varepsilon)$, which can be done by counting how many of its q neighbors $\Gamma(f, y)$ are in T.

All of these steps can be done in time $\mathrm{poly}(|T|, n, m, q, \log h)$. □

Plugging our expanders into Theorem 7.2, we obtain the following:

Theorem 7.4. *For every N, $L \leqslant N$, and $\varepsilon, \alpha > 0$, there is a randomized one-query bitprobe data structure for subsets of $[N]$ of size at most L with error probability at most ε and length*

$$M = \left(\frac{\log N}{\varepsilon}\right)^{O(1+1/\alpha)} \cdot L^{1+\alpha}.$$

Proof. We show how to achieve the claimed length with error probability at most 4ε for any $\varepsilon > 0$, which is equivalent to the above theorem up to a change in the hidden constant. We will apply Theorem 7.2 with our expander Γ defined in Equation (1). We will set the parameters n, m, q, and h as in the proof of Theorem 3.5, for $K_{max} = \lceil L/3\varepsilon \rceil$. (Note that the upper bound on α is not a problem, since here we may assume $\alpha \leqslant 1$ wlog.) This gives a right-hand side of size

$$M \leqslant D^2 \cdot K_{max}^{1+\alpha} = \left(\frac{\log N}{\varepsilon}\right)^{O(1+1/\alpha)} \cdot L^{1+\alpha},$$

since $D = ((\log N)/\varepsilon)^{O(1+1/\alpha)}$.

Since $K_{max} \geqslant 2L$, we have an explicit $(\leqslant 2L, (1-\varepsilon)D)$ expander and the first condition of Theorem 7.2 is satisfied. For the second condition, we will use Lemma 7.3 to ensure that we can efficiently compute $\mathrm{LIST}(T, 4\varepsilon)$ for every T of size at most LD. Recalling that $D = q$, this imposes the constraint $Lq \leqslant Ah^m - 1$, where $A = 4\varepsilon q - (n-1)(h-1)m$. The settings in Theorem 3.5 ensure that $q \geqslant (n-1)(h-1)m/\varepsilon$, so we have $A \geqslant 3\varepsilon q$. They also ensure that $h^m \geqslant K_{max}$. Thus, we have

$$Ah^m \geqslant 3\varepsilon q K_{max} > Lq + 1,$$

as desired. Thus, we can compute $\mathrm{LIST}(T, 4\varepsilon)$ for $|T| \leqslant LD$ in time $\mathrm{poly}(|T|, n, m, q, \log h) = \mathrm{poly}(M)$. □

The optimal setting of α in the above theorem is $\alpha = \Theta(\sqrt{(\log\log N + \log(1/\varepsilon))/\log L})$, which leads to a bound of

$$M = L \cdot \left(\frac{\log N}{\varepsilon}\right)^{O(1)} \cdot \exp\left(\sqrt{(\log\log N + \log(1/\varepsilon)) \cdot \log L}\right).$$

Previous explicit constructions achieved $M = O(L^2 \cdot (\log N)/\varepsilon^2)$ [BMRV] and $M = L \cdot \exp((\log\log N + \log(1/\varepsilon))^3)$ [Ta-]. Our bound is an improvement when

$$((\log N)/\varepsilon)^{\omega(1)} \leqslant L \leqslant \exp(o((\log\log N + \log(1/\varepsilon))^5)).$$

8 Conclusions

The "list-decoding" view of expanders and condensers used in this paper seems to be quite powerful, leading to constructions that are more direct, achieve improved parameters. It is thus natural to ask how far this approach can be pushed. Constructing unbalanced expanders with expansion close to the degree where the degree and/or size of the right-hand side are within *constant factors* of optimal is a natural next goal. This is closely related to question of constructing truly optimal extractors, ones that are optimal up to *additive* constants in the seed length and/or output length. Towards this end, we wonder if there is some variant of our construction with a better entropy rate – the next natural threshold is to have entropy *deficiency* only $k^{o(1)}$. Another interesting question is whether some variant of these constructions can give a block-wise source directly. Depending on the actual parameters, either of these two improvements have the potential to lead to extractors with optimal output length (i.e. ones extract all the min-entropy). Alternatively, if we can find an extractor with optimal output length for high min-entropy (say $.99n$), then, by composing it with our condenser, we would get one for arbitrary min-entropy. Yet another approach is to eliminate the entropy loss in our recursion construction; see Remark 4.23.

We also wonder whether these new techniques can help in other settings. For example, can we use them to argue about *computational* analogues of the objects in this paper – pseudorandom generators and pseudoentropy generators? Or, can variants of our constructions yield so-called "2-source" objects, in which both the source and the seed are only weakly random? In recent work [RZ], a 3-source extractor was constructed using the techniques from this paper, for the case when one of the sources is much shorter than the other two. Whether one can remove this length restriction and construct a general 3-source (or even 2-source) extractor remains open.

Acknowledgements. This paper began with a conversation at the BIRS workshop "Recent Advances in Computation Complexity" in August 2006. We thank the organizers for inviting us, and BIRS for hosting the workshop. We also thank Kai-Min Chung, Ariel Gabizon, Oded Goldreich, Prahladh Harsha, Farzad Parvaresh, Jaikumar Radhakrishnan, Omer Reingold, Ronen Shaltiel, Prasad Tetali, Dieter van Melkebeek, Michael von Korff, and the anonymous CCC reviewers for helpful comments. We thank Ariel Gabizon for an important observation that enabled the construction in Subsection 6.2.

References

[BMRV] H. Buhrman, P. B. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bitvectors optimal? *SIAM Journal on Computing*, 31(6):1723–1744 (electronic), 2002.

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

[CRVW] M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree expansion beyond the degree/2 barrier. In *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, pages 659–668, 2002.

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

[GG] O. Gabber and Z. Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, June 1981.

[GHSZ] V. Guruswami, J. Hastad, M. Sudan, and D. Zuckerman. Combinatorial bounds for list decoding. *IEEE Transactions on Information Theory*, 48(5):1021–1035, 2002.

[Gil] D. Gillman. A Chernoff bound for random walks on expander graphs. *SIAM J. Comput.*, 27(4):1203–1220 (electronic), 1998.

[GR] V. Guruswami and A. Rudra. Explicit codes achieving list decoding capacity: Error-correction with optimal redundancy. *IEEE Transactions on Information Theory*, 54(1):135–150, January 2008. Preliminary version appeared in STOC 2006.

[GS] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and Algebraic-Geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.

[GT] D. Galvin and P. Tetali. Slow mixing of Glauber dynamics for the hard-core model on regular bipartite graphs. *Random Structures & Algorithms*, 28(4):427–443, 2006.

[GUV1] V. Guruswami, C. Umans, and S. Vadhan. Extractors and condensers from univariate polynomials. Technical Report TR06-134, Electronic Colloquium on Computational Complexity, October 2006.

[GUV2] V. Guruswami, C. Umans, and S. Vadhan. Unbalanced expanders and randomness extractors from Parvaresh–Vardy codes. In *Proceedings of the 22nd Annual IEEE Conference on Computational Complexity (CCC '07)*, pages 96–108, 12–16 June 2007.

[GW] O. Goldreich and A. Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

[HLW] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bull. Amer. Math. Soc. (N.S.)*, 43(4):439–561 (electronic), 2006.

[ILL] R. Impagliazzo, L. A. Levin, and M. Luby. Pseudo-random generation from one-way functions (extended abstracts). In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 12–24, Seattle, Washington, 15–17 May 1989.

[ISW] R. Impagliazzo, R. Shaltiel, and A. Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing*, pages 1–10, 2000.

[IZ] R. Impagliazzo and D. Zuckerman. How to recycle random bits. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 248–253, 1989.

[Kah] N. Kahale. Eigenvalues and expansion of regular graphs. *Journal of the ACM*, 42(5):1091–1106, September 1995.

[KU] S. Kalyanaraman and C. Umans. On obtaining pseudorandomness from error-correcting codes. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS*, volume 4337 of *Lecture Notes in Computer Science*, pages 105–116. Springer, 2006.

[LN] R. Lidl and H. Niederreiter. *Introduction to Finite Fields and their applications*. Cambridge University Press, 1986.

[LPS] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[LRVW] 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.

[Mar1] G. A. Margulis. Explicit constructions of expanders. *Problemy Peredači Informacii*, 9(4):71–80, 1973.

[Mar2] G. A. Margulis. Explicit group-theoretic constructions of combinatorial schemes and their applications in the construction of expanders and concentrators. *Problemy Peredachi Informatsii*, 24(1):51–60, 1988.

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

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

[PV] F. Parvaresh and A. Vardy. Correcting errors beyond the Guruswami-Sudan radius in polynomial time. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 285–294, 2005.

[RRV] R. Raz, O. Reingold, and S. Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. *Journal of Computer and System Sciences*, 65(1):97–128, August 2002. Special Issue on STOC '99.

[RSW] O. Reingold, R. Shaltiel, and A. Wigderson. Extracting randomness via repeated condensing. *SIAM J. Comput.*, 35(5):1185–1209, 2006.

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

[RVW] O. Reingold, S. P. Vadhan, and A. Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. *Electronic Colloquium on Computational Complexity (ECCC)*, 8(18), 2001.

[RZ] A. Rao and D. Zuckerman. Extractors for three uneven-length sources. *Manuscript*, April 2008.

[Sha] R. Shaltiel. Recent developments in explicit constructions of extractors. *Bulletin of the European Association for Theoretical Computer Science*, 77:67–, June 2002. Columns: Computational Complexity.

[Sho] V. Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990.

[STV] M. Sudan, L. Trevisan, and S. Vadhan. Pseudorandom generators without the xor lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[SU] R. Shaltiel and C. Umans. Simple extractors for all min-entropies and a new pseudorandom generator. *Journal of the ACM*, 52(2):172–216, 2005. Conference version appeared in FOCS 2001.

[Sud] M. Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *J. Complexity*, 13(1):180–193, 1997.

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

[Ta-] A. Ta-Shma. Storing information with extractors. *Inform. Process. Lett.*, 83(5):267–274, 2002.

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

[TU] A. Ta-Shma and C. Umans. Better lossless condensers through derandomized curve samplers. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006. To appear.

[TUZ] 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.

[TZ] A. Ta-Shma and D. Zuckerman. Extractor codes. *IEEE Transactions on Information Theory*, 50(12):3015–3025, 2004.

[TZS] A. Ta-Shma, D. Zuckerman, and S. Safra. Extractors from Reed-Muller codes. *J. Comput. Syst. Sci.*, 72(5):786–812, 2006.

[Uma] C. Umans. Pseudo-random generators for all hardnesses. *J. Comput. Syst. Sci.*, 67(2):419–440, 2003.

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

[Zuc1] D. Zuckerman. Simulating BPP using a general weak random source. *Algorithmica*, 16(4-5):367–391, 1996.

[Zuc2] D. Zuckerman. Randomness-optimal oblivious sampling. *Random Struct. Algorithms*, 11(4):345–367, 1997.

[Zuc3] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 681–690, 2006.