Entropy Waves, the Zig-Zag Graph Product, and New Constant-Degree Expanders and Extractors[*]

Omer Reingold[†] Salil Vadhan[‡] Avi Wigderson[§]

February 23, 2001

Abstract

The main contribution of this work is a new type of graph product, which we call the **zig-zag product**. Taking a product of a large graph with a small graph, the resulting graph inherits (roughly) its size from the large one, its degree from the small one, and its expansion properties from both! Iteration yields simple explicit constructions of constant-degree expanders of every size, starting from one constant-size expander.

Crucial to our intuition (and simple analysis) of the properties of this graph product is the view of expanders as functions which act as "entropy wave" propagators — they transform probability distributions in which entropy is concentrated in one area to distributions where that concentration is dissipated. In these terms, the graph product affords the constructive interference of two such waves.

A variant of this product can be applied to extractors, giving the first explicit extractors whose seed length depends (poly)logarithmically on only the entropy deficiency of the source (rather than its length) and that extract almost all the entropy of high min-entropy sources. These high min-entropy extractors have several interesting applications, including the first constant-degree explicit expanders which beat the "eigenvalue bound."

Keywords: expander graphs, extractors, dispersers, samplers, graph products

[*] This work was begun at the DIMACS Workshop on Pseudorandomness and Explicit Combinatorial Constructions in October 1999, and an extended abstract of it appeared in FOCS '00 [RVW00]. **This version mentions some results not contained in previous versions.**

[†] AT&T Labs - Research. Room A243, 180 Park Avenue, Bldg. 103, Florham Park, NJ, 07932, USA. E-mail: omer@research.att.com. Part of this research was performed while visiting the Institute for Advanced Study, Princeton, NJ.

[‡] Harvard University, Cambridge, MA, and Institute for Advanced Study, Princeton, NJ. E-mail: salil@deas.harvard.edu. URL: http://deas.harvard.edu/~salil. Work done while at MIT, supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

[§] Institute for Advanced Study, Princeton and the Hebrew University, Jerusalem. Address: Institute for Advanced Study, School of Mathematics, Einstein Drive, Princeton, NJ 08540. E-mail: avi@ias.edu. Partially supported by Israel – US Binational Science Foundation grant number 97-00188.

Contents

1 Introduction

1.1 Expanders

Expanders are graphs which are sparse but nevertheless highly connected. Such graphs have been used to address many fundamental problems in computer science, on topics including network design (e.g. [Pip87, PY82, AKS83]), complexity theory ([Val77, Sip88, Urq87]), derandomization ([NN93, INW94, IW97]), coding theory ([SS96, Spi96]), and cryptography ([GIL$^+$90]). Standard probabilistic arguments ([Pin73]) show that almost every constant-degree (≥ 3) graph is an expander. However, explicitly constructing such graphs seemed to be much harder and this led to an exciting and extensive body of research, developed mainly by mathematicians intrigued by this computer science challenge.

Most of this work was guided by the sufficient[1] condition for the expansion of (infinite families of constant-degree regular) graphs discovered by Tanner [Tan84] (see also [AM85]): the second largest eigenvalue of the adjacency matrix should be strictly smaller than the degree. This naturally led researchers to consider algebraic constructions, where this eigenvalue can be estimated. The celebrated sequence of papers [Mar73, GG81, AM85, AGM87, JM87, LPS88, Mar88, Mor94] provided such constant-degree expanders. All these graphs are extremely simple to describe: given the name of a vertex (in binary), its neighbors can be computed in polynomial time (or even logarithmic space). This level of explicitness is essential for many of the applications. However, the analysis bounding the eigenvalue is quite sophisticated (and often based on deep mathematical results). Thus, it is hard to intuitively understand why these graphs are expanders.

A deviation from this path was taken by Ajtai [Ajt94], who proposed a combinatorial construction of cubic expanders. It starts with an arbitrary cubic N-vertex graph and applies a sequence of polynomially many local operations which gradually increase the girth and turn it into an expander. However, the resulting graph does not have any simply described form and lacks the explicitness level (and hence applicability) of the algebraic constructions mentioned above.

In this work, we give a simple, combinatorial construction of constant-degree expander graphs.[2] Moreover, the analysis proving expansion (via the second eigenvalue) is as simple and follows a clear intuition. The construction is iterative, and needs as a basic building block a *single, almost arbitrary* expander of constant size. The parameters required from it can be easily obtained explicitly, but exhaustive search is an equally good solution since it requires only constant time. Simple operations applied to this graph generate another whose size is increased but whose degree and expansion remain unchanged. This process continues, yielding arbitrarily large expanders.

The heart of the iteration is our new "zig-zag" graph product. Informally, taking a product of a large graph with a small graph, the resulting graph inherits (roughly) its size from the large one, its degree from the small one, and its expansion properties from both! (That is, the composed graph has good expansion properties as long as the two original graphs have good expansion properties.)

Below we sketch the construction and the new graph product.

1.2 Overview of Expander Construction

In this section, we describe a simplified, but less efficient, version of our expander construction and omit formal proofs. Our full construction is described in detail in Section 3. Throughout this section, all graphs are undirected and may have loops and parallel edges.

The Basic Operations. Three essential parameters play a role in an expander — size, degree and expansion. We classify graphs accordingly.

[1] This condition turned out to be necessary as well [Alo86a].

[2] We can even achieve degree 3, which is clearly the smallest possible.

Definition 1.1 *An (N, D, λ)-**graph** is any D-regular graph on N vertices, whose 2nd largest eigenvalue (of the associated random walk) has absolute value at most λ.*[3]

We use two operations on (the adjacency matrices of) graphs — the standard matrix squaring, and our new zig-zag graph product. Here is their effect on the above three parameters.

SQUARING: Let G^2 denote the square of G. That is, the edges in G^2 are paths of length 2 in G. Then

Fact 1.2 $(N, D, \lambda)^2 \to (N, D^2, \lambda^2)$

THE ZIG-ZAG PRODUCT: Let $G \textcircled{z} G_2$ denote the zig-zag product of G_1 and G_2. Then,

Theorem 1.3 $(N_1, D_1, \lambda_1) \textcircled{z} (D_1, D_2, \lambda_2) \to (N_1 \cdot D_1, D_2^2, \lambda_1 + \lambda_2 + \lambda_2^2)$

(The eigenvalue bound of $\lambda_1 + \lambda_2 + \lambda_2^2$ is improved somewhat in Sections 3 and 4.2.)

The Iterations. Let H be any $(D^4, D, 1/5)$-graph, which will serve as the building block for our construction.[4] We define a sequence of graphs G_i as follows.

- $G_1 = H^2$
- $G_{i+1} = G_i^2 \textcircled{z} H$

From Fact 1.2 and Theorem 1.3 above, it is easy to conclude that this sequence is indeed an infinite family of expanders:

Theorem 1.4 *For every i, G_i is an $(N_i, D^2, 2/5)$-graph with $N_i = D^{4i}$*

As mentioned above, this construction is not as efficient as we would like — computing neighborhoods in G_i takes time $\mathrm{poly}(N_i)$ rather than $\mathrm{polylog}(N_i)$. As we show in Section 3, this is easily overcome by augmenting the iterations with another standard graph operation.

1.3 The Zig-Zag Graph Product

The new product mentioned above takes a large graph and a small one, and produces a graph that (roughly speaking) inherits the size of the large one but the degree of the small one. This was the key to creating arbitrarily large graphs with bounded degrees. Naturally, we are concerned with maintaining the expansion properties of the two graphs. First, we describe the product.

For simplicity, we assume that the edges in our D-regular graphs are actually partitioned to D perfect matchings (or color classes). For a color $i \in [D]$ and a vertex v let $v[i]$ be the neighbor of v along the edge colored i.[5] With this simple notation, we can formally define the zig-zag product$\textcircled{z}$ (and then explain it).

Definition 1.5 *Let G_1 be an D_1-regular graph on $[N_1]$ and G_2 a D_2-regular graph on $[D_1]$. Then $G_1 \textcircled{z} G_2$ is a D_2^2-regular graph on $[N_1] \times [D_1]$ defined as follows: For all $v \in [N_1], k \in [D_1], i, j \in [D_2]$, the edge (i, j) connects the vertex (v, k) to the vertex $(v[k[i]], k[i][j])$.*

[3] The transition matrix of the random walk on G is the adjacency matrix divided by D, and we are looking at the second largest eigenvalue $\in [0, 1]$ of this matrix.

[4] For completeness, in Section 3.4 we include two simple explicit constructions of such an H based on [Alo86b, AR94].

[5] This assumption causes some loss of generality, because our "zig-zag" product does not preserve the property that the label (color) of an edge (u, v) is the same from the perspective of both u and v. The formal construction uses something we call a "rotation map" (introduced in Section 2) to keep track of how the label of an edge changes when moving from one endpoint to the other.

What is going on? Note that the size of the small graph G_2 is the degree of the large graph G_1. Thus a vertex name in $G_1 \textcircled{z} G_2$ has a first component which is a vertex of the large graph, and a second which is viewed both as a vertex of the small graph *and* an edge color of the large one. The edge label in $G_1 \textcircled{z} G_2$ is just a pair of edge labels in the small graph. One step in the new product graph from a vertex (v, k) along the edge (i, j) can be broken into three substeps.

1. $(v, k) \rightarrow (v, k[i])$ — A step ("zig") in the small graph moving k to $k[i]$. This affects only the second component, according to the first edge label.

2. $(v, k[i]) \rightarrow (v[k[i]], k[i])$ — A step in the large graph, changing the first component according to the second, viewed as an edge color.

3. $(v[k[i]], k[i]) \rightarrow (v[k[i]], k[i][j])$ – A step ("zag") in the small graph moving $k[i]$ to $k[i][j]$. This affects only the second component, according to the second edge label.

1.4 Analysis of the Zig-Zag Product

Intuition. Why does it work? More precisely, why does Theorem 1.3 hold? What this theorem says intuitively, is that $G_1 \textcircled{z} G_2$ is a good expander as long as both G_1 and G_2 are good expanders. Consider the above three steps as a random walk on $G_1 \textcircled{z} G_2$. Then Steps 1 and 3 are independent random steps on the small graph. If at least one of them "works" as well as it does in the small graph, this would guarantee that the new graph is as good expander as the small one. So let's argue (very intuitively) that indeed one of them "works".

A random step in an expander increases the (H_2-) entropy of a distribution on the vertices, *provided that it is not already too close to uniform.* Let us consider a distribution on the vertices of the new graph (v, k). Roughly speaking, there are two cases.

- If the distribution of the second component k (conditioned on v) is not too uniform, then Step 1 "works". Since Step 2 is just a permutation and Step 3 is a random step on a regular graph, these steps cannot make the distribution less uniform and undo the progress made in Step 1.

- If k (conditioned on v) is very close to uniform, then Step 1 is a "waste". However, Step 2 is then like a real random step in the large expander G_1! This means that the entropy of the first component v increases. Note that Step 2 is a permutation on the vertices of $G_1 \textcircled{z} G_2$, so if entropy increases in the first component, it decreases in the second. That means that in Step 3 we are in the good case (the conditional distribution on the second component is far from uniform), and the entropy of the second component will increase by the expansion of the small graph.

The key to this product is that Step 2 is simultaneously a permutation (so that any progress made in Step 1 is preserved) and an operation whose "projection" to the first component is simply a random step on the large graph (when the second component is random). All previous discussions of expanders focused on the increase of entropy to the vertex distribution by a step along a random edge. We insist on keeping track of that edge name, and consider the joint distribution! In a good expander, if the edge is indeed random, the entropy propagates from it to the vertex. This reduces the (conditional) entropy in the edge. Thus the "entropy wave" in Step 2, in which no fresh randomness enters the distribution on vertices of $G_1 \textcircled{z} G_2$, is what facilitates entropy increase in Steps 1 or 3. Either the "zig" step does it, if there is room for more entropy in k, or if not (which may be viewed as destructive interference of the large and small waves in Step 1), Step 2 guarantees constructive interference in Step 3. Moreover, Step 1 is not redundant as, if there is no or little initial entropy in k, the wave of Step 2 (being a permutation) may flood k with entropy, destroying the effect of Step 3. It is important to note that we discovered this viewpoint (of keeping track of the edge name) while trying to find expander analogies of constructions in the world of “extractors” (mainly from [RR99]) that preserve entropy in a condensed form. The formal linear algebra proof, given in Section 3.3, follows the above intuition very closely.

1.5 Extensions to the Expander Construction

Having achieved our basic goal of simple explicit construction of constant-degree expanders, we turn to study some refinements and variants.

Smaller Degree. A naive and direct implementation of our graph product yields expanders whose degree is reasonable, but not that small (something under 1000). In Section 3.2, we show how to combine this construction, together with *one, constant-size* cycle, to obtain an infinite family of explicit degree 4 expanders. Again, this combination uses the zig-zag product. Actually, using the simpler product mentioned below, we can obtain degree 3 expanders.[6]

Better Degree vs. Eigenvalue Relation. It was shown by Alon and Boppana that D-regular expanders cannot have their 2nd eigenvalue smaller than $\Theta(1/D^{1/2})$ (cf., [Alo86a, LPS88, Nil91]). Graphs achieving this optimum were called **Ramanujan**, and were first constructed by Lubotzky, Phillips, and Sarnak [LPS88] and Margulis [Mar88] (with more recent constructions by Morgenstern [Mor94]). Our basic zig-zag product, applied recursively to one fixed Ramanujan graph will yield D-regular expanders of 2nd largest eigenvalue $O(1/D^{1/4})$. A slightly more sophisticated zig-zag product, given in Section 4.1, improves this relation and achieves second eigenvalue $O(1/D^{1/3})$.

Expansion of Min-Entropy. Our basic zig-zag theorem analyzes the effect the composition has on the second eigenvalue as expansion measure, or equivalently, how the graph increases Renyi's H_2-entropy. But the intuition for the proof of the theorem suggests it should work for other entropy measures. Attempting to apply it for the standard definition of set expansion (namely the support of a distribution — a very naive “entropy” measure) fails. However, with hindsight, a natural choice strengthening this standard notion is **min-entropy expansion**. In a subsequent paper, we will introduce this notion, relate it to standard expansion on the one hand, and show that the zig-zag product affects it as it does the 2-entropy expansion. Furthermore, this definition appears to lead to a better relationship between degree and (vertex) expansion than that achieved by the zig-zag product for 2-entropy. The reason is that this definition allows us to incorporate extractors into the zig-zag product.

A Simpler Product. In the zig-zag product, the traversal of an edge consists of three substeps. An more natural and simpler graph product $G_1 * G_2$ is obtained by taking the *union* of the edges corresponding to the substeps rather than combining them into a single edge. In the notation of Section 1.2, connect vertex $(v,k) \in [N_1] \times [D_1]$ to vertex $(v[k],k)$ and the vertices $(v,k[i])$ for all $i \in D_2$. This product has the advantage that the degree is only D_2+1, rather than D_2^2. (The “cube-connected cycle” is a well-known example of this product, obtained by taking G_1 to be a hypercube of 2^n vertices and degree n, and G_2 to be a cycle on n vertices.) Moreover, we can use the zig-zag analysis to show that if G_1 and G_2 are good expanders then so is $G_1 * G_2$.[7] We defer further details of this product to a subsequent paper.

[6] As pointed out to us by Noga Alon, any family of constant-degree expanders can be easily turned into a family of degree 3 expanders, but our methods give a way to reduce the degree of any expander with good control over the effect on the eigenvalue.

[7] One way to see this is to observe that the *cube* of $G_1 * G_2$ contains $G_1 \textcircled{z} G_2$ as a regular subgraph. However, we can obtain a better bound on the eigenvalue via a direct analysis. Actually, to optimize the degree-eigenvalue relationship, it turns out to be better to give the edges of the form $(v[k],k)$ multiplicity D_2, for total degree of $2D_2$.

A Connection to Algebra. Lubotzky and Wigderson [LW00] have shown that, under certain algebraic conditions, if we take the zig-zag product of Cayley graphs for groups G_1 and G_2, we obtain a Cayley graph for the *semidirect product* of G_1 and G_2. They have used this connection to disprove the conjecture of Lubotzky and Weiss [LW93] that the property of a Cayley graph being an expander is independent of the set of generators.

1.6 Extractors

Like expanders, extractors are a fundamental construct in theoretical computer science. They originate in three independent research tracks regarding probabilistic algorithms:

Deterministic Amplification The attempts to reduce their error with few random bits (initiated in [KPS85, CG89, Sip88, CW89, IZ89],

Weak Random Sources The attempts to perform them using a source of biased, correlated coin flips (initiated in [Blu86, SV86, CG88, VV85]), and

Derandomization The attempts to derandomize completely probabilistic small-space algorithms which use a few random bits (initiated in [AKS87])

In 1990, Zuckerman (cf., [Zuc96]) proposed the following definition of a weak random source (parameterized by a number k and termed a k-**source**): It is a probability distribution on n bits in which no string has probability larger than 2^{-k}. So, intuitively, the distribution has k bits of randomness, and this particular (H_∞) notion of entropy turns out to be the most appropriate in this setting. With this definition and the subsequent paper of Nisan and Zuckerman [NZ96], it became clear that the same construct — the **extractor**, which they defined — addresses all of the above three problems. Moreover, it turned out to be fundamental derandomization tool and found other applications, such as sampling [Zuc97] and various combinatorial constructions (including certain kinds of expander graphs and other networks) [WZ99].

For the purpose of the introduction, we will use a simplified definition. Intuitively, an extractor is a function that converts the k bits of entropy “hidden” in every k-source on n bits into an (almost) uniform distribution on k bits. It is not hard to see that this task is impossible if the extractor is completely deterministic, so we allow a few additional truly random d bits of randomness as another input to the extractor. This input, sometimes referred to as the **seed**, serves as a catalyst to the process of extraction. So formally, denoting by (t) the set of all strings of length t and U_t the uniform distribution on this set, we can give a simplified definition of extractors. (The more general definition, which can be found in Section 5, allows the output of fewer bits, and on the other hand may demand subconstant error ε.)

Definition 1.6 (extractors, simplified) *A function* $\mathrm{E} : (n) \times (d) \mapsto (k)$ *is an* **extractor** *if for* every k*-source* X*, the distribution* $\mathrm{E}(X, U_d)$ *has statistical difference* ε *from* U_k *(for some small constant* ε*, e.g.* $\varepsilon = .01$*).*

Minimizing the **seed length** d (and trading it off with the other parameters we hid under the rug in this definition) is the goal in constructing extractors. Intuitively, in most derandomization tasks, a deterministic algorithm will enumerate all 2^d possible values of the seed, thus relying only on the randomness of the source. So, the efficiency of such applications depends crucially on d.

Nisan and Zuckerman [NZ96] proved that the seed length d is at least $\Omega(\log(n-k))$. This is easily matched by a nonconstructive upper bound, applying the Probabilistic Method to a random function E as above. However, all applications clearly need an **explicit** extractor, namely a function E which can be computed in polynomial time. An impressive body of work has developed on the problem of explicitly constructing extractors over the last few years (see, e.g., [Zuc97, NT99, Tre99, RSW00] and the references therein).

1.7 The Case of High Min-Entropy

Almost all the previous work on extractors focused on the case that the source min-entropy k is smaller than cn for some constant $c < 1$. (This is indeed the case for most applications.) In this case, the lower bound on the seed length d is $\Omega(\log n)$ and efforts concentrated on achieving, or coming close to, this bound with explicit extractors. However, when k is much closer to n (which is natural in several applications mentioned below), even smaller values of d are possible. It is natural in this case to define $\Delta = n - k$ to be the **entropy deficiency** of the k-source and look for explicit extractors whose seed length depends only on Δ, but not on n. Goldreich and Wigderson [GW97] studied this “high min-entropy” case, giving an explicit extractor whose seed length is $d = O(\Delta)$. They also gave extractors (under the more general definition) with shorter seeds; however, these extractors lose more than Δ bits of entropy (i.e. their output length is less than $k - \Delta$, rather than being k).

In this paper, we continue that work, giving explicit extractors with the optimal (up to constant factors) $d = O(\log \Delta)$ for small values of Δ (below loglog n) and nearly optimal $d = O(\log^3 \Delta)$ for every value of Δ (without losing entropy as in [GW97]). Stated differently, we give a reduction from the problem of constructing high min-entropy extractors for long sources (of length n) to that of constructing extractors for sources of length $O(\Delta)$, the deficiency! When Δ is sufficiently small, optimal extractors can be obtained in polynomial time by brute force. Otherwise, we use the best existing explicit constructions. This reduction is achieved using a “Zig-Zag Composition Theorem” for extractors (which is analogous to the zig-zag product for expanders). It is interesting to note that we obtained this composition theorem before the one for expanders.

1.8 Applications of Our Extractors

The significance of the improved bounds we obtain is illustrated by several applications described below (with more details in Section 7). Below and throughout the paper, all logarithms are base 2.

Averaging Samplers. A function $f : (m) \to [0,1]$ is given by a black box and we want to efficiently estimate its average, up to an additive error $\varepsilon = .01$ (for simplicity). An **averaging sampler** (also known as an **oblivious sampler** [BR94]) uses some n random bits to compute some t sample points in (m), and returns the average of their f-values. The probability that this average deviates too much from the truth should be at most γ, regardless of which f is in the box. The goal is to simultaneously minimize both n, the number of random bits used, and t, the number of samples. Nonconstructively, it can be done with $n = m + \log(1/\gamma) + O(1)$ random bits and $t = O(\log(1/\gamma))$ samples [CEG95, Zuc97]. The most randomness-efficient **explicit** (i.e., polynomial-time) construction is due to Zuckerman [Zuc97]. He observed that averaging samplers are essentially equivalent to extractors, and using his extractor construction obtained $n = (1+\alpha)\cdot(m+\log(1/\gamma))$ for an arbitrarily small constant α, with $t = \mathrm{poly}(m, \log(1/\gamma))$. Using one of our extractors (which makes use of of Zuckerman's extractor), we improve this to $n = m + (1+\alpha)\cdot\log(1/\gamma)$ and $t = \mathrm{poly}(\log(1/\gamma))$. Most notably, the polynomial dependence of t on m and $\log(1/\gamma)$ in Zuckerman's construction has turned into a polynomial dependence on just $\log(1/\gamma)$; this corresponds to the fact that number of truly random bits in our extractor depends only on the entropy deficiency Δ rather than the source length n.

Expanders Beating the Eigenvalue Bound. What is the smallest degree needed to ensure that in a graph of N vertices every two sets of size N/A have an edge between them? Random graphs show that degree $O(A \cdot \log A)$ suffices, but explicit constructions have failed to match this bound. An application of the best known relation between eigenvalues and vertex expansion [Tan84] shows that Ramanujan graphs (e.g., as given by [LPS88, Mar88, Mor94]) of degree $\Theta(A^2)$ suffice. To beat this “eigenvalue bound,” Wigderson and

Zuckerman [WZ99] suggested to build such graphs from extractors and obtained degree $A \cdot N^{o(1)}$, which was important for many applications where A is a fixed power of N. However, for very small A, even much better dependence on N obtained in subsequent work (e.g., [NT99]) does not beat the eigenvalue bound. We show for any constant A, that degree $O(A \cdot \log^4 A)$ suffices, almost matching the random graph bound.

Error Reduction for Dispersers. Dispersers are the one-sided analogue of extractors — instead of inducing a distribution that is ε-close to uniform on their output, they are only required to hit all but an ε fraction of their range with nonzero probability. They were introduced by Sipser [Sip88] and predate the notion of extractors. For simplicity above, we treated the error ε of extractors as a small constant, but in general one wants dispersers and extractors whose parameters have a near-optimal dependence on ε. An optimal disperser's parameters have a better dependence on ε than an optimal extractor — in a disperser, to achieve an error of ε, the seed length need only grow by an additive $\log(1/\varepsilon)$ and the "entropy loss" [8] need only be $\operatorname{loglog}(1/\varepsilon)$, whereas for extractors both the seed length and entropy loss must be at least $2\log(1/\varepsilon))$ [RT97]. Using our high min-entropy extractors, we give the first explicit constructions of dispersers which achieve a better dependence on ε than can be achieved with extractors (in both the seed length and entropy loss). More generally, we use our high min-entropy extractors to give a method to reduce the error of any disperser from a constant to an arbitrary ε paying an essentially optimal price in terms of seed length and entropy loss. (A related error reduction technique for dispersers was independently discovered by Ta-Shma and Zuckerman.)

2 Expander Preliminaries

2.1 Graphs and Rotations

All graphs we discuss may have self loops and parallel edges. They are best described by their (nonnegative, integral) adjacency matrix. Such a graph is **undirected** iff the adjacency matrix is symmetric. It is D-**regular** if the sum of entries in each row (and column) is D (so exactly D edges are incident on every vertex).

Let G be a D-regular undirected graph on N vertices. Suppose that the edges leaving each vertex of G are labeled from 1 to D in some arbitrary, but fixed, way. Then for $v, w \in [N]$ and $i \in [D]$, it makes sense (and is standard) to say "the i'th neighbor of vertex v is w". In this work, we make a point to always keep track of the edge traversed to get from v to w. This is formalized as follows:

Definition 2.1 *For a D-regular undirected graph G, the* **rotation map** $\mathrm{Rot}_G : [N] \times [D] \to [N] \times [D]$ *is defined as follows:* $\mathrm{Rot}_G(v, i) = (w, j)$ *if the i'th edge incident to v leads to w, and this edge is the j'th edge incident to w.*

This definition enables us to remove the simplifying assumption made in the introduction, which was that the label of an edge is the same from the perspective of both endpoints, i.e. $\mathrm{Rot}_G(v, i) = (w, j) \Rightarrow i = j$. From Definition 2.1, it is clear that Rot_G is a permutation, and moreover $\mathrm{Rot}_G \circ \mathrm{Rot}_G$ is the identity map. We will always view graphs as being specified by their rotation maps. Hence we call a family $\mathcal{G}$ of graphs **explicit** if for every $G \in \mathcal{G}$, Rot_G is computable in time polylog N, where N is the number of vertices of G. That is, graphs in $\mathcal{G}$ are indexed by some parameters (such as the number of vertices and the degree, which may be required to satisfy some additional relations) and there should be a single algorithm which efficiently computes Rot_G for any $G \in \mathcal{G}$ when given these parameters as an additional input. We will often

[8] This is the total randomness invested ($k + d$) minus the output length. Definition 1.6 assumes the output is k, but the more general definitions allow it to vary.

informally refer to an individual graph as explicit, as shorthand for saying that the graph comes from an explicit family.

2.2 Eigenvalues and Expansion

The **normalized adjacency matrix** M of G is the adjacency matrix of G divided by D. In terms of the rotation map, we have:

$$M_{u,v} = \frac{1}{D} \cdot \left|\{(i,j) \in [D]^2 : \mathrm{Rot}_G(u,i) = (v,j)\}\right| .$$

M is simply the transition matrix of a random walk on G. By the D-regularity of G, the all-1's vector $1_N = (1, 1, \ldots, 1) \in \mathbb{R}^N$ is an eigenvector of M of eigenvalue 1. It is turns out that all the other eigenvalues of M have absolute value at most 1, and it is well-known that the second largest eigenvalue of G is a good measure of G's expansion properties [Tan84, AM85, Alo86a]. We will use the following variational characterization of the second largest eigenvalue.

Definition 2.2 $\lambda(G)$ *denotes the* **second largest eigenvalue** *(in absolute value) of G's normalized adjacency matrix. Equivalently,*

$$\lambda(G) = \max_{\alpha \perp 1_N} \frac{|\langle \alpha, M\alpha \rangle|}{\langle \alpha, \alpha \rangle} = \max_{\alpha \perp 1_N} \frac{\|M\alpha\|}{\|\alpha\|}.$$

Above, $\langle \cdot, \cdot \rangle$ refers to the standard inner product in $\mathbb{R}^N$ and $\|\alpha\| = \sqrt{\langle \alpha, \alpha \rangle}$.

The meaning of $\lambda(G)$ can be understood as follows: Suppose $\pi \in [0,1]^N$ is a probability distribution on the vertices of G. By linear algebra, π can be decomposed as $\pi = u_N + \pi^{\perp}$, where $u_N = 1_N/N$ is the uniform distribution and $\pi^{\perp} \perp u_N$. Then $M\pi = u_N + M\pi^{\perp}$ is the probability distribution on vertices obtained by selecting a vertex v according to π and then moving to a uniformly selected neighbor of v. By Definition 2.2, $\|M\pi^{\perp}\| \leq \lambda(G) \cdot \|\pi^{\perp}\|$. Thus $\lambda(G)$ is a measure of how quickly the random walk on G converges to the uniform distribution. Intuitively, the smaller $\lambda(G)$ is, the better the expansion properties of G. Accordingly, an (infinite) family $\mathcal{G}$ of graphs is called a family of **expanders** if these eigenvalues are bounded away from 1, i.e. there is a constant $\lambda < 1$ such that $\lambda(G) \leq \lambda$ for all $G \in \mathcal{G}$. It was shown by Tanner [Tan84] and Alon and Milman [AM85] that this implies (and is in fact equivalent to [Alo86a]) the standard notion of **vertex expansion**: there is a constant $\varepsilon > 0$ such that for every $G \in \mathcal{G}$ and for any set S of at most half the vertices in G, at least $(1+\varepsilon) \cdot |S|$ vertices of G are connected to some vertex in S.

As mentioned in the introduction, we refer to a D-regular undirected graph G on N vertices such that $\lambda(G) \leq \lambda$ as an (N, D, λ)-**graph**. Clearly, achieving expansion is easier as the degree gets larger. The main goal in constructing expanders is to minimize the degree, and, more generally, obtain the best degree-expansion tradeoff. Using the Probabilistic Method, Pinsker [Pin73] showed that most 3-regular graphs are expanders (in the sense of vertex expansion), and this result was extended to eigenvalue bounds in [Alo86a, BS87, FKS89, Fri91]. The best known bound on the eigenvalues of random graphs is due to Friedman [Fri91], who showed that most D-regular graphs have second largest eigenvalue at most $2/\sqrt{D} + O((\log D)/D)$ (for even D). In fact, the bound of $\approx 2/\sqrt{D}$ is the best possible for an infinite family of graphs, as shown by Alon and Boppana (cf., [Alo86a, LPS88, Nil91]). Infinite families of graphs whose second largest eigenvalues are bounded by $O(1/\sqrt{D})$ are referred to as **Ramanujan graphs**.[9]

While these probabilistic arguments provide strong existential results, applications of expanders in computer science often require *explicit* families of constant-degree expanders. The first such construction was given by Margulis [Mar73], with improvements and simplifications by Gabber and Galil [GG81], Jimbo and

[9] In order for this big-Oh notation to make sense, we must consider families of graphs containing not just infinitely many graphs of a fixed degree D, but families in which both the number of vertices and the degree can be arbitrarily large.

Maruoka [JM87], Alon and Milman [AM85], and Alon, Galil, and Milman [AGM87]. Explicit families of Ramanujan graphs were first constructed by Lubotzky, Phillips, and Sarnak [LPS88] and Margulis [Mar88], with more recent constructions given by Morgenstern [Mor94].The best eigenvalues we know how to achieve using our approach are $O(1/D^{1/3})$.

2.3 Squaring and Tensoring

In addition to the new zig-zag product, our expander construction makes use of two standard operations on graphs — squaring and tensoring. Here we describe these operations in terms of rotation maps and state their effects on the eigenvalues.

Let G be a D-regular multigraph on $[N]$ given by rotation map Rot_G. The **t'th power** of G is the D^t-regular graph G^t whose rotation map is given by $\mathrm{Rot}_{G^t}(v_0, (k_1, k_2, \ldots, k_t)) = (v_t, (\ell_t, \ell_{t-1}, \ldots, \ell_1))$, where these values are computed via the rule $(v_i, \ell_i) = \mathrm{Rot}_G(v_{i-1}, k_i)$.

Proposition 2.3 *If G is an (N, D, λ)-graph, then G^t is an (N, D^t, λ^t)-graph. Moreover, Rot_{G^t} is computable in time $\mathrm{poly}(\log N, \log D, t)$ with t oracle queries to Rot_G.*

Proof: The normalized adjacency matrix of G^t is the t'th power of the normalized adjacency matrix of G, so all the eigenvalues also get raised to the t'th power. ■

Let G_1 be a D_1-regular multigraph on $[N_1]$ and let G_2 be a D_2-regular multigraph on $[N_2]$. Define the **tensor product** $G_1 \otimes G_2$ to be the $D_1 \cdot D_2$-regular multigraph on $[N_1] \times [N_2]$ given by $\mathrm{Rot}_{G_1 \otimes G_2}((v, w), (i, j)) = ((v', w'), (i', j'))$, where $(v', i') = \mathrm{Rot}_G(v, i)$ and $(w', j') = \mathrm{Rot}_G(w, j)$. In order to analyze this construction (and our new graph product), we need some concepts from linear algebra. For vectors $\alpha \in \mathbb{R}^{N_1}$ and $\beta \in \mathbb{R}^{N_2}$, their **tensor product** is the vector $\alpha \otimes \beta \in \mathbb{R}^{N_1 \cdot N_2}$ whose (i, j)'th entry is $\alpha_i \cdot \beta_j$. If A is an $N_1 \times N_1$ matrix and B is an $N_2 \times N_2$ matrix, there is a unique $N_1 N_2 \times N_1 N_2$ matrix $A \otimes B$ (again called the **tensor product**) such that $(A \otimes B)(\alpha \otimes \beta) = (A\alpha) \otimes (B\beta)$ for all α, β.

Proposition 2.4 *If G_1 is an (N_1, D_1, λ_1)-graph and G_2 is an (N_2, D_2, λ_2)-graph, then $G_1 \otimes G_2$ is an $(N_1 \cdot N_2, D_1 \cdot D_2, \max(\lambda_1, \lambda_2))$-graph. Moreover, $\mathrm{Rot}_{G_1 \otimes G_2}$ is computable in time $\mathrm{poly}(\log N_1 N_2, \log D_1 D_2)$ with one oracle query to Rot_{G_1} and one oracle query to Rot_{G_2}.*

Proof: The normalized adjacency matrix of $G_1 \otimes G_2$ is the tensor product of the normalized adjacency matrices of G_1 and G_2. Hence its eigenvalues are the pairwise products of eigenvalues of G_1 and G_2. The largest eigenvalue is $1 \cdot 1$, and the second largest eigenvalue is either $1 \cdot \lambda_2$ or $\lambda_1 \cdot 1$. ■

3 Expander Construction

In the introduction, we described how to obtain a family of expanders by iterating two operations on graphs — squaring and the new "zig-zag" product. That description used a simplifying assumption about the edge labelings. In terms of rotation maps, the assumption was that $\mathrm{Rot}(v, i) = (w, j) \Rightarrow i = j$. In this section, we describe the construction in terms of arbitrary rotation maps and prove its properties. The expander construction given here will also use tensoring to improve the efficiency to polylogarithmic in the number of vertices.

3.1 The Zig-Zag Graph Product

We begin by describing the new graph product in terms of rotation maps. Let G_1 be a D_1-regular multigraph on $[N_1]$ and G_2 a D_2-regular multigraph on $[D_1]$. Their **zig-zag product** is a D_2^2-regular multigraph $G_1 \textcircled{z} G_2$ on $[N_1] \times [D_1]$. We view every vertex v of G_1 being blown up to a "cloud" of $D_{\ 1}$ vertices $(v,1),\ldots,(v,D_1)$, one for each edge of G_1 leaving v. Thus for every edge $e=(v,w)$ of G_1, there are two associated vertices of $G_1 \textcircled{z} G_2$ — (v,k) and (w,ℓ), where e is the k'th edge leaving v and the ℓ'th edge leaving w. Note that these pairs satisfy the relation $(w,\ell) = \mathrm{Rot}_{G_1}(v,k)$. Since G_2 is a graph on $[D_1]$, we can also imagine connecting the vertices of each such cloud using the edges of G_2. Now, the edges of $G_1 \textcircled{z} G_2$ are defined (informally) as follows: we connect two vertices (v,k) and (w,ℓ) if it is possible to get from (v,k) to (w,ℓ) by a sequence of moves of the following form:

1. Move to a neighboring vertex (v,k') within the initial cloud (using an edge of G_2).

2. Jump across clouds (using edge k' of G_1) to get to (w,ℓ').

3. Move to a neighboring vertex (w,ℓ) within the new cloud (using an edge of G_2).

To make this precise, we describe how to compute the $\mathrm{Rot}_{G_1 \textcircled{z} G_2}$ given Rot_{G_1} and Rot_{G_2}.

Definition 3.1 *If G_1 is a D_1-regular graph on $[N_1]$ with rotation map Rot_{G_1} and G_2 is a D_2-regular graph on $[D_1]$ with rotation map Rot_{G_2}, then their* **zig-zag product** *$G_1 \textcircled{z} G_2$ is defined to be the D_2^2-regular graph on $[N_1] \times [D_1]$ whose rotation map $\mathrm{Rot}_{G_1 \textcircled{z} G_2}$ is as follows:*

$\mathrm{Rot}_{G_1 \textcircled{z} G_2}((v,k),(i,j))$**:**

1. *Let $(k',i') = \mathrm{Rot}_{G_2}(k,i)$.*

2. *Let $(w,\ell') = \mathrm{Rot}_{G_1}(v,k')$.*

3. *Let $(\ell,j') = \mathrm{Rot}_{G_2}(\ell',j)$.*

4. *Output $((w,\ell),(j',i'))$.*

The important feature of this graph product is that $G_1 \textcircled{z} G_2$ is a good expander if both G_1 and G_2 are, as shown by the following theorem.

Theorem 3.2 *If G_1 is an (N_1,D_1,λ_1)-graph and G_2 is a (D_1,D_2,λ_2)-graph, then $G_1 \textcircled{z} G_2$ is a $(N_1 \cdot D_1, D_2^2, f(\lambda_1,\lambda_2))$-graph, where $f(\lambda_1,\lambda_2) \le \lambda_1 + \lambda_2 + \lambda_2^2$ and $f(\lambda_1,\lambda_2) < 1$ when $\lambda_1,\lambda_2 < 1$. Moreover, $\mathrm{Rot}_{G_1 \textcircled{z} G_2}$ can be computed in time $\mathrm{poly}(\log N, \log D_1, \log D_2)$ with one oracle query to Rot_{G_1} and two oracle queries to Rot_{G_2}.*

Stronger bounds on the function $f(\lambda_1,\lambda_2)$ are given in Section 4.2. Before proving Theorem 3.2, we show how it can be used to construct an infinite family of constant-degree expanders starting from a constant-size expander.

3.2 The Recursion

The construction is like the construction in the introduction, except that we use tensoring to reduce the depth of the recursion and thereby make the construction run in polylogarithmic time (in the size of the graph).

Let H be a (D^8, D, λ)-graph for some D and λ. (We describe two ways of obtaining such H in Section 3.4.) For every $t \geq 0$, we will define a (D^{8t}, D^2, λ_t)-graph G_t. G_1 is H^2 and G_2 is $H \otimes H$. For $t > 1$, G_t is recursively defined by

$$G_t = \left(G_{\lceil \frac{t-1}{2} \rceil} \otimes G_{\lfloor \frac{t-1}{2} \rfloor} \right)^2 \textcircled{z} H.$$

Theorem 3.3 *For every $t \geq 0$, G_t is an (D^{8t}, D^2, λ_t)-graph with $\lambda_t = \lambda + O(\lambda^2)$. Moreover,* Rot_{G_t} *can be computed in time* $\mathrm{poly}(t, \log D)$ *with* $\mathrm{poly}(t)$ *oracle queries to* Rot_H.

Proof: A straightforward induction establishes that the number of vertices in G_t is D^{8t} and that its degree is D^2. To analyze the eigenvalues, define $\mu_t = \max\{\lambda_1, \ldots, \lambda_t\}$. Then we have $\mu_t \leq \max\{\mu_{t-1}, \mu_{t-1}^2 + \lambda + \lambda^2\}$ for all $t \geq 2$. Solving this recurrence gives $\mu_t \leq \lambda + O(\lambda^2)$ for all t. For the efficiency, note that the depth of the recursion is at most $\log_2 t$ and evaluating the rotation maps for G_t requires 4 evaluations of rotation maps for smaller graphs, so the total number of recursive calls is at most $4^{\log_2 t} = t^2$. ■

In order for Theorem 3.3 to guarantee that graphs $\{G_t\}$ are expanders, the second largest eigenvalue λ of the building block H must be sufficiently small (say, $\lambda \leq 1/5$). This forces the degree of H and hence the degree of the expander family to be rather large, though still constant. However, by zig-zagging the family $\{G_t\}$ with a cycle, we can obtain a family of degree 4 expanders. More generally, we can use this method convert any family of odd-degree expanders into a family of degree 4 expanders:

Corollary 3.4 *For every $\lambda < 1$ and every odd D, there exists a $\lambda' < 1$ such that if G is an (N, D, λ)-graph and C is the cycle on D vertices, then $G \textcircled{z} C$ is a $(ND, 4, \lambda')$-graph.*

As mentioned in Section 1.5, we can obtain degree 3 expanders using a simpler, but related graph product.

3.3 Analysis of the Zig-Zag Product

Now we prove Theorem 3.2. Recall the intuition behind the zig-zag product. We aim to show that for any (non-uniform) initial probability distribution π on the vertices of $G_1 \textcircled{z} G_2$, taking a random step on $G_1 \textcircled{z} G_2$ results in a distribution that is more uniform. We argued this intuitively in the introduction, by considering two extreme cases, based on the conditional distributions induced by π on the N_1 "clouds" of D_1 vertices each: one in which these conditional distributions are far from uniform, and the second in which they are uniform. The actual linear algebra proof below will restrict itself to these two cases by decomposing any other vector into a linear combination of the two. Also, the argument in the introduction was not symmetric in the first and second steps on the small graph. Using the variational definition of the second largest eigenvalue, we get a cleaner analysis than by following that intuition directly.

Let M be the normalized adjacency matrix of $G_1 \textcircled{z} G_2$. According to Definition 2.2, we must show that, for every vector $\alpha \in \mathbb{R}^{N_1 \cdot D_1}$ such that $\alpha \perp 1_{N_1 D_1}$, $|\langle M\alpha, \alpha \rangle|$ is smaller than $\langle \alpha, \alpha \rangle$ by a factor $f(\lambda_1, \lambda_2)$.[10] For every $v \in [N_1]$, define $\alpha_v \in \mathbb{R}^{D_1}$ by $(\alpha_v)_k = \alpha_{vk}$. Also define a (linear) map $C : \mathbb{R}^{N_1 \cdot D_1} \to \mathbb{R}^{N_1}$ by $(C\alpha)_v = \sum_{k=1}^{D_1} \alpha_{vk}$. Thus, for a probability distribution π on the vertices of $G_1 \textcircled{z} G_2$, π_v is a multiple of the conditional distribution on "cloud v" and $C\pi$ gives the marginal distribution on set of clouds. By definition, $\alpha = \sum_v e_v \otimes \alpha_v$, where e_v denotes the v'th standard basis vector in $\mathbb{R}^{N_1}$. By basic linear algebra, every α_v

[10] For intuition, α should be thought of as the nonuniform component of the probability distribution π referred to above, i.e. $\pi = u_{N_1 D_1} + \alpha$, where $u_{N_1 D_1} = 1_{N_1 D_1}/N_1 D_1$ is the uniform distribution on $[N_1 D_1]$. Thus, we are showing that π becomes more uniform after a random step on $G_1 \textcircled{z} G_2$.

can be decomposed (uniquely) into $\alpha_v = \alpha_v^{||} + \alpha_v^{\perp}$ where $\alpha_v^{||}$ is parallel to 1_{D_1} (i.e., all of its entries are the same) and $\alpha_v^{\perp}$ is orthogonal to 1_{D_1} (i.e., the sum of its entries are 0). Thus, we obtain a decomposition of α:

$$\begin{aligned}\alpha &= \sum_v e_v \otimes \alpha_v \\ &= \sum_v e_v \otimes \alpha_v^{||} + \sum_v e_v \otimes \alpha_v^{\perp} \\ &\stackrel{\text{def}}{=} \alpha^{||} + \alpha^{\perp}\end{aligned}$$

This decomposition corresponds to to the two cases in our intuition: $\alpha^{||}$ corresponds to a probability distribution on the vertices of $G_1 \textcircled{z} G_2$ such that the conditional distributions on the clouds are all uniform. $\alpha^{\perp}$ corresponds to a distribution such that the conditional distributions on the clouds are all far from uniform. Another way of matching $\alpha^{||}$ with the intuition is to note that $\alpha^{||} = C\alpha \otimes 1_{D_1}/D_1$. Since α and $\alpha^{\perp}$ are both orthogonal to $1_{N_1 D_1}$, so is $\alpha^{||}$ and hence also $C\alpha$ is orthogonal to 1_{N_1}.

To analyze how M acts on these two vectors, we relate M to the normalized adjacency matrices of G_1 and G_2, which we denote by A and B, respectively. First, we decompose M into the product of three matrices, corresponding to the three steps in the definition of $G_1 \textcircled{z} G_2$'s edges. Let $\tilde{B}$ be the (normalized) adjacency matrix of the graph on $[N_1] \times [D_1]$ where we connect the vertices within each cloud according to the edges of G_2. $\tilde{B}$ is related to B by the relation $\tilde{B} = I_{N_1} \otimes B$, where I_{N_1} is the $N_1 \times N_1$ identity matrix. Let $\tilde{A}$ be the permutation matrix corresponding to Rot_{G_1}. The relationship between $\tilde{A}$ and A is somewhat subtle, so we postpone describing it until later. By the definition of $G_1 \textcircled{z} G_2$, we have $M = \tilde{B}\tilde{A}\tilde{B}$. Note that both $\tilde{B}$ and $\tilde{A}$ are symmetric matrices, due to the undirectedness of G_1 and G_2.

Recall that we want to bound $|\langle M\alpha, \alpha\rangle|/\langle \alpha, \alpha\rangle$. By the symmetry of $\tilde{B}$, we have

$$\langle M\alpha, \alpha\rangle = \langle \tilde{B}\tilde{A}\tilde{B}\alpha, \alpha\rangle = \langle \tilde{A}\tilde{B}\alpha, \tilde{B}\alpha\rangle. \tag{1}$$

Now note that $\tilde{B}\alpha^{||} = \alpha^{||}$, because $\alpha^{||} = C\alpha \otimes 1_{D_1}/D_1$, $\tilde{B} = I_{N_1} \otimes B$, and $B1_{D_1} = 1_{D_1}$. This corresponds to the fact that if the conditional distribution within each cloud is uniform, then taking a random G_2-step does nothing. Hence, $\tilde{B}\alpha = \tilde{B}(\alpha^{||} + \alpha^{\perp}) = \alpha^{||} + \tilde{B}\alpha^{\perp}$. Substituting this into (1), we have

$$\langle M\alpha, \alpha\rangle = \langle \tilde{A}(\alpha^{||} + \tilde{B}\alpha^{\perp}), \alpha^{||} + \tilde{B}\alpha^{\perp}\rangle. \tag{2}$$

Expanding and using the fact that $\tilde{A}$ is length-preserving (because it is a permutation matrix), we have

$$|\langle M\alpha, \alpha\rangle| \quad \leq \quad |\langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle| + 2\|\alpha^{||}\| \cdot \|\tilde{B}\alpha^{\perp}\| + \|\tilde{B}\alpha^{\perp}\|^2. \tag{3}$$

Now we apply the expansion properties of G_1 and G_2 to bound each of these terms. First, we bound $\|\tilde{B}\alpha^{\perp}\|$, which corresponds to the intuition that when the conditional distributions within the clouds are far from uniform, they become more uniform when we take a random G_2-step.

Claim 3.5 $\|\tilde{B}\alpha^{\perp}\| \leq \lambda_2 \cdot \|\alpha^{\perp}\|$.

Proof of claim:

$$\begin{aligned}\tilde{B}\alpha^{\perp} &= \tilde{B}\left(\sum_v e_v \otimes \alpha_v^{\perp}\right) \\ &= \sum_v e_v \otimes B\alpha_v^{\perp}.\end{aligned}$$

By the expansion of G_2, $\|B\alpha_v^{\perp}\| \leq \lambda_2 \cdot \|\alpha_v^{\perp}\|$ for all v. Hence, $\|\tilde{B}\alpha^{\perp}\| \leq \lambda_2 \cdot \|\alpha^{\perp}\|$. □

Next, we bound $|\langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle|$, which corresponds to the intuition that when the conditional distribution within each cloud is uniform, the jump between the clouds makes the marginal distribution on clouds themselves more uniform.

Claim 3.6 $|\langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle| \leq \lambda_1 \cdot \langle \alpha^{||}, \alpha^{||}\rangle$.

> **Proof of claim:** To prove this, we must first relate $\tilde{A}$ to A. Recall that, when k is uniformly distributed, $\mathrm{Rot}_{G_1}(v, k)$ gives a pair (w, ℓ) where w is a uniformly selected neighbor of v. Similarly, if $e_v \in \mathbb{R}^{N_1}$ is the v'th standard basis vector, then Ae_v gives the uniform distribution over the neighbors of v. This similarity is captured by the formula $C\tilde{A}(e_v \otimes 1_{D_1}/D_1) = Ae_v$ for all v. (Tensoring e_v with $1_{D_1}/D_1$ corresponds to taking the uniform distribution over k and applying C corresponds to discarding ℓ and looking just at w.) Because the e_v's form a basis, this formula extends to all vectors $\beta \in \mathbb{R}^{N_1}$: $C\tilde{A}(\beta \otimes 1_{D_1}/D_1) = A\beta$. Applying this formula to $\alpha^{||} = C\alpha \otimes 1_{D_1}/D_1$, we have $C\tilde{A}(\alpha^{||}) = AC\alpha$. Thus,
>
> $$\begin{aligned} \langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle &= \langle \tilde{A}\alpha^{||}, C\alpha \otimes 1_{D_1}\rangle / D_1 \\ &= \langle C\tilde{A}\alpha^{||}, C\alpha\rangle / D_1 \\ &= \langle AC\alpha, C\alpha\rangle / D_1. \end{aligned}$$
>
> Recalling that $C\alpha$ is orthogonal to 1_{N_1}, we may apply the expansion of G_1 to obtain:
>
> $$\begin{aligned} |\langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle| &\leq \lambda_1 \cdot \langle C\alpha, C\alpha\rangle / D_1 \\ &= \lambda_1 \cdot \langle C\alpha \otimes 1_{D_1}, C\alpha \otimes 1_{D_1}\rangle / D_1^2 \\ &= \lambda_1 \cdot \langle \alpha^{||}, \alpha^{||}\rangle, \end{aligned}$$
>
> □

Substituting the bounds of Claim 3.5 and 3.6 into (3), we have:

$$|\langle M\alpha, \alpha\rangle| \leq \lambda_1 \cdot \|\alpha^{||}\|^2 + 2\lambda_2 \cdot \|\alpha^{||}\| \cdot \|\alpha^{\perp}\| + \lambda_2^2 \cdot \|\alpha^{\perp}\|^2 \tag{4}$$

If we let $p = \|\alpha^{||}\|/\|\alpha\|$ and $q = \|\alpha^{\perp}\|/\|\alpha\|$, then $p^2 + q^2 = 1$, and the above expression can be rewritten as:

$$\frac{|\langle M\alpha, \alpha\rangle|}{\langle \alpha, \alpha\rangle} \leq \lambda_1 \cdot p^2 + 2\lambda_2 \cdot pq + \lambda_2^2 \cdot q^2 \leq \lambda_1 + \lambda_2 + \lambda_2^2.$$

This shows that we can take $f(\lambda_1, \lambda_2) \leq \lambda_1 + \lambda_2 + \lambda_2^2$. It remains to show that we can set $f(\lambda_1, \lambda_2) < 1$ as long as $\lambda_1, \lambda_2 < 1$. We consider two cases, depending on the length of $\|\alpha^{\perp}\|$. First, suppose that $\|\alpha^{\perp}\| \leq \frac{1-\lambda_1}{3\lambda_2} \cdot \|\alpha\|$. Then, from (4), we have

$$|\langle M\alpha, \alpha\rangle| \leq \lambda_1 \cdot \|\alpha\|^2 + 2\lambda_2 \cdot \left(\frac{1-\lambda_1}{3\lambda_2}\right) \|\alpha\|^2 + \lambda_2^2 \cdot \left(\frac{1-\lambda_1}{3\lambda_2}\right)^2 \|\alpha\|^2 < \left(1 - \frac{1-\lambda_1}{9}\right) \cdot \|\alpha\|^2.$$

Now suppose that $\|\alpha^{\perp}\| \geq \frac{1-\lambda_1}{3} \cdot \|\alpha\|$. Notice that $\tilde{B}\alpha^{\perp}$ is orthogonal to $\alpha^{||}$: $\langle \tilde{B}\alpha^{\perp}, \alpha^{||}\rangle = \langle \alpha^{\perp}, \tilde{B}\alpha^{||}\rangle = \langle \alpha^{\perp}, \alpha^{||}\rangle = 0$. Using this, we can bound (2) as follows:

$$\begin{aligned} |\langle M\alpha, \alpha\rangle| &= |\langle \tilde{A}(\alpha^{||} + \tilde{B}\alpha^{\perp}), \alpha^{||} + \tilde{B}\alpha^{\perp}\rangle| \leq \|\alpha^{||} + \tilde{B}\alpha^{\perp}\|^2 = \|\alpha^{||}\|^2 + \|\tilde{B}\alpha^{\perp}\|^2 \\ &\leq \|\alpha\|^2 - \|\alpha^{\perp}\|^2 + \lambda_2^2 \cdot \|\alpha^{\perp}\|^2 \leq \|\alpha\|^2 - (1 - \lambda_2^2) \cdot \left(\frac{1-\lambda_1}{3\lambda_2^2}\right)^2 \cdot \|\alpha\|^2. \end{aligned}$$

Thus, we can take

$$f(\lambda_1, \lambda_2) \leq 1 - \frac{1-\lambda_1}{9} \cdot \min\left\{1, \frac{1-\lambda_2^2}{\lambda_2^2}\right\} < 1.$$

3.4 The Base Graph

In this section, we describe two simple graphs that can be used as the building block H for our expander construction. The first is simpler and more intuitive, but the second yields a construction with better parameters.

The Affine Plane

The first construction is based on the "projective plane" construction of Alon [Alo86b], but we instead use the affine plane in order to make N exactly D^2 and then use the zig-zag product to obtain a graph with $N = D^8$. For a prime power $q = p^t$, let $\mathbb{F}_q$ be the finite field of size q; an explicit representation of such a field can be found deterministically in time $\mathrm{poly}(p, t)$ [Sho90]. We define a graph AP_q with vertex set $\mathbb{F}_q^2$, and edge set $\{((a, b), (c, d)) : ac = b + d\}$. That is, we connect the vertex (a, b) to all points on the line $L_{a,b} = \{(x, y) : y = ax - b\}$. (Note that we have chosen the sign of b to make the graph undirected.)

Lemma 3.7 AP_q *is an* $(q^2, q, 1/\sqrt{q})$*-graph. Moreover, a rotation map for* AP_q *can be computed in time* $\mathrm{poly}(\log q)$ *given a representation of the field* $\mathbb{F}_q$.

Proof: The expansion of AP_q will follow from the fact the square of AP_q is almost the complete graph, which in turn is based on the fact that almost all pairs of lines in the plane $\mathbb{F}_q^2$ intersect. Let M be the $q^2 \times q^2$ normalized adjacency matrix of AP_q; we will now calculate the entries of M^2. The entry of M^2 in row (a, b) and column (a', b') is exactly the number of common neighbors of (a, b) and (a', b') in AP_q divided by q^2, *i.e.*, $|L_{a,b} \cap L_{a',b'}|/q^2$. If $a \neq a'$, then $L_{a,b}$ and $L_{a',b'}$ intersect in exactly one point. If $a = a'$ and $b \neq b'$, then their intersection is empty, and if $a = a'$ and $b = b'$, then their intersection is of size q. Thus, if we let I_q denote the $q \times q$ identity matrix and J_q the $q \times q$ all-one's matrix, we have

$$M^2 = \frac{1}{q^2}\begin{pmatrix} qI_q & J_q & \cdots & J_q \\ J_q & qI_q & & J_q \\ \vdots & & \ddots & J_q \\ J_q & J_q & \cdots & qI_q \end{pmatrix} = \frac{I_q \otimes qI_q + (J_q - I_q) \otimes J_q}{q^2}.$$

Now we can calculate the eigenvalues explicitly. J_q has eigenvalues q (multiplicity 1) and 0 (multiplicity $q - 1$). So $(J_q - I_q) \otimes J_q$ has eigenvalues $(q - 1) \cdot q$, $-1 \cdot q$, and 0. Adding $I_q \otimes qI_q$ increases all these eigenvalues by q, and then we divide by q^2. Hence the eigenvalues of M^2 are 1 (multiplicity 1), 0 (multiplicity $q - 1$), and $1/q$ (multiplicity $(q - 1) \cdot q$). Therefore, the second largest eigenvalue of M has absolute value $1/\sqrt{q}$.

A rotation map for AP_q is given by

$$\mathrm{Rot}_q((a, b), t) = \begin{cases} ((t/a, t - b), t) & \text{if } a \neq 0 \text{ and } t \neq 0, \\ ((t, -b), a) & \text{if } a = 0 \text{ or } t = 0, \end{cases}$$

where $a, b, t \in \mathbb{F}_q$. ■

Now, define the following graphs inductively:

$$\begin{aligned} \mathrm{AP}_q^1 &= \mathrm{AP}_q \otimes \mathrm{AP}_q \\ \mathrm{AP}_q^{i+1} &= \mathrm{AP}_q^i \textcircled{z} \mathrm{AP}_q \end{aligned}$$

From Proposition 2.4 and Theorem 3.2, we immediately deduce:

Proposition 3.8 AP_q^i *is a* $(q^{2(i+1)}, q^2, O(i/\sqrt{q}))$*-graph.*[11] *Moreover, a rotation map for* AP_q^i *can be computed in time* $\mathrm{poly}(i, \log q)$ *given a representation of* $\mathbb{F}_q$*.*

Taking $i = 7$ and a sufficiently large q gives a graph suitable for the expander construction in Section 3.2.

Low-Degree Polynomials

The graphs we describe here are derived from constructions of Alon and Roichman [AR94], which are Cayley graphs derived from the generator matrix of an error-correcting code. In order to give a self-contained presentation, we specialize the construction to a Reed-Solomon code concatenated with a Hadamard code (as used in, e.g. [AGHP92]).

For a prime power q and $d \in \mathbb{N}$, we define a graph $\mathrm{LD}_{q,d}$ on vertex set $\mathbb{F}_q^{d+1}$ with degree q^2. For a vertex $a \in \mathbb{F}_q^{d+1}$ and $x, y \in \mathbb{F}_q$, the the (x, y)'th neighbor of a is $a + (y, yx, yx^2, \ldots, yx^d)$.

Proposition 3.9 $\mathrm{LD}_{q,d}$ *is a* $(q^{d+1}, q^2, d/q)$*-graph. Moreover, a rotation map for* $\mathrm{LD}_{q,d}$ *can be computed in time* $\mathrm{poly}(\log q, d)$ *given a representation of* $\mathbb{F}_q$*.*

As above, taking $d = 7$ and sufficiently large q gives a graph suitable for our expander construction. These graphs are better than those of Proposition 3.8 because the the eigenvalue-degree relationship is the optimal $\lambda = O(1/\sqrt{D})$ (as q grows).

Proof: To simplify notation, let $\mathbb{F} = \mathbb{F}_q$. Let M be the $q^{d+1} \times q^{d+1}$ normalized adjacency matrix of $\mathrm{LD}_{q,d}$. We view vectors in $\mathbb{C}^{q^{d+1}}$ as functions $f : \mathbb{F}^{d+1} \to \mathbb{C}$. We will now explicitly describe the eigenvectors of M. Let p be the characteristic of $\mathbb{F}$, let $\zeta = e^{2\pi i/p}$ be a primitive p'th root of unity, and let $L : \mathbb{F} \to \mathbb{F}_p$ be any full-rank $\mathbb{F}_p$-linear map. (For simplicity, one can think of the special case that $p = q$ and L is the identity map.)

For every sequence $a = (a_0, \ldots, a_d) \in \mathbb{F}^{d+1}$, define the function $\chi_a : \mathbb{F}^{d+1} \to \mathbb{C}$ by $\chi_a(b) = \zeta^{L(\sum a_i b_i)}$. Clearly, $\chi_a(b + c) = \chi_a(b)\chi_a(c)$ for any $b, c \in \mathbb{F}^{d+1}$. Moreover, it can be verified that the $\{\chi_a\}$ are orthogonal under the standard inner product $\langle f, g\rangle = \sum_b f(b)g(b)^*$, and thus form a basis for $\mathbb{C}^{q^{d+1}}$. Hence, if we show that each χ_a is an eigenvector of M, then they are all the eigenvectors of M. This can be done by direct calculation:

$$\begin{aligned}
(M\chi_a)(b) &= \frac{1}{q^2} \sum_{c \in \mathbb{F}^{d+1}} M_{bc} \cdot \chi_a(c) \\
&= \frac{1}{q^2} \sum_{x,y \in \mathbb{F}} \chi_a(b + (y, yx, \ldots, yx^d)) \\
&= \left(\frac{\sum_{x,y\in\mathbb{F}} \chi_a(y, yx, \ldots, yx^d)}{q^2} \right) \cdot \chi_a(b) \\
&\stackrel{\text{def}}{=} \lambda_a \cdot \chi_a(b).
\end{aligned}$$

Thus, χ_a is an eigenvector of M with eigenvalue λ_a and all eigenvectors of M are of this form. So we simply need to show that $|\lambda_a| \le d/q$ for all but one $a \in \mathbb{F}^{d+1}$. To do this, note that

$$\lambda_a = \frac{1}{q^2} \sum_{x,y\in\mathbb{F}} \chi_a((y, yx, \ldots, yx^d)) = \frac{1}{q^2} \sum_{x,y\in\mathbb{F}} \zeta^{L(y \cdot p_a(x))},$$

[11] The hidden constant in $O(i/\sqrt{q})$ can be reduced to 1 using the improved analysis of the zig-zag product in Theorem 4.3.

where $p_a(x)$ is the polynomial $a_0 + a_1 x + \cdots + a_d x^d$. When x is a root of p_a, then $\zeta^{L(yp_a(x))} = 1$ for all y, and hence x contributes $q/q^2 = 1/q$ to λ_a. When x is not a root of $p_a(x)$, $yp_a(x)$ takes on all values in $\mathbb{F}$ as y varies, and hence $\zeta^{L(yp_a(x))}$ varies uniformly over all p'th roots of unity. Since the sum of all p'th roots of unity is 0, these x's contribute nothing to λ_a. When $a \neq 0$, p_a has at most d roots, so $|\lambda_a| \leq d/q$. ■

4 Extensions to the Expander Construction

4.1 Better Eigenvalue-Degree Relation

Recall that the optimal second-largest eigenvalue for an infinite family of D-regular graphs is $\Theta(1/D^{1/2})$, and families of graphs meeting this bound are referred to as Ramanujan. Starting with a constant-size Ramanujan graph,[12] our basic construction of Theorem 3.3 gives an achieves a second-largest eigenvalue of $O(1/D^{1/4})$. Here, we define a variant of the zig-zag product which leads to a better dependence of the eigenvalue on the degree. Specifically, using it in a construction like that of Theorem 3.3 together a constant-size Ramanujan graph (*e.g.,* as given by Proposition 3.9), we obtain a second-largest eigenvalue of $O(1/D^{1/3})$. It is an interesting problem to construct families of graphs achieving the optimal eigenvalue $O(1/D^{1/2})$ using a similar graph product.

Definition 4.1 *Let G_1 be a D_1-regular graph on $[N_1]$ with rotation map Rot_{G_1} and let G_2 be a D_2-regular graph on $[D_1]$ with rotation map Rot_{G_2}. Suppose that for every $i \in [D_2]$, $\mathrm{Rot}_{G_2}(\cdot, i)$ is a permutation on $[D_1]$. Then the* **modified zig-zag product** *of G_1 and G_2 is defined to be the D_2^3-regular graph $G_1 \textcircled{z}' G_2$ on $[N_1] \times [D_1]$ whose rotation map $\mathrm{Rot}_{G_1 \textcircled{z}' G_2}$ is as follows:*

$\mathrm{Rot}_{G_1 \textcircled{z}' G_2}((v,k),(h,i,j))$:

1. *Let $(k', h') = \mathrm{Rot}_{G_2}(k, h)$.*
2. *Let $(k'', i') = \mathrm{Rot}_{G_2}(k', i)$.*
3. *Let $(w, \ell'') = \mathrm{Rot}_{G_1}(v, k'')$.*
4. *Find the unique $\ell' \in [D_1]$ such that $(\ell'', i'') = \mathrm{Rot}_{G_2}(\ell', i)$ for some i''. (ℓ' exists by the assumption on Rot_{G_2}.)*
5. *Let $(\ell, j') = \mathrm{Rot}_{G_2}(\ell', j)$.*
6. *Output $((w,\ell),(j', i, h'))$.*

So, in this graph product we do *two* random steps on the small graph in both the zig and the zag parts. However, to save random bits (namely decrease the degree) we use *the same* random bits for the second move of the zig part and the first move of the zag part. Thus the degree of the new graph is D_2^3. However, we will show that the bound on the eigenvalue will be as if these moves were independent.

Theorem 4.2 *If G_1 is an (N_1, D_1, λ_1)-graph and G_2 is a (D_1, D_2, λ_2)-graph, then $G_1 \textcircled{z}' G_2$ is a $(N_1 \cdot D_1, D_2^3, \lambda_1 + 2\lambda_2^2)$-graph. Moreover, $\mathrm{Rot}_{G_1 \textcircled{z}' G_2}$ can be computed in time $\mathrm{poly}(\log N, \log D_1, D_2)$ with one oracle query to Rot_{G_1} and $D_2 + 2$ oracle queries to Rot_{G_2}.*

[12] Analogous to Footnote 9, the notion of a “constant-size Ramanujan graph” is not well-defined. What we mean is that the size of the Ramanujan graph we need as a building block is a (polynomial) function of only the degree of the graph we are constructing, and not the number of vertices.

Proof: We use the same notation as in the proof of Theorem 3.2. Like there, we need to bound $|\langle M\alpha, \alpha\rangle|/\langle \alpha, \alpha\rangle$, where M is the normalized adjacency matrix of $G_1 \textcircled{z}' G_2$ and $\alpha \perp 1_{N_1 D_1}$. Let B_i be the $D_1 \times D_1$ permutation matrix corresponding to $\mathrm{Rot}_{G_2}(\cdot, i)$, and let $\tilde{B}_i = I_{N_1} \otimes B_i$. Then

$$\tilde{B} = \frac{1}{D_1} \sum_{i=1}^{D_1} \tilde{B}_i.$$

Note that the normalized adjacency matrix corresponding to Steps 2–4 in the definition of $G_1 \textcircled{z}' G_2$ is given by

$$M' = \frac{1}{D_1} \sum_i \tilde{B}_i \tilde{A} \tilde{B}_i^T,$$

where $\tilde{B}_i^T$ is the transpose (equivalently, inverse) of $\tilde{B}_i$. Thus, $M = \tilde{B} M' \tilde{B}$. The main observation is that not only does $\tilde{B}\alpha^{||} = \alpha^{||}$ (as we used in the original analysis), but also $\tilde{B}_i^T \alpha^{||} = \alpha^{||}$ for every i (because B_i is a permutation matrix). Hence,

$$M'\alpha^{||} = \frac{1}{D_1} \sum_i \tilde{B}_i \tilde{A} \tilde{B}_i^T \alpha^{||} = \frac{1}{D_1} \sum_i \tilde{B}_i \tilde{A} \alpha^{||} = \tilde{B} \tilde{A} \alpha^{||}.$$

Applying this (and the symmetry of $\tilde{B}$ and M'), we get

$$\begin{aligned} \langle M\alpha, \alpha\rangle &= \langle M\alpha^{||}, \alpha^{||}\rangle + 2\langle M\alpha^{||}, \alpha^{\perp}\rangle + \langle M\alpha^{\perp}, \alpha^{\perp}\rangle \\ &= \langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle + 2\langle \alpha^{||}, \tilde{B}^2\alpha^{\perp}\rangle + \langle M'\tilde{B}\alpha^{\perp}, \tilde{B}\alpha^{\perp}\rangle. \end{aligned}$$

Being the normalized adjacency matrix of an undirected, regular graph, M' has no eigenvalues larger than 1 and hence does not increase the length of any vector. Using this together with Claims 3.5 and 3.6, we have

$$\begin{aligned} |\langle M\alpha, \alpha\rangle| &\leq |\langle \tilde{A}\alpha^{||}, \alpha^{||}\rangle| + 2\|\alpha^{||}\| \cdot \|\tilde{B}^2\alpha^{\perp}\| + \|\tilde{B}\alpha^{\perp}\|^2 \\ &\leq \lambda_1 \cdot \|\alpha^{||}\|^2 + 2\lambda_2^2 \cdot \|\alpha^{||}\| \cdot \|\alpha^{\perp}\| + \lambda_2^2 \cdot \|\alpha^{\perp}\|^2. \end{aligned}$$

As in the the proof of Theorem 3.2, using the fact that $\|\alpha^{||}\|^2 + \|\alpha^{\perp}\|^2 = \|\alpha\|^2$ yields the desired bound. ■

4.2 Improved Analysis of the Zig-Zag Graph Product

Theorem 4.3 (Thm. 3.2, improved) *If G_1 is an (N_1, D_1, λ_1)-graph and G_2 is a (D_1, D_2, λ_2)-graph, then $G_1 \textcircled{z} G_2$ is a $(N_1 \cdot D_1, D_2^2, f(\lambda_1, \lambda_2))$-graph, where*

$$f(\lambda_1, \lambda_2) = \frac{1}{2}(1 - \lambda_2^2)\lambda_1 + \frac{1}{2}\sqrt{(1 - \lambda_2^2)^2\lambda_1^2 + 4\lambda_2^2}.$$

Although the function $f(\lambda_1, \lambda_2)$ looks ugly, it can be verified that it has the following nice properties:

1. $f(\lambda, 0) = f(0, \lambda) = \lambda$ and $f(\lambda, 1) = f(1, \lambda) = 1$ for all $\lambda \in [0, 1]$.

2. $f(\lambda_1, \lambda_2)$ is a strictly increasing function of both λ_1 and λ_2 (except when one of them is 1).

3. When $\lambda_1 < 1$ and $\lambda_2 < 1$, then $f(\lambda_1, \lambda_2) < 1$.

4. $f(\lambda_1, \lambda_2) \leq \lambda_1 + \lambda_2$ for all $\lambda_1, \lambda_2 \in [0, 1]$.

Proof: The proof proceeds along the same lines as the proof of Theorem 3.2, except that we will use a geometric argument to directly bound (2) rather than first passing to (3). That is, we we must bound (using the same notation as in that proof)

$$\frac{\langle M\alpha, \alpha\rangle}{\langle \alpha, \alpha\rangle} = \frac{\langle \tilde{A}(\alpha^{||} + \tilde{B}\alpha^{\perp}), \alpha^{||} + \tilde{B}\alpha^{\perp}\rangle}{\|\alpha^{||} + \alpha^{\perp}\|^2}.$$

The key observation is:

Claim 4.4 *$\tilde{A}$ is a reflection through a linear subspace S of $\mathbb{R}^{N_1 D_1}$. Hence, for any any vector v, $\langle Av, v\rangle = (\cos 2\theta) \cdot \|v\|^2$, where θ is the angle between v and S.*

> **Proof of claim:** By the symmetry of $\tilde{A}$, we can decompose $\mathbb{R}^{N_1 D_1}$ into the sum of orthogonal eigenspaces of $\tilde{A}$. Since $\tilde{A}^2 = I_{N_1 D_1}$, the only eigenvalues of $\tilde{A}$ are ± 1. Take S to be the 1-eigenspace of $\tilde{A}$. □

Thus, the expression we want to bound is

$$\frac{|\langle M\alpha, \alpha\rangle|}{\langle \alpha, \alpha\rangle} = |\cos 2\theta| \cdot \frac{\|\alpha^{||} + \tilde{B}\alpha^{\perp}\|^2}{\|\alpha^{||} + \alpha^{\perp}\|^2} = |\cos 2\theta| \cdot \frac{\cos^2\phi}{\cos^2\phi'},$$

where θ is the angle between $\alpha^{||} + \tilde{B}\alpha^{\perp}$ and S, $\phi \in [0, \pi/2]$ is the angle between $\alpha^{||}$ and $\alpha^{||} + \alpha^{\perp}$, and $\phi' \in [0, \pi/2]$ is the angle between $\alpha^{||}$ and $\alpha^{||} + \tilde{B}\alpha^{\perp}$. If we also let ψ be the angle between $\alpha^{||}$ and S, then we clearly have $\theta \in [\psi - \phi', \psi + \phi']$.

Now we translate Claims 3.5 and 3.6 into this geometric language. Claim 3.5 constrains the relationship between ϕ' and ϕ by

$$\frac{\tan\phi'}{\tan\phi} = \frac{\|\tilde{B}\alpha^{\perp}\|}{\|\alpha^{\perp}\|} \leq \lambda_2.$$

Claim 3.6 says $|\cos 2\psi| \leq \lambda_1$. For notational convenience, we will denote the exact values of $(\tan\phi')/(\tan\phi)$ and $|\cos 2\psi|$ by μ_2 and μ_1, respectively. We will work with these values until the end of the proof, at which point we will upper bound them by λ_2 and λ_1.

To summarize, we want to maximize

$$|\cos 2\theta| \cdot \frac{\cos^2\phi}{\cos^2\phi'}. \tag{5}$$

over the variables θ, ϕ, ϕ', and ψ, subject to the following constraints:

1. $\phi, \phi', \psi \in [0, \pi/2]$.
2. $\theta \in [\psi - \phi', \psi + \phi']$.[13]
3. $\tan\phi' / \tan\phi = \mu_2$.
4. $|\cos 2\psi| = \mu_1$.

There are two cases, depending on whether $|\cos 2x|$ ever achieves the value 1 in the interval $[\psi - \phi', \psi + \phi']$.

[13] We do not require $\theta \in [0, \pi/2]$ so that we do not have to worry about "wraparound" in the interval $[\psi - \phi\ ', \psi + \phi']$. Adding a multiple of $\pi/2$ to θ does not change the value of (5).

Case I: $\phi' \le \min\{\psi, \pi/2 - \psi\}$**.** Then

$$\begin{aligned} |\cos 2\theta| &= \max\{|\cos 2(\psi + \phi')|, |\cos 2(\psi - \phi')|\} \\ &= |\cos 2\psi \cdot \cos 2\phi'| + |\sin 2\psi \cdot \sin 2\phi'|. \end{aligned}$$

After some trigonometric manipulations, we have

$$|\cos 2\theta| \cdot \frac{\cos^2 \phi}{\cos^2 \phi'} = \frac{1}{2}\left|(1-\mu_2^2)\cos 2\psi + (1+\mu_2^2)\cos 2\psi \cos 2\phi\right| + \frac{1}{2}|2\mu_2 \sin 2\psi \sin 2\phi|$$

The choice of ϕ which maximizes this is to have $(\cos 2\phi, \sin 2\phi)$ be a unit vector in the direction of $(\pm(1 + \mu_2^2)\cos 2\psi, 2\mu_2 \sin 2\psi)$, so

$$\begin{aligned} |\cos 2\theta| \cdot \frac{\cos^2 \phi}{\cos^2 \phi'} &\le \frac{1}{2}(1-\mu_2^2)|\cos 2\psi| + \frac{1}{2}\sqrt{(1+\mu_2^2)^2 \cos^2 2\psi + 4\mu_2^2 \sin^2 2\psi} \\ &= \frac{1}{2}(1-\mu_2^2)\mu_1 + \frac{1}{2}\sqrt{(1+\mu_2^2)^2\mu_1^2 + 4\mu_2^2(1-\mu_1^2)}. \end{aligned}$$

Case II: $\phi' > \min\{\psi, \pi/2 - \psi\}$**.** In this case, we cannot obtain any nontrivial bound on $|\cos 2\theta|$, so, after some trigonometric manipulations, the problem is reduced to bounding:

$$|\cos 2\theta| \cdot \frac{\cos^2 \phi}{\cos^2 \phi'} \le \frac{\cos^2 \phi}{\cos^2 \phi'} = \mu_2^2 + (1-\mu_2^2)\cos^2 \phi. \tag{6}$$

The condition $\phi' > \min\{\psi, \pi/2 - \psi\}$ implies that $\cos 2\phi' < |\cos 2\psi| = \mu_1$. After some trigonometric manipulations, we have

$$\cos 2\phi' = \frac{(1+\mu_2^2)\cos^2 \phi - \mu_2^2}{(1-\mu_2^2)\cos^2 \phi + \mu_2^2},$$

and the condition $\cos 2\phi' < \mu_1$ is equivalent to

$$\cos^2 \phi < \frac{\mu_2^2(1+\mu_1)}{(1-\mu_1) + \mu_2^2(1+\mu_1)}.$$

Substituting this into (6) and simplifying, we conclude that

$$|\cos 2\theta| \cdot \frac{\cos^2 \phi}{\cos^2 \phi'} < \frac{2\mu_2^2}{1 - \mu_1 + \mu_2^2(1+\mu_1)}.$$

It can be verified that the bound obtained in Case I is an increasing function of μ_1 and μ_2 and is always greater than or equal to the bound in Case II. Therefore, replacing μ_1 and μ_2 by λ_1 and λ_2 in the Case I bound proves the theorem.

■

5 Extractor Preliminaries

5.1 Definition of Extractors

Extractors are procedures for obtaining "almost" uniformly distributed bits from an arbitrary source that contains some k bits of "hidden randomness". The definition of extractors employs a very general measure of the randomness in such a "weak" random source: Let X and Y be random variables over a set S. The **min-entropy** of X is defined to be

$$\mathrm{H}_\infty(X) \overset{\text{def}}{=} -\log(\max_{a\in S} \Pr[X = a]),$$

where here and throughout this paper, all logarithms are base 2. X is a k**-source** if $\mathrm{H}_\infty(X) \geq k$. We say that X and Y are ε**-close** if the statistical difference between X and Y is at most ε. That is, if

$$\max_{P\subseteq S} |\Pr[X \in P] - \Pr[Y \in P]| = \frac{1}{2}\sum_{a\in S} |\Pr[X = a] - \Pr[Y = a]| \leq \varepsilon$$

Note that the statistical difference is a metric and therefore it obeys the triangle inequality. For any integer n, denote by (n) the set of all n-bit strings, $\{0,1\}^n$. Denote by U_n the uniform distribution over (n).

Definition 5.1 (extractors) *A function* $\mathrm{E} : (n) \times (d) \mapsto (m)$ *is a* (k,ε)**-extractor** *if for any k-source X over (n), the distribution* $\mathrm{E}(X, U_d)$ *is ε-close to U_m.*

In other words, E "extracts" m (almost) truly random bits from a source with k bits of hidden randomness, using a random d-bit **seed** as a catalyst. The original definition of extractors in [NZ96] is stronger than the one above (from [NT99]) in that it requires the seed to be explicitly included in the output. As we discuss later (in Remark 6.8), our results also apply to such **strong extractors**.[14]

As noted in the defining paper of [NZ96], an extractor can be viewed as a bipartite graph with left-hand side (n) and right-hand side (m), where we place an edge between $x \in (n)$ and $y \in (m)$ for every $r \in (d)$ such that $\mathrm{E}(x,r) = y$. The definition of extractors can be viewed as an expansion property of this graph, and our extractor constructions in the following sections can be viewed as a generalization of our expander construction and intuition (from Section 3) to unbalanced bipartite graphs.

5.2 Previous Constructions

The goal in the design of extractors is, given n, k, and ε, to simultaneously minimize the **seed** length d and maximize the output length m. Nisan and Zuckerman [NZ96] proved bounds for both these values which were later improved by Radhakrishnan and Ta-Shma [RT97]: (1) The seed length d is at least $\log(n-k) + 2\log 1/\varepsilon - O(1)$. (2) The **entropy loss** $\Lambda = k + d - m$ is at least $2\log 1/\varepsilon - O(1)$.[15] It can also be shown that (up to an additive constant factor) these bounds can nonconstructively be matched:

Proposition 5.2 ([Sip88, RT97]) *For any $n \geq k$ and ε there exists a (k,ε)-extractor* $\mathrm{E} : (n) \times (d) \mapsto (k + d - \Lambda)$*, where $d = \log(n-k) + 2\log 1/\varepsilon + O(1)$ and $\Lambda = 2\log(1/\varepsilon) + O(1)$.*

Most applications, however, require **explicit extractors**: That is, a *family* of extractors (parameterized by n, k, m and ε) that are computable in polynomial time (in their input length, $n + d$). In recent years, a substantial body of work has provided steady progress towards the goal of constructing explicit extractors that achieve optimal seed length for all settings of parameters (see, e.g., [Zuc97, NT99, Tre99, ISW00] and the references therein). However, this goal has not yet been achieved.

In this work, we give explicit extractors for sources of very high min-entropy. For such sources, our extractors have a substantially shorter seed length than all previous constructions (that do not lose much entropy). Let X be a random variable over (n) with min-entropy k. The **entropy deficiency** Δ of X is defined to be $n - k$. Note that the optimal seed length of extractors does not depend on the input length n of the source but rather on its deficiency Δ. The only previous *explicit* extractors with such a dependence were provided by Goldreich and Wigderson:

[14] In fact, it was recently shown in [RSW00] that there exists a simple transformation of (standard) extractors to strong extractors.

[15] These bounds hold whenever $\varepsilon \leq 1/2$, $k \leq n - O(1)$, and $d \leq m - 2$.

Theorem 5.3 ([GW97]) *For any $\varepsilon > 0$ and $0 < k \leq n$ there exists an explicit $(n - \Delta, \varepsilon)$-extractor*

$$\mathrm{GW} : (n) \times (d) \mapsto (n),$$

where $d = \Delta + 2\log(1/\varepsilon) + 2$.

Our extractor composition theorem, described in the next section, gives a method for reducing this linear dependence of d on the deficiency Δ to the (optimal) logarithmic dependence while preserving the small entropy loss of $O(\log(1/\varepsilon))$. As we will discuss in more detail in the next section, a method to make d logarithmic in Δ was already suggested in [GW97], but resulted in an entropy loss greater than Δ (which negates effect of the smaller d in all the applications we describe in Section 7).

The extractors of Theorem 5.3 are essentially obtained by using the d truly random bits to do a random walk on an expander graph on (2^n) vertices. To achieve the parameters listed, one must use a "Ramanujan" graph, i.e. an expander whose second largest eigenvalue is optimal up to a constant factor. One could use our expander graphs (which do not have optimal second largest eigenvalue) instead at the price of increasing d by a constant factor (and thereby incurring an entropy loss of $O(\Delta + \log(1/\varepsilon))$). These costs are not troublesome because our composition theorem will give a way to both reduce d and the entropy loss to near-optimal values.

We note that the theorem in [GW97] entitled "Extractors for High Min-Entropy" does not claim the parameters listed above; rather, both d and the entropy loss are only claimed to be $O(\Delta + \log(1/\varepsilon))$. These worse parameters are a result of modifications made in order to make their extractors *strong* (i.e., where the seed is explicitly part of the output). For ease of exposition in this preliminary version, we do not require our extractors to be strong, though all our results extend to strong extractors.

5.3 Condensers

Let $\mathrm{E} : (n) \times (d) \mapsto (m)$ be a (k, ε)-**extractor**. As mentioned above, the entropy loss of E (i.e. the quantity $\Lambda = n + k - m$) is $\Omega(\log 1/\varepsilon)$. This means that even if a random variable X has min-entropy exactly k, the distribution of the output $\mathrm{E}(X, U_d)$ has less min-entropy than that of the input (X, U_d). In case X has a larger min-entropy (e.g., X is uniform over (n)), this entropy loss may be quite significant. In our extractor composition, it is crucial that we keep track of the entropy lost in intermediate applications of the extractors. Following Raz and Reingold [RR99], we extend each extractor E used in the composition to an extractor-condenser pair $\langle \mathrm{E}, \mathrm{C} \rangle$ such that $\mathrm{C}(X, U_d)$ produces a relatively short buffer "containing" the entropy lost by the application $\mathrm{E}(X, U_d)$. One way to formalize this idea is by requiring $\langle \mathrm{E}, \mathrm{C} \rangle$ to be a permutation:

Definition 5.4 (permutation extractor) *A pair of functions $\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (m) \times (b)$ with $b = n + d - m$ is a (k, ε) permutation extractor if* E *is a (k, ε)-extractor and* $\mathrm{E} \times \mathrm{C}$ *is 1-to-1 on $(n + d)$.*

Note that if $\langle \mathrm{E}, \mathrm{C} \rangle$ is permutation extractor, then E must be **regular**, i.e. for every $y \in (m)$, there are the same number of pairs $x \in (n)$ and $r \in (d)$ such that $\mathrm{E}(x, r) = y$. Nonconstructive regular extractors with optimal parameters (i.e. those in Proposition 5.2) can easily be extended to optimal permutation extractors. This also holds for some explicit extractors. (We say that a permutation extractor $\langle \mathrm{E}, \mathrm{C} \rangle$ is **explicit** if both $\langle \mathrm{E}, \mathrm{C} \rangle$ and its inverse are polynomial-time computable.) A natural example is implied by the Leftover Hash Lemma [HILL99]: Let $\mathcal{H}'$ be the family of pairwise independent permutations defined by $\mathcal{H}' \stackrel{\mathrm{def}}{=} \{h'_{\alpha,\beta} | \alpha \neq 0, \beta \in \mathrm{GF}(2^n)\}$, where for every $\alpha \neq 0, \beta, x \in \mathrm{GF}(2^n), h'_{\alpha,\beta}(x) = \alpha x + \beta$. Let $d = 2n$ and let $\mathcal{H}$ be an arbitrary extension of $\mathcal{H}'$ to a family of size 2^d of permutations on $\mathrm{GF}(2^n)$. Then one can define a (k, ε) permutation extractor $\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (d + m) \times (b)$, where $m = k - 2\log 1/\varepsilon - O(1)$ as follows (we identify here $\mathrm{GF}(2^n)$ with (n)):

$$\mathrm{E}(x, h) \stackrel{\mathrm{def}}{=} h, h(x)|_{[1..m]} \text{ and } \mathrm{C}(x, h) \stackrel{\mathrm{def}}{=} h(x)|_{[m+1..n]}$$

(where h is interpreted as a (uniform) permutation in H and for any string $z = z_1 z_2 \cdots z_\ell$ we denote by $z|_{[i..j]}$ the substring $z_i \cdots z_j$). Another example of an explicit extractor that can be extended into a permutation extractor is the high min-entropy extractor of Goldreich and Wigderson [GW97] (described in Theorem 5.3). Recall that those extractors simply take a random walk on an expander graph, so any rotation map (recall Definition 2.1) for the graph extends it to a permutation. In general, if we view a regular extractor E as a regular *bipartite* graph, the definition of a permutation extractor is equivalent to the natural extension of rotation maps to bipartite graphs.

However, not every extractor implies an efficient permutation extractor (e.g., if E is not regular). Fortunately, our composition theorem is still applicable even if we weaken Definition 5.4 in two ways: (1) By allowing a larger buffer size b, and (2) By requiring $\langle \mathrm{E}, \mathrm{C} \rangle$ to be 1-to-1 only on *most* inputs (rather then on all inputs). This suggests the following definition (which is still less general than that of [RR99] but is sufficiently general for our needs):

Definition 5.5 (extractor-condenser pairs) *A pair of functions* $\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (m) \times (b)$ *is a* (k, ε)-**ECP** *if* E *is a* (k, ε)*-extractor and for any* k*-source* X *over* (n)*, there exists a "bad" set of inputs* $\mathbf{B}_X \subseteq (n + d)$ *such that the following holds:*

1. $\Pr[\langle X, U_d \rangle \in \mathbf{B}_X] \leq \varepsilon$
2. $\mathrm{E} \times \mathrm{C}$ *is 1-to-1 on* $(n + d) \setminus \mathbf{B}_X$.

It turns out that given any extractor one can (easily) define an extractor-condenser pair (ECP) with comparable parameters:

Lemma 5.6 (a corollary of [RR99]) *Let* $\mathrm{E}' : (n) \times (d') \mapsto (m)$ *be an explicit* $(k, \varepsilon/3)$*-extractor. Let* $b' = n + d' - m$ *(*b' *is a lower bound on the buffer size). Then there exist two integers* $b = O(b' + \log 1/\varepsilon)$ *and* $d = O(\log(n + d') + \log 1/\varepsilon)$ *and an explicit* (k, ε)*-ECP* $\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d' + d) \mapsto (m + d) \times (b)$

The definition of $\langle \mathrm{E}, \mathrm{C} \rangle$ in the proof of Lemma 5.6 is essentially given by

$$\forall x \in (n), r \in (d'), h \in (d), \quad \mathrm{E}(z, r \circ h) = \mathrm{E}'(z, r) \circ h \text{ and } \mathrm{C}(z, r \circ h) \stackrel{\text{def}}{=} h(z, r),$$

where h is an almost 2-universal function of the right parameters and $\circ$ stands for concatenation of strings (for more details see [RR99]).

6 High Min-Entropy Extractors

In this section, we extend our new zig-zag product to extractors (which can be viewed as directed unbalanced graphs) and show how to obtain improved high min-entropy extractors using this product. As we will describe in a subsequent paper, the same product is helpful in the design of expanders whose expansion is measured *in terms of min-entropy*.

6.1 Block Sources

The starting point for our extractor composition is the construction of Goldreich and Wigderson [GW97] (not the one referred to in Theorem 5.3, but rather their method of modifying it to make the extractor strong and reduce d). Their basic observation is that any high min-entropy source is also a "block source" (as defined by Chor and Goldreich [CG88]): when the source is divided into a prefix and suffix (of arbitrary lengths), each one of these values contains a lot of "independent" randomness. (A formal statement will shortly follow.) It turns out that this simple observation along with the standard extractor composition for block sources, already has nontrivial implications. For example, for any constant error ε, there exist explicit (k, ε)-extractors with seed length $O(\log \Delta)$ that extract $k - O(\Delta)$ bits (recall that Δ is the deficiency of the source). Unfortunately, this entropy loss of $O(\Delta)$ is significant in the applications we have in mind. Our main contribution is a new extractor composition which, using condensers, can achieve an almost optimal entropy loss. In order to motivate our construction, we begin by describing the method of [GW97].

Definition 6.1 ([CG88]) *Two random variables (X_1, X_2) form a (k_1, k_2)* **block source** *if X_1 is a k_1-source, and for every possible value x_1 of X_1 the distribution of X_2, conditioned on $X_1 = x_1$, is a k_2-source.*

Lemma 6.2 (implicit in [GW97]) *Let X be any $(n - \Delta)$-source over (n) Then for any integers n_1 and n_2 such that $n = n_1 + n_2$ and any $\varepsilon > 0$, X is ε-close to some $(n_1 - \Delta, n_2 - \Delta - \log 1/\varepsilon)$ block source (X_1, X_2), where X_i is a random variable over (n_i) for $i = 1, 2$.*

The task of extraction is usually much easier for block sources. Let (X_1, X_2) be a (k_1, k_2) block source, then it is possible to extract a few (up to k_2) random bits out of X_2 and use this randomness as a seed for the extraction of many additional bits (up to k_1) out of X_1 (this appealing strategy does not work for a general source):

Lemma 6.3 ([NZ96]) *Let $\mathrm{E}_1 : (n_1) \times (d_1) \mapsto (m_1)$ be a (k_1, ε_1)-extractor and let $\mathrm{E}_2 : (n_2) \times (d_2) \mapsto (d_1)$ be a (k_2, ε_2)-extractor. Let (X_1, X_2) be a (k_1, k_2) block source on $(n_1) \times (n_2)$. Then the distribution $\mathrm{E}_1(X_1, \mathrm{E}_2(X_2, U_{d_2}))$ is $(\varepsilon_1 + \varepsilon_2)$-close to U_{m_1}.*

Lemma 6.3 combined with Lemma 6.2 immediately implies a simple composition of high min-entropy extractors:

Lemma 6.4 (implicit in [GW97]) *Let $\mathrm{E}_1 : (n_1) \times (d_1) \mapsto (m_1)$ be an $(n_1 - \Delta, \varepsilon)$-extractor with entropy loss Λ_1 and let $\mathrm{E}_2 : (n_2) \times (d_2) \mapsto (d_1)$ be an $(n_2 - \Delta - \log 1/\varepsilon, \varepsilon)$-extractor with entropy loss Λ_2. Set $n = n_1 + n_2$ and define $\mathrm{E} : (n) \times (d_2) \mapsto (m_1)$ such that for any $x_1 \in (n_1)$, $x_2 \in (n_2)$ and $r_2 \in (d_2)$,*

$$\mathrm{E}(x_1 \circ x_2, r_2) \stackrel{\mathrm{def}}{=} \mathrm{E}_1(x_1, \mathrm{E}_2(x_2, r_2))$$

Then E is an $(n - \Delta, 3\varepsilon)$-extractor with entropy loss $\Lambda_1 + \Lambda_2 + \Delta + \log(1/\varepsilon)$.

As suggested in [GW97], applying Lemma 6.4 with the high min-entropy extractors of Theorem 5.3 as E_1 gives a way to obtain new high min-entropy extractors with a *much shorter* seed.

Proposition 6.5 *If $\mathrm{E}_2 : (n_2) \times (d_2) \mapsto (d_1)$ is an $(n_2 - \Delta - \log 1/\varepsilon, \varepsilon)$-extractor for $d_1 = \Delta + 2\log(1/\varepsilon) + 2$ with entropy loss Λ_2 then for any $n = n_1 + n_2$ there exist an $(n - \Delta, 3\varepsilon)$-extractor $\mathrm{E} : (n) \times (d_2) \mapsto (n_1)$ with entropy loss $\Lambda_2 + \Delta + 3\log(1/\varepsilon) + O(1)$ such that E is computable in polynomial time with one oracle query to E_2.*

One can even get an almost optimal seed length using this composition since the seed length of E equals the seed length of E_2, whose input length n_2 may be as small as $O(\Delta + \log(1/\varepsilon))$. Unfortunately, this composition always produces extractors of entropy loss at least Δ. Our aim in the next section is to remedy this.

6.2 Using Condensers

The reason the composition described in Lemma 6.4 must lose at least Δ bits is that when we divide our high min-entropy source into a prefix and suffix, either one of these parts may be missing Δ bits of entropy. Since we do not know how the source behaves, we must take a pessimistic approach and view our source as a block source where *each of the blocks* has deficiency Δ (for a total deficiency of 2Δ instead of Δ).[16] One approach for reducing the entropy loss of this composition is to apply a third extractor E_3 with fresh randomness to our source in order to extract the remaining entropy. However, conditioned on the randomness already extracted, the source now has a rather large deficiency. Therefore, the seed length of E_3 is at least $\log n$, which defeats the entire goal of the construction. The solution is rather simple: when applying E_1 and E_2 on the two parts of the source, also collect two relatively short buffers z_1 and z_2 with the remaining entropy. Now, E_3 can be applied to these buffers (instead of the source) in order to extract the missing entropy.

The result of the composition just sketched can be viewed as an extension of the zig-zag product to unbalanced bipartite graphs, where the application of E_2 corresponds to the "zig" step on the "small" graph G_2 and E_3 to the "zag" step. More formally, we have:

Definition 6.6 (zig-zag product for extractors) *Let* $\langle \mathrm{E}_1, \mathrm{C}_1\rangle : (n_1) \times (d_1) \mapsto (m_1) \times (b_1)$, $\langle \mathrm{E}_2, \mathrm{C}_2\rangle : (n_2) \times (d_2) \mapsto (d_1) \times (b_2)$, *and* $\langle \mathrm{E}_3, \mathrm{C}_3\rangle : (b_1 + b_2) \times (d_3) \mapsto (m_3) \times (b_3)$ *be three functions. Set the parameters*

$$\begin{aligned} n &= n_1 + n_2, \\ d &= d_2 + d_3, \\ m &= m_1 + m_3, \\ b &= b_3 \end{aligned}$$

and define the **zig-zag product**

$$\langle \mathrm{E}, \mathrm{C}\rangle : (n) \times (d) \mapsto (m) \times (b)$$

of these functions as follows: For any $x_1 \in (n_1)$, $x_2 \in (n_2)$,$r_2 \in (d_2)$ *and* $r_3 \in (d_3)$ *define*

$$\langle \mathrm{E}, \mathrm{C}\rangle(x_1 \circ x_2, r_2 \circ r_3) \stackrel{\text{def}}{=} \langle y_1 \circ y_2, z\rangle,$$

where

$$\begin{aligned} \langle r_1, z_1\rangle &\stackrel{\text{def}}{=} \langle \mathrm{E}_2, \mathrm{C}_2\rangle(x_2, r_2) \\ \langle y_1, z_2\rangle &\stackrel{\text{def}}{=} \langle \mathrm{E}_1, \mathrm{C}_1\rangle(x_1, r_1), \text{ and} \\ \langle y_2, z\rangle &\stackrel{\text{def}}{=} \langle \mathrm{E}_3, \mathrm{C}_3\rangle(z_1 \circ z_2, r_3). \end{aligned}$$

Theorem 6.7 *Let* $\langle \mathrm{E}_1, \mathrm{C}_1\rangle$, $\langle \mathrm{E}_2, \mathrm{C}_2\rangle$, $\langle \mathrm{E}_3, \mathrm{C}_3\rangle$ *and* $\langle \mathrm{E}, \mathrm{C}\rangle$ *be as in Definition 6.6. Let* $k = n - \Delta$, $k_1 = n_1 - \Delta$, $k_2 = n_2 - \Delta - \log(1/\varepsilon)$ *and* $k_3 = k + d_2 - m_1 - 1$. *If for* $i = 1, 2, 3$, $\langle \mathrm{E}_i, \mathrm{C}_i\rangle$ *is a* (k_i, ε)*-ECP with entropy loss* Λ_i *then* $\langle \mathrm{E}, \mathrm{C}\rangle$ *is a* $(k, O(\varepsilon))$*-ECP with entropy loss* $\Lambda_3 + 1$.

The key improvement over Lemma 6.4 is that the entropy loss of E no longer depends on Δ, but only on the entropy loss of E_3.

[16] There is a simpler solution to this problem if we only want to construct a *disperser* (cf., Definition 7.8): Following Ta-Shma [Ta-98], we can simply "guess" how the entropy is divided between the two blocks. Since there are only Δ bits of entropy whose location we are unsure of, the guessing only requires $\log \Delta$ additional random bits.

Remark 6.8 Theorem 6.7 implies a similar composition theorem of *strong* extractors (as originally defined in [NZ96]). Loosely speaking, E' is a **strong extractor** if E, defined by $\mathrm{E}(x, r) \overset{\text{def}}{=} \mathrm{E}'(x, r) \circ r$, is an extractor in the sense of Definition 5.1. To deduce the composition theorem of strong extractors, consider three ECPs, $\langle \mathrm{E}'_i, \mathrm{C}_i \rangle$ for $i = 1, 2, 3$, such that E'_i is a strong extractor. Let E_i be the standard extractor that corresponds to E'_i. If the three pairs $\langle \mathrm{E}_i, \mathrm{C}_i \rangle$ satisfy the conditions of Theorem 6.7 we can apply the composition of extractors and get the resultant ECP, $\langle \mathrm{E}, \mathrm{C} \rangle$. By Definition 6.6 and the definition of the extractors E_i it is easy to verify that for every x, r the seed r is part of the output $\mathrm{E}(x, r)$. This naturally implies the corresponding strong ECP, $\langle \mathrm{E}', \mathrm{C} \rangle$.

We now sketch the proof of Theorem 6.7.

Proof Sketch: For all possible values $x \in (n)$, $r_2 \in (d_2)$ and $r_3 \in (d_3)$, the computation of $\langle \mathrm{E}, \mathrm{C} \rangle(x, r_2 \circ r_3)$ produces the following intermediate values: $x_1, x_2, r_1, z_1, y_1, z_2, y_2$ and z. Let $X_1, X_2, R_1, Z_1, Y_1, Z_2, Y_2$ and Z be the corresponding random variables in the computation $\langle \mathrm{E}, \mathrm{C} \rangle(X, R_2 \circ R_3)$, where X is some k-source over (n), R_2 is uniformly distributed in (d_2) and R_3 is uniformly distributed in (d_3).

Lemma 6.3 directly implies that:

Claim 6.9 *Y_1 is $O(\varepsilon)$-close to U_{m_1}.*

It is also not hard to verify that the definitions of ECPs and $\langle \mathrm{E}, \mathrm{C} \rangle$ imply:

Claim 6.10 *(Y_1, Z_1, Z_2) is a 1-to-1 function of all but an $O(\varepsilon)$-fraction of $\langle X, R_2 \rangle$.*

By Claims 6.9 and 6.10, plus the facts that (X, R_2) has min-entropy $k + d_2$ and $k_3 = k + d_2 - m_1 - 1$, it follows that:

Claim 6.11 *$\langle Y_1, Z_1, Z_2 \rangle$ is $O(\varepsilon)$-close to some $\langle Y'_1, Z'_1, Z'_2 \rangle$, where Y'_1 is uniform over (m_1) and for every possible value y'_1 of Y'_1 the distribution of $\langle Z'_1, Z'_2 \rangle$, conditioned on $Y'_1 = y'_1$ is a k_3-source.*

By the definition of $\langle \mathrm{E}_3, \mathrm{C}_3 \rangle$ it is now immediate that $\langle Y_1, Y_2 \rangle$ is $O(\varepsilon)$-close to $U_{m_1+m_2}$. It is also not hard to verify that Z is a 1-to-1 function of all but an $O(\varepsilon)$-fraction of $\langle X, R_2, R_3 \rangle$. □

6.3 Applying the New Composition

A natural candidate for $\langle \mathrm{E}_1, \mathrm{C}_1 \rangle$ in Theorem 6.7 is again the high min-entropy extractors of Theorem 5.3 *when extended into a permutation extractor* (which can be achieved using a rotation map for the underlying expander, as described in Section 5). Using this permutation extractor one gets the following transformation from two "fixed"-sized ECPs to an arbitrary-sized, high min-entropy ECP:

Theorem 6.12 *For any $\varepsilon > 0$ and Δ, let $d_1 = \Delta + 2\log(1/\varepsilon) + 2$. Let $\langle \mathrm{E}_2, \mathrm{C}_2 \rangle : (n_2) \times (d_2) \mapsto (d_1) \times (b_2)$ be some $(n_2 - \Delta - \log 1/\varepsilon, \varepsilon)$-ECP. Let $\langle \mathrm{E}_3, \mathrm{C}_3 \rangle : (d_1 + b_2) \times (d_3) \mapsto (m_3) \times (b_3)$ be some $(n_2 + d_2 - \Delta - 1, \varepsilon)$-ECP with entropy loss Λ_3. Then for any $n = n_1 + n_2$ there exists an $(n - \Delta, O(\varepsilon))$-ECP*

$$\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d_2 + d_3) \mapsto (n_1 + m_3) \times (b_3)$$

with entropy loss $\Lambda_3 + 1$ such that E is computable in polynomial time with one oracle query to E_2 and one oracle query to E_3.

As opposed to Proposition 6.5, this result can imply ECPs that *simultaneously obtain a short seed* (provided that $\langle \mathrm{E}_2\ \mathrm{C}_2 \rangle$ and $\langle \mathrm{E}_3\ \mathrm{C}_3 \rangle$ have short seeds) *and a small entropy-loss* (provided that E_3 has a small entropy-loss). One way to apply Theorem 6.12 is by exhaustively searching for $\langle \mathrm{E}_2, \mathrm{C}_2 \rangle$ and $\langle \mathrm{E}_3, \mathrm{C}_3 \rangle$ of optimal parameters (i.e. those in Proposition 5.2). This method is applicable as long as Δ and $\log 1/\varepsilon$ are sufficiently small. In fact, when $\log 1/\varepsilon$ is larger than Δ then Theorem 5.3 already gives good ECPs. We therefore have:

Corollary 6.13 *For any $0 \leq \Delta < n$ and $\varepsilon > 0$, there exists a $(n-\Delta, \varepsilon)$ permutation extractor*

$$\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (m) \times (b),$$

with seed length $d = 2\log\Delta + 4\log(1/\varepsilon) + O(1)$ and entropy loss $\Lambda = b - \Delta = 2\log(1/\varepsilon) + O(1)$, such that $\langle \mathrm{E}, \mathrm{C} \rangle$ is computable in time $2^{2^{O(\Delta)}} \cdot \mathrm{poly}(n)$.

In case exhaustive search is too expensive, one can still use "off the shelf" explicit extractors and get significant improvements. In order to do this, we need to extend these existing extractors to ECP's. It turns out that Lemma 5.6 is too expensive in terms of seed length for us, but a more trivial conversion suffices — simply let the buffer be the entire input and seed! This does not cost anything in the seed length, and the buffers are not too large since E_2 and E_3 have short inputs in Theorem 6.12. Taking E_2 and E_3 to be the extractors of Zuckerman [Zuc97] extended to ECP's in this way, we get:

Corollary 6.14 *Let $\alpha > 0$ be an arbitrarily small constant. For any $0 \leq \Delta < (1-\alpha)n$ and $\varepsilon > \exp(-\Delta/2^{O(\log^* \Delta)})$,*[17] *there exists an explicit $(n-\Delta, \varepsilon)$-ECP*

$$\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (m) \times (b),$$

with seed length $d = O(\log\Delta + \log(1/\varepsilon))$, buffer size $b = O(\Delta + \log(1/\varepsilon))$, and entropy loss $\Lambda \leq \alpha\Delta$.

Similarly, using the extractors of Reingold, Shaltiel, and Wigderson [RSW00], we get:

Corollary 6.15 *For any $0 \leq \Delta < n$ and $\varepsilon > \exp(-\Delta/(\log^* \Delta)^{O(\log^* \Delta)})$, there exists an explicit $(n-\Delta, \varepsilon)$-ECP*

$$\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (m) \times (b),$$

with seed length $d = O(\log^2\Delta \cdot \mathrm{polyloglog}\Delta + \log\Delta \cdot \log(1/\varepsilon))$, and entropy loss $\Lambda \leq 2\log(1/\varepsilon) + O(1)$.

Extractors from Elementary Building Blocks. Section 3 gives a simple construction of a *family* of constant-degree expanders out of a single fixed-sized expander. In this section, we have seen a similar construction of extractors. However, this construction also uses the extractors of [GW97] described in Theorem 5.3. Still, since these extractors are essentially a walk on an expander, and we already have expanders out of elementary building blocks one may argue that we also obtain high min-entropy extractors out of elementary building blocks. We also note that, using recursive applications of Theorem 6.7, one can also construct good high min-entropy extractors that are only based on two fixed-sized extractors (i.e. of input length $O(\Delta + \log(1/\varepsilon))$ and *pairwise independent permutations*. We find some aesthetic value in this construction given the fundamental role of pairwise independent permutations in the development of extractors.

7 Applications of the Extractors

7.1 Averaging Samplers

In this section, we describe how our high min-entropy extractors yield improved sampling algorithms, as pointed out to us by Ronen Shaltiel. The reader is referred to the survey of Goldreich [Gol97] and the references therein for a detailed description of previous work on samplers.

A **sampler** is a randomized algorithm which, given any function $f : (m) \to [0,1]$ as an oracle, estimates (with high probability) the average value of f up to some desired accuracy ε. It is desirable to minimize both the number of random bits and the number of queries to f made by such a procedure. Bellare and Rompel [BR94] noted that, in some applications, it is important to have samplers of the following (natural) form (called **oblivious samplers** in [BR94]):

[17] This restriction on ε is inherited from an analogous restriction on the relationship between ε and the input length in [Zuc97].

Definition 7.1 (averaging samplers [BR94]) *A function* $\mathrm{S} : (n) \to (m)^t$ *is a* (γ, ε)-**averaging sampler** *if, for every function* $f : (m) \to [0,1]$,

$$\Pr_{(z_1,\ldots,z_t)\leftarrow \mathrm{S}(U_n)}\left[\left|\frac{1}{t}\sum_{i=1}^{t} f(x_i) - \mu(f)\right| \leq \varepsilon\right] \geq 1-\gamma,$$

where $\mu(f)$ *is the average of* f *over* (m). S *is said to* **explicit** *if it can be evaluated in time* $\mathrm{poly}(n, m, t)$.

Nonconstructively, there are averaging samplers using only $t = O(\log(1/\gamma)/\varepsilon^2)$ samples and $n = m + \log(1/\gamma) + O(1)$ random bits [CEG95, Zuc97], and these bounds are essentially tight [CEG95]. (See [Gol97] for precise statements.)

Zuckerman [Zuc97] has shown that averaging samplers are essentially equivalent to extractors. We will only use the transformation from extractors to samplers: From an extractor $\mathrm{E} : (n) \times (d) \to (m)$, define an averaging sampler $\mathrm{S}_{\mathrm{E}} : (n) \to (m)^t$ with $t = 2^d$ by

$$\mathrm{S}_{\mathrm{E}}(x) = (\mathrm{E}(x, y_1), \ldots, \mathrm{E}(x, y_t)),$$

where $y_1, \ldots, y_t$ are all the strings of length d. The parameters of S_{E} are related to those of E as follows.

Lemma 7.2 ([Zuc97]) *If* E *is a* $(n - \Delta, \varepsilon)$*-extractor, then* S_{E} *is* $(1/2^{\Delta-1}, \varepsilon)$*-averaging sampler.*

Applying this lemma to his extractor construction, Zuckerman [Zuc97] obtained an explicit sampler which uses $n = (1+\alpha)(m+\log(1/\gamma))$ truly random bits and $t = \mathrm{poly}(m, 1/\varepsilon, \log(1/\gamma))$ samples for an arbitrarily small constant $\alpha > 0$.

Observe that the confidence of the averaging sampler in the above lemma corresponds to the extractor's entropy deficiency, and the number of samples used by the sampler corresponds to the number of truly random bits used by the extractor. Hence, our high min-entropy extractors in which the number of truly random bits depends only on the entropy deficiency of the source translate to samplers in which the number of samples depends only on the confidence. In particular, applying Lemma 7.2 to Corollary 6.14 we obtain the following improvement to Zuckerman's samplers:

Corollary 7.3 *Let* $\alpha > 0$ *be an arbitrarily small constant. For any* m, γ*, and* $\varepsilon > \exp(\log(\gamma)/2^{O(\log^* \gamma^{-1})})$*, there exists an explicit* (γ, ε)*-averaging sampler* $\mathrm{S} : (n) \to (m)^t$ *with* $n = m + (1+\alpha) \cdot \log(1/\gamma)$ *and* $t = \mathrm{poly}(1/\varepsilon, \log(1/\gamma))$.

Our other extractor constructions given in Corollaries 6.13 and 6.15 similarly yield averaging samplers (with different parameters), but we omit the exact statements for sake of brevity.

7.2 Expanders that Beat the Eigenvalue Bound

There are many different measures for the expansion properties of a graph. In Sections 2 and 3, we worked with an eigenvalue measure, which is convenient for many purposes. Another measure which is appropriate in other settings is the following one, due to Pippenger.

Definition 7.4 ([Pip87]) *Let* G *be an undirected graph on* $[N]$. G *is* A-**expanding,**[18] *if every two subsets of at least* N/A *vertices each are joined by an edge.*

[18] Actually, Pippenger's definition refers to such graphs as N/A-expanding, but this version is more convenient for our purposes.

The goal, of course, is to minimize the degree (as a function of the other parameters) as a function of A and N. A standard nonconstructive probabilistic argument shows that degree $D = O(A \cdot \log A)$ suffices, but, as usual, we seek explicit constructions. Ideally, given $v \in [N]$ and $i \in [D]$, we would like to be able to compute the i'th neighbor of v (or even evaluate $\mathrm{Rot}_G(v, i)$ in case G is regular) efficiently (e.g., in time $\mathrm{polylog} N$), but such constructions are still useful (and nontrivial) even if they work in time $\mathrm{poly}(N)$.

Explicit constructions have not yet matched the optimal $O(A \cdot \log A)$ degree bound. The A-expanding property does follow from bounds on the second largest eigenvalue [Tan84, AM85], but best degree that can be obtained via this relationship is $\Theta(A^2)$ (cf., discussion in [WZ99]). Wigderson and Zuckerman [WZ99] proposed to beat this barrier using extractors as follows, and thereby obtained degree $A \cdot N^{o(1)}$, which is very close to optimal when A is a constant power of N. Later extractors reduced the dependence on N, with Ta-Shma [NT99] achieving degree $A \cdot 2^{\mathrm{poly} \log \log N}$ (with reductions in the degree of the polylog in [RRV99b, RSW00]). Here we show how our high min-entropy extractors can completely remove the dependence in these results, and hence obtain "constant-degree" expanders that beat the eigenvalue bound.

We recall the Wigderson–Zuckerman [WZ99] method for constructing A-expanding graphs from extractors. First, suppose we have a *regular* extractor $\mathrm{E} : (n) \times (d) \to (m)$ (as some of our extractors are). Define a graph G_{E} on $N = 2^n$ vertices by placing one edge between $x_1, x_2 \in (n)$ for each pair $r_1, r_2 \in (d)$ such that $\mathrm{E}(x, r_1) = \mathrm{E}(x, r_2)$.

Lemma 7.5 ([WZ99]) *If* E *is a regular* $(n-\Delta, 1/4)$*-extractor with entropy loss* $\Lambda = (n-\Delta)+d-m$*, then* G_{E} *is an* A*-expanding graph for* $A = 2^{\Delta}$*, and is of degree* $D = A \cdot 2^{\Lambda+d}$*. Moreover, if* E *can be extended to a permutation extractor* $\langle \mathrm{E}, \mathrm{C} \rangle$*, then a rotation map for* G_{E} *can be computed in time* $\mathrm{poly}(\log N, \log D)$ *with oracle access to* $\langle \mathrm{E}, \mathrm{C} \rangle$ *and its inverse.*

Proof: Let S and T be any subsets of $[N]$ of size $\geq N/A = 2^{n-\Delta}$. Since E is a $(n-\Delta, 1/4)$-extractor, $\mathrm{E}(S, U_d)$ and $\mathrm{E}(T, U_d)$ are each $1/4$-close to uniform (where S and T denote the uniform distributions over the corresponding sets). In particular, the supports of $\mathrm{E}(S, U_d)$ and $\mathrm{E}(T, U_d)$ intersect, which implies that there is an edge between S and T in G_{E}.

By the regularity of E, the degree of every vertex in G_{E} is $2^d \cdot 2^{n+d-m} = 2^d \cdot 2^{\Delta} \cdot 2^{(n-\Delta)+d-m} = 2^d \cdot A \cdot 2^{\Lambda}$. Now suppose that E can be extended to a permutation extractor $\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \to (m) \times (b)$. Then we can obtain a rotation map for G_{E} as follows: For $x \in (n)$, $r \in (d)$, $s \in (b)$, set $\mathrm{Rot}_G(x, (r, s)) = (x', (r', s'))$ where $(y, s') = \langle \mathrm{E}, \mathrm{C} \rangle (x, r)$ and $(x', r') = \langle \mathrm{E}, \mathrm{C} \rangle^{-1}(y, s)$. ■

Observe that, if the seed length d and entropy loss Λ of the extractor depend only on Δ (rather than n), then the degree of the graph depends only on A, as desired. Using our high min-entropy extractors from Corollary 6.13 in this construction, we obtain:

Corollary 7.6 *For every* N *and* A*, there is a (regular)* A*-expanding graph* $G_{N,A}$ *of degree* $O(A \cdot \log^4 A)$ *such that the rotation map for* $G_{N,A}$ *can be computed in time* $\mathrm{poly}(\log N, 2^A)$.

This construction is significantly closer to the optimal degree bound of $O(A \log A)$ than previous constructions, and is quite efficient when A is small (e.g. a constant independent of N). In order to reduce the dependence of the computation time on A, we use our high min-entropy extractors from Corollary 6.15. However, these extractors are not regular and hence the above construction can result in graphs whose degree is much higher than stated in Lemma 7.5. The method Wigderson and Zuckerman [WZ99] suggest to overcome this is to "throw out" extractor outputs which have more than twice the average indegree 2^{n+d-m} (when E is viewed as a bipartite graph). That is, G_{E} is defined by connecting $x_1, x_2 \in (n)$ if there exists a $y \in (m)$ and $r_1, r_2 \in (d)$ such that $\mathrm{E}(x, r_1) = y = \mathrm{E}(x, r_2)$ *and* the number of pairs (z, r) such that $\mathrm{E}(z, r) = y$ is at most $2^{n+d-m+1}$. Wigderson and Zuckerman show that this still results in an A-expanding graph of degree $A \cdot 2^{\Lambda+d+1}$. We must also analyze the computational complexity of the construction. A straightforward upper bound on the time of the construction, as given in [WZ99], is $\mathrm{poly}(N)$. However, using the fact that our extractors were constructed using an explicit permutation extractor (the one extending Theorem 5.3) as the large graph in a the zig-zag product, we can show that the complexity is actually only $\mathrm{poly}(A)$. We thereby obtain the following graphs:

Corollary 7.7 *For every N and $A \leq N$, there is an A-expanding graph $G_{N,A}$ of degree*

$$A \cdot (\log A)^{\mathrm{loglog}A \cdot \mathrm{polyloglogloglog}A}$$

such that all the neighbors of a vertex in $G_{N,A}$ can be enumerated in time $\mathrm{poly}(\log N, A)$.

More details for these constructions will be given in a subsequent paper.

7.3 Error Reduction for Dispersers

Dispersers, defined by Sipser [Sip88], are the one-sided analogue of extractors. Rather than inducing the uniform distribution on their output, they are only guaranteed to hit most points with nonzero probability.

Definition 7.8 *A function* $\mathrm{D} : (n) \times (d) \to (m)$ *is a* (k, ε)-**disperser** *if for every distribution X on (n) of min-entropy k, at most a ε fraction of points in (m) have zero probability under* $\mathrm{D}(X, U_d)$.

Clearly, every (k, ε)-extractor is also a (k, ε)-disperser. However, the parameters of dispersers can be somewhat better than those of extractors, specifically with respect to the error ε. In optimal (nonconstructive) dispersers, the number of truly random bits need only be $d = \log(n-k) + \log(1/\varepsilon) + O(1)$ and the entropy loss need only be $\Lambda = k + d - m = \mathrm{loglog}(1/\varepsilon) + O(1)$, whereas for extractors, $d \geq \log(n-k) + 2\log(1/\varepsilon) - O(1)$ and $\Lambda \geq 2\log(1/\varepsilon) - O(1)$ [RT97]. However, none of the existing explicit constructions of dispersers have managed to achieve the parameters that are impossible for extractors.

Here, we show how a generalization of the Wigderson–Zuckerman construction described in the previous section yields a general error-reduction technique for dispersers — we can compose two constant-error dispersers and obtain a disperser with a small error ε. Taking one of the initial dispersers to be one of our high min-entropy extractors, we are (in some cases) able to get a dispersers whose seed length and entropy loss have a better dependence on ε than is possible for extractors. Previously, an error-reduction procedure was given for extractors in [RRV99a]. An error-reduction technique for dispersers related to ours was independently discovered by Ta-Shma and Zuckerman.

To see the connection between dispersers and A-expanding graphs, note that a regular 2^a-expanding graph on 2^n vertices is exactly the same as a regular $(n-a, 1/2^a)$-disperser $\mathrm{D} : (n) \times (d) \to (n)$ (when viewed as a bipartite graph). Hence, dispersers are simply a generalization of the A-expanding property to unbalanced, bipartite graphs.

The first observation in our error reduction is that to reduce the error of a constant-error disperser D_1 to ε, it suffices to compose it with ε-error disperser D_2 with constant entropy deficiency.

Lemma 7.9 *Suppose* $\mathrm{D}_1 : (n_1) \times (d_1) \to (m_1)$ *is a* $(k_1, 1/4)$*-disperser with entropy loss* Λ_1 *and* $\mathrm{D}_2 : (m_1) \times (d_2) \to (m_2)$ *is a* $(m_1 - 1, \varepsilon_2)$*-disperser with entropy loss* Λ_2. *Define* $\mathrm{D}_3 : (n_1) \times (d_1 + d_2) \to (m_2)$ *by* $\mathrm{D}(x, (r_1, r_2)) = \mathrm{D}_2(\mathrm{D}_1(x, r_1), r_1)$. *Then* D_3 *is a* (k_1, ε_2)*-disperser with entropy loss* $\Lambda_1 + \Lambda_2 + 1$.

Thus, to reduce the error of any arbitrary disperser D_1, we only need good dispersers for entropy deficiency 1. Using the extractor of Goldreich and Wigderson in Theorem 5.3 as D_2, yields a transformation which, in going from D_1 to D_3, increases both the seed length and entropy loss of by $2\log(1/\varepsilon) + O(1)$, which is quite good (but not optimal for dispersers). This is the error-reduction approach suggested by Ta-Shma and Zuckerman.

We improve on this by using our high min-entropy extractors. The key observation is that if we "flip" a disperser (viewing it is a bipartite graph), the roles of the entropy deficiency and error are switched. For simplicity, we focus on permutation dispersers (defined analogously to permutation extractors).

Lemma 7.10 *Let* $\mathrm{D} : (n) \times (d) \to (m) \times (b)$ *be a* $(n - \Delta, \varepsilon)$ *permutation disperser with entropy loss* Λ*. Then* D^{-1} *is a* $(m - \log(1/\varepsilon), 1/2^{\Delta})$ *permutation disperser with seed length* $b = \Lambda + \Delta$ *and entropy loss* $d - \log(1/\varepsilon)$*.*

Applying this to our extractor in Corollary 6.13, we get:

Lemma 7.11 *For any* $0 \le \Delta < n$ *and* $\varepsilon > 0$*, there exists a* $(n - \Delta, \varepsilon)$ *permutation extractor*

$$\langle \mathrm{E}, \mathrm{C} \rangle : (n) \times (d) \mapsto (m) \times (b),$$

with seed length $d = 2\Delta + \log(1/\varepsilon) + O(1)$ *and entropy loss* $\Lambda = 2\mathrm{loglog}(1/\varepsilon) + 3\Delta + O(1)$*, such that* $\langle \mathrm{E}, \mathrm{C} \rangle$ *is computable in time* $\mathrm{poly}(n, 2^{1/\varepsilon})$*.*

Thus, we already see the dependence on ε in both the seed length and entropy loss beating what is possible for extractors. Using this extractor as D_2 in Lemma 7.9, we get

Corollary 7.12 *Suppose* $\mathrm{D} : (n) \times (d) \to (m)$ *is a* $(k, 1/4)$*-disperser with entropy loss* Λ*. Then, for any* $\varepsilon > 0$*, there exists a* (k, ε)*-disperser* $\mathrm{D}' : (n) \times (d') \to (m')$ *with seed length* $d' = d + \log(1/\varepsilon) + O(1)$ *and entropy loss* $\Lambda + 2\mathrm{loglog}(1/\varepsilon) + O(1)$*. Moreover,* D' *is computable in time* $\mathrm{poly}(n, d, 2^{1/\varepsilon})$ *with one oracle query to* D*.*

Note that the complete transformation taking D and the extractor of Corollary 6.13 and applying Lemmas 7.10 and 7.9 to yield D' is simply the Wigderson–Zuckerman construction we used in the previous section, generalized to using two different dispersers with the same output length.

Corollary 7.12 is essentially an optimal error-reduction for dispersers, except for the exponential computation time as a function of ε. As in the previous section, to overcome this we can use our extractors from Corollaries 6.14 and 6.15, and we need to do similar tricks to deal with the fact that they are not regular. We defer the details of the proof to the final version of the paper, but the results obtained are as follows:

Corollary 7.13 *Let* $\alpha > 0$ *be an arbitrarily small constant. Suppose* $\mathrm{D} : (n) \times (d) \to (m)$ *is a* $(k, 1/4)$*-disperser with entropy loss* Λ*. Then, for any* $\varepsilon > \exp((1 - \alpha)n)$*, there exists a* (k, ε)*-disperser* $\mathrm{D}' : (n) \times (d') \to (m')$ *with seed length* $d' = d + (1 + \alpha)\log(1/\varepsilon)$ *and entropy loss* $\Lambda + O(\mathrm{loglog}(1/\varepsilon))$*. Moreover,* D' *is computable in time* $\mathrm{poly}(n, d, 1/\varepsilon)$ *with one oracle query to* D*.*

Corollary 7.14 *Suppose* $\mathrm{D} : (n) \times (d) \to (m)$ *is a* $(k, 1/4)$*-disperser with entropy loss* Λ*. Then, for any* $\varepsilon > 0$*, there exists a* (k, ε)*-disperser* $\mathrm{D}' : (n) \times (d') \to (m')$ *with seed length* $d' = d + \log(1/\varepsilon) + O(1)$ *and entropy loss* $\Lambda + \mathrm{polyloglog}(1/\varepsilon)$*. Moreover,* D' *is computable in time* $\mathrm{poly}(n, d, 1/\varepsilon)$ *with one oracle query to* D*.*

Acknowledgments

We are indebted to David Zuckerman for illuminating discussions and a number of useful suggestions early in the stages of this work. We thank the organizers of the DIMACS Workshop on Pseudorandomness and Explicit Combinatorial Constructions in October 1999, where we began this research. We are grateful to Peter Winkler for suggesting the terminology "zig-zag product" and Ronen Shaltiel for pointing out the application to averaging samplers. We also thank Noga Alon, Oded Goldreich, and Dan Spielman for helpful conversations and pointers.

References

[Ajt94] M. Ajtai. Recursive construction for 3-regular expanders. *Combinatorica*, 14(4):379–416, 1994.

[AKS87] Miklós Ajtai, János Komlós, and E. Szemerédi. Deterministic simulation in LOGSPACE. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, pages 132–140, New York City, 25–27 May 1987.

[AKS83] Miklós Ajtai, János Komlós, and Endre Szemerédi. Sorting in $c \log n$ parallel steps. *Combinatorica*, 3(1):1–19, 1983.

[AGM87] N. Alon, Z. Galil, and V. D. Milman. Better expanders and superconcentrators. *J. Algorithms*, 8(3):337–347, 1987.

[AM85] N. Alon and V. D. Milman. λ_1, isoperimetric inequalities for graphs, and superconcentrators. *J. Combin. Theory Ser. B*, 38(1):73–88, 1985.

[Alo86a] Noga Alon. Eigenvalues and expanders. *Combinatorica*, 6(2):83–96, 1986.

[Alo86b] Noga Alon. Eigenvalues, geometric expanders, sorting in rounds, and Ramsey theory. *Combinatorica*, 6(3):207–219, 1986.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta. Simple constructions of almost k-wise independent random variables. *Random Structures & Algorithms*, 3(3):289–304, 1992.

[AR94] Noga Alon and Yuval Roichman. Random Cayley graphs and expanders. *Random Structures Algorithms*, 5(2):271–284, 1994.

[BR94] Mihir Bellare and John Rompel. Randomness-efficient oblivious sampling. In *35th Annual Symposium on Foundations of Computer Science*, pages 276–287, Santa Fe, New Mexico, 20–22 November 1994. IEEE.

[Blu86] M. Blum. Independent unbiased coin flips from a correlated biased source—a finite state Markov chain. *Combinatorica*, 6(2):97–108, 1986. Theory of computing (Singer Island, Fla., 1984).

[BS87] Andrei Broder and Eli Shamir. On the second eigenvalue of random regular graphs (preliminary version). In *28th Annual Symposium on Foundations of Computer Science*, pages 286–294, Los Angeles, California, 12–14 October 1987. IEEE.

[CEG95] Ran Canetti, Guy Even, and Oded Goldreich. Lower bounds for sampling algorithms for estimating the average. *Information Processing Letters*, 53(1):17–25, 13 January 1995.

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

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

[CW89] Aviad Cohen and Avi Wigderson. Dispersers, deterministic amplification, and weak random sources (extended abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 14–19, Research Triangle Park, North Carolina, 30 October–1 November 1989. IEEE.

[Fri91] Joel Friedman. On the second eigenvalue and random walks in random d-regular graphs. *Combinatorica*, 11(4):331–362, 1991.

[FKS89] Joel Friedman, Jeff Kahn, and Endre Szemerédi. On the second eigenvalue in random regular graphs. In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, pages 587–598, Seattle, Washington, 15–17 May 1989.

[GG81] Ofer Gabber and Zvi Galil. Explicit constructions of linear-sized superconcentrators. *Journal of Computer and System Sciences*, 22(3):407–420, June 1981.

[Gol97] Oded Goldreich. A sample of samplers: A computational perspective on sampling. Technical Report TR97-020, Electronic Colloquium on Computational Complexity, May 1997.

[GIL$^+$90] Oded Goldreich, Russell Impagliazzo, Leonid Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *31st Annual Symposium on Foundations of Computer Science*, volume I, pages 318–326, St. Louis, Missouri, 22–24 October 1990. IEEE.

[GW97] Oded Goldreich and Avi Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Structures & Algorithms*, 11(4):315–343, 1997.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396 (electronic), 1999.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on the Theory of Computing*, pages 356–364, Montréal, Québec, Canada, 23–25 May 1994.

[ISW00] Russell Impagliazzo, Ronen Shaltiel, and Avi Wigderson. Extractors and pseudo-random generators with optimal seed length. In *Proceedings of the Thirty-Second Annual ACM Symposium on the Theory of Computing*, pages 1–10, Portland, Oregon, May 2000. See also ECCC TR00-009.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 220–229, El Paso, Texas, 4–6 May 1997.

[IZ89] Russell Impagliazzo and David Zuckerman. How to recycle random bits. In *30th Annual Symposium on Foundations of Computer Science*, pages 248–253, Research Triangle Park, North Carolina, 30 October–1 November 1989. IEEE.

[JM87] Shuji Jimbo and Akira Maruoka. Expanders obtained from affine transformations. *Combinatorica*, 7(4):343–355, 1987.

[KPS85] Richard Karp, Nicholas Pippenger, and Michael Sipser. A time-randomness tradeoff. In *AMS Conference on Probabilistic Computational Complexity*, Durham, New Hampshire, 1985.

[LPS88] A. Lubotzky, R. Phillips, and P. Sarnak. Ramanujan graphs. *Combinatorica*, 8(3):261–277, 1988.

[LW93] A. Lubotzky and B. Weiss. Groups and expanders. In *Expanding graphs (Princeton, NJ, 1992)*, pages 95–109. Amer. Math. Soc., Providence, RI, 1993.

[LW00] Alex Lubotzky and Avi Wigderson. Personal communication, September 2000.

[Mar73] G. A. Margulis. Explicit constructions of expanders. *Problemy Peredači Informacii*, 9(4):71–80, 1973.

[Mar88] 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.

[Mor94] Moshe Morgenstern. Existence and explicit constructions of $q + 1$ regular Ramanujan graphs for every prime power q. *J. Combin. Theory Ser. B*, 62(1):44–62, 1994.

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

[Nil91] A. Nilli. On the second eigenvalue of a graph. *Discrete Math.*, 91(2):207–210, 1991.

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

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

[Pin73] M. Pinsker. On the complexity of a concentrator. In *7th Annual Teletraffic Conference*, pages 318/1–318/4, Stockholm, 1973.

[PY82] N. Pippenger and A.C. Yao. Rearrangeable networks with limited depth. *SIAM J. Algebraic and Discrete Methods*, 3:411–417, 1982.

[Pip87] Nicholas Pippenger. Sorting and selecting in rounds. *SIAM Journal on Computing*, 16(6):1032–1038, December 1987.

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

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

[RRV99a] Ran Raz, Omer Reingold, and Salil Vadhan. Error reduction for extractors. In *Proceedings of the 40th Annual Symposium on the Foundations of Computer Science*, New York, NY, October 1999. IEEE.

[RRV99b] Ran Raz, Omer Reingold, and Salil Vadhan. Extracting all the randomness and reducing the error in Trevisan's extractors. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 149–158, Atlanta, GA, 1999.

[RSW00] Omer Reingold, Ronen Shaltiel, and Avi Wigderson. Extracting randomness via repeated condensing. In *41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, California, 12–14 November 2000. IEEE.

[RVW00] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors (extended abstract). In *41st Annual Symposium on Foundations of Computer Science*, Redondo Beach, California, 12–14 November 2000. IEEE.

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

[Sho90] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. *Mathematics of Computation*, 54(189):435–447, 1990.

[Sip88] Michael Sipser. Expanders, randomness, or time versus space. *Journal of Computer and System Sciences*, 36(3):379–383, June 1988.

[SS96] Michael Sipser and Daniel A. Spielman. Expander codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1710–1722, 1996. Codes and complexity.

[Spi96] Daniel A. Spielman. Linear-time encodable and decodable error-correcting codes. *IEEE Trans. Inform. Theory*, 42(6, part 1):1723–1731, 1996. Codes and complexity.

[Ta-98] Amnon Ta-Shma. Almost optimal dispersers. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pages 196–202, Dallas, TX, May 1998. ACM.

[Tan84] Michael R. Tanner. Explicit concentrators from generalized n-gons. *SIAM Journal on Algebraic Discrete Methods*, 5(3):287–293, 1984.

[Tre99] Luca Trevisan. Construction of extractors using pseudo-random generators. In *Proceedings of the Thirty-First Annual ACM Symposium on the Theory of Computing*, pages 141–148, Atlanta, GA, May 1999. See also ECCC TR98-55.

[Urq87] Alasdair Urquhart. Hard examples for resolution. *J. Assoc. Comput. Mach.*, 34(1):209–219, 1987.

[Val77] Leslie G. Valiant. Graph-theoretic arguments in low-level complexity. In *Mathematical foundations of computer science (Proc. Sixth Sympos., Tatranská Lomnica, 1977)*, pages 162–176. Lecture Notes in Comput. Sci., Vol. 53. Springer, Berlin, 1977.

[VV85] Umesh V. Vazirani and Vijay V. Vazirani. Random polynomial time is equal to slightly-random polynomial time. In *26th Annual Symposium on Foundations of Computer Science*, pages 417–428, Portland, Oregon, 21–23 October 1985. IEEE.

[WZ99] Avi Wigderson and David Zuckerman. Expanders that beat the eigenvalue bound: explicit construction and applications. *Combinatorica*, 19(1):125–138, 1999.

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

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