Tiny Families of Functions with Random Properties: A Quality–Size Trade–off for Hashing[*]

Oded Goldreich[†]
Department of Computer Science
and Applied Mathematics
Weizmann Institute of Science
Rehovot, Israel.

Avi Wigderson[‡]
Institute for Computer Science
Hebrew University
Givat Ram
Jerusalem, Israel.

March 21, 1997

Abstract

We present three explicit constructions of hash functions, which exhibit a trade-off between the size of the family (and hence the number of random bits needed to generate a member of the family), and the quality (or error parameter) of the pseudo-random property it achieves. Unlike previous constructions, most notably universal hashing, the size of our families is essentially independent of the size of the domain on which the functions operate.

The first construction is for the *mixing* property – mapping a proportional part of any subset of the domain to any other subset. The other two are for the *extraction* property – mapping any subset of the domain almost uniformly into a range smaller than it. The second and third constructions handle (respectively) the extreme situations when the range is very large or very small.

We provide lower bounds showing that our constructions are nearly optimal, and mention some applications of the new constructions.

Keywords: Randomness and Computation, Randomness Extractors, Sampling Algorithms, Random-Looking Functions, Expander Graphs, Ramanujan Graphs, Universal Hashing, Small-Biased Probability Spaces, Lindsey's Lemma.

[*] An extended abstract of this paper has appeared in the *26th ACM Symposium on Theory of Computing* (STOC 94) held in Montreal, Quebec, Canada, May 23-25, 1994.

[†] Research was supported in part by grant No. 92-00226 from the United States – Israel Binational Science Foundation (BSF), Jerusalem, Israel.

[‡] Research was supported in part by the Wolfson Research Awards, administered by the Israel Academy of Sciences and Humanities.

1 Introduction

In 1979, Carter and Wegman introduced the notion of universal hashing functions [9]. Though these functions were introduced with data storage application in mind, they found many applications to complexity theory [30, 32, 34, 18, 17, 21, 22, 19, 20, 27, 28, 36]. This wide range of applications owns its existence to two related 'random' properties of these succinct and efficiently computable functions: the *extraction* and the *mixing* properties.

For a family F of functions, each mapping n-bit strings to m-bit strings, the *extraction* property asserts the following. Every subset of $K \cdot 2^m$ strings in the domain $\{0,1\}^n$, is mapped almost uniformly to the range $\{0,1\}^m$, by all but a small fraction of the functions in the family. The parameter $K > 1$ determines the quality of the approximation to the uniform distribution and the fraction of bad functions in F (i.e. those that don't achieve this approximation). The extraction property is the heart of the Leftover Hash Lemma [21] and its precursors, which were key to numerous results, e.g. in saving randomness [22], weak random sources [36], pseudorandom generators [17, 21] and interactive proofs [18]. (Alternative function families with extraction property were previously constructed in [29], with a variety of other applications.)

The *mixing* property is meaningful also in case $m = n$, and in fact it is commonly used with this choice. Hence, we assume for simplicity that $m = n$. Loosely speaking, the mixing property asserts that, for all but a small fraction of the functions f in the family F, the membership in $A \times B$ of a pair $(r, f(r))$ with r being a random element from the domain, is essentially the same as that of a random pair (r, s) of elements. The prime use of the mixing property is in the logspace pseudorandom generators of Nisan [27, 28].

In the definitions above, there is an error parameter ϵ (e.g. the fraction of bad functions, the distance from the uniform distribution etc.), which determines the quality of the mixing or extraction achieved by the family F. All the applications mentioned above take F to be a universal family of hash functions. This family achieves the best possible quality parameter: ϵ is exponentially small in m. However, while small enough for these applications, a universal family has to be large: exponential in n.

But in some applications we may be content with a larger ϵ (i.e. lower quality), say constant or $1/poly(n)$. Can we use much smaller families F in this case and achieve similar random properties? A straightforward counting argument shows that there exist families F of size $poly(1/\epsilon)$ (resp., $poly(n/\epsilon)$) achieving the mixing (resp., extraction) properties with quality ϵ. Note that these bounds depend essentially only on the quality required, and not on the size of the domain.

1.1 Our Results

The main contribution of this paper is in presenting explicit constructions of such families, thus yielding a trade-off between the size of the family and the desired quality. The first construction is for mixing, where we obtain a complete trade-off. The second and third constructions are for extraction, where we (respectively) handle two extreme cases: when $n - m \ll n$ and when $m \ll n$. Our constructions are relatively simple. The first two of them combine universal hashing and expander graphs. (It is interesting to note that despite the similarity in these two constructions, the proofs are completely different). The third construction uses small-bias probability spaces of small size, and its analysis utilizes a new generalization of Lindsey's Lemma. We provide lower bounds to show that the first construction is nearly optimal, and the third is nearly optimal for $m = O(\log n)$. By nearly optimal here we mean that the number of bits needed to describe a member of the family in our constructions is within a constant factor of the lower bound.

Using the first construction we reduce the randomness complexity of two generic procedures as follows:

1. For sampling procedures, which use an asymptotically optimal number of sample points, the amount of randomness required to generate the sample points is reduced by a *factor* of 2, yielding an optimal result up-to a small *additive term*; and

2. The randomness complexity of Nisan's "generalized logspace" generator [27], is reduced by a logarithmic factor.

The second construction implies a randomness–efficient leftover hash lemma, which is particularly appealing in case $n - m \ll n$. The third construction turned out to be the main technical tool in the recent advances on constructing nearly optimal *extractors*[1] for any $m = \Theta(n)$, on which we elaborate below.

1.2 Previous, Concurrent and Subsequent Work

Despite the general interest in reducing the size of sample spaces achieving various random properties, very little was done for the properties provided by universal hashing. The only previous result achieving such a quality-size trade-off is by Nisan and Zuckerman [29]. They deal with the extraction problem in the difficult range $m = \Theta(n)$ (which we cannot handle), via an ingenious construction, following earlier work of Zuckerman [36]. In addition, they applied their extractors to show that poly(S) many random bits add no power at all to space(S) Turing machines. (Actually, they showed how to simulate poly(S) many random bits, in space(S) computations by using $O(S)$ many random coins.)

Srinivasan and Zuckerman [33] have independently discovered a construction similar to our third construction. Their construction is different and its analysis is simpler than the analysis of our construction. Furthermore, they have used such a construction as the main technical tool in reducing the size of extractors, for the range $m = \Theta(n)$.

Subsequently, Zuckerman [37], using ideas from [35, 33], obtained nearly optimal results for the extraction problem in the range $m = \Theta(n)$. This construction has numerous applications which we shall not elaborate here.

We stress that although all the above results improve on our second construction in case $m = \Theta(n)$, our second construction is better in case $n - m \ll n$ (specifically, in case $n - m \leq O(\log 1/\epsilon)$).

1.3 Conventions

Most probabilistic expressions refer to explicitly defined random variables which are typically denoted X, Y, Z, U_n and ζ. In this case we write

$$\text{Prob}(\text{Boolean expression in these random variables})$$

[1] An extractor is a family of functions having the extraction property.

and it is understood that the probability space is the one used in the definition of these random variables. In a few cases, the probabilistic expression also involves a uniformly selected object, such as $f \in F$. In such a case we write

$$\mathrm{Prob}_{f \in F}(\text{Boolean expression in these random variables and in } f)$$

1.4 Organization

We start by recalling the technical tools used in our proofs. The following three sections (Sec. 3–5) are devoted to the corresponding three constructions mentioned above. Each section starts with a brief intuitive summary of the results obtained. Next, comes a formal statement of the result, a description of the construction which achieves it and an analysis of this construction. We conclude each section with a relevant lower bound. In Section 6 we present the new sampling procedure mentioned as an application above.

2 Technical Tools

Universal Hashing and Expanders are used in our first two constructions, whereas Small Bias Probability Spaces are used in the third. Expanders are also used in Section 6.

2.1 Universal Hashing

Loosely speaking, universal families of hashing functions consist of functions operating on the same domain-range pair so that a function uniformly selected in the family maps each pair of points in a pairwise independent and uniform manner. Specifically, a family, $H_{n,m}$, of functions from $\{0,1\}^n$ to $\{0,1\}^m$, is called *universal* if for every $x \neq y \in \{0,1\}^n$ and $\alpha, \beta \in \{0,1\}^m$ it holds

$$\mathrm{Prob}_{h \in H_{n,m}}(h(x) = \alpha \wedge h(y) = \beta) = 2^{-2m}$$

where the probability is taken over all choices of $h \in H_{n,m}$ with uniform probability distribution.

Several efficient families of universal hashing functions are known [9]. The functions in these families can be described using $O(n+m)$ bits and posses an efficient (e.g., polynomial-time and even logspace) evaluating algorithms. For example, linear transformations with Toeplitz[2] matrices require only $n + 2m - 1$ bits. The two main facts we will use about universal hash families are:

Pairwise Independence. The set of random variables $\{h(x) | x \in \{0,1\}^n\}$ defined by a random $h \in H$ are pairwise independent and uniformly distributed in $\{0,1\}^m$.

Leftover Hash Lemma. This fundamental lemma of [21] asserts that a random hash function from a universal family will smooth min-entropy k whenever the range parameter m is smaller than k. More precisely

Lemma 2.1 (Leftover Hash Lemma [21]): *Let X be any random variable on $\{0,1\}^n$ with min-entropy k* (i.e., $\mathrm{Prob}(X = x) \leq 2^{-k}$ for all x's). *Then the distribution $(h, h(X))$, with h chosen at random from $H_{n,m}$, has* (norm-1) *distance $2^{-(k-m)/3}$ from the uniform distribution.*

[2] A Toeplitz matrix, $T = (t_{i,j})$, satisfies $t_{i,j} = t_{i+1,j+1}$, for all i, j.

2.2 Expanders

An (N, d, λ)-expander is a d-regular graph with N vertices so that the absolute value of all eigenvalues (except the biggest one) of its adjacency matrix is bounded by λ. A (d, λ)-family is an infinite sequence of graphs so that the n^{th} graph is a $(2^n, d, \lambda)$-expander. We say that such a family is *efficiently constructible* if there exists a log-space algorithm which given a vertex, v, in the expander and an index $i \in [d] \stackrel{\text{def}}{=} \{1, ..., d\}$, returns the i^{th} neighbor of v. We first recall that for $d = 16$ and some $\lambda < 16$, efficiently constructible $(16, \lambda)$-families do exist (cf., [16])[3].

In our applications we use (parameterized) expanders satisfying $\frac{\lambda}{d} < \alpha$ and $d = \text{poly}(1/\alpha)$, where α is an application-specific parameter. Such (parameterized) expanders are also efficiently constructible. For example, we may obtain them by taking paths of length $O(\log(1/\alpha)$ on an expander as above. Specifically, given a parameter $\alpha > 0$, we obtain an efficiently constructible (D, Λ)-family satisfying $\frac{\Lambda}{D} < \alpha$ and $D = \text{poly}(1/\alpha)$ as follows. We start with a constructible $(16, \lambda)$-family, set $k \stackrel{\text{def}}{=} \log_{16/\lambda}(1/\alpha) = O(\log 1/\alpha)$ and consider the paths of length k in each graph. This yields a constructible $(16^k, \lambda^k)$-family, and both $\frac{\lambda^k}{16^k} < \alpha$ and $16^k = \text{poly}(1/\alpha)$ indeed hold.

To obtain the best constants in Sections 3, 4 and 6.2, one may use efficiently constructible Ramanujan Graphs [24]. Furthermore, using Ramanujan Graphs is essential for our proof of Theorem 6.5. Ramanujan Graphs satisfy $\lambda \leq 2\sqrt{d}$ and so, setting $d = 4/\alpha$, we obtain $\frac{\lambda}{d} < \alpha$, where α is an application-specific parameter. Here some minor technicalities arise as these graphs are given only for certain degrees and certain sizes. Specifically, they can be efficiently constructed for $\frac{1}{2} \cdot q^k \cdot (q^{2k} - 1)$ vertices, where $q \equiv d - 1 \equiv 1 \bmod 4$ and $d - 1$ being a quadratic residue modulo q (cf., [3, Sec. II]). Still, fixing d and ϵ, N, we may find a q satisfying the above conditions with $\frac{1}{2} \cdot q^k \cdot (q^{2k} - 1) \in [(1 - \epsilon) \cdot N, N]$, in time polynomial in $1/\epsilon$. This defines a Ramanujan Graph which is adequate for all our applications (and specifically, for the proof of Theorem 6.5).

The Expander Mixing Lemma. The following lemma is folklore and has appeared in many papers. Loosely speaking, the lemma asserts that expander graphs (for which $d \gg \lambda$) have the property that the fraction of edges between two large sets of vertices approximately equals the product of the densities of these sets. This property is called *mixing*.

Lemma 2.2 (Expander Mixing Lemma): *Let $G = (V, E)$ be an expander graph of degree d and λ be an upper bound on the absolute value of all eigenvalues, save the biggest one, of the adjacency matrix of the graph. Then for every two subsets, $A, B \subseteq V$, it holds*

$$\left| \frac{|(A \times B) \cap E|}{|E|} - \frac{|A|}{|V|} \cdot \frac{|B|}{|V|} \right| \leq \frac{\lambda\sqrt{|A| \cdot |B|}}{d \cdot |V|} < \frac{\lambda}{d}$$

The lemma (and a proof) appears as Corollary 2.5 in [6, Chap. 9].

[3] The are minor technicalities which can be easily fixed. Firstly, the Gaber–Galil expanders are defined (only) for graph sizes which are perfect squares [16]. This suffices for even n's. For odd n's, we may use a trivial modification, such as taking two copies of the graph of size 2^{n-1} and connecting each pair of corresponding vertices. Finally, we add multiple edges so that the degree becomes 16, rather than being 14 for even n's and 15 for odd n's.

The Expander Smoothing Lemma. Random walks on expander graphs are known to increase the entropy of a distribution very fast. That is, if one starts with some (non-uniform) distribution on the vertices of the expander and takes a short random walk then one arrives at a distribution which is closer to uniform. The following lemma refers to the effect of a single random step. It follows easily by the standard techniques of dealing with random walks on expander graphs (cf., [1, 6]).

Lemma 2.3 (Expander Smoothing Lemma): *Let $G=(V,E)$, d and λ be as in the previous lemma. Let X be a random variable, distributed over V, so that* $\mathrm{Prob}(X=v) \leq \frac{K}{|V|}$, *for every $v \in V$, and Y denote the vertex reached from X by following a uniformly chosen edge. Then*

$$\sum_{v \in V} \left| \mathrm{Prob}(Y=v) - \frac{1}{|V|} \right| \leq \frac{\lambda}{d} \cdot \sqrt{K-1}$$

Proof: Let $N \stackrel{\mathrm{def}}{=} |V|$, and let x denote the N-dimensional probability vector defined by X (i.e., $x_i \stackrel{\mathrm{def}}{=} \mathrm{Prob}(X=i)$). Let A denote the Markov process defined by traversing a uniformly selected edge in G; namely, the matrix A is the adjacency matrix of the graph G, normalized by division by d. Denote the eigenvalues of A by $\lambda_1, \ldots, \lambda_N$, and note that $\lambda_1 = 1$ and $|\lambda_i| \leq \frac{\lambda}{d}$, for every $i > 1$. We consider the orthogonal eigenvector basis, $e_1, \ldots, e_N$, where $e_i e_i^\top = \frac{1}{N}$ for each i, $e_1 = (\frac{1}{N}, \ldots, \frac{1}{N})$, and write each vector as a linear combination of the vectors in this basis. Denote by c_i the coefficient of x in the direction of e_i. We start by bounding $\sum_i c_i^2$ as follows

$$\begin{aligned} \sum_i c_i^2 \cdot \frac{1}{N} &= (\sum_i c_i e_i^\top) \cdot (\sum_i c_i e_i^\top)^\top \\ &= x \cdot x^\top \\ &= \sum_i x_i^2 \\ &\leq \frac{N}{K} \cdot \left(\frac{K}{N}\right)^2 \end{aligned}$$

getting $\sum_i c_i^2 \leq K$. It is also easy to see that $c_1 = 1$. We now consider the differences vector, denoted z, representing the deviation of the random variable Y from the uniform distribution.

$$\begin{aligned} z^\top &\stackrel{\mathrm{def}}{=} Ax^\top - e_1^\top \\ &= A(\sum_i c_i e_i)^\top - e_1^\top \\ &= \sum_{i>1} \lambda_i c_i e_i^\top \end{aligned}$$

Recall that the lemma claims an upper bound on the norm-1 of z. Instead, we start by providing a bound on its norm-2:

$$\begin{aligned} \sum_i z_i^2 &= \sum_{i>1} \lambda_i^2 c_i^2 e_i e_i^\top \\ &\leq \frac{1}{N} \cdot \left(\frac{\lambda}{d}\right)^2 \sum_{i>1} c_i^2 \\ &\leq \frac{1}{N} \cdot \left(\frac{\lambda}{d}\right)^2 \cdot (K-1) \end{aligned}$$

Maximizing the sum of the $|z_i|$'s, subject to the above bound, the lemma follows. ■

2.3 Small Probability Spaces with the Small Bias Property

The following definition of small-bias sample spaces implies the informal definition in Section 5. Both definitions are legitimate generalizations of the definition of small-biased sample spaces for the binary case (and indeed they are equivalent for $p = 2$).

Definition 2.4 *Let t be an integer, p be a prime and ω be a p^{th} root of unity (in the complex field). A set $S \subseteq GF(p)^t$ is said to have ϵ* bias (sample space for $GF(p)^t$) *if, for every t-long sequence $(a_1, ..., a_t)$ of elements in $GF(p)$, so that not all a_i's are zero, the expectation of (the magnitude of) $\omega^{\sum_{i=1}^{t} a_i s_i}$, taken over all $(s_1, ..., s_t) \in S$ with uniform distribution, is bounded above by ϵ. That is,*

$$\left\| \text{Exp}_{(s_1,\ldots,s_t)\in S} \left(\omega^{\sum_{i=1}^{t} a_i s_i} \right) \right\| \;\leq\; \epsilon$$

The following theorem, due to G. Even [14], is obtained by generalizing a construction of Alon et. al. [4]. Specifically, Even generalizes the LFSR construction by considering sequences over $GF(p)$ (rather than over $GF(2)$).

Theorem 2.5 [14, 15]: *For every integer t, prime p and $\epsilon > 0$, there exists an efficiently constructible ϵ-bias sample space for $GF(p)^t$ of size $(2t/\epsilon)^2$.*

3 Tiny Families of Functions with Mixing Properties

Recall that a function f is mixing for subsets A and B of the domain if membership in $A \times B$ of a pair $(r, f(r))$, with r being a random element in the domain, occurs roughly as often as it would for a random pair (r, s) of elements. The main result of this section is the explicit construction of an ϵ-mixing family of size poly$(1/\epsilon)$. Here ϵ stands both for distance from truly random behavior, as well as the fraction of bad functions which do not achieve this distance. We state the precise theorem, then describe the construction. We prove that our family has optimal size up to a polynomial, and present an application: saving randomness in the generalized logspace model of [27]. We conclude with a different perspective of this result, advocated by Linial.

3.1 Main result

Theorem 3.1 (The Mixing Family): *Let n be an integer and $\epsilon > 2^{-n/O(1)}$. Then, there exists a family of functions, each mapping $\{0,1\}^n$ to itself, satisfying the following properties.*

- succinctness: *the family contains a polynomial in $\frac{1}{\epsilon}$ number of functions, and each function is represented by a unique string of length $l(\epsilon) = O(\log \frac{1}{\epsilon})$.*
- efficient evaluation: *There exists a logspace algorithm that, on input a description of a function f and a string x, returns $f(x)$.*

- mixing property: *For every two subsets $A, B \subseteq \{0,1\}^n$, all but at most an ϵ fraction of the functions f in the family satisfy*

$$|\mathrm{Prob}(U_n \in A \wedge f(U_n) \in B) - \rho(A)\rho(B)| \leq \epsilon$$

where $\rho(S) \overset{\mathrm{def}}{=} \frac{|S|}{2^n}$ denotes the density of the set S and U_n is a random variable uniformly distributed over $\{0,1\}^n$.

Using Ramanujan Graphs as expanders and Toeplitz matrices as Universal Hashing, in the construction below, one may obtain $l(\epsilon) = 7\log_2(1/\epsilon) + O(1)$.

3.2 The Construction

The construction make used of two basic tools which are frequently used for saving randomness: universal hashing functions and expander graphs (see Section 2). We start by setting the parameters for the expander graph and the universal hashing family to be used.

The expander graph. We use an efficiently constructible expander graph, denoted G, of degree d, second eigenvalue λ, and vertex set $\{0,1\}^n$, so that $\frac{\lambda}{d} \leq \frac{\epsilon}{2}$ and $d = \mathrm{poly}(1/\epsilon)$. For every $i \in [d] \overset{\mathrm{def}}{=} \{1, 2..., d\}$ and $v \in \{0,1\}^n$, denote by $g_i(v)$ the vertex reached by moving along the i^{th} edge of the vertex v.

The universal hashing family. We consider a universal family, denoted H, of hash functions, each mapping $l \overset{\mathrm{def}}{=} 3\log_2(2/\epsilon)$-bit long strings to $[d]$ (where $[d] = \{0,1\}^m$, for some m, as d is a power of 2). Namely, a *uniformly chosen function* $h \in H$ maps each string $\alpha \in \{0,1\}^l$ uniformly into $[d]$ so that every two strings are mapped in an independent manner.

Our construction. We now define the functions in our family, denoted F. For each hashing function $h \in H$, we introduce a function $f \in F$ defined by

$$f(v) \overset{\mathrm{def}}{=} g_{h(\mathrm{lsb}(v))}(v)$$

where $\mathrm{lsb}(v)$ returns the l least significant bits of $v \in \{0,1\}^n$. Namely, $f(v)$ is the vertex reached from v by following the i^{th} edge of v, where i is the image of the l least significant bits of v under the function h. (We remark that our choice of using the l least significant bits is arbitrary and any other efficient partition of $\{0,1\}^n$ into 2^l parts, of approximately the same size, will do.)

3.3 Analysis

The technical tools used in our analysis are the Expander Mixing Lemma (Lemma 2.2) and the pairwise independence of images under Universal Hashing functions.

Clearly, the family F satisfies the succinctness and efficiency requirements (of Theorem 3.1). We now turn to prove that it satisfies the mixing property. Towards this end we fix two arbitrary sets A and B. We first observe that by the Expander Mixing Lemma, it holds that

$$\left|\frac{|\{(x,i) : x \in A \wedge g_i(x) \in B\}|}{d \cdot 2^n} - \rho(A)\rho(B)\right| < \frac{\lambda}{d} \leq \frac{\epsilon}{2} \tag{1}$$

Let

$$e_i(A,B) \stackrel{\rm def}{=} 2^{-n} \cdot |\{x \in A : g_i(x) \in B\}| \tag{2}$$

Thus, Eq. (1) can be rewritten as

$$\left| \frac{1}{d} \cdot \sum_{i=1}^{d} e_i(A,B) - \rho(A)\rho(B) \right| \leq \frac{\epsilon}{2} \tag{3}$$

Overview. Eq. (3) states that $\frac{1}{d}\sum_i e_i(A,B)$ is a good approximation of $\rho(A)\rho(B)$. If, for most $i \in [d]$, each $e_i(A,B)$ were a good approximation to $\rho(A)\rho(B)$ then we would be done. But, we don't know whether this property holds. Instead, we partition A into a small number of subsets, denoted A_α's, associate a random $i_\alpha \in [d]$ with each such A_α and consider how well $\sum_\alpha e_{i_\alpha}(A_\alpha, B)$ approximates $\sum_\alpha \rho(A_\alpha)\rho(B) = \rho(A)\rho(B)$. We show that when the i_α's are uniformly distributed in a pairwise independent manner, as is the case when setting $i_\alpha = h(\alpha)$ for one uniformly chosen $h \in H$, the approximation is good with high probability.

Returning to the formal proof, we consider a partition of A into $L \stackrel{\rm def}{=} 2^l$ subsets so that $A_\alpha = \{x \in A : {\rm lsb}(x) = \alpha\}$, for every $\alpha \in \{0,1\}^l$. We define L random variables, $\zeta_{0^l}, ..., \zeta_{1^l}$, so that ζ_α represents the density of the set $\{x \in A_\alpha : g_{h(\alpha)}(x) \in B\}$ (in $\{0,1\}^n$). Note that the ζ_α's are all determined by the choice of h, and thus the probability space is uniform over all choices of $h \in H$. Alternatively, since ${\rm lsb}(x) = \alpha$ for every $x \in A_\alpha$, we can write

$$\zeta_\alpha = 2^{-n} \cdot |\{x \in A_\alpha : g_{h({\rm lsb}(x))}(x) \in B\}|$$

Observe that the set $\{x \in A : g_{h({\rm lsb}(x))}(x) \in B\}$ can be written as $\dot{\cup}_\alpha \{x \in A_\alpha : g_{h({\rm lsb}(x))}(x) \in B\}$. Thus, $\sum_\alpha \zeta_\alpha = 2^{-n} \cdot |\{x \in A : g_{h({\rm lsb}(x))}(x) \in B\}|$ and the mixing property (to be proven) can be rephrased as asserting

$${\rm Prob}\left(\left| \sum_{\alpha \in \{0,1\}^l} \zeta_\alpha - \rho(A) \cdot \rho(B) \right| > \epsilon \right) \leq \epsilon \tag{4}$$

where the probability space is over all possible choices of $h \in H$ (or, equivalently of $f \in F$). This claim is very appealing since each ζ_α is expected to approximate $\rho(A_\alpha) \cdot \rho(B)$. Indeed, Eq. (4) follows by combining the two items of the next lemma.

Lemma 3.2 *Let the ζ_α's be as above and $I \stackrel{\rm def}{=} \{0,1\}^l$. Then*

$$\left| \sum_{\alpha \in I} {\rm Exp}(\zeta_\alpha) - \rho(A) \cdot \rho(B) \right| < \frac{\epsilon}{2} \tag{5}$$

$${\rm Prob}\left(\left| \sum_{\alpha \in I} \zeta_\alpha - \sum_{\alpha \in I} {\rm Exp}(\zeta_\alpha) \right| > \frac{\epsilon}{2} \right) < \epsilon \tag{6}$$

Proof: Using the fact that $h(\alpha)$ is uniformly distributed on $[d]$ and recalling the definition of ζ_α, we have

$${\rm Exp}(\zeta_\alpha) = \frac{1}{d} \cdot \sum_{i=1}^{d} \frac{|\{x \in A_\alpha : g_i(x) \in B\}|}{2^n}$$

Using $A = \cup_\alpha A_\alpha$ and Eq. (2), we have

$$\begin{aligned}\sum_{\alpha\in I} \mathrm{Exp}(\zeta_\alpha) &= \frac{1}{d}\cdot\sum_{i=1}^{d}\frac{\sum_{\alpha\in I}|\{x\in A_\alpha : g_i(x)\in B\}|}{2^n} \\ &= \frac{1}{d}\cdot\sum_{i=1}^{d} e_i(A,B)\end{aligned}$$

Using Eq. (3), we establish Eq. (5).

Next we use Chebyshev Inequality to prove Eq. (6): here we use the fact that the ζ_α's are pairwise independent (since the $h(\alpha)$'s are pairwise independent).

$$\begin{aligned}\mathrm{Prob}\left(\left|\sum_{\alpha\in I}\zeta_\alpha - \sum_{\alpha\in I}\mathrm{Exp}(\zeta_\alpha)\right| > \frac{\epsilon}{2}\right) &< \frac{\mathrm{Var}(\sum_\alpha \zeta_\alpha)}{(\epsilon/2)^2} \\ &\leq \frac{4\cdot\sum_\alpha \mathrm{Exp}(\zeta_\alpha^2)}{\epsilon^2}\end{aligned}$$

Using $\sum_\alpha \mathrm{Exp}(\zeta_\alpha^2) \leq \sum_\alpha \rho(A_\alpha)^2 \leq L\cdot(\frac{1}{L})^2$, and the definition of L ($= 8/\epsilon^3$), we upper bound the above by $\frac{4}{\epsilon^2}\cdot\frac{1}{L} = \epsilon/2$, and so Eq. (6) follows. ■

3.4 Lower Bound

Theorem 3.3 *A family with mixing property of accuracy ϵ, must have size at least $\sqrt{\frac{4}{\epsilon}}$.*

Proof: Otherwise, let $F = \{f_i : 1\leq i\leq t\}$ be a family of functions over $\{0,1\}^n$, contradicting the claim. We construct a graph with vertex set $\{0,1\}^n$ and edges set $\{(x, f(x)) : x\in\{0,1\}^n \wedge f\in F\}$. Clearly, the graph has an independent set of size N/t, where $N \stackrel{\mathrm{def}}{=} 2^n$. Consequently, there are two sets, A and B, each of cardinality $N/2t$, so that there exists no function $f\in F$ for which both $x\in A$ and $f(x)\in B$. On the other hand, $\rho(A)\cdot\rho(B) = (1/2t)^2$, and the theorem follows (even for the special case of hitting $A\times B$ when $\rho(A)\cdot\rho(B)\geq\epsilon$). ■

3.5 Application to Generalized Random Logspace

In [27], Nisan considered the problem of saving randomness in a context in which m randomized algorithms are executed and their output is fed to an s-space bounded machine which then produces a final Boolean output. (Actually, the problem is not affected if the s-space machine is allowed to have output of length bounded by $O(s)$.) For simplicity, assume that each of the algorithms uses n coin flips. The obvious way of running the entire procedure requires $m\cdot n$ coin flips. In case we are willing to tolerate an ϵ additive error/deviation in the final output, more randomness-efficient solutions are possible. In particular, Nisan showed [27] that the randomness complexity can be decreased to

$$O(\max\{n, s+\log(m/\epsilon)\}\cdot\log m)$$

Replacing the universal hash functions used in [27] by our family of mixing functions, we note that *the above problem can be solved with randomness complexity*

$$n + O((s+\log(m/\epsilon))\cdot\log m)$$

We remark that in many applications $n \gg s+\log(m/\epsilon)$. For these cases, our improvement yields a logarithmic reduction in the randomness complexity.

Remark: Theorem 6.2 follows as a special case of the above (alas with a more complicated construction).

3.6 A Different Perspective

The mixing property of families of functions should not be confused with the mixing property of graphs. Yet, the two are related as we shall see below. We say that a graph has a good mixing property if for every two subsets of vertices the fraction of edges connecting these subsets is approximately equal to the product of the densities of these subsets. Clearly, a family of functions over $\{0,1\}^n$, with good mixing, induces a regular multi-graph[4] with good mixing. The converse is not obvious. Specifically, it was not even known whether the edges of some small degree graph with good mixing property (e.g., an expander) can be so colored that they induce a family of functions with a good mixing property.

Let us try to clarify the nature of this problem. Consider a d-degree expander with vertex-set $V \stackrel{\rm def}{=} \{0,1\}^n$, and some d-coloring of its edges. For every two sets of vertices, A and B, denote by $E_i(A,B)$ the set of edges of color i that connect a vertex in A to a vertex in B. By the Expander Mixing Lemma, it follows that the *average* of $\frac{|E_i(A,B)|}{|V|}$, taken over all $1 \leq i \leq d$, is approximately $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$. The question is whether $\frac{|E_i(A,B)|}{|V|}$ is approximately $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$, *for almost all* $1 \leq i \leq d$. One can easily verify that, in general, the answer is negative. Specifically, for Cayley Graph expanders (e.g., [25, 5, 16, 24]), there are sets A and B for which *there exist no* i such that $\frac{|E_i(A,B)|}{|V|}$ approximates $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$ (e.g., consider the cosets obtained by omitting one generator). The problem raised by Nati Linial was to construct an expander for which the mixing property holds for most colors (and not only on the average).

We resolve the above problem by presenting a transformation which takes an arbitrary (edge-colored) expander and produces an (edge-colored) expanders for which the mixing property holds for most colors (as required above). Our transformation preserves the vertex set and the expansion properties of the original expander, but increases the degree by a polynomial factor (i.e., from d to poly(d)). Although the transformation is not explicitly presented in this paper, it can be easily derived from the description above.

4 Tiny Families Extracting High Min-entropy

Recall that the *extraction* property, for a family of functions each mapping n-bit strings to m-bit strings, means that each subset of $K \cdot 2^m$ strings in $\{0,1\}^n$ is mapped almost uniformly to $\{0,1\}^m$, by all but a small fraction of the functions in the family. We consider the extraction problem in two special cases: the case where m is very small (in the next section) and the case m is very close to n (in this section). Actually, we consider a generalization of the extraction problem to random variables with an upper bound, of $\frac{1}{K \cdot 2^m}$, on the probability function. Such a bound is called *min-entropy* (cf., Chor and Goldreich [10]).

Definition 4.1 (min-entropy): *Let X be a random variable. We say that X has* min-entropy *k if* $\text{Prob}(X = x) \leq 2^{-k}$*, for each x.*

[4] A multi-graph is a graph in which parallel edges are allowed.

Here we treat the case of random variables with min-entropy $n-k$ with $k \ll n$. We construct a family of $\mathrm{poly}(2^k/\epsilon)$ functions mapping $\{0,1\}^n$ to $\{0,1\}^m$, where $m = n - O(k)$. For each such random variable, all but a ϵ fraction of the functions, when applied to it, yield a random variable which is ϵ-close to uniform (in norm-1). Loosely speaking, this means that these functions are able to "smoothen" almost the entire min-entropy; specifically, min-entropy $n-k$ is mapped to almost uniform distribution over the strings of length $n - O(k)$.

In a typical use of this extraction, most notably the applications of the leftover hash lemma, $\epsilon = 2^{-\Omega(k)}$. In these cases the size of our family is $\mathrm{poly}(1/\epsilon)$ which is optimal by the lower bound below.

4.1 Main Result

Theorem 4.2 (Extractors for High Min-Entropy): *Let $k < n$ and $m < n - k$ be integers, and $\epsilon > \max\{2^{-(m-O(k))/O(1)}, 2^{-(n-m-O(k))/O(1)}\}$.* (In particular, $m < n - O(k)$.) *Then, there exists a family of functions, each mapping $\{0,1\}^n$ to $\{0,1\}^m$, satisfying the following properties.*

- succinctness: *the family contains a polynomial in $\frac{2^k}{\epsilon}$ number of functions, and each function is represented by a unique string of length $l(k,\epsilon) = O(k + \log\frac{1}{\epsilon})$.*
- efficient evaluation: *There exists a logspace algorithm that, on input a description of a function f and a string x, returns $f(x)$.*
- extraction property: *For every random variable $X \in \{0,1\}^n$ of min-entropy $n-k$, all but an ϵ fraction of the functions f in the family satisfy*

$$\frac{1}{2} \cdot \sum_{\alpha \in \{0,1\}^m} |\mathrm{Prob}(f(X) = \alpha) - \frac{1}{2^m}| \leq \epsilon$$

That is, we may extract $m = n - O(k) - O(\log(1/\epsilon))$ bits from min-entropy $n-k$, and the extracted distribution is ϵ-close to uniform. Using Ramanujan Graphs as expanders and Toeplitz matrices as Universal Hashing, in the construction below, one may obtain $l(k,\epsilon) = 4k + 20\log_2(1/\epsilon) + O(1)$ and $m = n - 2k - 12\log_2(1/\epsilon) - O(1)$.

4.2 The construction

Again, we use universal hashing functions and expander graphs. This time we use an efficiently constructible expander graph, G, of degree d (power of two), second eigenvalue λ, and vertex set $\{0,1\}^m$, so that $\frac{\lambda}{d} \leq \frac{\epsilon^2}{4 \cdot 2^{k/2}}$ (and $d = \mathrm{poly}(2^k/\epsilon)$). As before, for every $i \in [d] \overset{\mathrm{def}}{=} \{1, 2..., d\}$ and $v \in \{0,1\}^m$, denote by $g_i(v)$ the vertex reached by moving along the i^{th} edge of the vertex v. The universal family, denoted H, contains hash functions each mapping $(n-m)$-bit long strings to $[d]$.

Our construction. We now define the functions in our family, denoted F. For each hashing function $h \in H$, we introduce a function $f \in F$ defined by

$$f(x) \overset{\mathrm{def}}{=} g_{h(\mathrm{lsb}(x))}(\mathrm{msb}(x))$$

where $\mathrm{lsb}(x)$ returns the $n-m$ least significant bits of $x \in \{0,1\}^n$, and $\mathrm{msb}(x)$ returns the m most significant bits of x. Namely, $f(x)$ is the vertex reached from the vertex $v \stackrel{\mathrm{def}}{=} \mathrm{msb}(x)$ by following the i^{th} edge of v, where i is the image of the $n-m$ least significant bits of x under the function h. (Again, our choice of using the $n-m$ least significant bits is arbitrary.)

We remark that one may use any family of extractors with the appropriate parameters instead of the universal family H used above. In fact, in preliminary versions of this work we have used the extractors of [29] in order to derive alternative constructions with size $k^{O(\log(1/\epsilon))}$. However, these alternative constructions are subsumed by Zuckerman's recent work [37].

4.3 Analysis

Despite the apparent similarity to the construction for mixing, the analysis of the current construction is completely different. In particular, it is based on "stronger" technical tools: the Expander Smoothing Lemma and the Leftover Hash Lemma.

Clearly, the family F satisfies the succinctness and efficiency requirements. We now turn to prove that it satisfies the extraction property. We fix an arbitrary random variable $X \in \{0,1\}^n$, of min-entropy $n-k$, and consider the distribution $(f, f(X))$, when f is randomly chosen in F. Once we bound the statistical difference between $(f, f(X))$ and (f, U_m) by ϵ^2, where U_m is the uniform distribution over $\{0,1\}^m$, the theorem follows (by a counting argument).

Lemma 4.3 *Let X and $f \in F$ be as above. Then, the statistical difference between $(f, f(X))$ and (f, U_m) is bounded above by ϵ^2.*

Proof: Let Z be a random variable representing the distribution on the m most significant bits of X; i.e., $Z = \mathrm{msb}(X)$. For each $z \in \{0,1\}^m$, let Y_z be a random variable representing the distribution on $\mathrm{lsb}(X)$ conditioned on $Z = z$; i.e., X is the concatenation of Z and Y_Z. We call *bad* those z's in $\{0,1\}^m$ for which Y_z has 'too small' min-entropy. Namely, for $\delta > 0$ to be fixed later, let the set of bad prefixes be denoted by

$$B_\delta \stackrel{\mathrm{def}}{=} \{z \in \{0,1\}^m : \exists y \text{ s.t. } \mathrm{Prob}(Y_z = y) > \delta\}$$

The reader can easily verify, using the min-entropy bound on X, that

$$\mathrm{Prob}(Z \in B_\delta) < \frac{2^{m-(n-k)}}{\delta} \tag{7}$$

(As otherwise, there exists $z \in B_\delta$ (the most probable $z \in B_\delta$) and $y \in \{0,1\}^{n-m}$ (the most probable y for Y_z) so that $\mathrm{Prob}(X = zy) = \mathrm{Prob}(Z = z) \cdot \mathrm{Prob}(Y_z = y) > \frac{2^{m-(n-k)}}{\delta \cdot 2^m} \cdot \delta = 2^{-(n-k)}$.) Also, it can be verified that for every z

$$\mathrm{Prob}(Z = z) \leq 2^{-(m-k)} \tag{8}$$

(As otherwise, there exists $y \in \{0,1\}^{n-m}$ so that $\mathrm{Prob}(X = zy) > 2^{-(m-k)} \cdot 2^{-(n-m)} = 2^{-(n-k)}$.)
We now turn to bound the statistical difference between the distributions $(f, f(X))$ and (f, U_m), where f is uniformly distributed in F. Denote the statistical difference between distributions D_1 and D_2 by $\Delta[D_1, D_2]$ (i.e., $\Delta[D_1, D_2] \stackrel{\mathrm{def}}{=} \frac{1}{2}\sum_\alpha |\mathrm{Prob}(D_1 = \alpha) - \mathrm{Prob}(D_2 = \alpha)|$). Some useful

inequalities used below are $\Delta[g(D_1), g(D_2)] \leq \Delta[D_1, D_2]$, for every function g, and $\Delta[D_1, D_3] \leq \Delta[D_1, D_2] + \Delta[D_2, D_3]$. We have

$$\begin{aligned} \mathrm{Exp}_{f \in F}(\Delta[(f, f(X)), (f, U_m)]) &= \mathrm{Exp}_{f \in F}(\Delta[f(X), U_m]) && (9) \\ &\leq \mathrm{Exp}_{f \in F}(\Delta[f(X'), U_m]) + \Delta[X, X'] && (10) \end{aligned}$$

where X' is the random variable induced by X conditioned on $Z \notin B_\delta$. By Eq. (7), $\Delta[X, X'] < \frac{2^{-(n-m-k)}}{\delta}$, and it is left only to bound the other term in Eq. (10).

Let A be the matrix representing the transition probabilities in a random step on the graph G; i.e., Ap describes the probability distribution after one random step on the graph G, starting with the distribution p. Here and in the sequel, we abuse notation and refer to random variables and distributions as vectors in the natural manner (i.e., the i^{th} component of the vector p is $p(i)$ and the i^{th} component of the vector X is the probability that $X = i$). Each column in A has d non-zero entries and each such entry holds the value $\frac{1}{d}$. For every $h \in H$, let A_h be the matrix that results from A by modifying the non-zero entries as follows. The i^{th} non-zero entry in column z is changed from $\frac{1}{d}$ to $\mathrm{Prob}(h(Y_z){=}i)$. Note that $A_h Z$ equals $g_{h(Y_Z)}(Z)$ which in turn equals $f(X)$ for the function f associated with the hashing function h. Thus, letting $Z' = \mathrm{msb}(X')$, we get

$$\begin{aligned} \mathrm{Exp}_{f \in F}(\Delta[f(X'), U_m]) &= \mathrm{Exp}_{h \in H}(\Delta[A_h Z', U_m]) && (11) \\ &\leq \Delta[AZ', U_m] + \mathrm{Exp}_{h \in H}(\Delta[A_h Z', AZ']) && (12) \\ &\leq \Delta[Z', Z] + \Delta[AZ, U_m] + \mathrm{Exp}_{h \in H}(\Delta[A_h Z', AZ']) && (13) \end{aligned}$$

The first term in Eq. (13) is bounded by Eq. (7). Fixing $\delta \stackrel{\text{def}}{=} \frac{\epsilon^6}{8d}$ (as the min-entropy bound of good Y_z's) and using the Leftover Hash Lemma (Lemma 2.1) we get, for each $z \notin B_\delta$,

$$\mathrm{Exp}_{h \in H}(\Delta[h(Y_z), \eta]) < \sqrt[3]{\delta d} = \frac{\epsilon^2}{2}$$

where η is uniformly distributed over $\{1, ..., d\}$. Recalling the definition of A_h, this means that the *expected* difference between corresponding entries in the matrices A and A_h is at most $\epsilon^2/2$. Thus, for every probability vector p (and in particular for p corresponding to Z'),

$$\mathrm{Exp}_{h \in H}(\Delta[A_h p, Ap]) < \frac{\epsilon^2}{2}$$

This yields a bound on the third term in Eq. (13). It is left to bound the second term; that is $\Delta[AZ, U_m]$. This is done using the Expander Smoothing Lemma (Lemma 2.3), while relying on the min-entropy bound of Eq. (8). We get

$$\Delta[AZ, U_m] \; < \; \frac{\lambda}{d} \cdot \sqrt{2^k} \; \leq \; \frac{\epsilon^2}{4}$$

Combining all the above bounds, we get

$$\mathrm{Exp}_{f \in F}(\Delta[(f, f(X)), (f, U_m)]) \; < \; 2 \cdot \frac{2^{-(n-m-k)}}{\delta} + \frac{\epsilon^2}{2} + \frac{\epsilon^2}{4} \qquad (14)$$

Recalling that $d = (\frac{2^k}{\epsilon})^c$ and $\delta = \frac{\epsilon^6}{8d} = \frac{\epsilon^{6+c}}{2^{3+ck}}$, and using $n - m - k = 6 + ck + (8 + c) \cdot \log(1/\epsilon)$, the first term in Eq. (14) is bounded by $\epsilon^2/4$, and the lemma follows. ■

4.4 Lower Bound

We conclude by observing that a lower bound of [29] (i.e., [29, Thm 3]) implies that, for $\epsilon = 2^{-\Omega(k)}$, our construction is optimal. This holds even when trying to extract just one bit. Specifically, *any family of functions from $\{0,1\}^n$ to $\{0,1\}$, with extraction property of accuracy $\epsilon < 0.5$ with respect to random variables of min-entropy $n-k \leq n-1$, must have size at least* $\max\{k+1, (1/\epsilon)-1\}$.

5 Tiny Families Extracting Low Min-Entropy

Here we treat the case of random variables with min-entropy k, for $k \ll n$. We construct a family of $\text{poly}(2^k n/\epsilon)$ functions mapping $\{0,1\}^n$ to $\{0,1\}^m$, where $m = \Omega(k)$. (Again, ϵ is the accuracy parameter.) Loosely speaking, this means that these functions are able to "smoothen" a constant fraction of the min-entropy; specifically, min-entropy k is mapped to almost uniform distribution over the strings of length $\Omega(k)$.

5.1 Main Result

Theorem 5.1 (Extractors for Low Min-Entropy): *Let $3m < k < n$ and $\epsilon > 2^{-(k-2m-9)/5}$. There exists a family of functions, each mapping $\{0,1\}^n$ to $\{0,1\}^m$, satisfying the following properties.*

- succinctness: *the family contains a polynomial in $\frac{2^m n}{\epsilon}$ number of functions, and each function is represented by a unique string of length $l(m,\epsilon,n) = O(m + \log\frac{n}{\epsilon})$.*
- efficient evaluation: *There exists a logspace algorithm that, on input a description of a function f and a string x, returns $f(x)$.*
- extraction property: *For every random variable $X \in \{0,1\}^n$ of min-entropy k, all but an ϵ fraction of the functions f in the family satisfy*

$$\frac{1}{2} \cdot \sum_{\alpha\in\{0,1\}^m} |\text{Prob}(f(X)=\alpha) - \frac{1}{2^m}| \leq \epsilon$$

That is, we may extract $m = \frac{1}{2}\cdot(k - 9 - 5\log_2(1/\epsilon))$ bits from min-entropy k, and the extracted distribution is ϵ-close to uniform. In fact, $l(m,\epsilon,n) = 4m + 10\log_2(1/\epsilon) + 2\log_2 n + O(1)$.

5.2 The Construction

We use a construction of small probability spaces with small bias. Intuitively, we consider a prime $p \approx 2^m$ and a construction of $t \stackrel{\text{def}}{=} \frac{n}{m}$ random variables, $(\xi_1, ..., \xi_t)$, each distributed over $GF(p)$ with the following *small bias* property:

> for every t-long sequence $(a_1, ..., a_t)$ of elements in $GF(p)$, so that not all a_i's are zero, the random variable $\sum_{i=1}^t a_i\xi_i$ is almost uniformly distributed over $GF(p)$ (i.e., its statistical distance from uniform is small).

The actual condition is given in Definition 2.4. Typically, such random variables are defined by the uniform distribution over some sample space $S \subseteq GF(p)^t$. We will use such a sample space, S, for bias $\epsilon' = \text{poly}(\epsilon/p)$ (to be specified later). Hence, using the sample space of [4, 14], we have $|S| = \text{poly}(n/\epsilon') = \text{poly}(n \cdot p/\epsilon) = \text{poly}(n2^m/\epsilon)$.

Our construction. The functions in our family, denoted F, correspond to the samples in the small-bias space. Namely, for each $(s_1, ..., s_t) \in S$, we introduce the function $f \in F$ defined by

$$f(x) \stackrel{\rm def}{=} \sum_{i=1}^{t} s_i x_i$$

where x_i is the $i^{\rm th}$ coordinate in $x \in GF(p)^t$ and the arithmetic is in $GF(p)$. The functions, so defined, map $GF(p)^t$ to $GF(p)$. Standard modifications can be applied to derive functions mapping $\{0,1\}^n$ to $\{0,1\}^m$ (recall $p \approx 2^m$ and $p^t \approx 2^n$).

5.3 Analysis

Our analysis uses the fact that the construction of small–bias spaces of [4, 14] satisfies a bound on an exponential sum related to the above intuitive motivation to small–bias spaces (see Definition 2.4). We then prove a Lindsey–like lemma on near-orthogonal vectors and combine it with the bound above to give the result.

Suppose, on the contrary to the extraction property, that for some random variable $X = (X_1, ..., X_t) \in GF(p)^t$ with min-entropy k, and for an ϵ fraction of the f's in F, the random variable $f(X)$ is ϵ-away (in variation distance) from the uniform distribution. Then, it follows that there is a subset $S' \subseteq S$ of $\epsilon \cdot |S|$ sequences so that, for each $\bar{s} \stackrel{\rm def}{=} (s_1, ..., s_t) \in S'$, the random variable $\sum_{i=1}^{t} X_i s_i$ is ϵ-away from the uniform distribution. Namely, for every $\bar{s} \stackrel{\rm def}{=} (s_1, ..., s_t) \in S'$,

$$\frac{1}{2} \cdot \sum_{j=0}^{p-1} \left| {\rm Prob}\left(\sum_{i=1}^{t} X_i s_i = j \right) - \frac{1}{p} \right| \geq \epsilon \qquad (15)$$

Let v be a *zero-sum* p-dimensional vector with norm-1 greater than 2ϵ (here v represents the difference between the probability function of $\sum_{i=1}^{t} X_i s_i$ and the uniform distributional function). Then, the norm-2 of $v \stackrel{\rm def}{=} (v_1, ..., v_p)$ is at least $\sqrt{p \cdot (2\epsilon/p)^2} = 2\epsilon/\sqrt{p}$. Passing to the Fourier basis (i.e., in which the $j^{\rm th}$ vector is $p^{-1/2} \cdot (\omega^j, \omega^{2j}, ..., \omega^{pj})$ with ω being a $p^{\rm th}$ root of unity), we represent v by $\hat{v} = (\hat{v}_1, ..., \hat{v}_p)$, where $\hat{v}_j = \frac{1}{\sqrt{p}} \sum_i \omega^{ij} \cdot v_i$. The norm-2 of v and $\hat{v}$ are equal, and thus the max-norm of $\hat{v}$ is at least $\frac{2\epsilon/\sqrt{p}}{\sqrt{p}} = 2\epsilon/p$. Let $\|c\|$ denote the magnitude of the complex number c. It follows that there exists a j so that $\sqrt{p} \cdot \|\hat{v}_j\| = \|\sum_i v_i \omega^{ji}\| \geq 2\epsilon/\sqrt{p}$ and this j cannot be p (since $\sum_i v_i \omega^{pi} = \sum_i v_i = 0$). Applying this argument to the vector representing the difference between the probability function of $\sum_{i=1}^{t} X_i s_i$ and the uniform distributional function, we conclude that for every $\bar{s} \stackrel{\rm def}{=} (s_1, ..., s_t) \in S'$ there exists some $j \in \{1, ..., p-1\}$, so that

$$\begin{aligned}
\frac{2\epsilon}{\sqrt{p}} &\leq \left\| \sum_i \left({\rm Prob}\left(\sum_{k=1}^{t} X_k s_k = i \right) - \frac{1}{p} \right) \cdot \omega^{ij} \right\| \\
&= \left\| \sum_i {\rm Prob}\left(\sum_{k=1}^{t} X_k s_k = i \right) \cdot \omega^{ij} \right\| \\
&= \left\| {\rm Exp}\left(\omega^{j \sum_{k=1}^{t} X_k s_k} \right) \right\|
\end{aligned}$$

It follows that for some $j \in \{1, ..., p-1\}$ there exists a subset $S'' \subseteq S'$ of cardinality $\frac{|S'|}{p-1} > \frac{\epsilon}{p} \cdot |S|$, so that for every $\bar{s} \stackrel{\rm def}{=} (s_1, ..., s_t) \in S''$

$$\left\| {\rm Exp}\left(\omega^{j \sum_{i=1}^{t} X_i s_i} \right) \right\| \geq \frac{2\epsilon}{\sqrt{p}} \qquad (16)$$

Assume, without loss of generality that $j = 1$. By partitioning these sequences according to the approximate direction of the exponential sum and applying a pigeon-hole argument, we obtain a set $B \subseteq S''$ of cardinality $\Omega(\epsilon|S|/p)$ so that

$$\left\| \frac{1}{|B|} \sum_{(s_1,\ldots,s_t)\in B} \mathrm{Exp}\left(\omega^{\sum_{i=1}^t X_{is_i}}\right) \right\| = \Omega(\epsilon/\sqrt{p})$$

Specifically, we may partition the vectors according quarters of the plain and consider the direction which resides in the middle of the quarter with the largest number of vectors. This yields, $B \subseteq S'$ so that

$$\begin{aligned} |B| &\geq \frac{1}{4}\cdot|S''| \geq \frac{\epsilon}{4p}\cdot|S| && (17)\\ \left\| \frac{1}{|B|} \sum_{(s_1,\ldots,s_t)\in B} \mathrm{Exp}\left(\omega^{\sum_{i=1}^t X_{is_i}}\right) \right\| &\geq \frac{\sqrt{2}}{2}\cdot\frac{2\epsilon}{\sqrt{p}} = \sqrt{2}\cdot\frac{\epsilon}{\sqrt{p}} && (18) \end{aligned}$$

Contradiction follows by contrasting Eq. (18) with the following lemma, which generalizes Lindsey's Lemma (cf., [13, p. 88] and [2]).

Lemma 5.2 (A Generalized Lindsey's Lemma): *Let A be an N-by-M matrix of complex numbers, so that each row has inner-product*[5] *with itself equal to M and each pair of different rows have inner-product bounded* (in magnitude) *by $\epsilon' M$. Let u be an N-dimensional probability vector with each components bounded above by δ, and v be an M-dimensional probability vector with each components being either $\frac{1}{K}$ or zero. Then,*

$$\|uAv^\top\| \leq \sqrt{(\epsilon'+\delta)\cdot\frac{M}{K}}$$

Lindsey's Lemma is obtained from the above by requiring the rows of A to be orthogonal (i.e., $\epsilon' = 0$) and considering only "flat" distributions (i.e., each u_i being either δ or 0).[6]

Proof: Denote, $\Delta \stackrel{\rm def}{=} \|uAv^\top\|$. Then, using Cauchy Schwartz Inequality, we get

$$\begin{aligned} \Delta^2 &\leq (v\cdot v^\top)\cdot((uA)\cdot(uA)^\top)\\ &= \frac{1}{K}\cdot((\sum_i u_iA_i)\cdot(\sum_i u_iA_i)^\top) \end{aligned}$$

where A_i is the $i^{\rm th}$ row of the matrix A and u_i is the $i^{\rm th}$ entry of the vector u. Using the hypothesis concerning the inner-product of the rows of A we obtain the bound

$$\Delta^2 \leq \frac{1}{K}\cdot\left(\sum_{i\neq j} u_iu_j\epsilon' M + \sum_i u_i^2 M\right)$$

[5] Note that inner-product of complex vectors is defined as component-wise complex multiplication of one vector by the conjugate of the other.

[6] The standard formulation refers to matrices with ±1 entries and asserts that the sum of elements in any L-times-K generalized sub-matrix is bounded by $\sqrt{LKM}$. Instead, our formulation bounds the sum normalized by the area of the sub-matrix (i.e., divided by $L\cdot K$, with $L = 1/\delta$).

$$< \frac{M}{K} \cdot \left(\epsilon' \sum_{i,j} u_i u_j + \sum_i u_i^2 \right)$$

Using $\sum_{i,j} u_i u_j = (\sum_i u_i)^2 = 1$ and $\sum_i u_i^2 \leq \sum_i u_i \cdot \delta = \delta$, we get $\Delta^2 \leq \frac{M}{K} \cdot (\epsilon' + \delta)$ and the lemma follows. ■

Contradiction to Eq. (18) follows by considering the p^t-by-$|S|$ matrix with rows corresponding to elements of $GF(p)^t$ and columns corresponding to elements of S. The $(x,s)^{\rm th}$ entry in this matrix consists of $\omega^{\sum_{i=1}^t x_i s_i}$, where $x = (x_1, ..., x_t) \in GF(p)^t$ and $s = (s_1, ..., s_t) \in S$. Let u be a vector describing the probability distribution of the random variable X (i.e., $u_x = \text{Prob}(X = x)$) and $\delta = 2^{-k}$ (the upper bound on probability for X). Let v be the (normalized) vector characterizing the set B (i.e., v_s equals $\frac{1}{|B|}$ if $s \in B$ and 0 otherwise). Note that the inner-product of different rows corresponding to sequences $x = (x_1, ..., x_t)$ and $y = (y_1, ..., y_t)$ equals $\sum_{s \in S} \omega^{\sum_{i=1}^t (x_i - y_i) \cdot s_i}$, which, by construction of the sample space S, has magnitude bounded by $\epsilon'|S|$. Letting $A = (\omega^{\sum_{i=1}^t x_i s_i})_{x,s}$, we have

$$\begin{aligned} \|uAv^\top\| &= \left\| \sum_{x \in GF(p)^t} \sum_{s \in S} u_x \cdot \omega^{\sum_{i=1}^t x_i s_i} \cdot v_s \right\| \\ &= \left\| \sum_{x \in GF(p)^t} \sum_{s \in B} \frac{1}{|B|} \cdot \text{Prob}(X = x) \cdot \omega^{\sum_{i=1}^t x_i s_i} \right\| \\ &= \left\| \sum_{s \in B} \frac{1}{|B|} \cdot \text{Exp}(\omega^{\sum_{i=1}^t X_i s_i}) \right\| \end{aligned}$$

We are now ready to apply Lemma 5.2: Here, $M = |S|$, $K = |B| \geq \frac{\epsilon}{4p} \cdot |S|$ (by Eq. (17)), $\delta = 2^{-k} \leq \frac{\epsilon^3}{4p^2}$ (by Theorem's hypothesis and assuming $p \approx 2^m$), and $\epsilon' \stackrel{\rm def}{=} \frac{\epsilon^3}{5p^2}$ (defined here preserving $\epsilon' = \text{poly}(\epsilon/p)$). Applying the lemma we get

$$\begin{aligned} \left\| \frac{1}{|B|} \sum_{(s_1,\ldots,s_t) \in B} \text{Exp}(\omega^{\sum_{i=1}^t X_i s_i}) \right\| &\leq \sqrt{(\epsilon' + \delta) \cdot \frac{M}{K}} \\ &< \sqrt{\frac{\epsilon^3}{2p^2} \cdot \frac{|S|}{\frac{\epsilon}{4p} \cdot |S|}} \\ &= \sqrt{2} \cdot \frac{\epsilon}{\sqrt{p}} \end{aligned}$$

which contradicts Eq. (18).

There is still a minor technicality to be addressed: how do we achieve a mapping to $\{0,1\}^m$ (rather than to $GF(p)$). This is resolved by letting $p \approx 2^m/\epsilon$ (still $p^t \approx 2^n$) and mapping $GF(p)$ to $\{0,1\}^m$ in the natural manner (i.e., each range element having $\epsilon^{-1} \pm 1$ elements). The increase in p only effects the condition $2^{-k} \leq \frac{\epsilon^3}{4p^2}$ which still holds assuming $p < 2^{m+1}/\epsilon$ and using the hypothesis $\epsilon > 2^{-(k-2m-4)/5}$. The approximate mapping of $GF(p)$ to $\{0,1\}^m$ yields and additional extraction deviation of ϵ (totaling to 2ϵ). Substituting $\epsilon/2$ for ϵ, the theorem follows.

5.4 Lower Bound

To illustrate that our construction is near optimal when $k = O(\log n)$ we recall a lower bound of [29] (already mentioned in Section 4.4). We stress that the bound holds even when trying to extract just one bit.

Theorem 5.3 [29, Thm. 3]: *A family of functions from* $\{0,1\}^n$ *to* $\{0,1\}$*, with extraction property of accuracy* $\epsilon < 0.5$ *with respect to random variables of min-entropy* $k \leq n-1$*, must have size at least* $\max\{n-k+1, (1/\epsilon)-1\}$.

We note that the *BPP* simulation of [33] mentioned in the introduction uses an extracting family for this value of the parameter k.

6 A New Sampling Procedure

In many settings repeated sampling is used to estimate the average value of a huge set of values. Namely, there is a value function ν defined over a huge space, say $\nu : \{0,1\}^n \mapsto [0,1]$, and one wishes to approximate $\bar{\nu} \stackrel{\text{def}}{=} \frac{1}{2^n}\sum_{x\in\{0,1\}^n} \nu(x)$. To this end, one may randomly select a small sample set S and compute $\frac{1}{|S|}\sum_{x\in S}\nu(x)$. Using a sample of $O(1/\epsilon^2)$ uniformly and independently selected points, one gets, with constant probability, an approximation that is within an additive factor of ϵ from the correct average. In fact, a set of $O(1/\epsilon^2)$ points selected in a pairwise-independent and uniform manner yields the same quality of approximation. Whereas generating t totally independent random points in $\{0,1\}^n$ requires $t \cdot n$ unbiased coin flips, one can generate t pairwise-independent random points using only $2 \cdot n$ unbiased coin flips [11]. Using the new family of mixing functions, we were able to reduce the randomness complexity of the approximation problem to $n + O(\log(1/\epsilon))$, while maintaining the number of sample points (up-to a multiplicative constant).

Definition 6.1 (sampler): *A* sampler *is a randomized algorithm that on input parameters* n (length), ϵ (accuracy) *and* δ (error), *and oracle access to* any *function* $\nu : \{0,1\}^n \mapsto [0,1]$, *outputs, with probability at least* $1-\delta$*, a value that is at most* ϵ *away from* $\bar{\nu} \stackrel{\text{def}}{=} \frac{1}{2^n}\sum_{x\in\{0,1\}^n}\nu(x)$*. Namely,*

$$\mathrm{Prob}(|\mathrm{sampler}^{\nu}(n,\epsilon,\delta) - \bar{\nu}| > \epsilon) \; < \; \delta$$

where the probability is taken over the internal coin tosses of the sampler.

Theorem 6.2 *There exists a* $\mathrm{poly}(n, \epsilon^{-1}, \delta^{-1})$*-time sampler which*

- *makes* $O(\frac{1}{\delta\epsilon^2})$ *oracle queries; and*
- *tosses* $n + O(\log(1/\epsilon)) + O(\log(1/\delta))$ *coins.*

Our original proof of Theorem 6.2 used the mixing functions guaranteed by Theorem 3.1. In retrospect, we realized that the construction amounts to uniformly selecting a vertex in an expander having vertex set $\{0,1\}^n$ and degree $\mathrm{poly}(1/\epsilon\delta)$, and averaging over the values observed in a *pairwise independent* $O(1/\epsilon^2\delta)$-long sequence of the neighbors of this vertex. Furthermore, for the special case of Boolean functions, an optimal sampler utilizes a Ramanujan graph of degree $O(1/\epsilon^2\delta)$ and

	lower bound [8]	upper bound [8]	algorithms (this paper)
Boolean functions	$n + \log_2(1/\delta)$ $-2\log_2(1/\epsilon) - \log_2\log_2(1/\delta) - O(1)$	$n + 2\log_2(2/\delta)$	$n + O(\log(1/\delta))$ (Thm. 6.5)
general functions	$n + \log_2(1/\delta)$ $-2\log_2(1/\epsilon) - \log_2\log_2(1/\delta) - O(1)$	$n + 2\log_2(2/\delta)$ $+\log_2\log_2(1/\epsilon)$	$n + O(\log(1/\delta))$ $+O(\log(1/\epsilon))$ (Cor. 6.3)

Figure 1: The randomness complexity of samplers which make $\Theta(\frac{\log(1/\delta)}{\epsilon^2})$ queries. The previous best algorithm (by [7]) had randomness complexity $2n + O(\log(1/\delta))$, provided $\epsilon > 2^{-n}$.

consists of uniformly selecting a single vertex and outputting the average of the function values on *all* its neighbors. Both these constructions are analyzed below.

The sampler guaranteed in Theorem 6.2 is not optimal in its query complexity (specifically, the dependency on δ is bad). However, this can be easily redeemed using a generic method of Bellare et al [7]: Given a sampler of query complexity $q(n,\epsilon,\delta)$ and randomness complexity $r(n,\epsilon,\delta)$ they obtain a sampler having query complexity $O(q(n,\epsilon,0.1)\cdot\log(1/\delta))$ and randomness complexity $r(n,\epsilon,0.1)+O(\log(1/\delta))$. Applying their method to the sampler of Theorem 6.2, we obtain

Corollary 6.3 *There exists a* $\mathrm{poly}(n,\epsilon^{-1},\log(1/\delta))$*-time sampler which*

- *makes* $O(\frac{\log(1/\delta)}{\epsilon^2})$ *oracle queries; and*
- *tosses* $n + O(\log(1/\epsilon)) + O(\log(1/\delta))$ *coins.*

This sampler is optimal (up to a multiplicative factor) in its sample-complexity [8, Thm. 1], and among the samplers with nearly optimal sample complexity it is optimal (up to the additive logarithmic factors) in its randomness-complexity [8, Thm. 2]. Previously, *efficient* samplers with optimal sample-complexity were known only for twice the optimal randomness-complexity [7] (yet, [8] have proved non-constructively that "samplers" with sample and randomness complexities as in the corollary do exist[7]). The known results are summarized in Figure 1.

Remark: The randomness-complexity of the sampler asserted in Corollary 6.3 can be improved from $n+O(\log(1/\epsilon))+O(\log(1/\delta))$ to $n+O(\log(1/\epsilon))+(2+o(1))\cdot\log_2(1/\delta))$. This is done by using a Ramanujan Expander (rather than an arbitrary expander) in the method of Bellare et al [7]. A similar comment holds with respect to Theorem 6.5 (below), where the randomness complexity can be improved to $n+(2+o(1))\cdot\log_2(1/\delta)$ (rather than $n+O(\log(1/\delta))$).

6.1 A Sampler for the Boolean Case

We start by presenting a sampler for the special case of Boolean functions. This simpler sampler has even lower randomness complexity (specifically n instead of $n+O(\log(1/\epsilon))$). Our sampling procedure is exactly the one suggested by Karp, Pippinger and Sipser for hitting a witness set [23], yet the analysis is somewhat more involved. Furthermore, to get an algorithm which samples the universe only on $O(1/\delta\epsilon^2)$ points, it is crucial to use a Ramanujan graph in role of the expander in the Karp-Pippinger-Sipser method.

[7] Actually, the non-constructive upper bound of [8, Cor. 2] is slightly better than the result of Corollary 6.3.

Definition 6.4 (Boolean sampler): *A* Boolean sampler *is a randomized algorithm, denoted A, which satisfies*

$$\mathrm{Prob}(|A^{\nu}(n,\epsilon,\delta)-\bar{\nu}|>\epsilon) \; < \; \delta$$

for every Boolean function $\nu:\{0,1\}^n \mapsto \{0,1\}$.

Theorem 6.5 *There exists a* $\mathrm{poly}(n,\epsilon^{-1},\log(1/\delta))$*-time Boolean sampler which*

- *makes $O(\frac{\log(1/\delta)}{\epsilon^2})$ oracle queries; and*
- *tosses $n+O(\log(1/\delta))$ coins.*

As in the general case, Theorem 6.5 will follow by employing the method of Bellare et. al. [7] to an even simpler sampler asserted in Lemma 6.6 (below). This latter sampling algorithm uses, in an essential way, an efficiently constructible Ramanujan (expander) Graph [24]; namely, d-regular expanders with second eigenvalue, λ, satisfying $\lambda \leq 2\sqrt{d}$ (rather than merely $\lambda < d^{1-\frac{1}{O(1)}}$). Specifically, we use an expander of degree $d = 4/\delta\epsilon^2$ and associate the vertex set of the expander with $\{0,1\}^n$. (This is slightly inaccurate as we do not have explicit constructions of Ramanujan Graphs of size 2^n. Still we can efficiently construct a Ramanujan Graph with N vertices where $(1-\epsilon')\cdot 2^n < N \leq 2^n$. Using this graph in our construction causes us to estimate the average value of ν taken over N of the 2^n possible strings, but this average value is within ϵ' of $\bar{\nu}$.)

The algorithm. The sampling algorithm consists of uniformly selecting a vertex, v, (of the expander) and averaging over the values assigned (by ν) to all the neighbors of v; namely,

$$\tilde{\nu} \stackrel{\mathrm{def}}{=} \frac{1}{d} \sum_{u\in\mathcal{N}(v)} \nu(u)$$

where $\mathcal{N}(v)$ denotes the set of neighbors of vertex v.

Lemma 6.6 *The above sampling algorithm constitutes a Boolean sampler. It makes $\frac{4}{\epsilon^2\delta}$ oracle queries, tosses n coins, and runs in time $\frac{\mathrm{poly}(n)}{\epsilon^2\delta}$.*

Proof: The complexity bounds are obvious from the description of the algorithm. We turn to the analysis of its estimates.

We denote by B the set of *bad* choices for the algorithm; namely, the set of vertices that once selected by the algorithm yield a wrong estimate. That is, $v \in B$ if

$$\left| \frac{1}{d} \sum_{u\in\mathcal{N}(v)} \nu(u) - \bar{\nu} \right| > \epsilon$$

Denote by B' the subset of $v \in B$ for which

$$\frac{1}{d} \sum_{u\in\mathcal{N}(v)} \nu(u) > \bar{\nu} + \epsilon \qquad (19)$$

It follows that each $v \in B'$ has ϵd too many neighbors in the set $A \stackrel{\mathrm{def}}{=} \{u : \nu(u)=1\}$; namely,

$$|\{u\in\mathcal{N}(v) : u\in A\}| > (\rho(A)+\epsilon)\cdot d \qquad (20)$$

where $\rho(A) \stackrel{\rm def}{=} \frac{|A|}{N}$ and $N \stackrel{\rm def}{=} 2^n$. Using the Expander Mixing Lemma (Lemma 2.2) ones gets that

$$\begin{aligned} \epsilon \cdot \rho(B') &= \left|\frac{|B'| \cdot (\rho(A)+\epsilon)d}{dN} - \rho(B') \cdot \rho(A)\right| \\ &\leq \left|\frac{|(B' \times A) \cap E|}{|E|} - \frac{|A|}{|V|} \cdot \frac{|B'|}{|V|}\right| \\ &\leq \frac{\lambda}{d} \cdot \frac{\sqrt{|A| \cdot |B'|}}{N} \\ &\leq \frac{2}{\sqrt{d}} \cdot \sqrt{\rho(A) \cdot \rho(B')} \end{aligned}$$

and $\rho(B') \leq \delta \cdot \rho(A)$ follows. Using a similar argument, we can show that $\rho(B \setminus B') \leq \delta \cdot (1-\rho(A))$. Thus, $\rho(B) \leq \delta$ and the claim follows. ■

6.2 Proof of Theorem 6.2

Here we may use weaker expanders than in the simpler sample (above). Specifically, we use an efficiently constructible expander graph of degree d and second eigenvalue λ so that $\frac{\lambda}{d} \leq \sqrt{\frac{\epsilon^5\delta}{64}}$ and $d = \text{poly}(1/\epsilon\delta)$. As above, we associate the vertex set of the expander with $\{0,1\}^n$. Here we also use a sample space for sequences of pairwise independent elements uniformly distributed in $[d]$. In particular, we use sequences of length $m \stackrel{\rm def}{=} \frac{8}{\epsilon^2\delta}$ which can be efficiently generated using $2\log_2 d = O(\log(1/\epsilon\delta))$ random bits (cf., [11]).

The sampler. We select uniformly a vertex v (in the expander graph) and a sequence of m pairwise-independent elements in $[d]$, denoted, $i_1, ..., i_m$. Together these define m sampling points, denoted $u_1, ..., u_j$, where u_j is the $i_j^{\rm th}$ neighbor of v. The estimate of the sampler is merely the average (of ν) over the values at these u_j's; namely,

$$\tilde{\nu} \stackrel{\rm def}{=} \frac{1}{m}\sum_{j=1}^{m} \nu(u_j)$$

Clearly, the complexities are as claimed in Theorem 6.2. It is left to show that for every $\nu : \{0,1\}^n \mapsto [0,1]$,

$$\text{Prob}(|\tilde{\nu} - \bar{\nu}| > \epsilon) \ < \ \delta$$

We first observe that it suffices to evaluate the behavior of the sampler on functions of the form $\nu : \{0,1\}^n \mapsto \{(i-\frac{1}{2}) \cdot \frac{\epsilon}{2} : i = 1,..,2\epsilon^{-1}\}$. Specifically, for every $\nu : \{0,1\}^n \mapsto [0,1]$, there exists a $\mu : \{0,1\}^n \mapsto \{(i-\frac{1}{2}) \cdot \frac{\epsilon}{2} : i = 1,..,2\epsilon^{-1}\}$ so that $|\nu(x) - \mu(x)| \leq \frac{\epsilon}{4}$, $\forall x \in \{0,1\}^n$. Thus, both $|\bar{\nu}(x) - \bar{\mu}(x)| \leq \frac{\epsilon}{4}$ and $|\tilde{\nu}(x) - \tilde{\mu}(x)| \leq \frac{\epsilon}{4}$, and so it suffices to prove the following

Lemma 6.7 *For every* $\mu : \{0,1\}^n \mapsto \{(i-\frac{1}{2}) \cdot \frac{\epsilon}{2} : i = 1,..,2\epsilon^{-1}\}$

$$\text{Prob}\left(|\tilde{\mu} - \bar{\mu}| > \frac{\epsilon}{2}\right) \ < \ \delta$$

Proof: We first mimic the proof of Lemma 6.6 and show that for all but $\delta/2$ of the vertices $v \in \{0,1\}^n$, $\frac{1}{d} \cdot \sum_{u \in \mathcal{N}(v)} \mu(u)$ is within $\epsilon/4$ of $\bar{\mu}$, where $\mathcal{N}(v)$ denotes the set of neighbors of vertex

v. We conclude by recalling that with probability at least $1 - \frac{\delta}{2}$ a pairwise independent sample of m vertices in $\mathcal{N}(v)$ approximates the above up-to $\epsilon/4$.

Suppose, for simplicity, that ϵ is such that $t \stackrel{\rm def}{=} 2\epsilon^{-1}$ is an integer. For every $i = 1, .., t$, let

$$\begin{aligned} A_i &\stackrel{\rm def}{=} \{v : \mu(v) = (i - 0.5) \cdot \epsilon/2\} &(21)\\ B_i &\stackrel{\rm def}{=} \left\{ v : \left| \frac{|\mathcal{N}(v) \cap A_i|}{d} - \rho(A_i) \right| > \frac{\epsilon^2}{4} \right\} &(22) \end{aligned}$$

Observe that if $v \notin \cup_{i=1}^t B_i$ then

$$\begin{aligned} \frac{1}{d} \cdot \sum_{u \in \mathcal{N}(v)} \mu(u) &= \frac{1}{d} \cdot \sum_{i=1}^{t} (i - 0.5) \cdot \frac{\epsilon}{2} \cdot |\mathcal{N}(v) \cap A_i| \\ &= \sum_{i=1}^{t} (i - 0.5) \cdot \frac{\epsilon}{2} \cdot \left(\rho(A_i) \pm \frac{\epsilon^2}{4} \right) \\ &= \bar{\mu} \pm \sum_{i=1}^{2/\epsilon} (i - 0.5) \cdot \frac{\epsilon^3}{8} \\ &= \bar{\mu} \pm \frac{\epsilon}{4} \end{aligned}$$

Thus, the average value of the neighbors of each $v \notin \cup_{i=1}^t B_i$ is within $\frac{\epsilon}{4}$ of the the correct value $\bar{\mu}$.

Next, we use the Expander Mixing Lemma (Lemma 2.2) as in the proof of Lemma 6.6 to derive a bound on the density of each B_i. Analogously, we partition B_i into $B_i^+ \stackrel{\rm def}{=} \{v : |\mathcal{N}(v) \cap A_i| > (\rho(A_i) + \epsilon^2/4) \cdot d\}$ and $B_i^- \stackrel{\rm def}{=} B_i \setminus B_i^+$, and get

$$\begin{aligned} \frac{\epsilon^2}{4} \cdot \rho(B_i^+) &= \left| \frac{|B_i^+| \cdot (\rho(A_i) + \frac{\epsilon^2}{4}) \cdot d}{dN} - \rho(B_i^+) \cdot \rho(A_i) \right| \\ &\leq \left| \frac{|(B_i^+ \times A_i) \cap E|}{|E|} - \frac{|A_i|}{|V|} \cdot \frac{|B_i^+|}{|V|} \right| \\ &\leq \frac{\lambda}{d} \cdot \frac{\sqrt{|A_i| \cdot |B_i^+|}}{N} \\ &\leq \sqrt{\frac{\epsilon^5 \delta}{64}} \cdot \sqrt{\rho(A) \cdot \rho(B_i^+)} \end{aligned}$$

It follows that $\rho(B_i^+) \leq \frac{\epsilon\delta}{4} \cdot \rho(A_i)$ and similarly $\rho(B_i^-) \leq \frac{\epsilon\delta}{4} \cdot (1 - \rho(A_i))$. Thus, $\rho(\cup_{i=1}^t B_i) \leq t \cdot \frac{\epsilon\delta}{4} = \frac{\delta}{2}$. Combined with the above we have

$$\mathrm{Prob}_{v \in \{0,1\}^n} \left(\left| \frac{1}{d} \cdot \sum_{u \in \mathcal{N}(v)} \mu(u) - \bar{\mu} \right| > \frac{\epsilon}{4} \right) \leq \mathrm{Prob}_{v \in \{0,1\}^n} (v \in \cup_{i=1}^t B_i) \leq \frac{\delta}{2} \tag{23}$$

where v is uniformly selected in $\{0,1\}^n$.

Finally, we note that, for every $v \in \{0,1\}^n$, if $u_i, ..., u_m$ are uniformly distributed in a pairwise-independent manner in $\mathcal{N}(v)$ then

$$\mathrm{Prob}_{u_1,...,u_m} \left(\left| \frac{1}{m} \cdot \sum_{i=1}^{m} \mu(u_i) - \frac{1}{d} \cdot \sum_{u \in \mathcal{N}(v)} \mu(u) \right| > \frac{\epsilon}{4} \right) \leq \frac{m \cdot \frac{1}{4}}{(m \cdot \frac{\epsilon}{4})^2} = \frac{\delta}{2} \tag{24}$$

where the equality is due to $m = \frac{8}{\epsilon^2 \delta}$. Combining Eq. (23) and Eq. (24), the lemma follows. ■

Acknowledgments

We are grateful to the anonymous referees for their useful comments, and to Noga Alon for helpful discussions.

References

[1] M. Ajtai, J. Komlos, E. Szemerédi, "Deterministic Simulation in LogSpace", *Proc. 19th STOC*, 1987, pp. 132–140.

[2] N. Alon, "Eigenvalues, Geometric Expanders, Sorting in Rounds and Ramsey Theory", *Combinatorica*, 6 (1986), pp. 231–243.

[3] N. Alon, J. Bruck, J. Naor, M. Naor and R. Roth, "Construction of Asymptotically Good, Low-Rate Error-Correcting Codes through Pseudo-Random Graphs", *IEEE Transactions on Information Theory* **38** (1992), pp. 509–516.

[4] N. Alon, O. Goldreich, J. Hastad, R. Peralta, "Simple Constructions of Almost k-wise Independent Random Variables", *Journal of Random structures and Algorithms*, Vol. 3, No. 3, (1992), pp. 289–304.

[5] N. Alon and V.D. Milman, λ_1, Isoperimetric Inequalities for Graphs and Superconcentrators, *J. Combinatorial Theory, Ser. B* 38 (1985), pp. 73–88.

[6] N. Alon and J.H. Spencer, *The Probabilistic Method*, John Wiley & Sons, Inc., 1992.

[7] M. Bellare, O. Goldreich, and S. Goldwasser "Randomness in Interactive Proofs", *Computational Complexity*, Vol. 4, No. 4 (1993), pp. 319–354.

[8] R. Canetti, G. Even and O. Goldreich, "Lower Bounds for Sampling Algorithms for Estimating the Average", *IPL*, Vol. 53, pp. 17–25, 1995.

[9] L. Carter and M. Wegman, "Universal Classes of Hash Functions", *J. Computer and System Sciences*, Vol. 18, pp. 143–154 (1979).

[10] B. Chor and O. Goldreich, "Unbiased Bits from Sources of Weak Randomness and Probabilistic Communication Complexity", *SIAM J. Comput.*, Vol. 17, No. 2, April 1988, pp. 230–261.

[11] B. Chor and O. Goldreich, "On the Power of Two–Point Based Sampling," *Jour. of Complexity*, Vol 5, 1989, pp. 96–106.

[12] A. Cohen and A. Wigderson, "Dispensers, Deterministic Amplification, and Weak Random Sources", *30th FOCS*, 1989, pp. 14–19.

[13] P. Erdös and J. Spencer, *Probabilistic Methods in Combinatorics*, Academic Press, 1974.

[14] G. Even, "Construction of Small Probability Spaces for Deterministic Simulation", M.Sc. thesis, Computer Science Department, Technion, Haifa, Israel, 1991. (In Hebrew, abstract in English)

[15] G. Even, O. Goldreich, M. Luby, N. Nisan, and B. Veličković, "Approximations of General Independent Distributions", *24th STOC*, pp. 10–16, 1992.

[16] O. Gaber and Z. Galil, "Explicit Constructions of Linear Size Superconcentrators", *JCSS*, **22** (1981), pp. 407-420.

[17] O. Goldreich, H. Krawcyzk and M. Luby, "On the Existence of Pseudorandom Generators", *SIAM J. on Computing*, Vol. 22-6 (Dec. 1993), pp. 1163–1175.

[18] S. Goldwasser and M. Sipser, "Private Coins versus Public Coins in Interactive Proof Systems", *Advances in Computing Research: a research annual*, Vol. 5 (Randomness and Computation, S. Micali, ed.), pp. 73–90, 1989.

[19] R. Impagliazzo and M. Luby, "One-Way Functions are Essential for Complexity Based Cryptography", *30th FOCS*, pp. 230–235, 1989.

[20] R. Impagliazzo and L.A. Levin, "No Better Ways to Generate Hard NP Instances than Picking Uniformly at Random ", *31st FOCS*, pp. 812-821, 1990.

[21] R. Impagliazzo, L.A. Levin, and M.G. Luby, "Pseudorandom Generators from any One-Way Functions", *21st STOC*, pp. 12–24, 1989.

[22] R. Impagliazzo and D. Zuckerman, "How to Recycle Random Bits", *30th FOCS*, 1989, pp. 248-253.

[23] R.M. Karp, N. Pippinger and M. Sipser, "A Time-Randomness Tradeoff", *AMS Conference on Probabilistic Computational Complexity*, Durham, New Hampshire (1982).

[24] A. Lubotzky, R. Phillips, P. Sarnak, "Explicit Expanders and the Ramanujan Conjectures", *Proc. 18th STOC*, 1986, pp. 240-246.

[25] G.A. Margulis, "Explicit Construction of Concentrators", *Prob. Per. Infor.* 9 (4) (1973), 71–80. (In Russian, English translation in *Problems of Infor. Trans.* (1975), 325–332.)

[26] J. Naor and M. Naor, "Small-bias Probability Spaces: Efficient Constructions and Applications", *SIAM J. on Computing*, Vol 22, 1993, pp. 838–856.

[27] N. Nisan, "Pseudorandom Generators for Space Bounded Computation", *Combinatorica* 12 (4), 1992, pp. 449–461.

[28] N. Nisan, "$\mathcal{RL} \subseteq \mathcal{SC}$", *Journal of Computational Complexity* 4, 1994, pp. 1–11.

[29] N. Nisan and D. Zuckerman, "Randomness is Linear in Space", to appear in *JCSS*. Preliminary version in *25th STOC*, pp. 235–244, 1993.

[30] M. Sipser, "A Complexity Theoretic Approach to Randomness", *15th STOC*, 1983, pp. 330–335.

[31] M. Sipser, "Expanders, Randomness or Time vs Space", *Structure in Complexity Theory* (proceedings), 1986.

[32] L. Stockmeyer, "The Complexity of Approximate Counting", *15th STOC*, 1983, pp. 118–126.

[33] A. Srinivasan and D. Zuckerman, "Computing with Very Weak Random Sources", *35th FOCS*, pp. 264–275, 1994.

[34] L. Valiant and V.V. Vazirani, "NP is as Easy as Detecting Unique Solutions", *Theoretical Computer Science*, Vol. 47, 1986, pp. 85–93.

[35] A. Wigderson, D. Zuckerman, " Expanders that Beat the Eigenvalue Bound, Explicit Construction and Applications", *Proc. of the 25th STOC*, pp. 245–251, 1993. To appear in *Combinatorica.*

[36] D. Zuckerman, "Simulating BPP Using a General Weak Random Source," *Algorithmica*, Vol. 16, pp. 367–391, 1996.

[37] D. Zuckerman, "Randomness-Optimal Sampling, Extractors, and Constructive Leader Election", *28th STOC*, 1996, pp. 286–295.