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.

December 11, 1996

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 our constructions are nearly optimal, and mention some applications of the new constructions.

*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 [7]. Though these functions were introduced with data storage application in mind, they found many applications to complexity theory [28, 30, 33, 16, 15, 19, 20, 17, 18, 25, 26, 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 [19] and its precursors, which were key to numerous results, e.g. in saving randomness [20], weak random sources [36], pseudorandom generators [15, 19] and interactive proofs [16]. (Alternative function families with extraction property were previously constructed in [27], with a variety of other applications.)

The *mixing* property is meaningful also in case $m = n$, and in fact it is usually 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 $(a, f(a))$ with a being a random element from the domain, is essentially the same as that of a random pair (a, b) of elements. The prime use of the mixing property is in the logspace pseudorandom generators [25, 26].

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.

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. 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 upto a small *additive term*; and

2. The randomness complexity of Nisan's "generalized logspace" generator [25], 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 optimal extractors for any $m = \Theta(n)$, on which we elaborate below.

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 [27]. 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 [32] have independently discovered a construction similar to our third construction. Actually, their construction and its analysis are simpler than ours. Furthermore, they have used such a construction as the main technical tool in reducing the size of extractors for the range $m = \Theta(n)$ to nearly optimal.

Recently, Zuckerman [37], using ideas from [35, 32], obtained the 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 construction is better in case $n - m \ll n$ (i.e., in case $n - m \leq O(\log 1/\epsilon)$).

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

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

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

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

Organization

We start by recalling the technical tools used in our proofs. The following three sections 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 addition, for the first construction, we describe two applications.

2 Technical Tools

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

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 [7]. 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. The two main facts we will use about universal hash families are:

Pairwise Independence

Lemma 2.1 *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 [19] asserts that a random hash function from a universal family will smooth min-entropy k (recall definition in the previous section) whenever the range parameter m is smaller than k. More precisely

Lemma 2.2 (Leftover Hash Lemma [19]): *Let X be any random variable on $\{0,1\}^n$ with min-entropy k. 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.*

2.2 Expanders

When we talk of expanders we refer to families of d-regular graphs where a bound, denoted $\lambda < d$, is given on the absolute value of all eigenvalues (except the biggest one) of the corresponding adjacency matrix.

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 2.3 (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*

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

For sake of self-containment, we provide a proof in Appendix A

The Expander Smoothing Lemma. The following lemma follows easily by the standard techniques of dealing with random walks on expander graphs.

Lemma 2.4 (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*

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

For sake of self-containment, we provide a proof in Appendix A

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 2.5 *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 norm-2 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.*

The following theorem, due to G. Even [12], is obtained by generalizing a construction of Alon et. al. [3]. Specifically, Even generalizes the LFSR construction by considering sequences over $GF(p)$ (rather than over $GF(2)$).

Theorem 2.6 [12, 13]: *For every integer t, prime p and $\epsilon > 0$, there exists an efficiently constructible ϵ-bias sample space of size $(2t/\epsilon)^2$ for $GF(p)^t$.*

3 Tiny Families of Functions with Mixing Properties

Recall that a function f is mixing for subsets A, B of the domain if membership in $A \times B$ of a pair $(a, f(a))$, with a being a random element in the domain, occurs roughly as often as it would for a random pair (a, b) of elements. The main result of this section is the explicit construction of an ϵ-mixing family of size $\mathrm{poly}(1/\epsilon)$. 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 two applications; one to saving randomness in sampling procedures and the other for saving randomness in the generalized logspace model of [25]. We conclude with a different perspective of this result, advocated by Linial.

3.1 Main result

Theorem 3.1 *Let n be an integer and $\epsilon > 2^{-O(n)}$. 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 $\rho(S) \stackrel{\mathrm{def}}{=} \frac{|S|}{2^n}$ denotes the density of the set S and U_n is a random variable uniformly distributed over $\{0,1\}^n$.

The following corollary will be used for the application to sampling.

Corollary 3.2 *Let F be as in Theorem 3.1. Then, for every subset $S \subseteq \{0,1\}^n$,*

1. *all but at most an 2ϵ fraction of the function pairs (f, f') in $F \times F$ satisfy*

$$|\mathrm{Prob}(f(U_n) \in S \wedge f'(U_n) \in S) - \rho(S)^2| \leq 2\epsilon$$

where ρ and U_n are as in the theorem.

2. *For a uniformly chosen $f \in F$*

$$|\mathrm{Prob}_{f \in F}(f(U_n) \in S) - \rho(S)| \leq 2\epsilon$$

where the probability is over the choices of f and U_n.

Proof: Let $R_f \stackrel{\rm def}{=} \{x : f(x) \in S\}$. Then, for every f and f'

$$\mathrm{Prob}(f(U_n) \in S \wedge f'(U_n) \in S) \;=\; \mathrm{Prob}(U_n \in R_f \wedge f'(U_n) \in S)$$

By the mixing property, for every f and all but at most ϵ fraction of the $f' \in F$ we have

$$|\mathrm{Prob}(f(U_n) \in S \wedge f'(U_n) \in S) - \rho(R_f)\rho(S)| \leq \epsilon$$

Clearly, for all but at most ϵ fraction of the f's we have $|\rho(R_f) - \rho(S)| \leq \epsilon$ and Part 1 follows. Now, let G denote the subset of f's in F for which $|\mathrm{Prob}_{f \in F}(f(U_n) \in S) - \rho(S)| \leq \epsilon$. By the mixing property, $\frac{|G|}{|F|} \geq 1 - \epsilon$. Thus,

$$\begin{aligned} |\mathrm{Prob}_{f \in F}(f(U_n) \in S) - \rho(S)| &\leq \max_{f \in G} |\mathrm{Prob}(f(U_n) \in S) - \rho(S)| \;+\; \mathrm{Prob}_{f \in F}(f \notin G) \\ &\leq \epsilon + \epsilon \end{aligned}$$

and Part 2 follows. ■

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 expander graph, denoted G, of degree d, second eigenvalue λ, and vertex set $\{0,1\}^n$, so that $\frac{\lambda}{d} \leq \frac{\epsilon}{2}$. Such expander graphs are efficiently constructible for $d = \frac{1}{\epsilon^{O(1)}}$ (cf., [14]).[1] Assume, without loss of generality, that d is a power of 2. For every $i \in [d] \stackrel{\rm def}{=} \{1, 2..., d\}$ and $v \in \{0,1\}^n$, denote by $g_i(v)$ the vertex reached by moving along the $i^{\rm th}$ edge of the vertex v.

The universal hashing family. We consider a universal family, denoted H, of hash functions, each mapping $l \stackrel{\rm def}{=} 3\log_2(2/\epsilon)$-bit long strings to $[d]$ (where $[d] = \{0,1\}^m$, for some m). 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

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

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 $i^{\rm th}$ 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.)

[1] Actually, using Ramanujan Graphs, it suffices to have $d = \frac{4}{\epsilon^2}$ (cf., [22]). One may prefer the Gaber–Galil expander since it allows to avoid problems such as generating large primes and embedding $\{0,1\}^n$ in $GF(p)$, for a suitably large prime p.

3.3 Analysis

The main technical tools used in our analysis are the Expander Mixing Lemma (Lemma 2.3) 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 fixed two arbitrary sets A and B. We first observe that by the Expander Mixing Lemma, it holds that

$$|\mathrm{Prob}(U_n \in A \wedge g_\eta(U_n) \in B) - \rho(A)\rho(B)| \;<\; \frac{\lambda}{d} \;\leq\; \epsilon/2 \tag{1}$$

where η is a random variable uniformly distributed over $[d]$, and U_n is uniformly distributed over $\{0,1\}^n$. That is, let $e_i(A,B) \stackrel{\text{def}}{=} 2^{-n} \cdot |\{x \in A : g_i(x) \in B\}|$. Then,

$$\begin{aligned} e_i(A,B) &= \mathrm{Prob}(U_n \in A \wedge g_i(U_n) \in B) && (2)\\ \frac{1}{d} \cdot \sum_{i=1}^{d} e_i(A,B) &= \mathrm{Prob}(U_n \in A \wedge g_\eta(U_n) \in B) && (3) \end{aligned}$$

Thus, Eq. (1) can be rewritten as

$$\left| \frac{1}{d} \cdot \sum_{i=1}^{d} e_i(A,B) - \rho(A)\rho(B) \right| \;\leq\; \epsilon/2 \tag{4}$$

Overview. Eq. (4) 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 $i_\alpha \in [d]$ for 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 note 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 $L \stackrel{\text{def}}{=} 2^l$ subsets so that $A_\alpha = \{x \in A : \mathrm{lsb}(x) = \alpha\}$, 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 choices of h, and thus the probability space is uniform over all choices of $h \in H$. Alternatively, since $\mathrm{lsb}(x) = \alpha$ for every $x \in A_\alpha$, we can write

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

Observe that the set $\{x \in A : g_{h(\mathrm{lsb}(x))}(x) \in B\}$ we can write as $\dot\cup_\alpha \{x \in A_\alpha : g_{h(\mathrm{lsb}(x))}(x) \in B\}$. Thus, the theorem can be rephrased as claiming

$$\mathrm{Prob}\left(\left| \sum_{\alpha \in \{0,1\}^l} \zeta_\alpha - \rho(A) \cdot \rho(B) \right| \geq \epsilon \right) \;\leq\; \epsilon \tag{5}$$

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. (5) follows by combining the two items of the next lemma.

Lemma 3.3 *Let the ζ_α's be as above and $I \stackrel{\rm def}{=} \{0,1\}^l$. Then*

$$\left|\sum_{\alpha\in I} \mathrm{Exp}(\zeta_\alpha) - \rho(A)\cdot\rho(B)\right| \;<\; \frac{\epsilon}{2} \tag{6}$$

$$\mathrm{Prob}\left(\left|\sum_{\alpha\in I}\zeta_\alpha - \sum_{\alpha\in I}\mathrm{Exp}(\zeta_\alpha)\right| > \frac{\epsilon}{2}\right) \;<\; \epsilon \tag{7}$$

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

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

Using $A = \dot\cup_\alpha A_\alpha$ 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. (4), we establish Eq. (6).

Next we use Chebyshev Inequality to prove Eq. (7): here we use the fact that the ζ_α's are pairwise independent (since the $h(\alpha)$'s are pairwise indepedent).

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

Using $\sum_\alpha \mathrm{Exp}(\zeta_\alpha^2) \leq \sum_\alpha \rho(A_\alpha)^2 \leq \sum_\alpha \frac{1}{L}\cdot\rho(A_\alpha) = \frac{1}{L}\cdot\rho(A) \leq \frac{1}{L}$, 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. (7) follows. ■

3.4 Lower Bound

Theorem 3.4 *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 $N \stackrel{\rm def}{=} 2^n$. 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 (in a strong sense!). ■

3.5 Application to Sampling

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 [9]. Using the new family of functions, we further reduce the randomness complexity of the approximation problem to $n + O(\log(1/\epsilon))$, while almost maintaining the number of sample points.

Definition 3.5 (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 $\bar{\nu}$. Namely,*

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

Theorem 3.6 *There exists a* $\text{poly}(n,\epsilon^{-1},\delta^{-1})$*-time sampler which*

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

The proof of Theorem 3.6 is given below. We remark that samplers for Boolean functions can be obtained in a more direct way; and furthermore, these samplers use only n coin tosses (see Appendix B).

Bellare et al [5] presented a generic method of reducing the dependence of the query complexity of a sampler on the error parameter δ. Given a sampler of query complexity $q(n,\epsilon,\delta)$ and randomness complexity $r(n,\epsilon,\delta)$ they obtain a sampler having query complexity $O(\log(1/\delta)\cdot q(n,\epsilon,0.1))$ and randomness complexity $r(n,\epsilon,0.1) + O(\log(1/\delta))$. Applying this result to the sampler of Theorem 3.6, we obtain

Corollary 3.7 *There exists a* $\text{poly}(n,\epsilon^{-1},\log(1/\delta))$*-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.*

The last sampler is optimal (up to a multiplicative factor) in its sample-complexity, and among the samplers with nearly optimal sample complexity the above is optimal (up to the additive logarithmic factors) in its randomness-complexity [6]. Previously, efficient samplers with optimal sample-complexity were known only for twice the randomness-complexity [5] (yet, [6] have proved non-constructively that "samplers" with sample and randomness complexities as in the corollary do exist[2]). The known results are summarized in Figure 1.

Proof of Theorem 3.6: The idea is to use a sequence of related random sample points. These sample points are generated by selecting a sequence of pairwise independent functions from a family

[2]Actually, the non-constructive upper bound is slightly better than the result of Corollary 3.7.

	lower bound [6]	upper bound [6]	algorithm (this paper)
Boolean functions	$n + \log_2(1/\delta)$ $-2\log_2(1/\epsilon) - O(\log\log(1/\delta))$	$n + 2\log_2(2/\delta)$	$n + O(\log(1/\delta))$ (Thm. B.2)
general functions	$n + \log_2(1/\delta)$ $-2\log_2(1/\epsilon) - O(\log\log(1/\delta))$	$n + 2\log_2(2/\delta)$ $+\log_2\log_2(1/\epsilon)$	$n + O(\log(1/\delta))$ $+O(\log(1/\epsilon))$ (Cor. 3.7)

Figure 1: The randomness complexity of samplers which make $\Theta(\frac{\log(1/\delta)}{\epsilon^2})$ queries.

as in Corollary 3.2 and applying each function to a *single* string that is uniformly selected in $\{0,1\}^n$. (We remark that the constructions of almost k-wise independent sample spaces, and specifically the ones in [24, 3, 13, 12], are of no help here as they would all require $O(n)$ random bits, whereas we aim at using $(1+o(1)) \cdot n$ random bits.)

We begin by considering the special case of Boolean functions; namely, we assume that $\nu : \{0,1\}^n \mapsto \{0,1\}$. Next, we define $S \stackrel{\rm def}{=} \{x : \nu(x) = 1\}$. Hence, the problem reduces to approximating $\rho(S) \stackrel{\rm def}{=} |S|/N$, where $N \stackrel{\rm def}{=} 2^n$. Using a family of functions, F, guaranteed by Corollary 3.2 (while replacing ϵ by ϵ' to be determined later), we know that all but at most a $4\epsilon'$ fraction of the pairs, (f, f'), of functions in F it holds

$$|\mathrm{Prob}(f(U_n)\in S \wedge f'(U_n)\in S) - \rho(S)^2| \;<\; 2\epsilon' \tag{8}$$

$$|\mathrm{Prob}(f(U_n)\in S) \cdot \mathrm{Prob}(f'(U_n)\in S) - \rho(S)^2| \;<\; 2\epsilon' \tag{9}$$

Intuitively, this means that as far as sampling S is concerned, setting $m = \frac{3}{\delta\epsilon^2}$ and generating m sample points by randomly selecting a sequence of pairwise independent functions, $f_1, ..., f_m \in F$, and a single random string $s \in \{0,1\}^n$, is as good as using a pairwise independent sequence of m sample points in $\{0,1\}^n$.

The actual sampling algorithm is now obvious. We pick uniformly in $\{0,1\}^n$ a *seed*, denoted s, and independently of it, we generate an m-long sequence of pairwise independent functions, $f_1, ..., f_m \in F$. (Recall that $m = \frac{2}{\delta\epsilon^2}$ and that the mixing parameter ϵ' of F is yet to be determined.) Our sample is the sequence $s_1, ..., s_m$, where $s_i = f_i(s)$. As our estimate, we output the average of ν over this sample; namely, $\tilde{\nu} \stackrel{\rm def}{=} \frac{1}{m}\sum_{i=1}^m \nu(s_i)$.

To analyze the performance of this algorithm, we use an analysis analogous to the one used for pairwise independent sampling. Namely, we define a sequence of m random variables, $\zeta_1, ..., \zeta_m$, so that

$$\zeta_i = \begin{cases} 1 & \text{if } f_i(U_n) \in S \\ 0 & \text{otherwise} \end{cases}$$

Namely, $\zeta_i = \nu(f_i(U_n))$, where f_i is uniformly chosen in F. Note that unlike the analysis of pairwise independent sampling, here the ζ_i's are not pairwise independent. Each ζ_i has expectation $\bar{\mu} \pm 2\epsilon'$ (due to Item 2 of Corollary 3.2). Using the Chebyshev Inequality and $\epsilon' < \epsilon/4$, we get

$$\begin{aligned} \mathrm{Prob}(|\tilde{\nu} - \bar{\nu}| > \epsilon) \;&<\; \mathrm{Prob}\left(\left|\sum_{i=1}^m \zeta_i - \sum_{i=1}^m \mathrm{Exp}(\zeta_i)\right| > (\epsilon - 2\epsilon')m\right) \\ &<\; \frac{\mathrm{Var}(\sum_i \zeta_i)}{(\epsilon m/2)^2} \end{aligned}$$

$$< \quad \frac{\sum_{i\in[m]} \mathrm{Exp}(\zeta_i^2)}{\epsilon^2 m^2/4} + \frac{\sum_{i\neq j\in[m]} (\mathrm{Exp}(\zeta_i\zeta_j) - \mathrm{Exp}(\zeta_i)\cdot\mathrm{Exp}(\zeta_j))}{\epsilon^2 m^2/4}$$

The first term in the last expression is bounded by $\delta/2$ (as usual), since $\mathrm{Exp}(\zeta_i^2) \leq 1$ and $m = 3/(\delta\epsilon^2)$. To bound the second term note that $\mathrm{Exp}(\zeta_i) = \mathrm{Prob}_{f_i}(U_n)$ and $\mathrm{Exp}(\zeta_i\zeta_j) = \mathrm{Prob}_{f_i,f_j}(f_i(U_n)\in S \wedge f_j(U_n)\in S)$, for every i, j.

Let G denote the set of pairs (f,g) for which Eq. (8) and (9) both hold. Then for every $(f,g)\in G$,

$$\Delta(f,g) \stackrel{\mathrm{def}}{=} |\mathrm{Prob}(f(U_n)\in S \wedge g(U_n)\in S) - \mathrm{Prob}(f(U_n)\in S)\cdot\mathrm{Prob}(g(U_n)\in S)| \;\leq\; 4\epsilon'$$

It follows that

$$\begin{aligned}
\mathrm{Prob}(|\tilde{\nu}-\bar{\nu}|>\epsilon) &< \frac{\delta}{2} + \frac{4}{\epsilon^2 m^2}\cdot\sum_{i\neq j\in[m]} (\mathrm{Exp}(\zeta_i\zeta_j) - \mathrm{Exp}(\zeta_i)\cdot\mathrm{Exp}(\zeta_j)) \\
&< \frac{\delta}{2} + \frac{4}{\epsilon^2 m^2}\cdot\sum_{i\neq j\in[m]} \left(\mathrm{Prob}_{f_i,f_j\in F}((f_i,f_j)\notin G) + \max_{(f_i,f_j)\in G}(\Delta(f_i,f_j))\right) \\
&< \frac{\delta}{2} + \frac{4m^2\cdot(4\epsilon'+4\epsilon')}{\epsilon^2 m^2}
\end{aligned}$$

Using $\epsilon' = \frac{\epsilon^2\delta}{64}$ the above is bounded by δ. Note that $\epsilon' = \mathrm{poly}(\epsilon,\delta)$ implies $\log(1/\epsilon') = O(\log(1/\epsilon)) + O(\log(1/\delta))$, and so our usage of randomness is as required. This concludes the analysis of the simple case of Boolean functions.

We now generalize the proof to deal with arbitrary functions ν, rather than Boolean ones. We use the same algorithm, except for a different setting of the parameter ϵ', and analyze it with more care. The definition of the random variables, ζ_i, is modified as follows. Let $q \stackrel{\mathrm{def}}{=} \lceil\frac{1}{\epsilon}\rceil + 1$. We define $q+1$ sets $S_r \stackrel{\mathrm{def}}{=} \{x : \frac{r-1}{q} \leq \nu(x) < \frac{r}{q}\}$, for $1\leq r\leq q+1$, and set $\zeta_i = \frac{r-1}{q}$ if $f_i(U_n)\in S_r$. We proceed by considering only functions $\nu : \{0,1\}^n \mapsto \{\frac{r-1}{q} : 1\leq r\leq q+1\}$, rounding-up any other function in the obvious manner. This, by itself, introduces an $\epsilon/2$ error in the approximation. As before, for the functions we consider we have $\zeta_i = \nu(f_i(U_n))$.

Following the same ideas as before, we show that for all but a $4q^2\epsilon'$ fraction of the pairs, (f,f'), of functions in F it holds for every $1\leq r, r'\leq q$

$$|\mathrm{Prob}(f(U_n)\in S_r \wedge f'(U_n)\in S_{r'}) - \rho(S_r)\rho(S_{r'})| \;<\; 2\epsilon' \tag{10}$$
$$|\mathrm{Prob}(f(U_n)\in S_r)\cdot\mathrm{Prob}(f'(U_n)\in S_{r'}) - \rho(S_r)\rho(S_{r'})| \;<\; 2\epsilon' \tag{11}$$

Here we will set $\epsilon' = \frac{\epsilon^2\delta}{64q^2} = \Theta(\epsilon^4\delta)$. The rest of the analysis now follows as before, since again we will have bound the sum

$$\sum_{i\neq j} \mathrm{Exp}(\zeta_i\zeta_j) - \mathrm{Exp}(\zeta_i)\cdot\mathrm{Exp}(\zeta_j)$$

by considering separately the pairs (f_i,f_j) for which Eq. (10) and (11) hold and the few for which one of these does not hold. Here the sum is bounded above by $m^2\cdot q^2\cdot 4\epsilon'$, and the theorem follows noting that the algorithm uses $m = O(\delta^{-1}\epsilon^{-2})$ samples and $n+O(\log(1/\epsilon') = n+O(\log(1/\epsilon))+O(\log(1/\delta))$ random bits. ■

Remark: In retrospect, one can describe the construction employed in the proof of Theorem 3.6 as follows. We start with an explicit construction of an expander with vertex set $\{0,1\}^n$ and degree $\text{poly}(1/\epsilon\delta)$. The sampler consists of uniformly selecting a vertex of the expander and considering a pairwise independent $O(1/\epsilon^2\delta)$-long sequence of the neighbors of this vertex. The output of the sampler is merely the average of the function value on these neighbors. Furthermore, for the special case of Boolean functions, an optimal sampler utilizes a Ramanujan graph of degree $O(1/\epsilon^2\delta)$ and consists of unifomly selecting a single vertex and outputting the average of the function values of all its neighbours (see Appendix B).

3.6 Application to Generalized Random Logspace

In [25], 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 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 [25] 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 [25] 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 3.6 follows as a special case case of the above (alas with a more complicated construction).

3.7 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[3] 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 $V \stackrel{\text{def}}{=} \{0,1\}^n$, and some d-coloring of its edges. For every two sets of vertices, A and B, denote by

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

$E_i(A,B)$ 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., [23, 4, 22]), there are sets A and B for which *there exist no* i such that $\frac{|E_i(A,B)|}{|V|}$ approximates $\frac{|A|}{|V|} \cdot \frac{|B|}{|V|}$. 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 [8]).

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.*

Here we treat the case of random variables with min-entropy $n-k$ with $k \ll n$. We construct a family of 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 poly$(1/\epsilon)$ which is optimal by the lower bound we give.

4.1 Main Result

Theorem 4.2 *Let $k < n$, $m < n-k$ and $\epsilon > 2^{-(n-m-O(k))/O(1)}$.* (Typically, $m = n - O(k)$ and $\epsilon = 2^{-\Theta(n-m)}$.) *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*

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

4.2 The construction

Again, we use universal hashing functions and expander graphs. This time we use an expander graph, G, degree d (power of two), second eigenvalue λ, and vertex set $\{0,1\}^m$, so that $\frac{\lambda}{d} \leq \frac{\epsilon^2}{8 \cdot 2^{k/2}}$. (Recall that such an expander can be constructed for $d = \mathrm{poly}(\frac{2^k}{\epsilon})$.) As before, for every $i \in [d] \stackrel{\mathrm{def}}{=} \{1, 2..., d\}$ 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

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

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 [27] 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. 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

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

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

$$\mathrm{Prob}(Z \in B_\delta) < \frac{2^{m-(n-k)}}{\delta} \tag{12}$$

(As otherwise, there exists $z \in B_\delta$ and $y \in \{0,1\}^{n-m}$ 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

$$\mathrm{Prob}(Z = z) \leq 2^{-(m-k)} \tag{13}$$

(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., $\Delta[D_1, D_2] \stackrel{\text{def}}{=} \frac{1}{2}\sum_\alpha |\mathrm{Prob}(D_1 = \alpha) - \mathrm{Prob}(D_2 = \alpha)|$). 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

$$\begin{aligned}
\mathrm{Exp}_{f\in F}(\Delta[(f, f(X)), (f, U_m)]) &= \mathrm{Exp}_{f\in F}(\Delta[f(X), U_m]) &(14)\\
&\leq \mathrm{Exp}_{f\in F}(\Delta[f(X'), U_m]) + \Delta[X, X'] &(15)
\end{aligned}$$

where X' is the random variable induced by X conditioned on $Z \notin B_\delta$. By Eq. (12), $\Delta[X, X'] < \frac{2^{-(n-m-k)}}{\delta}$, and it is left only to bound the other term in Eq. (15).

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

$$\begin{aligned}
\mathrm{Exp}_{f\in F}(\Delta[f(X'), U_m]) &= \mathrm{Exp}_{h\in H}(\Delta[A_h Z', U_m]) &(16)\\
&\leq \Delta[AZ', U_m] + \mathrm{Exp}_{h\in H}(\Delta[A_h Z', AZ']) &(17)\\
&\leq \Delta[Z', Z] + \Delta[AZ, U_m] + \mathrm{Exp}_{h\in H}(\Delta[A_h Z', AZ']) &(18)
\end{aligned}$$

The first term in Eq. (18) is bounded by Eq. (12). Fixing $\delta \stackrel{\text{def}}{=} \frac{\epsilon^6}{8d}$ and using the Leftover Hash Lemma (Lemma 2.2) we get, for each $z \notin B_\delta$,

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

where D 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 induced by 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. (18). It is left to bound the second term; that is $\Delta[AZ, U_m]$. This is done using the Expander Smoothing Lemma (Lemma 2.4), while relying on the min-entropy bound of Eq. (13). We get

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

Combining all the above bounds, we get

$$\mathrm{Exp}_{f\in F}(\Delta[(f, f(X)), (f, U_m)]) \;<\; 2\cdot\frac{2^{-(n-m-k)}}{\delta} + \frac{\epsilon^2}{2} + \frac{\epsilon^2}{4} \tag{19}$$

Recalling that $\delta = \frac{\epsilon^6}{8d}$ and $d = (\frac{2^k}{\epsilon})^c$, and using $n - m - k = 6 + ck + (8+c)\cdot\log(1/\epsilon)$, the first term in Eq. (19) is bounded by $\epsilon^2/4$, and the lemma follows. ■

4.4 Lower Bound

We conclude with the lower bound. It shows, that for $\epsilon = 2^{-\Omega(k)}$, our construction is optimal. We stress that the bound holds even when trying to extract just one bit.

Theorem 4.4 *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 $n-k \le n-1$, must have size at least $\max\{k+1, (1/\epsilon)-1\}$.*

Proof: Let $F = \{f_i : 1 \le i \le t\}$ be a family of Boolean functions as in the hypothesis of the theorem. First, we assume, on the contrary that $t \le k$. Our argument proceeds in t iterations. In the first iteration we consider the function f_1 and omit all the strings $x \in \{0,1\}^n$ which are mapped by f_1 to the value with less preimages. In the i^{th} iteration we omit the strings according to the mapping induced by f_i. Thus, in each iteration, we omit at most half of the remaining strings while preserving that the remaining strings have the same image under each function considered so far. After t iterations we are left with a set B of at least $\frac{2^n}{2^t} \ge 2^{n-k}$ strings such that for every $x, y \in B$ and $f \in F$ it holds that $f(x) = f(y)$. Considering the uniform distribution on B, we derive a contradiction (as long as $\epsilon < 1$).

We now turn to the second inequality. Assume, on the contrary that $t < (1/\epsilon)$. Without loss of generality, we assume that t is odd (otherwise consider $t-1$ of the functions in F). It follows that for every x, there exists a bit σ, so that $\mathrm{Prob}_{f\in F}(f(x)=\sigma) \ge \frac{(t+1)/2}{t} > \frac{1+\epsilon}{2}$. Thus, there exists a bit σ, so that for at least half of the x's (in $\{0,1\}^n$) the above holds. Letting X be a random variable uniformly distributed on these "bad" x's, we get $\mathrm{Exp}_{f\in F}(\mathrm{Prob}(f(X)=\sigma) > \frac{1+\epsilon}{2}$. Since X has min-entropy at least $n-1$ and the family is not ϵ-extracting (for X) we reach a contradiction and the theorem follows. ■

5 Tiny Families Extracting Low Min-Entropy

Here we treat the case of random variables with min-entropy k, with $k \ll n$. 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 *Let $3m < k < n$ and $\epsilon > 2^{-(k-3m)/3}$.* (Typically, $m = \frac{k}{6}$ and $\epsilon = 2^{-m}$.) *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 $l(m, \frac{n}{\epsilon}) = O(m + \log \frac{n}{\epsilon})$.*
- 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*

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

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 2.5. 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 $\epsilon' \stackrel{\text{def}}{=} \frac{\epsilon^3}{p^3}$. (Hence, using the sample space of [3, 12], $|S| = \text{poly}(\frac{n2^m}{\epsilon})$.)

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

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

where x_i is the i^{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 [3, 12] satisfies a bound on an exponential sum related to the above intuitive motivation to small–bias spaces. 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 $X = (X_1, ..., X_t)$ with min-entropy k, for an ϵ fraction of the f's in F, the random variable $f(X)$ is ϵ-away (in norm-1) from the uniform distribution. Then, it follows that there is a subset $S' \subseteq S$ of $\epsilon \cdot |S|$ sequences so that, for each $\bar{s} \stackrel{\text{def}}{=} (s_1, ..., s_t) \in S'$, the random variable $\sum_{i=1}^t X_i s_i$ is ϵ-away from the uniform distribution. Namely, for every $\bar{s} \stackrel{\text{def}}{=} (s_1, ..., s_t) \in S'$,

$$\frac{1}{2} \cdot \sum_{j=0}^{p-1} \left| \text{Prob}\left(\sum_{i=1}^{t} X_i s_i = j \right) - \frac{1}{p} \right| > \epsilon \tag{20}$$

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 $v \stackrel{\text{def}}{=} (v_1, ..., v_p)$ 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^{th} vector is $p^{-1/2} \cdot (\omega^j, \omega^{2j}, ..., \omega^{pj})$ with ω being a p^{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 $\bar{s} \stackrel{\text{def}}{=} (s_1, ..., s_t) \in S'$ there exists some $j \in \{1, ..., p-1\}$, so that

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

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 $\bar{s} \stackrel{\text{def}}{=} (s_1, ..., s_t) \in S''$

$$\left\| \text{Exp}\left(\omega^{j \sum_{i=1}^{t} X_i s_i} \right) \right\| \geq \frac{2\epsilon}{\sqrt{p}} \tag{21}$$

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[4], we obtain a set $B \subseteq S''$ of cardinality $\Omega(\epsilon|S|/p)$ so that

$$\left\| \frac{1}{|B|} \sum_{(s_1, ..., s_t) \in B} \text{Exp}\left(\omega^{\sum_{i=1}^{t} X_i s_i} \right) \right\| = \Omega(\epsilon/\sqrt{p}) \tag{22}$$

[4] E.g., 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, $|B| \geq \frac{1}{4} \cdot |S''|$ and sum bounded below by $\frac{\sqrt{2}}{2} \cdot \frac{2\epsilon}{\sqrt{p}} > \frac{\epsilon}{\sqrt{p}}$.

Contradiction follows by contrasting Eq. (22) with the following lemma, which generalizes Lindsey's Lemma (cf., [11, 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 norm-2) *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

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

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

$$\begin{aligned} \Delta^2 &\leq \frac{1}{K} \cdot \left(\sum_{i \neq j} u_i u_j \epsilon' M + \sum_i u_i^2 M \right) \\ &< \frac{M}{K} \cdot \left(\epsilon' \sum_{i,j} u_i u_j + \sum_i u_i^2 \right) \end{aligned}$$

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. (22) 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,

[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., by $L \cdot K$, with $L = 1/\delta$).

by construction of the sample space S, has norm-2 bounded by $\epsilon'|S|$. Letting $A = (\omega^{\sum_{i=1}^t x_i s_i})_{x,s}$, we have

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

Applying Lemma 5.2 and using the definition of ϵ', δ, M and K (i.e., $\epsilon' = \frac{\epsilon^3}{p^3}$, $\delta = 2^{-k} \leq \epsilon^3 \cdot 2^{-3m} \approx \frac{\epsilon^3}{p^3}$, $|M| = |S|$ and $K = |B| = \Omega(\epsilon|S|/p)$) we get

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

which, for sufficiently large p, contradicts Eq. (22).

There are two minor technicalities to be addressed: how can we assume that p is sufficiently large (i.e., bigger than some unspecified constant which is actually 9) and how do we achive a mapping to $\{0,1\}^m$ (rather than to $GF(p)$). Both issues are 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). We may also assume $\epsilon < 1/9$, and so the theorem follows.

5.4 Lower Bound

To illustrate that this construction is near optimal when $k = O(\log n)$ we restate Theorem 4.4 with the necessary change of parameters.

> A family of functions from $\{0,1\}^n$ to $\{0,1\}$, with extraction property of accuracy $\epsilon < o.5$ 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 [32] mentioned in the introduction indeed uses this construction for this value of the parameter k.

Acknowledgements

We are grateful to the anonymous referees for their useful comments.

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, 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.

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

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

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

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

[8] 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.

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

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

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

[12] 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)

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

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

[15] 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.

[16] S. Goldwasser and M. Sipser, "Private Coins versus Public Coins in Interactive Proof Systems", *18th STOC*, pp. 59–68, 1986.

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

[18] 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.

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

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

[21] R.M. Karp, N. Pippinger and M. Sipser, "A Time-Randomness Tradeoff", *AMS Conference on Probabilistic Computational Complexity*, Durham, New Hampshire (1985).

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

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

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

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

[26] N. Nisan, "$\mathcal{RL} \subseteq \mathcal{SC}$", *Journal of Computational Complexity* 4, 1994.

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

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

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

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

[31] M. Santha and U. Vazirani, "Generating Quasi-Random Sequences from Slightly Random Sources", *JCSS*, Vol. 33, No. 1, pp. 75–87, 1986.

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

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

[34] U. Vazirani and V. Vazirani, "Random Polynomial Time Equal to Semi-Random Polynomial Time", *Proc. 26th FOCS*, pp. 417–428, 1985.

[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.

A Proofs for Section 2

Recall that $G = (V, E)$ is a d-regular (expander) graph and λ is an upper bound on the absolute value of all eigenvalues, save the biggest one, of the adjacency matrix of the graph.

A.1 Proof of Lemma 2.3

Lemma A.1 (Lemma 2.3 restated): *For every two subsets, $A, B \subseteq V$, it holds*

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

Proof: Let $A, B \subseteq V$ be two sets and denote $N \stackrel{\rm def}{=} |V|$, $\rho(A) \stackrel{\rm def}{=} |A|/N$ and $\rho(B) \stackrel{\rm def}{=} |B|/N$. Denote by M the adjacency matrix of the graph G, and let us denote it eigenvalues by $\lambda_1, ..., \lambda_N$, where $|\lambda_i| \geq |\lambda_{i+1}|$. Note that $\lambda_1 = d$, whereas, by the statement of the lemma, $\lambda \geq |\lambda_2|$. Hence, the claim of the lemma is restated as

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

We proceed by bounding the value of $|(A \times B) \cap E|$ (from both directions). To this end we let $\bar{a}$ denote the N-dimensional Boolean vector having 1 in the $i^{\rm th}$ component iff $i \in A$. The vector $\bar{b}$ is defined similarly. Clearly, $|(A \times B) \cap E|$ equals $\bar{a}M\bar{b}^\top$. We consider the orthogonal eigenvector basis, $e_1, ..., e_N$, where $e_i e_i^\top = N$ for each i, and write each vector as a linear combination of the vectors in the basis, denoting by a_i the coefficient of $\bar{a}$ in the direction of e_i (i.e., $\bar{a} = \sum_i a_i e_i$). One can easily verify that $a_1 = \rho(A)$ and $\sum_{i=1}^N a_i^2 = \rho(A)$. Similarly for $\bar{b}$. It now follows that

$$\begin{aligned}
|(A \times B) \cap E| &= \bar{a}M\bar{b}^\top \\
&= \bar{a}M(b_1 e_1^\top + \sum_{i=2}^N b_i e_i^\top) \\
&= d \cdot \rho(B) \cdot |A| + \bar{a}\sum_{i=2}^N \lambda_i b_i e_i^\top \\
&= \rho(B)\rho(A) \cdot dN + \sum_{i=2}^N \lambda_i a_i b_i N \\
&\in \left[\rho(B)\rho(A) \cdot dN \pm \lambda \cdot N \sum_{i=2}^N a_i b_i\right]
\end{aligned}$$

Using $\sum_{i=1}^{N} a_i^2 = \rho(A)$ and $\sum_{i=1}^{N} b_i^2 = \rho(B)$, and applying Cauchy-Schwartz Inequality, we bound $\sum_{i=2}^{N} a_i b_i$ by $\sqrt{\rho(A)\rho(B)}$. The lemma follows. ■

A.2 Proof of Lemma 2.4

Lemma A.2 (Lemma 2.4 restated): *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*

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

Proof: Let $N \stackrel{\text{def}}{=} |V|$, and x denote the N-dimensional probability vector defined by X (i.e., $x_i \stackrel{\text{def}}{=} \mathrm{Prob}(X = i)$). 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 $\lambda_1, ..., \lambda_N$, and note that $\lambda_1 = 1$ and $|\lambda_i| \leq \frac{\lambda}{d}$, for every $i > 1$. We consider the orthogonal eigenvector basis, $e_1, ..., e_N$, where $e_i e_i^\top = \frac{1}{N}$ for each i, $e_1 = (\frac{1}{N}, ..., \frac{1}{N})$, 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

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

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.

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

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:

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

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

B A Simpler Sampler for the Boolean Case

For the case of Boolean functions, a much simpler sampler, meeting the complexity bounds of the sampler presented above, exists. In fact, this simpler sampler has even lower randomness complexity (specifically n instead of $n + O(\log(1/\epsilon))$). Our sampling procedure is exactly the one that was presented by Karp, Pippinger and Sipser for hitting a witness set [21], yet the analysis is somewhat more involved. Furthermore, to get an algorithm which samples the universe only on $O(\delta/\epsilon^2)$ points, it is crucial to use a Ramanujan graph in role of the expander in the Karp-Pippinger-Sipser method. Again, we present a sampler for constant δ and derive the result for general δ using the method of Bellare et. al. [5]. Namely,

Definition B.1 (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 B.2 *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.*

B.1 Construction

As said, the sampling algorithm uses, in an essential way, an explicit construction of a Ramanujan (expander) graph [22]; namely, expanders with second eigenvalue, λ, satisfying $\lambda \leq 2\sqrt{d}$, where d denotes the degree. Specifically, we use an expander of degree $d = 4/\delta\epsilon^2$ and associate the vertex set of the expander with $\{0,1\}^n$.

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

B.2 Analysis

Lemma B.3 *The above 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

$$\frac{1}{d} \sum_{u \in \mathcal{N}(v)} \nu(u) > \bar{\nu} + \epsilon \tag{23}$$

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

$$|\{u \in \mathcal{N}(v) : u \in A\}| > (\rho(A) + \epsilon) \cdot d \tag{24}$$

where $\rho(A) \stackrel{\rm def}{=} \frac{|A|}{N}$ and $N \stackrel{\rm def}{=} 2^n$. Using the Expander Mixing Lemma 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 $\rho(B') \leq \delta \rho(A)$ follows. Using a similar argument, we can show that $\rho(B - B') \leq \delta(1 - \rho(A))$. Thus, $\rho(B) \leq \delta$ and the claim follows. ∎