Randomness-Efficient Sampling within $\mathbf{NC}^1$

Alexander Healy*

March 27, 2007

Abstract

We construct a randomness-efficient *averaging sampler* that is computable by uniform constant-depth circuits with parity gates (i.e., in uniform $\mathbf{AC}^0[\oplus]$). Our sampler matches the parameters achieved by random walks on constant-degree expander graphs, allowing us to apply a variety expander-based techniques within $\mathbf{NC}^1$. For example, we obtain the following results:

- Randomness-efficient error-reduction for uniform probabilistic $\mathbf{NC}^1, \mathbf{TC}^0, \mathbf{AC}^0[\oplus]$ and $\mathbf{AC}^0$: Any function computable by uniform probabilistic circuits with error $1/3$ using r random bits is computable by circuits of the same type with error δ using $r + O(\log(1/\delta))$ random bits.
- An optimal bitwise ϵ-biased generator in $\mathbf{AC}^0[\oplus]$: There exists a $1/2^{\Omega(n)}$-biased generator $G : \{0,1\}^{O(n)} \to \{0,1\}^{2^n}$ for which poly(n)-size uniform $\mathbf{AC}^0[\oplus]$ circuits can compute $G(s)_i$ given $(s,i) \in \{0,1\}^{O(n)} \times \{0,1\}^n$. This resolves a question raised by Gutfreund and Viola (*Random 2004*).
- uniform $\mathbf{BP} \cdot \mathbf{AC}^0 \subseteq$ uniform $\mathbf{AC}^0/O(n)$.

Our sampler is based on the *zig-zag graph product* of Reingold, Vadhan and Wigderson (*Annals of Math 2002*) and as part of our analysis we give an elementary proof of a generalization of Gillman's *Chernoff Bound for Expander Walks* (*FOCS 1994*).

1 Introduction

Over the last three decades, *expander graphs* have found a wide variety of applications in Theoretical Computer Science. They have been used in designing novel algorithms (e.g., [AKS83], [Rei05]), in the study of circuit complexity (e.g., [Val77], [IW97]) and to derandomize probabilistic computation (e.g., [CW89], [IZ89]), just to name a few notable examples from this vast literature.

Many of these applications involve a *random walk* on an expander. That is, we choose a random starting node v in an expander graph G, take a k-step random walk and use the k nodes visited by this walk in some way – often as a substitute for k independently-chosen nodes. Despite its simplicity, this processes has some remarkable sampling properties which we discuss shortly. For the moment, we address the computational efficiency of expanders walks.

*Division of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, `ahealy@fas.harvard.edu`.
Research supported by NSF grant CCR-0205423 and a Sandia Fellowship.

1.1 The Complexity of Walks on Expander Graphs

In applications, one often requires an expander graph that is exponentially large, say on 2^n nodes. In this case, a random walk on the graph is performed using a *strongly explicit* representation – that is, a representation in which each node is identified with an n-bit string and it is possible to efficiently (e.g., in time $\mathrm{poly}(n)$) find all the neighbors of a given node $v \in G$. Several beautiful constructions [Mar73, GG81, LPS88, RVW02] are known of such explicit constant-degree expander graphs of exponential size.

At first glance, the act of taking a random walk on an expander graph seems like an inherently *sequential* process – indeed, each step of the walk seems to rely on the previous step in an essential way. A natural question, therefore, is whether the wealth of expander-based techniques from the literature can be applied within highly-*parallel* models of computation, such as log-depth circuits (i.e., $\mathbf{NC}^1$) or even constant-depth circuits.

The main technical contribution of this work is a *sampler* that is just as good as a random walk on an expander graph (in a sense that is made precise in the next section), but which is computable in parallel time $O(\log n)$, i.e. computable by uniform $\mathbf{NC}^1$ circuits. In fact, our sampler is computable by uniform constant-depth circuits with parity gates (i.e. $\mathbf{AC}^0[\oplus]$), a class that is strictly weaker than $\mathbf{NC}^1$ as it cannot even compute the majority of n bits [Raz87].

1.2 The Properties of Walks on Expander Graphs

We now discuss the important sampling properties of random walks on expander graphs in order to better understand what properties we require of our sampler. A more formal definition of expander graphs is given in Section 2, but for the moment the reader may simply think of an expander graph as a constant-degree undirected graph, G, that is "highly-connected".

A fundamental sampling property of expander walks is the *hitting* property, first shown by Ajtai, Komlós and Szemerédi [AKS87]:

The Hitting Property: For any subset S of half the nodes of G, the probability that a k-step random walk never visits a node in S is at most $2^{-\Omega(k)}$.

This hitting property is quite useful (e.g., to reduce the error of **RP** algorithms), but some applications require an even stronger property, which we call the *strong hitting* property:

The Strong Hitting Property: For any sequence of subsets $S_1, \ldots, S_k$, each consisting of half the nodes of G, the probability that a k-step random walk does not pass through S_i on the i-th step for any $i \in \{1, \ldots, k\}$ is at most $2^{-\Omega(k)}$.

It turns out that this strong hitting property is what is necessary for the randomness-efficient error reduction techniques of [CW89] and [IZ89], the amplification technique of [GIL$^+$90] and for the derandomized XOR Lemma of [IW97].

Clearly, the strong hitting property is a generalization of the (non-strong) hitting property. Another natural generalization of the hitting property is the following, first proved by Gillman [Gil94]:

The Chernoff Bound for Expander Walks: For any subset S of half the nodes of G, the fraction of time that a k-step random walk spends in S is $1/2 \pm \epsilon$ with probability $1 - 2^{-\Omega(\epsilon^2 k)}$.

This Chernoff Bound is quite powerful and has applications to constructing randomness extractors (see [Zuc97]) and to Markov-Chain Monte Carlo algorithms (see [Gil94]); although, it is not clear that it subsumes the *strong* hitting property. The following property, however, generalizes both the strong hitting property *and* the Chernoff bound:

The Strong Chernoff Bound for Expander Walks: Fix a sequence of subsets $S_1, \ldots, S_k$, each consisting of half the nodes of G. Then for a k-step random walk on G, the fraction of indices i such that the i-th step of the walk lands in S_i is $1/2 \pm \epsilon$ with probability $1 - 2^{-\Omega(\epsilon^2 k)}$.

Thus, the Strong Chernoff Bound for Expander Walks subsumes all the aforementioned sampling properties, and it seems to represent the essential abstract property of random walks on expanders that is necessary for most natural applications. This bound has only been proved recently – it follows from the work of Wigderson and Xiao [WX05]. (Although a subsequent manuscript of Wigderson and Xiao [WX06] points out an error in [WX05], this only affects the case of sampling d-dimensional matrices for $d \geq 2$. Their proof remains valid for the case of sampling 1-dimensional matrices, which is all that is needed for the Strong Chernoff Bound stated here.)

In this paper, we give a direct and elementary proof of the Strong Chernoff Bound for Expander Walks (Theorem 1). In contrast to most of the proofs in this area, our proof uses only basic linear algebra and, in particular, does not require any perturbation theory or complex analysis in order to obtain a bound that matches the parameters of Gillman's (non-strong) Chernoff bound.[1] Since this bound is important to our analysis, we give a formal statement before describing our results in more detail. (In the following, a λ-*expander* is a regular graph whose normalized second-largest eigenvalue (in absolute value) is at most λ – see Section 2 for a precise definition.)

Theorem 1 (Implicit, up to constants, in [WX05])**.** *Let G be a regular λ-expander on V and fix a sequence of functions $f_i : V \to [0, 1]$ each with mean $\mu_i = \mathrm{E}_v[f_i(v)]$. If we consider a random walk $v_1, \ldots, v_k$ on G, then for all $\epsilon > 0$,*

$$\Pr\left[\left|\sum_{i=1}^{k} f_i(v_i) - \sum_{i=1}^{k} \mu_i\right| \geq \epsilon k\right] \leq 2e^{-\frac{\epsilon^2(1-\lambda)k}{4}}.$$

In particular, by taking the functions f_i to be the characteristic functions of the sets S_i we obtain the Strong Chernoff Bound for Expander Walks (informally) stated above.

We also give a multiplicative form of the Chernoff bound (Theorem 22) that is sharper than Theorem 1 when the sets we are sampling is small (i.e., when the μ_i are small in the notation of Theorem 1). While Kahale [Kah97] has also improved Gillman's Chernoff bound in this setting, his techniques only

[1] [WX05] also gives a proof of a (strong) Chernoff bound using no perturbation theory but this bound does not match Gillman's. In particular, Theorem A.1 of [WX05] has a cubic dependence on the spectral gap $1 - \lambda$ in the exponent, as opposed to the (optimal) linear dependence; moreover, even when the spectral gap $1 - \lambda$ is constant, the dependence on ϵ is (slightly) worse than quadratic.

address the case of sampling a single set; i.e., they give a non-strong Chernoff bound. As a corollary to the proof of Theorem 1, we obtain a strong Chernoff bound that improves upon Theorem 1 when the μ_i and λ are small (see Theorem 22 and Corollary 23).

1.3 Our Sampler

Our main result is the construction of a *sampler* that is computable by $\mathbf{AC}^0[\oplus]$ circuits and possesses all the "sampling properties" of a random walk on a constant-degree expander graphs of size 2^n. To make this notion precise, we recall the following definition (essentially from [Zuc97]):

Definition 2. *A function* $\Gamma : \{0,1\}^m \to (\{0,1\}^n)^k$ *is said to be a* strong (γ, ϵ)-averaging[2] sampler *if: for any sequence of functions* $f_i : \{0,1\}^n \to [0,1]$ *each with mean* $\mu_i = \mathrm{E}_x[f_i(x)]$,

$$\Pr_s\left[\left|\sum_{i=1}^k f_i(\Gamma(s)_i) - \sum_{i=1}^k \mu_i\right| \le \epsilon k\right] \ge 1 - \gamma.$$

We call m *the* seed-length *of the sampler, and we call* k *the* sample complexity *of the sampler.*

It is not hard to check that Theorem 1 implies that a random walk on a constant-degree expander (where λ is a constant less than 1) of size 2^n is a strong averaging sampler with seed-length $m = n + O(\log(1/\gamma)/\epsilon^2)$ and sample complexity $k = O(\log(1/\gamma)/\epsilon^2)$. Moreover, this sample complexity is known to be optimal up to constant factors, and when $\epsilon = \Omega(1)$ the seed-length is also optimal up to constant factors [CEG95]. Our main theorem is that uniform $\mathbf{AC}^0[\oplus]$ can compute a sampler that is just as good:

Theorem 3. *There exists a strong* (γ, ϵ)*-averaging sampler* $\Gamma : \{0,1\}^m \to (\{0,1\}^n)^k$ *with seed-length* $m = n + O(\log(1/\gamma)/\epsilon^2)$ *and sample complexity* $k = O(\log(1/\gamma)/\epsilon^2)$ *such that* Γ *is computable by* uniform $\mathbf{AC}^0[\oplus]$ *circuits of size* $\mathrm{poly}(n, 1/\epsilon, \log(1/\gamma))$.

On the one hand, Theorem 3 is superior to a random walk of length k on a constant-degree expander of size 2^n in the very low computational complexity of Γ; indeed, we do not know of any constant-degree expander walks computable in such low complexity. On the other hand, the sampler of Theorem 3 is potentially a weaker object than an expander walk: there may exist applications of expander walks in which one cannot simply substitute an arbitrary sampler. We note, however, that many applications of expander walks rely only on the fact that an expander walk is a good sampler; thus, when computational complexity is of the essence, we may employ our sampler in lieu of the expander walk.

As discussed in Section 3.1, the proof of Theorem 3 relies upon the *zig-zag graph product* of [RVW02] to build a sampler in $\mathbf{AC}^0[\oplus]$. In Section 3.3, we also mention an alternate construction of a sampler in $\mathbf{AC}^0[\oplus]$ that is inspired by the paradigm of sampler *composition* [BGG93, Gol97].

Gutfreund and Viola have shown [GV04] that walks on the Margulis/Gabber-Galil expander graph [Mar73, GG81] with 2^n nodes are computable in space $O(\log n)$ (and therefore that logspace has strong

[2] [Zuc97] uses the term "oblivious sampler". We follow [Gol97] and use the more-accurate "averaging sampler".

samplers that match the parameters of Theorem 3). To the best of our knowledge, ours is the first work that implies the existence of such strong samplers within the class $\mathbf{NC}^1$ of log-depth circuits; in fact, our construction is in the strictly-weaker class $\mathbf{AC}^0[\oplus] \subsetneq \mathbf{TC}^0 \subseteq \mathbf{NC}^1 \subseteq \mathbf{L}$.

Since expander walks are a powerful and widely-applicable tool it is not surprising that our sampler construction should have a variety of applications. Indeed, we apply our construction to obtain the new results described in the remainder of this section.

Randomness-Efficient Error Reduction within $\mathbf{NC^1}$ One important application of random walks on expander graphs is in reducing the error of probabilistic algorithms. Such error reduction was achieved for **BPP** by Cohen and Wigderson [CW89] and Impagliazzo and Zuckerman [IZ89]. Bar-Yosef, Goldreich and Wigderson [BYGW99] show how to achieve modest-but-optimal error reduction for probabilistic logspace (i.e., the class **BPL**); this is accomplished by a careful implementation of *short* random walks the Margulis/Gabber-Galil expander that can be computed with *one-way* access to the random bits describing the walk. In contrast, Gutfreund and Viola [GV04] show how to compute *long* random walks on the Margulis/Gabber-Galil expander when given *two-way* access to the random bits describing the walk – this implies randomness-efficient error reduction for the class $\mathbf{BP}\cdot\mathbf{L}$. ($\mathbf{BP}\cdot\mathbf{L}$ refers to probabilistic logspace computation that allows for *two-way* access to the random bits, whereas the result of Bar-Yosef et al. concerns the standard model of probabilistic logspace computation, i.e. **BPL**, which only allows *one-way* access to the random bits. See the survey of Saks [Sak96] for a discussion of the subtleties surrounding different notions of probabilistic space-bounded computation.) As an application of our sampler construction, we obtain analogous error-reduction for a variety of classes below logspace (see Section 2 for the definitions of $\mathbf{BP}\cdot\mathbf{NC}^1$, $\mathbf{BP}\cdot\mathbf{TC}^0$ and $\mathbf{BP}\cdot\mathbf{AC}^0[\oplus]$):

Corollary 4. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by polynomial-size* uniform $\mathbf{BP}\cdot\mathbf{AC}^0[\oplus]$ *(respectively, $\mathbf{BP}\cdot\mathbf{TC}^0$ or $\mathbf{BP}\cdot\mathbf{NC}^1$) circuits with error at most $1/3$ using $r = r(n)$ random bits. Then for any $\delta = \delta(n) > 1/2^{O(\mathrm{poly}(n))}$, f has polynomial-size* uniform $\mathbf{BP}\cdot\mathbf{AC}^0[\oplus]$ *(respectively, $\mathbf{BP}\cdot\mathbf{TC}^0$ or $\mathbf{BP}\cdot\mathbf{NC}^1$) circuits with error at most δ using $r + O(\log(1/\delta))$ random bits.*

Combining our sampler with Nisan's unconditional pseudorandom generator for constant-depth circuits [Nis91], we obtain an even stronger result for $\mathbf{BP}\cdot\mathbf{AC}^0$ (see Section 2 for the definition of $\mathbf{BP}\cdot\mathbf{AC}^0$):

Corollary 5. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by polynomial-size* uniform $\mathbf{BP}\cdot\mathbf{AC}^0$ *circuits with error at most $1/3$ using $r = r(n)$ random bits. Then for any $\delta = \delta(n) > 1/2^{O(\mathrm{poly}(n))}$, f has polynomial-size* uniform $\mathbf{BP}\cdot\mathbf{AC}^0$ *circuits with error at most δ using* $\min\{r, \mathrm{polylog}(n)\} + O(\log(1/\delta))$ *random bits.*

Derandomization with Linear Advice Recently, Fortnow and Klivans [FK06] have proved that $\mathbf{RL} \subseteq \mathbf{L}/O(n)$ – that is, one can derandomize probabilistic logspace computation at the cost of only a linear amount of non-uniform advice. Their approach is based on a clever combination of Nisan's pseudorandom generator for space-bounded computation [Nis92] and the logspace expander walks of Gutfreund and Viola [GV04]. Our techniques yield an analogous result for uniform probabilistic constant-depth circuits:

Corollary 6. uniform $\mathbf{BP} \cdot \mathbf{AC}^0 \subseteq$ uniform $\mathbf{AC}^0/O(n)$.

Ajtai and Ben-Or [ABO84] have shown that nonuniform $\mathbf{BP} \cdot \mathbf{AC}^0 =$ nonuniform $\mathbf{AC}^0$; however, even for derandomizing uniform $\mathbf{BP} \cdot \mathbf{AC}^0$ [Ajt93] their technique seems to require an arbitrary polynomial amount of non-uniform advice. Corollary 6 quantifies the amount of nonuniformity that is necessary to derandomize a probabilistic $\mathbf{AC}^0$ circuit, and therefore can be viewed as a refinement of their result. The same approach, together with a new pseudorandom generator of Viola [Vio05], yields similar results for circuits with a bounded number of parity or majority gates – see Corollary 17 in Section 4.2.

An Optimal Bitwise ϵ-Biased Generator in $\mathbf{AC^0}[\oplus]$ Gutfreund and Viola [GV04] study the complexity of constructing *bitwise*[3] ϵ-biased generators (see Definition 8). They give a construction in uniform $\mathbf{AC}^0[\oplus]$ whose seed-length is optimal for $\epsilon = \Omega(1/\text{poly} \log \log(m))$ (where m is the number of output bits) and sub-optimal for smaller ϵ. Healy and Viola [HV06] give an optimal construction in uniform $\mathbf{TC}^0$ and a sub-optimal construction in uniform $\mathbf{AC}^0[\oplus]$ whose parameters are incomparable to those of [GV04]. In this work, we resolve this question entirely – using our sampler (and [NN90, GV04]), we construct an *optimal* bitwise ϵ-biased generator in uniform $\mathbf{AC}^0[\oplus]$:

Corollary 7. *For every $\epsilon > 0$ and m, there is an ϵ-biased generator $G : \{0,1\}^n \to \{0,1\}^m$ with $n = O(\log m + \log(1/\epsilon))$ such that uniform $\mathbf{AC}^0[\oplus]$ circuits of size $\text{poly}(n, \log m) = \text{poly}(n)$ can compute $G(s)_i$ given $(s,i) \in \{0,1\}^n \times [m]$.*

It is known that ϵ-biased generators require seed length $\Omega(\log m + \log(1/\epsilon))$ [AGHP92], and it can be shown that bitwise ϵ-biased generators achieving the parameters of Corollary 7 require $\mathbf{AC}^0$ circuits of exponential size [GV04, MNT90]. Therefore, the construction of Corollary 7 is tight both in terms of seed-length and computational complexity.

1.4 Organization

The remainder of this paper is organized as follows. Some technical preliminaries are recalled in Section 2. In Section 3 we prove Theorem 3 and also describe an alternate sampler construction. The proofs of the applications described above can be found in Section 4. Section 5 is devoted to proving Theorem 1 as well as an alternate Chernoff bound, and some open questions are discussed in Section 6.

2 Preliminaries

For a positive integer n, we denote the set $\{1, \ldots, n\}$ by $[n]$.

[3] [GV04] calls such generators *explicitly computable.*

2.1 ϵ-Biased Sets and Generators

Small-biased spaces appear in two ways in this work. First, poly-size ϵ-biased sets are used to construct expander graphs on which our sampler construction is based (Lemma 10). Second, one of the applications of our sampler is to build exponential-size ϵ-biased sets that are *bitwise* computable (see the definition below and Corollary 7).

Definition 8. *For $a, b \in \mathbb{Z}_2^m$, let $\langle a, b\rangle_2$ denote the inner product of a and b modulo 2.*

A multi-set $S \subseteq \mathbb{Z}_2^m$ is ϵ-biased *if for all non-zero $y \in \mathbb{Z}_2^m$, $\Pr_{x\in S}[\langle x, y\rangle_2 = 1] \in [1/2 - \epsilon, 1/2 + \epsilon]$.*

An ϵ-biased generator *is a function $g : \{0,1\}^\ell \to \{0,1\}^m$ such that the multi-set $\{g(s) \mid s \in \{0,1\}^\ell\}$ is an ϵ-biased multi-set.*

A bitwise *ϵ-biased generator is a function $g : \{0,1\}^\ell \times [m] \to \{0,1\}$ such that the function $g'(s) = (g(s,1), g(s,2), \ldots, g(s,m))$ is an ϵ-biased generator.*

2.2 Expander Graphs

Informally, expander graphs are sparse-yet-highly-connected graphs. While there are a variety of equivalent notions of graph expansion (see, e.g., [AS00, Gol99, HLW06]), it is most convenient for us to work with the following spectral definition. (Recall that a directed graph is *d-regular* if every node has in-degree and out-degree equal to d, and a directed graph is *regular* if it is d-regular for some d.)

Definition 9. *Let G be a regular directed graph on N nodes with transition matrix P, and let $\mathbf{u} = (1/N, \ldots, 1/N) \in \mathbb{R}^N$ denote the uniform distribution on G. We say that G is a* λ-expander *if*

$$\max_{\substack{x\in\mathbb{R}^N \\ \langle x,\mathbf{u}\rangle=0}} \frac{\|Px\|}{\|x\|} \le \lambda.$$

When G is undirected, this definition is equivalent to the second-largest eigenvalue of P being at most λ in absolute value – see, e.g., [Mih89, Fil91].

We often abuse language and refer to a "λ-expander", when we really mean a "family of $\lambda(n)$-expanders of size $s(n)$" for some function $s(n)$. Also, when we simply refer to an "expander graph", without mention of λ, it is understood that we mean a family of λ-expanders for some constant $\lambda < 1$.

By a *random walk* on a d-regular graph G, we mean the following process: choose a random starting vertex $v_0 \in G$, and for $i = 1, \ldots, k$, let v_i be a uniformly random neighbor of v_{i-1} in G and output $v_1, \ldots, v_k$. Note that we are discarding the starting vertex v_0, although it is easy to see that the distribution is unchanged even if we keep v_0. We prefer this convention as it simplifies our notation and presentation. We also note that such a walk is described by a tuple $(v_0, s_1, \ldots, s_k) \in [|G|] \times [d] \times \cdots \times [d]$, and hence by a string of $\log|G| + O(k \log d)$ bits.

2.3 Circuits

Recall that $\mathbf{NC}^1$ denotes the class of functions $f : \{0,1\}^n \to \{0,1\}$ computable by circuits of size $\text{poly}(n)$ and depth $O(\log n)$ over the basis $\{\wedge, \vee, \neg\}$ (where all the gates have fan-in 2). We also consider three classes of unbounded fan-in *constant-depth circuits* of polynomial size: circuits over the bases $\{\wedge, \vee \neg\}$ (i.e., $\mathbf{AC}^0$), $\{\wedge, \vee, \mathit{Parity}, \neg\}$ (i.e., $\mathbf{AC}^0[\oplus]$), and $\{\wedge, \vee, \mathit{Majority}, \neg\}$ (i.e., $\mathbf{TC}^0$). Unless explicitly stated otherwise, all circuits are of polynomial size and *uniform* – specifically, we adopt the standard of *Dlogtime*-uniformity [BIS90], a notion of uniformity which is even more restrictive than logspace-uniformity and which has become the generally-accepted convention for uniformity in constant-depth circuits. Informally, a circuit is *Dlogtime*-uniform if, given the indices of two gates in the circuit, one can determine the types of the gates and whether they are connected in linear time in the length of the indices (which is logarithmic in the size of the circuit).

When referring to non-uniform circuits, we always indicate this explicitly using *slash* notation: for example, $\mathbf{AC}^0/O(n)$ is the class of boolean functions $f : \{0,1\}^n \to \{0,1\}$ such that there exists a *Dlogtime*-uniform $\mathbf{AC}^0$ circuit family $C_n : \{0,1\}^n \times \{0,1\}^{O(n)} \to \{0,1\}$ for which there is a single advice string a_n of length $O(n)$ such that $C_n(x, a_n) = f(x)$ for all $x \in \{0,1\}^n$.

The probabilistic classes $\mathbf{BP}\cdot\mathbf{AC}^0, \mathbf{BP}\cdot\mathbf{AC}^0[\oplus], \mathbf{BP}\cdot\mathbf{TC}^0$ and $\mathbf{BP}\cdot\mathbf{NC}^1$ are all defined in the natural way: the circuit takes two inputs, one of n bits and one of $r(n)$ random bits for some polynomially-bounded function $r(n)$, and for any fixed input $x \in \{0,1\}^n$, the circuit should correctly compute the function with probability at least $2/3$ over the $r(n)$ random bits.

Recall that $\mathbf{AC}^0 \subsetneq \mathbf{AC}^0[\oplus] \subsetneq \mathbf{TC}^0 \subseteq \mathbf{NC}^1 \subseteq \mathbf{L}$, where the last inclusion holds under logspace uniformity and the separations follow from works by Furst et al. [FSS84] and Razborov [Raz87], respectively (and hold even for non-uniform circuits). Despite these lower-bounds, $\mathbf{AC}^0$ can compute the *approximate* majority of n bits [Ajt93] – in particular, for any constant $\epsilon > 0$, there exists a family of $\mathbf{AC}^0$ circuits that correctly computes the majority function for all inputs with at most a $n/2 - \epsilon n$ ones and for all inputs with at least $n/2 + \epsilon n$ ones. See, e.g., [Hås87, Vol99] for additional background on constant-depth circuits.

3 The Sampler Construction

In this section, we describe our sampler construction and prove Theorem 3. Recall that our goal is to construct a sampler $\Gamma : \{0,1\}^m \to (\{0,1\}^n)^k$ that matches the parameters of random walks on expander graphs. Naturally, one way to achieve this would be to exhibit a family of constant-degree expander graphs on 2^n nodes and show that walks of length k on these expanders can be computed in $\mathbf{AC}^0[\oplus]$ of size $\text{poly}(n,k)$. Unfortunately, we do not know of any such family of expanders. Indeed, Gutfreund and Viola [GV04] observe that $\mathbf{AC}^0[\oplus]$ cannot compute walks on the Margulis/Gabber-Galil expander, and the same argument can easily be extended to rule out the possibility of $\mathbf{AC}^0[\oplus]$ circuits that compute walks on a variety of other natural expander graphs. (Nevertheless, it does seems plausible that $\mathbf{AC}^0[\oplus]$ circuits could compute walks on some constant-degree expander family – see the discussion in Section 6.)

In light of this, we begin instead with a family of expander graphs of degree $\mathrm{poly}(n)$ where walks are computable in $\mathbf{AC}^0[\oplus]$ – note that a walk of length k on such a graph is described by a seed of length $n + O(k \cdot \log n)$ – and then we *derandomize* the walk on this graph to achieve the optimal seed length $n + O(k)$. This derandomization uses random walks on a smaller expander graph, and its analysis is based on the *zig-zag graph product* of [RVW02].

In the sequel, we focus on the case where $k = \Omega(\log n)$ since by [GV04] it is known that $\mathbf{AC}^0$ circuits can compute walks of length $\log n$ on the Margulis/Gabber-Galil graph of size 2^n.

3.1 The Construction

Our first graph, G, is a Cayley graph on the group $\mathbb{Z}_2^n$. Specifically, we construct a $1/n$-biased set $S \subset \mathbb{Z}_2^n$ of size $\mathrm{poly}(n)$ (see Definition 8) and let $\{v, w\}$ be an edge if and only if $v \oplus w \in S$. The following well-known fact guarantees that G has second-largest eigenvalue at most $2/n$ (e.g., see [AR94]).

Lemma 10. *A Cayley graph on $\mathbb{Z}_2^n$ with generators $S \subset \mathbb{Z}_2^n$ is a 2ϵ-expander if and only if S is ϵ-biased.*

Before continuing, let us see how walks on G can be computed in $\mathbf{AC}^0[\oplus]$. First, we note that a $1/n$-biased set S of size $\mathrm{poly}(n)$ can be constructed in $\mathbf{AC}^0$. For instance, we may use the "Powering Construction" of an ϵ-biased generator from [AGHP92] together with the results on field arithmetic of [HV06].[4] (Note that for a non-uniform construction, we could simply hard-wire such an ϵ-biased set into the circuit.)

Next, we observe that the neighbors of a node $v \in \{0,1\}^n$ are the nodes $\{v \oplus g(s) \mid s \in \{0,1\}^\ell\}$, where $g : \{0,1\}^\ell \to \{0,1\}^n$ is the ϵ-biased generator defining the ϵ-biased set S and $\ell = O(\log n)$. Thus, a random walk starting at v is obtained by letting $v_0 = v$ and $v_i = v_{i-1} \oplus g(s_i)$ for randomly chosen seeds $s_i \in \{0,1\}^\ell$, and in particular $v_i = v_0 + \bigoplus_{j=0}^{i} g(s_j)$.

Hence, given the description a walk $(v, s_1, \ldots, s_k) \in \{0,1\}^n \times \{0,1\}^\ell \times \cdots \times \{0,1\}^\ell$, to determine the i-th vertex visited by the walk, the circuit need only compute from each seed s_j (in parallel) the appropriate vector $g(s_j) \in S$ and then compute the sum $v + \bigoplus_{j=1}^i g(s_j)$. This is clearly computable by $\mathbf{AC}^0[\oplus]$ circuits of size $\mathrm{poly}(n,k)$. In fact, the parity gates only appear at the outputs and each parity has fan-in at most $k+1$.

Now we turn to the problem of producing a *pseudorandom* sequence of steps s_j, with the goal of reducing the seed length of a walk on G, while at the same time preserving the sampling properties

[4] Specifically, let $m = \log n$ (assuming that $\log n$ is an integer for simplicity) and consider the finite field $\mathbb{F}_{2^{2m}}$ with 2^{2m} elements (viewed as the ring of polynomials over $\mathbb{F}_2$ modulo an irreducible polynomial of degree $2m$). The generator outputs $2^{4m} = n^4$ vectors $v_{\alpha,\beta}$ of dimension $2^m = n$, indexed by pairs of elements $\alpha, \beta \in \mathbb{F}_{2^{2m}}$, where the i-th bit of $v_{\alpha,\beta}$ is given by $\langle \alpha^i, \beta \rangle_2$. It is shown in [AGHP92] that such a generator has bias less than $2^m/2^{2m} = 1/n$, and it is shown in [HV06] that all the necessary field arithmetic can be carried out in uniform $\mathbf{AC}^0$ of size $\mathrm{poly}(n)$ for this range of parameters.

of such walks. Our approach is motivated by the zig-zag product of Reingold, Vadhan and Wigderson [RVW02]. Roughly speaking, one may interpret their results as saying the following: to derandomize a walk on a graph G of degree d, it suffices to choose the steps in G according to a random walk on a constant-degree expander graph H of size d. (For technical reasons, their result requires the graph H to be the *square* of an expander graph, but we ignore this for the moment.) Specifically, to take a pseudorandom k-step walk in G:

1. Choose a random starting vertex $v_0 \in G$.
2. Choose a random $w_0 \in H$ and take a random walk of length k on H, visiting nodes $w_1, \ldots, w_k$.
3. View $w_1, \ldots, w_k$ as indices in $[d]$ (recalling that $|H| = d$).
4. Use $w_1, \ldots, w_k$ as the steps of a walk (starting at v_0) in G.
5. Output the nodes $v_1, \ldots, v_k \in G$ visited by the walk from Step 4.

Note that the seed length of such a sampler is $|v_0| + (|w_0| + O(k)) = n + \log|H| + O(k) = n + O(k)$ (since we assume $k = \Omega(\log n)$), as desired. Moreover, one can show (using the results of [RVW02]) that the forgoing construction is a strong averaging sampler. What is not clear, however, is how to compute this sampler in $\mathbf{AC}^0[\oplus]$. The reason is that it requires a long walk on the graph H, and while H is small (only poly(n) nodes) compared to G (which has 2^n nodes), we do not know how to take such a long walk (on any constant-degree expander family) in $\mathbf{AC}^0[\oplus]$, or even in $\mathbf{NC}^1$ for that matter.

In order to circumvent this obstacle, we derandomize the walk on G by using many short walks on H, rather than a single long walk.

Construction 11.

1. *Choose a random starting vertex* $v_0 \in G$*.*
2. *Take* $k/\log n$ *independent random walks each of length* $\log n$ *in* H*, where the* i*-th walk visits* $w_1^{(i)}, \ldots, w_{\log n}^{(i)} \in H$*.*
3. *View* $w_1^{(1)}, \ldots, w_{\log n}^{(1)}, w_1^{(2)}, \ldots, w_{\log n}^{(2)}, \ldots, w_1^{(k/\log n)}, \ldots, w_{\log n}^{(k/\log n)}$ *as indices in* $[d]$*.*
4. *Use* $w_1^{(1)}, \ldots, w_{\log n}^{(1)}, \ldots, w_1^{(k/\log n)}, \ldots, w_{\log n}^{(k/\log n)}$ *as the steps of a walk (starting at* v_0*) in* G*.*
5. *Output the nodes* $v_1, \ldots, v_k \in G$ *visited by the walk from Step* 4*.*

This sampler has seed length $|v_0| + (k/\log n) \cdot (\log|H| + O(\log n)) = n + O(k)$ (again, since we assume that $k = \Omega(\log n)$). Furthermore, we show below that this construction is a strong averaging sampler, achieving the same parameters as a random walk on a constant-degree expander graph. Before proving this, however, we observe that this walk is computable in $\mathbf{AC}^0[\oplus]$. Indeed, it is known how to compute walks of length $O(\log n)$ on poly-sized explicit expanders of constant degree in $\mathbf{AC}^0$ [Ajt93, GV04], and thus each of the five steps above is computable in constant depth. (As with the $1/n$-biased set S above, the non-trivial issue here is the uniformity of the circuits; if we only wish to give a nonuniform construction we could simply hard-wire all the possible walks of length $\log n$ into the circuit.)

3.2 The Analysis

We now show that Construction 11 is a strong averaging sampler. In particular, Theorem 3 is a consequence of the following lemma:

Lemma 12. *Let* $H = \tilde{H}^2$ *where* $\tilde{H}$ *is a constant-degree expander graph on* $\mathrm{poly}(n)$ *nodes. Then Construction 11 is a strong* (γ, ϵ)*-averaging sampler with seed length* $n + O(\log(1/\gamma)/\epsilon^2)$ *and sample complexity* $O(\log(1/\gamma)/\epsilon^2)$*.*

Proof. Our proof relies on the zig-zag product of [RVW02], so we briefly recall that construction.

Zig-Zag Product Let G be a regular graph of degree d on vertices V_G whose edges are labeled with the names $1, \ldots, d$ in such a way that no two incident edges share the same label.[5] (Note that under such a labeling, if w is the "i-th neighbor of v", then v is the "i-th neighbor of w" – the graph G, defined above, clearly has this property, as it is a Cayley graph on a group of characteristic 2.) Then if g is a regular graph on vertices V_g where $|V_g| = d$, we may form the *zig-zag product* graph $G \circledz g$ where:

- $G \circledz g$ has vertices $V_G \times V_g$
- $\{(v, w), (v', w')\}$ is an edge if there is an $x \in V_g$ such that (w, x, w') is a path in g and v' is the x-th neighbor of v in G. (Note that the labeling condition on G ensures this is symmetric.)

Thus, if we start at $(v, w) \in G \circledz g$, a step to a random neighbor (v', w') has following form:

- Choose a random neighbor x of w in g.
- Set v' to be the x-th neighbor of v in G.
- Choose a random neighbor w' of x in g.

In particular, if we only consider the V_G-coordinate of a random walk of length ℓ in $G \circledz g$ (starting at a random vertex), it has the same distribution as the following process:

- Choose a random start vertex $v_0 \in V_G$.
- Take a random walk $w_1, w_2 \ldots, w_\ell$ in g^2.
- For $i > 0$, let v_i to be the w_i-th neighbor of v_{i-1} in G.
- Output $v_1, v_2, \ldots, v_\ell$.

Thus, each of of the segments of length $k/\log n$ in our sampler construction corresponds to a random walk on $G \circledz \tilde{H}$, projected onto the V_G-coordinate. But what about the boundaries between these segments? In this case, Construction 11 says we choose a new, entirely-random node of $\tilde{H}$ and then continue the walk on G. This is equivalent to taking a step on $G \circledz K_d$, i.e., the zig-zag product of G with a complete graph (with self-loops) on d nodes. Therefore, the output of our sampler is the projection onto the V_G-coordinate of a random walk on a *time-varying* graph that is $G \circledz \tilde{H}$ most of

[5] The zig-zag product of [RVW02] actually applies in much greater generality; however, this simplification suffices for our application.

the time, and $G ⓩ K_d$ once every $\log n$ steps. We now show that this output satisfies Definition 2 for the desired parameters.

First we note for any function $f : V_G \to [0,1]$ there is a natural *lift* of f to $\hat{f} : V_G \times V_{\tilde{H}} \to [0,1]$, defined by $\hat{f}(v,w) = f(v)$, and it is clear that the lift $\hat{f}$ has the same average as f. Therefore, to conclude that the projection of a random walk yields a strong averaging sampler, it suffices to show that a random walk on the forgoing time-varying graph is a strong averaging sampler. By Remark 21 (following the proof of Theorem 1), it does not matter that the graph is varying over time: as long as the graph is a λ-expander at every point in time, the random walk is a good sampler. Thus, we are left with the task of showing that $G ⓩ \tilde{H}$ and $G ⓩ K_d$ are expanders. For this, we apply the following consequence of the main theorem of [RVW02]:

Lemma 13 ([RVW02], Corollary to Theorem 4.3)**.** *Let G be a regular graph of degree d whose edges are labeled with $1, \dots, d$ in such a way that no two incident edges share the same label, and let g be a regular graph on d nodes. If G is a λ_G-expander and g is a λ_g-expander, then $G ⓩ g$ is a $(\lambda_G + \lambda_g)$-expander.*

By Lemma 10, G is a $2/n$-expander, and by assumption $\tilde{H}$ is a λ-expander for some constant $\lambda < 1$. So by Lemma 13, $G ⓩ \tilde{H}$ is a $(2/n + \lambda)$-expander, i.e. a λ'-expander for some constant $\lambda' < 1$ (when $n > 2/(1-\lambda)$).

It is not hard to see that K_d, the complete graph (with self-loops) on d nodes, is a 0-expander, and therefore by Lemma 13, $G ⓩ K_d$ is a $(2/n)$-expander, i.e. a λ''-expander for some constant $\lambda'' < 1$ (when $n > 2$).

Thus our sampler stretches a seed of length $n + O(k)$ into k samples (of n bits each) that satisfy the bound from Theorem 1 for some constant $\lambda < 1$. Specifically, the sampler approximates the mean of the f_i's with error ϵ and confidence $1 - \gamma = 1 - e^{-\Omega(\epsilon^2 k)}$; in other words, the seed length is $n + O(k) = n + O(\log(1/\gamma)/\epsilon^2)$ and the sample complexity is $k = O(\log(1/\gamma)/\epsilon^2)$. Lemma 12 follows. $\square$

3.3 An Alternate Sampler Construction

In this section we describe an alternate implementation of a sampler in $\mathbf{AC}^0[\oplus]$. While this construction uses many of the same tools as Construction 11, the fundamental approach is different and is inspired by the paradigm of sampler *composition* [BGG93, Gol97], rather than the zig-zag graph product.

We note that the general *median of averages* composition of [BGG93, Gol97] does not result in an *averaging* sampler, which is the kind of sampler we consider in this work. Nonetheless, the same ideas can be employed here to obtain an averaging sampler, albeit with weaker parameters. For constant error ϵ, this sampler (Construction 14) matches the parameters of Construction 11. In fact, it is possible to generalize this construction to handle sub-constant ϵ; however, if one insists that the sampler be in $\mathbf{AC}^0[\oplus]$, then this approach cannot handle ϵ smaller than $1/\text{polylog}(n)$. Thus, to simplify the presentation we only treat the case of constant ϵ; moreover the case of constant ϵ is most common in applications.

Recall that Construction 11 employed short walks on a small expander, H, to select the steps to be made in the large expander G. Thus, H was used to derandomize the long walk on G. For the present construction, however, we shall instead use a long walk on a large auxiliary graph (denoted G' below) to select seeds for short walks on a large expander graph (the Margulis/Gabber-Galil expander).

Recall the ϵ-biased expander G from the proof of Theorem 3. Here we define G' in the same way, but on the vertex set $\{0,1\}^{n+3\log n}$ instead of $\{0,1\}^n$; that is, we construct a $1/n$-biased set $S \subset \{0,1\}^{n+3\log n}$ of size $\mathrm{poly}(n)$, and take G' to be the Cayley graph on $\mathbb{Z}_2^{n+3\log n}$ with generators S.

Construction 14.

1. *Choose a random starting vertex* $v'_0 \in G'$.
2. *Take a* $(k/\log n)$*-step random walk* $v'_1, \ldots, v'_{k/\log n}$ *on* G'.
3. *View each* $v'_i \in \{0,1\}^{n+3\log n}$ *as a* $(\log n)$*-step walk on the Margulis/Gabber-Galil expander of size* 2^n *and degree* 8.
4. *Expand each such walk* v'_i *into the nodes* $v_1^{(i)}, \ldots, v_{\log n}^{(i)} \in \{0,1\}^n$ *that it visits.*
5. *Output the* k *samples* $v_1^{(1)}, \ldots, v_{\log n}^{(1)}, \ldots, v_1^{(k/\log n)}, \ldots, v_{\log n}^{(k/\log n)}$.

This generator is computable in uniform $\mathbf{AC}^0[\oplus]$ since each of the required ingredients is computable in uniform $\mathbf{AC}^0[\oplus]$, as discussed in Section 3.1. Moreover, it is a good sampler for constant ϵ:

Proposition 15. *For any constant* $\epsilon > 0$*, Construction 14 is a strong* (γ, ϵ)*-averaging sampler with seed-length* $n + O(k) = n + O(\log(1/\gamma))$ *and sample complexity* $k = O(\log(1/\gamma))$ *(where the hidden constants depend on* ϵ*).*

Proof. We begin by noting that the number of random bits used by Construction 14 is $n + O(\log n) + (k/\log n) \cdot O(\log n) = n + O(k)$ (since we assume that $k = \Omega(\log n)$) and its sample complexity is k by construction. We show below that this sampler has error at most $\gamma = 2^{-\Omega(k)}$ when $\epsilon > 0$ is a constant; in other words, Construction 14 has seed length $n + O(k) = n + O(\log(1/\gamma))$ and sample complexity $k = O(\log 1/\gamma)$, as claimed.

In the following analysis, we shall confine ourselves to the case of sampling a single function (i.e., showing that Construction 14 is a non-strong averaging sampler). The proof that it is a strong sampler is completely analogous and simply follows from the fact that all the Chernoff bounds we apply are strong Chernoff bounds.

Let $f : \{0,1\}^n \to [0,1]$ be the function that is being sampled, and let $\rho = \mathrm{E}_x[f(x)]$. We first observe that a $(\log n)$-step random walk on the Margulis/Gaber-Galil Expander of size 2^n is likely to estimate ρ to within additive error $\epsilon/2$. Indeed, if we let $x_1, \ldots, x_{\log n}$ be a $(\log n)$-step random walk on this expander, then by Theorem 1,

$$\Pr\left[\left|\frac{1}{\log n}\sum_{j=1}^{\log n} f(x_j) - \mu\right| \geq \epsilon/2\right] \leq e^{-\frac{\epsilon^2(1-\lambda)\log n}{4}} = 1/n^c,$$

for some positive constant $c < 1/4$ (since we assume ϵ is constant).

Construction 14 then says to choose the seeds to these $(\log n)$-step walks according to a walk of length $k/\log n$ on a $2/n$-expander G' of degree $\mathrm{poly}(n)$. We expect at most a $1/n^c$ fraction of the short walks chosen in this way to yield poor estimates of ρ (i.e. not estimate ρ within $\pm\epsilon/2$); however, it is enough to hope that at most an $\epsilon/2$ fraction of the short walks are poor estimates in order to conclude that the average over all $\Theta(\log n) \cdot k/\log n = k$ samples will be $\rho \pm \epsilon$. Moreover, we would like this to happen with probability $1 - 2^{-\Omega(k)}$. However, to conclude this we need to apply a sharper Chernoff bound than Theorem 1. Indeed, for a walk of length $k/\log n$ Theorem 1 will never yield an failure probability smaller than $2^{-O(k/\log n)}$ and we would like to bound the failure probability by $2^{-\Omega(k)}$.

Fortunately, G' is a very good expander (in particular, a $2/n$-expander) and so we may apply Corollary 23. Indeed, we are interested in accurately sampling a set of density $1/n^c$ (the bad $(\log n)$-step walks), and G' has eigenvalue $2/n \leq (1/n^c)^2/3$ (for sufficiently large n), as required to apply Corollary 23. Specifically, we let X be the random variable that counts how many of the $(\log n)$-step walks are not $\rho \pm \epsilon/2$, so that X has expectation at most $\frac{1}{n^c} \cdot \frac{k}{\log n}$, and then by Corollary 23

$$\Pr\left[X \geq \frac{\epsilon}{2} \cdot \frac{k}{\log n}\right] \leq \left(\frac{2e}{\epsilon \cdot n^c}\right)^{\frac{1}{2}\cdot\frac{\epsilon}{2}\cdot\frac{k}{\log n}} = \left(\frac{1}{n^{\Omega(1)}}\right)^{\frac{k}{\log n}} = 2^{-\Omega(k)},$$

since we assume ϵ is a constant.

That is, with probability $1 - 2^{-\Omega(k)}$, at least a $1 - \epsilon/2$ fraction of the $(\log n)$-step walks estimate ρ to within additive error $\epsilon/2$, and hence the average over all the samples is $\rho \pm \epsilon$. In other words, the probability that Construction 14 does not estimate ρ within additive error ϵ is at most $\gamma = 2^{-\Omega(k)}$, and the result follows. □

3.4 Sampling vs. Hitting

Many applications of expander walks do not require the full power of the Chernoff bound. For example, the randomness-efficient error reduction of [CW89, IZ89], ϵ-biased sets of [NN90], the amplification of [GIL$^+$90] and the derandomized XOR lemma of [IW97] only require the *hitting property* of expander walks; i.e., they require that for any set $T \subseteq V$ of size at most $|V|/2$, the probability that a k-step random walk never leaves T is at most $2^{-\Omega(k)}$. (Strictly speaking, some of these results seem to require the *strong hitting property* of expander walks discussed in the introduction.) The latter three applications use the hitting property in a very natural way: in each case, the construction requires a sequence of objects that are combined in some way (e.g., addition, concatenation or XOR) and the proof of correctness only requires that at least one of these objects is "good" – furthermore, it is shown that "good" objects are abundant. Thus, by choosing these objects according to an expander walk and applying the hitting property, at least one of them will be "good" with high probability. For error reduction, it is less obvious that the (strong) hitting property suffices, although it does.[6]

[6]Roughly, this is proved as follows: we suppose the algorithm of interest uses r-bits of randomness and has error probability at most $1/20$. The new algorithm chooses k random r-bit strings according to an expander walk and outputs the majority vote of the k executions of the algorithm using these random strings. For the analysis, we fix an input x and consider the set of random strings T_x that cause the original algorithm to err on x, and thus we have $|T_x| \leq 2^r/20$;

In light of this, it would have been sufficient to simply show that Constructions 11 and 14 satisfy the strong *hitting* property in order to prove the results discussed in Section 4. Nonetheless, we choose to show that these constructions are strong samplers – our motivation for doing so is twofold. Firstly, for certain applications (especially error-reduction) the Chernoff-like behavior of the sampler makes for simpler and, we feel, more natural proofs than the approach based on a strong hitting generator. Secondly, we would like to say that our sampler can be used in place of an expander walk for "any conceivable application", and some applications of expander walks do seem to require the strong sampling property – for instance, constructing *randomness extractors*.

Loosely speaking, an *extractor* $\mathrm{Ext} : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n$ is a function that takes an m-bit input that is somewhat random, together with a short d-bit *seed* that is truly random and outputs an n-bit string that is very close to random. (For background on extractors, see the survey of Shaltiel [Sha02].)

One useful construction of an extractor (for sources of high, constant min-entropy) is based on random walks on expanders. Specifically, suppose that $W : \{0,1\}^m \to (\{0,1\}^n)^k$ computes a walk of length $k = \Theta(n)$ on a constant-degree expander (and thus $m = n + O(k) = \Theta(n)$). Then the function $\mathrm{Ext} : \{0,1\}^m \times [k] \to \{0,1\}^n$ defined by $\mathrm{Ext}(x,s) \stackrel{\text{def}}{=} W(x)_s$ is a strong extractor for sources x of min-entropy at least $(1-\beta)m$ for some constant $\beta > 0$; this follows from Theorem 1 (see [Zuc97] and [Zuc06]). Furthermore, the analysis of this extractor only depends on the fact that an expander walk is a strong sampler; therefore we may replace W with the sampler Γ of Theorem 3 to obtain such an extractor that is computable by uniform $\mathbf{AC}^0[\oplus]$ circuits. In particular, by the same proof as Proposition 4.2 of [Zuc06], we have the following corollary.

Corollary 16. *For all $\epsilon, \alpha > 0$, there exists $\beta > 0$ such that there is a family of strong $((1-\beta)m, \epsilon)$-extractors* $\mathrm{Ext} : \{0,1\}^m \times \{0,1\}^d \to \{0,1\}^n$ *with $n \geq (1-\alpha)m$ and $d \leq \log(\alpha m)$ that is computable by $\mathbf{AC}^0[\oplus]$ circuits of size* $\mathrm{poly}(m)$. *That is, for any m-bit source X with min-entropy at least $(1-\beta)m$ and an independent uniform d-bit seed Y, the distribution $(\mathrm{Ext}(X,Y), Y)$ is ϵ-close (in total variation distance) to the uniform distribution on $n+d$ bits.*

4 Proofs of Other Results

4.1 Error Reduction

Corollary 4 (restated)**.** *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by polynomial-size* uniform $\mathbf{BP} \cdot \mathbf{AC}^0[\oplus]$ *(respectively, $\mathbf{BP} \cdot \mathbf{TC}^0$ or $\mathbf{BP} \cdot \mathbf{NC}^1$) circuits with error at most $1/3$ using $r = r(n)$ random bits. Then for any $\delta = \delta(n) > 1/2^{O(\mathrm{poly}(n))}$, f has polynomial-size* uniform $\mathbf{BP} \cdot \mathbf{AC}^0[\oplus]$ *(respectively, $\mathbf{BP} \cdot \mathbf{TC}^0$ or $\mathbf{BP} \cdot \mathbf{NC}^1$) circuits with error at most δ using $r + O(\log(1/\delta))$ random bits.*

---

to bound the probability that at least $k/2$ of the sampled strings land in T_x, we consider all sequences of sets $S_1, \ldots, S_k$ where each S_i is either T_x or its complement and where at least $k/2$ of the S_i's are T_x. It is easy to see that there are $2^k/2$ such sequences, and by an appropriate version of the strong hitting property and a suitably good expander graph, one can show that the probability that a walk exactly follows such a given sequence of sets is less than $(1/4)^k$. Therefore, the probability that a random walk hits any of these $2^k/2$ sequences is less than $(2^k/2) \cdot (1/4)^k < 2^{-k}$.

Proof sketch. Let C_f be a circuit computing f. Construct the circuit that, on input $x \in \{0,1\}^n$, runs $k = \Theta(\log(1/\delta))$ copies of C_f in parallel using independent random r-bit blocks of randomness, and then computes the $(5/12, 7/12)$-approximate majority of the outputs [Ajt93]. (For $\mathbf{BP} \cdot \mathbf{TC}^0$ and $\mathbf{BP} \cdot \mathbf{NC}^1$ we can just compute the majority exactly.) Now, instead of using independent random bits for each block, we apply the construction of $\Gamma : \{0,1\}^{r+O(k)} \to (\{0,1\}^r)^k$ from Theorem 3 (with $\epsilon = 1/12$ and $\gamma = \delta$) to generate the necessary random bits from a seed of length $r + O(k)$.

For any fixed input x, the probability that a randomly chosen $r + O(k)$-bit random string causes the algorithm to fail (i.e., that more than 5/12 of the outputs of Γ fall in the set of random strings that cause C_f to fail) is at most $2^{-\Omega(k)} = 2^{-\Omega(\Theta(\log 1/\delta))}$ since Γ is an averaging sampler (and the latter set has density at most 1/3). By choosing the constants appropriately, this is at most δ and the result follows. $\square$

Corollary 5 (restated)**.** *Let* $f : \{0,1\}^n \to \{0,1\}$ *be a function computable by polynomial-size* uniform $\mathbf{BP} \cdot \mathbf{AC}^0$ *circuits with error at most* $1/3$ *using* $r = r(n)$ *random bits. Then for any* $\delta = \delta(n) > 1/2^{O(\text{poly}(n))}$, f *has polynomial-size* uniform $\mathbf{BP} \cdot \mathbf{AC}^0$ *circuits with error at most* δ *using* $\min\{r, \text{polylog}(n)\} + O(\log(1/\delta))$ *random bits.*

Proof sketch. Let C_f be a circuit computing f. By applying Nisan's pseudorandom generator for $\mathbf{BP} \cdot \mathbf{AC}^0$ [Nis91] (which has been shown to be computable in $\mathbf{AC}^0$ in [Vio04]), we may assume, with no loss of generality, that C_f uses only $r' = r'(n) = \min\{r(n), \log^c(n)\}$ random bits for some constant c that may depend on f.

If $\delta \geq 1/2^{r'}$, then we may apply the construction of Corollary 4 to obtain a $\mathbf{BP} \cdot \mathbf{AC}^0$ circuit that has error at most δ and uses $r' + O(\log(1/\delta))$ bits of randomness. (The circuit is in $\mathbf{BP} \cdot \mathbf{AC}^0$, and not just $\mathbf{BP} \cdot \mathbf{AC}^0[\oplus]$ because one can readily check that all the necessary parities are on at most $O(r') = O(\log^c n)$ bits, and can therefore be computed by a constant-depth circuit of size poly(n).)

If, on the other hand, $\delta < 1/2^{r'}$, then we apply Corollary 4 with $\delta(n) = 2^{-r'}$ to obtain an $\mathbf{AC}^0$ circuit that has error at most $2^{-r'}$ and uses $r' + O(r') \leq O(r')$ random bits. We now apply $\Theta(\log(1/\delta)/r')$ such circuits in parallel (on the same input, but independent random strings), and take the approximate majority of their $\Theta(\log(1/\delta)/r')$ outputs. Thus we have a circuit taking $O(r') \cdot \Theta(\log(1/\delta)/r') = O(\log(1/\delta)) \leq r' + O(\log(1/\delta))$ random bits and having error less than $(2^{-r'})^{\Theta(\log(1/\delta)/r')}$ (by a multiplicative Chernoff bound, such as Theorem 4.1 on p. 68 of [MR95]). This is at most δ by an appropriate setting of constants, and the result follows. $\square$

4.2 Derandomization with Linear Advice

Corollary 6 (restated)**.** uniform $\mathbf{BP} \cdot \mathbf{AC}^0 \subseteq$ uniform $\mathbf{AC}^0/O(n)$.

Proof. Apply Corollary 5 to obtain a $\mathbf{BP} \cdot \mathbf{AC}^0$ circuit with error less than 2^{-n} using $r = O(n)$ random bits. By a union bound, at least one r-bit string causes the circuit to correctly decide all inputs. Fix one such string as the non-uniform advice and the result follows. $\square$

Corollary 17. *Let* $\mathbf{AC}^0[\oplus_{\log}]$ *be the class of boolean functions computable by* $\mathrm{poly}(n)$*-size* $\mathbf{AC}^0$ *circuits having* $O(\log n)$ *parity gates, and similarly let* $\mathbf{AC}^0[\mathrm{SYM}_{\log}]$ *be the class of boolean functions computable by* $\mathrm{poly}(n)$*-size* $\mathbf{AC}^0$ *circuits having* $O(\log n)$ *arbitrary symmetric gates (e.g., parity and majority gates). Then the following inclusions hold:*

1. $\mathbf{BP} \cdot \mathbf{AC}^0[\oplus_{\log}] \subseteq \mathbf{AC}^0[\oplus]/O(n)$
2. $\mathbf{BP} \cdot \mathbf{AC}^0[\mathrm{SYM}_{\log}] \subseteq \mathbf{TC}^0/O(n)$

Proof sketch. The proof is similar to the proofs of Corollaries 5 and 6, except that we use the generator of Viola [Vio04] instead of Nisan's. Specifically, the generator from [Vio04] allows us to assume, without loss of generality, that any function $f \in \mathbf{BP} \cdot \mathbf{AC}^0[\oplus_{\log}]$ (respectively, $\mathbf{BP} \cdot \mathbf{AC}^0[\mathrm{SYM}_{\log}]$) can be computed by a $\mathbf{BP} \cdot \mathbf{AC}^0[\oplus]$ (respectively, $\mathbf{BP} \cdot \mathbf{TC}^0$) circuit using only $n^{o(1)}$ random bits. By applying Corollary 4, we may reduce the error to less than 2^{-n} using only $n^{o(1)} + O(n) = O(n)$ random bits. Finally, a union bound yields a single advice string of $O(n)$ bits that works for all inputs. □

4.3 An Optimal bitwise ϵ-biased generator in $\mathbf{AC}^0[\oplus]$

Corollary 7. *For every* $\epsilon > 0$ *and* m*, there is an* ϵ*-biased generator* $G : \{0,1\}^n \to \{0,1\}^m$ *with* $n = O(\log m + \log(1/\epsilon))$ *such that uniform* $\mathbf{AC}^0[\oplus]$ *circuits of size* $\mathrm{poly}(n, \log m) = \mathrm{poly}(n)$ *can compute* $G(s)_i$ *given* $(s,i) \in \{0,1\}^n \times [m]$.

Proof idea. We follow the approach of [GV04] and implement the ϵ-biased generator of Naor and Naor [NN90]. This generator employs a 7-wise independent generator to construct a small set of "distinguishers" (an object that is weaker than an ϵ-biased set), and from these it is possible to obtain an ϵ-biased generator by choosing many such distinguishers (independently) and taking a random linear combination of them. However, to improve the seed-length of their generator, [NN90] choose these distinguishers according to a walk on an expander graph. Thus, to construct a bit-wise ϵ-biased generator in $\mathbf{AC}^0[\oplus]$, [GV04] require a bit-wise 7-wise independent generator and long expander walks that are each computable by $\mathbf{AC}^0[\oplus]$ circuits. Constructions of bit-wise 7-wise independent generators in $\mathbf{AC}^0[\oplus]$ are known [GV04, HV06], and since the analysis of [NN90] only uses the fact that an expander walk is a good hitting generator, our sampler construction from Section 3 can be used in place of the expander walk. □

5 Strong Chernoff Bounds for Expander Walks

In this section we give an elementary proof of a generalization of Gillman's *Chernoff Bound for Expander Walks* [Gil94] (Theorem 1) as well as a (strong) multiplicative Chernoff bound for expander walks that is sharper than Theorem 1 when the eigenvalue λ is small (Theorem 22 and Corollary 23).

5.1 The Proof of Theorem 1

This section is devoted to proving the following theorem.

Theorem 1 (restated)**.** *Let G be a regular λ-expander on V and fix a sequence of functions $f_i : V \to [0,1]$ each with mean $\mu_i = \mathrm{E}_v[f_i(v)]$. If we consider a random walk $v_1, \ldots, v_k$ on G, then for all $\epsilon > 0$,*

$$\Pr\left[\left|\sum_{i=1}^{k} f_i(v_i) - \sum_{i=1}^{k} \mu_i\right| \geq \epsilon k\right] \leq 2e^{-\frac{\epsilon^2(1-\lambda)k}{4}}.$$

Wigderson and Xiao [WX05] have recently established essentially the same bound (up to constants) using techniques from perturbation theory. Gillman's proof (which treats the case where $f_1 = \cdots = f_k$) also employs results from perturbation and complex analysis to obtain a similar bound. In contrast, the proof presented here has only very modest prerequisites, which are summarized in the following paragraph.

Background We work with a regular λ-expander G on N nodes (see Definition 9). In particular, if we denote G's transition matrix by P and write $\mathbf{u} = (1/N, \ldots, 1/N) \in \mathbb{R}^N$, then $P\mathbf{u} = \mathbf{u}$ and

$$\max_{\substack{x \in \mathbb{R}^N \\ \langle x, \mathbf{u}\rangle = 0}} \frac{\|Px\|}{\|x\|} \leq \lambda.$$

For any $\mathbf{z} \in \mathbb{R}^N$, we let $\mathbf{z}^{\|} = \langle \mathbf{1}, \mathbf{z}\rangle\mathbf{u}$ denote the component of $\mathbf{z}$ in the direction of $\mathbf{1} = (1, \ldots, 1) \in \mathbb{R}^N$ and we let $\mathbf{z}^{\perp} = \mathbf{z} - \mathbf{z}^{\|} = \mathbf{z} - \langle \mathbf{1}, \mathbf{z}\rangle\mathbf{u}$ denote the component of $\mathbf{z}$ orthogonal to $\mathbf{1}$. Thus, for any $\mathbf{z} \in \mathbb{R}^N$, we have that $P\mathbf{z}^{\|} = \mathbf{z}^{\|}$ and $\|P\mathbf{z}^{\perp}\| \leq \lambda \cdot \|\mathbf{z}^{\perp}\|$. Another useful fact is that for any $\mathbf{z} \in \mathbb{R}^N$, the vector $P\mathbf{z}^{\perp}$ is orthogonal to $\mathbf{1}$ simply because $\langle \mathbf{1}, P\mathbf{z}^{\perp}\rangle = \mathbf{1}^T P\mathbf{z}^{\perp}$ and $\mathbf{1}^T P = \mathbf{1}^T$ since we assume G is regular.

Proof of Theorem 1. Define the random variable $X = \sum_i f_i(v_i)$ where $v_1, \ldots, v_k$ is a random walk on G, and let $\mu = \sum_i \mu_i = E[X]$. We shall bound the quantity $\Pr[X \geq \mu + \epsilon k]$ and the same bound will follow for $\Pr[X \leq \mu - \epsilon k]$ by replacing the functions $f_i(v)$ with $1 - f_i(v)$. Let $r \leq \log(1/\lambda)/2$ be a positive parameter to be specified later.

$$\Pr[X \geq \mu + \epsilon k] = \Pr\left[e^{rX} \geq e^{r\mu + r\epsilon k}\right] \leq \frac{\mathrm{E}\left[e^{rX}\right]}{e^{r\mu + r\epsilon k}} \tag{1}$$

where the last step follows by applying Markov's inequality.

We now focus on bounding $\mathrm{E}\left[e^{rX}\right]$. Let P be the probability transition matrix for G, and for each function f_i let E_i be a diagonal matrix with diagonal entries $\left(e^{rf_i(v)}\right)_{v \in V}$. It is not hard to see that

$$\mathrm{E}\left[e^{rX}\right] = \mathbf{1}^T E_k P E_{k-1} P \cdots E_1 P\mathbf{u}, \tag{2}$$

since every non-zero cross-term in this matrix product corresponds to exactly one walk $v_1, \ldots, v_k$ on G and each such term is exactly the probability of the walk times $e^{\sum_i f_i(v_i)}$.

Thus far, the techniques are quite standard. Indeed, the typical recipe for proving a Chernoff bound begins by reducing the task to bounding the moment-generating function $E\left[e^{rX}\right]$, and many previous tail bounds for Markov chains make use of the identity (2) (albeit with $E_1 = \cdots = E_k$) to bound $E\left[e^{rX}\right]$ [Gil94, Din95, Kah97, L98́, LP04, WX05].

At this point, however, the proof diverges from previous approaches in that we bound (2) inductively using elementary manipulations. This is in contrast to the previous works that rely heavily on the machinery of perturbation theory (with the exception of [Kah97] which uses a novel eigenvalue argument), and also allows us to treat the case of sampling distinct function $f_1, \ldots, f_k$ (which is not readily amenable to previous techniques, except in the case of [WX05]).

Specifically, to bound the quantity (2), we study the sequence of vectors $\mathbf{z}_0 = \mathbf{u}$, $\mathbf{z}_1 = E_1 P\mathbf{u}$, $\mathbf{z}_2 = E_2 P E_1 P\mathbf{u}, \ldots$ inductively. Indeed, we note that

$$\mathrm{E}\left[e^{rX}\right] = \mathbf{1}^T E_k P E_{k-1} P \cdots E_1 P\mathbf{u} = \langle \mathbf{1}, \mathbf{z}_k\rangle = \langle \mathbf{1}, \mathbf{z}_k^{\|}\rangle = \sqrt{N}\cdot \|\mathbf{z}_k^{\|}\|, \tag{3}$$

and so our goal is to bound $\|\mathbf{z}_k^{\|}\|$.

We bound $\|\mathbf{z}_k^{\|}\|$ by first showing (in Lemma 19) that each of the $\mathbf{z}_i$'s remains nearly parallel to $\mathbf{u}$ (since E_i is close to the identity matrix when r is small, and moreover P helps shrink any component of $\mathbf{z}_i$ that is not parallel to $\mathbf{u}$). Then we observe (in Lemma 20) that E_i stretches $\mathbf{u}$ (and hence the $\mathbf{z}_i$'s, since they are nearly parallel to $\mathbf{u}$) by a factor of $\mathrm{E}_v\left[e^{rf_i(v)}\right] \approx e^{r\,\mathrm{E}_v[f_i(v)]} = e^{r\mu_i}$ (again, when r is small) which in turn ensures that $\mathrm{E}\left[e^{rX}\right] \approx e^{r\mu}$; more precisely, we find that $\mathrm{E}\left[e^{rX}\right] \leq e^{r\mu + r^2 k/(1-\lambda)}$. This bounds the probability in (1) by $e^{(r^2/(1-\lambda)-\epsilon r)k}$, and the result follows by choosing r to minimize this probability, i.e. $r = (1-\lambda)\epsilon/2$.

In the manipulations that follow, it may be worthwhile to bear in mind that we ultimately choose r to be small and therefore $e^r - 1 \approx r$ is also small.

The following lemma bounds how long $\mathbf{z}_{i+1}^{\|}$ and $\mathbf{z}_{i+1}^{\perp}$ can be relative to $\mathbf{z}_i^{\|}$ and $\mathbf{z}_i^{\perp}$.

Lemma 18. *Let P and $0 < r \leq \log(1/\lambda)/2$ be as above, and let E be a diagonal matrix with diagonal entries $(e^{rf(v)})_{v\in V}$ for some function $f : V \to [0,1]$ with mean $\rho = E_v[f(v)]$. Then for any $\mathbf{z} \in \mathbb{R}^N$:*

1. $\|(EP\mathbf{z}^{\|})^{\|}\| \leq (1 + (e^r - 1)\rho)\cdot \|\mathbf{z}^{\|}\|$.
2. $\|(EP\mathbf{z}^{\|})^{\perp}\| \leq \frac{e^r-1}{2}\cdot \|\mathbf{z}^{\|}\|$.
3. $\|(EP\mathbf{z}^{\perp})^{\|}\| \leq \frac{e^r-1}{2}\cdot \lambda \cdot \|\mathbf{z}^{\perp}\|$.
4. $\|(EP\mathbf{z}^{\perp})^{\perp}\| \leq \sqrt{\lambda}\cdot \|\mathbf{z}^{\perp}\|$.

Proof. (1): $(EP\mathbf{z}^{\|})^{\|} = (E\mathbf{z}^{\|})^{\|} = \langle \mathbf{1}, E\mathbf{z}^{\|}\rangle \mathbf{u} = \langle \mathbf{1}, E\mathbf{u}\rangle \mathbf{z}^{\|} = \mathrm{E}_v\left[e^{rf(v)}\right]\cdot \mathbf{z}^{\|}$, and using the fact that $e^{rx} \leq 1 + (e^r-1)x$ for all $0 \leq x \leq 1$, we have

$$\|(EP\mathbf{z}^{\|})^{\|}\| = \mathop{\mathrm{E}}_v\left[e^{rf(v)}\right]\cdot \|\mathbf{z}^{\|}\| \leq \mathop{\mathrm{E}}_v\left[1 + (e^r-1)f(v)\right]\cdot \|\mathbf{z}^{\|}\| = (1 + (e^r-1)\rho)\cdot \|\mathbf{z}^{\|}\|.$$

(2): Recalling that $(\mathbf{z}^{\|})^{\perp} = 0$ for all $\mathbf{z}$, we note that for any $\alpha \in \mathbb{R}$,

$$(EP\mathbf{z}^{\|})^{\perp} = (E\mathbf{z}^{\|})^{\perp} = ((E - \alpha\cdot I)\mathbf{z}^{\|})^{\perp} + (\alpha\cdot \mathbf{z}^{\|})^{\perp} = ((E-\alpha\cdot I)\mathbf{z}^{\|})^{\perp}.$$

Thus, we choose $\alpha = \frac{e^r+1}{2}$ so that $E - \alpha \cdot I$ is diagonal with entries bounded by $\frac{e^r-1}{2}$ in absolute value (since $e^r - \alpha = \frac{e^r-1}{2}$ and $e^0 - \alpha = -\frac{e^r-1}{2}$). Then,

$$\|(EP\mathbf{z}^{\|})^{\perp}\| = \|((E - \alpha \cdot I)\mathbf{z}^{\|})^{\perp}\| \le \frac{e^r - 1}{2} \cdot \|\mathbf{z}^{\|}\|.$$

(3): Recalling that $(P\mathbf{z}^{\perp})^{\|} = 0$ for all $\mathbf{z}$, we note that for any $\alpha \in \mathbb{R}$,

$$(EP\mathbf{z}^{\perp})^{\|} = ((E - \alpha \cdot I)P\mathbf{z}^{\perp})^{\|} + (\alpha \cdot P\mathbf{z}^{\perp})^{\|} = ((E - \alpha \cdot I)P\mathbf{z}^{\perp})^{\|}.$$

Again, we choose $\alpha = \frac{e^r+1}{2}$ so that $E - \alpha \cdot I$ is diagonal with entries bounded by $\frac{e^r-1}{2}$ in absolute value, and get

$$\|(EP\mathbf{z}^{\perp})^{\|}\| = \|((E - \alpha \cdot I)P\mathbf{z}^{\perp})^{\|}\| \le \frac{e^r - 1}{2} \cdot \|P\mathbf{z}^{\perp}\| \le \frac{e^r - 1}{2} \cdot \lambda \cdot \|\mathbf{z}^{\perp}\|,$$

where the last inequality uses the fact that $\|P\mathbf{z}^{\perp}\| \le \lambda \cdot \|\mathbf{z}^{\perp}\|$ for any vector $\mathbf{z} \in \mathbb{R}^N$.

(4): $\|(EP\mathbf{z}^{\perp})^{\perp}\| \le \|EP\mathbf{z}^{\perp}\| \le e^r \cdot \|P\mathbf{z}^{\perp}\| \le e^r\lambda \cdot \|\mathbf{z}^{\perp}\|$, and since we assume that $r \le \log(1/\lambda)/2$, this is at most $\sqrt{\lambda} \cdot \|\mathbf{z}^{\perp}\|$. □

Recall that $\mathbf{z}_0 = \mathbf{u}$ and $\mathbf{z}_{i+1} = E_{i+1}P\mathbf{z}_i$. We now show that $\mathbf{z}_i^{\perp}$ remains short compared to the previous $\mathbf{z}_j^{\|}$'s.

Lemma 19. $\|\mathbf{z}_i^{\perp}\| \le \frac{e^r-1}{1-\lambda} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}$ *for* $1 \le i \le k$.

Proof. By the triangle inequality,

$$\|\mathbf{z}_i^{\perp}\| = \|(E_iP\mathbf{z}_{i-1})^{\perp}\| = \|(E_iP\mathbf{z}_{i-1}^{\|})^{\perp} + (E_iP\mathbf{z}_{i-1}^{\perp})^{\perp}\| \le \|(E_iP\mathbf{z}_{i-1}^{\|})^{\perp}\| + \|(E_iP\mathbf{z}_{i-1}^{\perp})^{\perp}\|.$$

Thus, by Items 2 and 4 of Lemma 18, we have $\|\mathbf{z}_i^{\perp}\| \le \frac{e^r-1}{2} \cdot \|\mathbf{z}_{i-1}^{\|}\| + \sqrt{\lambda} \cdot \|\mathbf{z}_{i-1}^{\perp}\|$.

Recursively applying this bound, and noting that $\|\mathbf{z}_0^{\perp}\| = 0$, we have

$$\|\mathbf{z}_i^{\perp}\| \le \frac{e^r - 1}{2} \cdot \sum_{j=0}^{i-1}(\sqrt{\lambda})^j\|\mathbf{z}_{i-j-1}^{\|}\| \le \frac{e^r - 1}{2(1 - \sqrt{\lambda})} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}.$$

The lemma follows by noting that $1/(1 - \sqrt{\lambda}) = (1 + \sqrt{\lambda})/(1 - \lambda) \le 2/(1 - \lambda)$ since $\lambda \in [0, 1]$. □

We now use Lemma 19 to bound $\|\mathbf{z}_i^{\|}\|$ inductively.

Lemma 20. $\|\mathbf{z}_i^{\|}\| \le \exp\left\{(e^r - 1)\mu_i + \frac{\lambda\cdot(e^r-1)^2}{2(1-\lambda)}\right\} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}$, *for* $1 \le i \le k$.

Proof. By the triangle inequality,

$$\|\mathbf{z}_i^{\|}\| = \|(E_iP\mathbf{z}_{i-1})^{\|}\| = \|(E_iP\mathbf{z}_{i-1}^{\|})^{\|} + (E_iP\mathbf{z}_{i-1}^{\perp})^{\|}\| \le \|(E_iP\mathbf{z}_{i-1}^{\|})^{\|}\| + \|(E_iP\mathbf{z}_{i-1}^{\perp})^{\|}\|.$$

Thus, by Items 1 and 3 of Lemma 18, we have $\|\mathbf{z}_i^{\|}\| \le (1 + (e^r - 1)\mu_i) \cdot \|\mathbf{z}_{i-1}^{\|}\| + \frac{e^r-1}{2} \cdot \lambda \cdot \|\mathbf{z}_{i-1}^{\perp}\|$, and so by Lemma 19,

$$\|\mathbf{z}_i^{\|}\| \le (1+(e^r-1)\mu_i)\cdot\|\mathbf{z}_{i-1}^{\|}\|+\frac{\lambda \cdot (e^r - 1)^2}{2(1 - \lambda)}\cdot \max_{j<i-1}\{\|\mathbf{z}_j^{\|}\|\} \le \left(1 + (e^r - 1)\mu_i + \frac{\lambda \cdot (e^r - 1)^2}{2(1 - \lambda)}\right)\cdot\max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}.$$

Finally, using the fact that $1+x \leq e^x$ for all $x \geq 0$, we conclude that this is at most

$$\exp\left\{(e^r-1)\mu_i + \frac{\lambda\cdot(e^r-1)^2}{2(1-\lambda)}\right\}\cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}.$$

□

Recalling that $\|\mathbf{z}_0^{\|}\| = 1/\sqrt{N}$, Lemma 20 implies that for all $j \geq 0$:

$$\|\mathbf{z}_j^{\|}\| \leq \frac{1}{\sqrt{N}}\prod_{i=1}^{j}\exp\left\{(e^r-1)\mu_i + \frac{\lambda\cdot(e^r-1)^2}{2(1-\lambda)}\right\},$$

and in particular, by (3),

$$\mathrm{E}\left[e^{rX}\right] = \sqrt{N}\cdot\|\mathbf{z}_k^{\|}\| \leq \prod_{i=1}^{k}\exp\left\{(e^r-1)\mu_i + \frac{\lambda\cdot(e^r-1)^2}{2(1-\lambda)}\right\} = \exp\left\{(e^r-1)\mu + \frac{\lambda\cdot(e^r-1)^2}{2(1-\lambda)}\cdot k\right\}. \quad (4)$$

To simplify this expression, we shall assume that $r \leq 1/2$ (and thus that $e^r - 1 \leq r + 2r^2/3 \leq 4r/3$) and we note that $\mu \leq k$:

$$\mathrm{E}\left[e^{rX}\right] \leq \exp\left\{(r+r^2)\mu + \frac{\lambda\cdot(4r/3)^2}{2(1-\lambda)}\cdot k\right\} \leq e^{r\mu + r^2\cdot\left(1+\frac{\lambda}{1-\lambda}\right)\cdot k} = e^{r\mu + \frac{r^2 k}{1-\lambda}}.$$

Thus, by (1) we have

$$\Pr\left[X \geq \mu + \epsilon k\right] \leq \frac{\mathrm{E}\left[e^{rX}\right]}{e^{r\mu + r\epsilon k}} \leq e^{\left(\frac{r^2}{1-\lambda} - r\epsilon\right)k}.$$

Finally, we minimize this probability by setting $r = (1-\lambda)\epsilon/2$, noting that r is indeed at most $\min\{1/2, \log(1/\lambda)/2\}$ simply because $\epsilon \leq 1$ and $1-\lambda \leq \log(1/\lambda)$ for all $\lambda \in [0,1]$. It follows that

$$\Pr\left[X \geq \mu + \epsilon k\right] \leq e^{-\frac{\epsilon^2(1-\lambda)k}{4}}.$$

□

Remark 21. *One can readily see that the same proof applies even if the graph is different for each of the k steps, as long as it is a λ-expander at each step. This is observation is important for the proof of correctness of our sampler (Theorem 3), as that construction concerns a walk on an expander graph that is varying from one step to the next step. This observation is not unique to our proof of the Chernoff bound, and this same property has been exploited before, most notably in the hardness amplification result of Goldreich et al. [GIL+90] (although there, they only require the hitting property of expander walks, and not the stronger sampling properties guaranteed here).*

5.2 A Multiplicative Strong Chernoff Bound

As exemplified in Section 3.3, it is sometimes useful to have tail bounds that are sharper than Theorem 1 when considering rare events and large deviations from the mean. In this section we prove bounds (Theorem 22 and Corollary 23) that improve upon Theorem 1 in this case, provided that the eigenvalue λ is sufficiently small.

The motivation for such bounds comes from the case of independent random variables. Indeed, it is well known that the standard *additive* Chernoff bound of the form $\Pr[X \geq E[X]+\epsilon k] \leq e^{-\Omega(\epsilon^2 k)}$ (where $X = X_1 + \cdots + X_k$ is a sum of i.i.d. Bernoulli random variables X_i) is suboptimal when the mean of the X_i's is very small and ϵ is large. For instance, if we take $E[X_i] = 1/k$ and we consider $\Pr[X \geq k/2]$, then the standard Chernoff bound (with $\epsilon = 1/2 - 1/k$) only bounds this probability by $e^{-\Omega(k)}$, when in fact it is possible to show that $\Pr[X \geq k/2] \leq e^{-\Omega(k \log k)}$ (e.g., via a *multiplicative* Chernoff bound, such as Theorem 4.1 of [MR95]).

The analogous case for expander walks is when the set we are trying to sample is very small (or more generally when, in the notation of Theorem 1, the μ_i are small). To obtain a sharper bound in this setting, however, one should have an eigenvalue λ that is quite small, and then it is possible to slightly modify the proof of Theorem 1 to obtain a bound analogous to Theorem 4.1 of [MR95]:

Theorem 22. *Fix a sequence of functions $f_i : V \to [0,1]$ each with mean $\mu_i = \mathrm{E}_v[f_i(v)]$ and let $\mu = \sum_{i=1}^k \mu_i$. If we consider a random walk $v_1, \ldots, v_k$ on a λ-expander G, then for all $\delta > 0$,*

$$\Pr\left[\sum_{i=1}^k f_i(v_i) \geq (1+\delta)\mu\right] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\left(1 - \frac{\lambda}{1-\lambda} \cdot \left(\frac{k}{\mu}\right)^2\right)\mu}.$$

In particular, when $\lambda = 0$ this matches Theorem 4.1 of [MR95]. We also note that $e^\delta/(1+\delta)^{(1+\delta)} = e^{\delta-(1+\delta)\log(1+\delta)} \leq 1$ for all $\delta \geq 0$, simply because $\delta \leq (1+\delta)\log(1+\delta)$ (as can easily be verified by comparing the derivatives of both sides). Thus, the bound is nontrivial whenever $\frac{\lambda}{1-\lambda} < (\mu/k)^2$. So, for instance, if we are interested in sampling a set of density α, then we should use an expander with $\lambda \lesssim \alpha^2$ in order to meaningfully apply Theorem 22. In particular, if λ is sufficiently small compared to α, say $\lambda \leq \alpha^2/3$, then the bound from Theorem 22 is never more than the square-root of the bound for independent random variables (i.e., Theorem 4.1 of [MR95]) – see also the proof of Corollary 23.

Consequently, Theorem 22 is sharper than Theorem 1 in the same way that a multiplicative Chernoff bound is sharper than the standard additive Chernoff bound, provided that λ is sufficiently small. For instance, analogously to the example of independent random variables described above, if we have a set $S \subseteq V$ of nodes of density $1/k$ and take a random walk of length k, then the probability of landing in S at least $k/2$ times is at most $e^{-\Omega(k \log k)}$ (provided that $\lambda \lesssim 1/k^2$), and not just $e^{-\Omega(k)}$ (which is all that Theorem 1 gives). When bounding the probability of large derivations from the mean (as in the previous example), the bound from Corollary 23 is often easier to work with and essentially as good as Theorem 22. Indeed, this is the bound that we employ in the analysis of our alternate sampler construction from Section 3.3 (i.e., Proposition 15).

Proof of Theorem 22. The proof is identical to the proof Theorem 1 up to the derivation of (4), at which point we make a different choice of the parameter r. In particular, we first equate the notation of Theorem 22 with that of Theorem 1 by taking $\epsilon = \delta\mu/k$. Indeed, then $\Pr[X \geq \mu + \epsilon k] = \Pr[X \geq (1+\delta)\mu]$, and so (1) now becomes

$$\Pr[X \geq (1+\delta)\mu] \leq \frac{\mathrm{E}\left[e^{rX}\right]}{e^{r(1+\delta)\mu}}. \tag{5}$$

Next, in contrast to the proof of Theorem 1, we choose $r = \log(1+\delta)$ (rather than $r = (1-\lambda)\epsilon/2 = (1-\lambda)\delta\mu/2k$). Before proceeding, however, we must check that $r \le \log(1/\lambda)/2$, as required throughout the proof of Theorem 1: we may assume, with no loss of generality, that $1+\delta \le k/\mu$ (indeed, the result is trivial if $\delta > k/\mu - 1$), and we may assume that $\lambda \le (\mu/k)^2$, since otherwise the bound stated in the Corollary is larger than 1. Therefore, we have $r \le \log(k/\mu) \le \log(1/\lambda)/2$.

Substituting $r = \log(1+\delta)$ into (4), we have

$$\mathrm{E}\left[e^{rX}\right] \le e^{\mu\delta + \frac{\lambda\delta^2 k}{2(1-\lambda)}},$$

and thus, by (5), we have

$$\Pr\left[X \ge (1+\delta)\mu\right] \le \frac{e^{\mu\delta + \frac{\lambda\delta^2 k}{2(1-\lambda)}}}{e^{\log(1+\delta)(1+\delta)\mu}} = e^{\left(\delta + \frac{\lambda\delta^2}{2(1-\lambda)}\cdot\frac{k}{\mu} - (1+\delta)\log(1+\delta)\right)\mu}. \tag{6}$$

To bound this expression, we first establish that

$$\frac{\delta^2}{2} \le \frac{k}{\mu}\cdot\left[(1+\delta)\log(1+\delta) - \delta\right]. \tag{7}$$

Indeed, both sides of this expression are equal to 0 when $\delta = 0$, and we shall verify that the derivative of the left-hand side (with respect to $\delta \in [0, k/\mu - 1]$) is always bounded by the derivative of the right-hand side: the derivative of the left-hand side is δ and the derivative of the right-hand side is $\frac{k}{\mu}\cdot\log(1+\delta)$, and $\delta \le (1+\delta)\log(1+\delta) \le \frac{k}{\mu}\cdot\log(1+\delta)$ for $\delta \in [0, k/\mu - 1]$, so (7) holds.

Thus, by applying (7) to bound the $\delta^2/2$ term that appears in (6), we have

$$\Pr\left[X \ge (1+\delta)\mu\right] \le e^{\left(\delta + \frac{\lambda}{1-\lambda}\cdot\left(\frac{k}{\mu}\right)^2\cdot[(1+\delta)\log(1+\delta)-\delta] - (1+\delta)\log(1+\delta)\right)\mu} = e^{\left(1 - \frac{\lambda}{1-\lambda}\cdot\left(\frac{k}{\mu}\right)^2\right)\cdot[\delta - (1+\delta)\log(1+\delta)]\mu},$$

and the result follows. □

As mentioned above, a bound like Theorem 22 is better than Theorem 1 when μ is small and δ is large (and when we can afford to choose λ to be sufficiently small). With this in mind we mention a simpler, albeit less general, form of the bound:

Corollary 23. *Fix a sequence of functions $f_i : V \to [0,1]$ each with mean $\mu_i = \mathrm{E}_v[f_i(v)]$ and let $\mu = \sum_{i=1}^k \mu_i$. Furthermore, let G be a regular λ-expander on V for some $\lambda \le (\mu/k)^2/3$. If we consider a random walk $v_1, \ldots, v_k$ on G then*

$$\Pr\left[\sum_{i=1}^k f_i(v_i) \ge t\right] \le \left(\frac{e\mu}{t}\right)^{t/2}.$$

Proof. We let $\delta = t/\mu - 1$ so that $t = (1+\delta)\mu$. Then by Theorem 22,

$$\Pr\left[\sum_{i=1}^k f_i(v_i) \ge t\right] \le \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\mu\left(1 - \frac{\lambda}{1-\lambda}\cdot\left(\frac{k}{\mu}\right)^2\right)}.$$

Since we assume $\lambda \leq (\mu/k)^2/3 \leq 1/3$, we have that $\frac{\lambda}{1-\lambda} \cdot \left(\frac{k}{\mu}\right)^2 \leq 1/2$, and thus

$$\Pr\left[\sum_{i=1}^{k} f_i(v_i) \geq t\right] \leq \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{\mu/2} < \left(\frac{e}{1+\delta}\right)^{(1+\delta)\mu/2} = \left(\frac{e\mu}{t}\right)^{t/2}.$$

□

6 Open Questions

In this work we construct an extremely efficient sampler that is in many respects "just as good" as random-walk sampling using a constant-degree expander graph; a natural question that is left open, however, is whether $\mathbf{AC}^0[\oplus]$ can actually compute long expander walks – that is, whether there exists a family of constant-degree expander graphs for which a family of $\mathbf{AC}^0[\oplus]$ circuits can compute walks $v_1, \ldots, v_k$ when given a starting node v_0 and steps $s_1, \ldots, s_k$. (Or, one could even remove the restriction on the input format and just ask for a generator whose output distribution is the same as – or even statistically close to – a random walk on an expander.) Our techniques come close: indeed, the approach of Section 3.1 can easily be modified to obtain a circuit that computes walks on the zig-zag product $G \textcircled{z} H$ when given a circuit for computing walks on H. (Recall that G has size 2^n and degree poly(n), and H has size poly(n).) Thus, if we take H to be an exponentially-smaller copy of G (of size poly(n) and degree polylog(n)) – rather than a constant-degree expander – $\mathbf{AC}^0[\oplus]$ can compute long walks on H and therefore can also compute walks on the graph $G \textcircled{z} H$ of degree polylog(n); in fact, by recursively applying a constant number of such zig-zag products, we obtain an expander G' of size at least 2^n and degree at most poly$(\log^{(t)} n)$ for any constant t. Alternatively, by repeating this recursion $\log^* n$ times, we obtain a constant-degree expander G' and a family of circuits of depth $O(\log^* n)$ for computing walks on G'. Can this depth be reduced to $O(1)$? Even less ambitiously, is there any family of constant-degree expander graphs for which a family of (nonuniform) $\mathbf{NC}^1$ circuits can compute long walks?

There is also the question of lower-bounds. We suspect that $\mathbf{AC}^0$ cannot compute samplers that match the parameters of our $\mathbf{AC}^0[\oplus]$ construction. One approach to showing this is to use the equivalence of samplers and extractors from [Zuc97] (see also the discussion in Section 3.4) and show that $\mathbf{AC}^0$ cannot compute a (strong) extractor for sources of high constant min-entropy. Viola [Vio04] has shown that $\mathbf{AC}^0$ cannot compute an extractor for sources of low min-entropy; however, his techniques do not seem to apply directly in this setting.

7 Acknowledgements

Thanks are due to Kai-Min Chung, Oded Goldreich, Danny Gutfreund, Salil Vadhan, Emanuele Viola and David Zuckerman for various discussions, suggestions and their encouragement. In particular, Emanuele offered some helpful comments on an early draft of this work, and Oded's thorough reading of this paper and his many comments are greatly appreciated. Thanks also to David Xiao for an email exchange about [WX05] and [WX06] and to the anonymous Random 2006 reviewers for their comments.

References

[ABO84] Miklós Ajtai and Michael Ben-Or. A theorem on probabilistic constant depth computation. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing*, pages 471–474, April 30 – May 2 1984.

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

[Ajt93] Miklós Ajtai. Approximate counting with uniform constant-depth circuits. In *Advances in computational complexity theory*, pages 1–20. Amererican Mathematical Society, 1993.

[AKS83] M. Ajtai, J. Komlos, and E. Szemeredi. An O(n log n) sorting network. *Combinatorica*, 3:1–19, 1983.

[AKS87] M. Ajtai, J. Komlos, and E. Szemeredi. Deterministic simulation in LOGSPACE. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing*, pages 132–140, May 25–27 1987.

[AR94] Noga Alon and Yuval Roichman. Random cayley graphs and expanders. *Random Structures & Algorithms*, 5:271–284, 1994.

[AS00] Noga Alon and Joel H. Spencer. *The Probabilistic Method.* Wiley-Interscience Series in Discrete Mathematics and Optimization. John Wiley and Sons, Inc., 2000.

[BGG93] M. Bellare, O. Goldreich, and S. Goldwasser. Randomness in interactive proofs. *Computational Complexity*, 4(1):319–354, 1993.

[BIS90] David A. Mix Barrington, Neil Immerman, and Howard Straubing. On uniformity within NC^1. *Journal of Computer and System Sciences*, 41(3):274–306, 1990.

[BYGW99] Z. Bar-Yossef, O. Goldreich, and A. Wigderson. Deterministic amplification of space-bounded probabilistic algorithms. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity*, pages 188–198, June 1999.

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

[CW89] Aviad Cohen and Avi Wigderson. Dispersers, deterministic amplification, and weak random sources. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 14–19, October 30 – November 1 1989.

[Din95] I. H. Dinwoodie. A probability inequality for the occupation measure of a reversible Markov chain. *Annals of Applied Probability*, 5(1):37–43, 1995.

[Fil91] J. A. Fill. Eigenvalue bounds on convergence to stationarity for nonreversible Markov chains with an application to the exclusion process. *Annals of Applied Probability*, 1:62–87, 1991.

[FK06] Lance Fortnow and Adam Klivans. Linear advice for randomized logarithmic space. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, pages 469 – 476. Springer, February 23–25 2006.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, April 1984.

[GG81] O. Gabber and Z. Galil. Explicit construction of linear size superconcentrators. *Journal of Computer and System Sciences*, 22:407–420, 1981.

[GIL+90] Oded Goldreich, Russell Impagliazzo, Leonid A. Levin, Ramarathnam Venkatesan, and David Zuckerman. Security preserving amplification of hardness. In *Proceedings of the 31st Annual IEEE Symposium on Foundations of Computer Science*, pages 318–326, 1990.

[Gil94] David Gillman. A Chernoff bound for random walks on expander graphs. In *Proceedings of the 34th Annual IEEE Symposium on Foundations of Computer Science*, pages 680–691, 1994.

[Gol97] Oded Goldreich. A sample of samplers - a computational perspective on sampling (survey). *Electronic Colloquium on Computational Complexity (ECCC)*, (TR97-020), May 1997.

[Gol99] Oded Goldreich. *Modern cryptography, probabilistic proofs and pseudorandomness*, volume 17 of *Algorithms and Combinatorics*. Springer-Verlag, Berlin, 1999.

[GV04] Dan Gutfreund and Emanuele Viola. Fooling parity tests with parity gates. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM)*, volume 3122 of *Lecture Notes in Computer Science*, pages 381–392, August 22–24 2004.

[Hås87] Johan Håstad. *Computational limitations of small-depth circuits*. MIT Press, 1987.

[HLW06] S. Hoory, N. Linial, and A. Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006.

[HV06] Alexander Healy and Emanuele Viola. Constant-depth circuits for arithmetic in finite fields of characteristic two. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science, pages 672 – 683. Springer, February 23–25 2006.

[IW97] Russell Impagliazzo and Avi Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 220–229, May 4–6 1997.

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

[Kah97] N. Kahale. Large deviation bounds for Markov chains. *Combinatorics, Probability and Computing*, 6(4):465–474, 1997.

[L9́8] P. Lézaud. Chernoff-type bound for finite Markov chains. *Annals of Applied Probability*, 8(3):849–867, 1998.

[LP04] Carlos A. León and François Perron. Optimal Hoeffding bounds for discrete reversible Markov chains. *Annals of Applied Probability*, 14(2):958–970, 2004.

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

[Mar73] G. A. Margulis. Explicit constructions of expanders. *Problemy Peredachi Informatssi; English translation, Problems of Information Transmission*, 9(4):71–80, 1973.

[Mih89] M. Mihail. Conductance and convergence of Markov chains: a combinatorial treatment of expanders. In *Proceedings of the 30th Annual IEEE Symposium on Foundations of Computer Science*, pages 526–531, October 30 – November 1 1989.

[MNT90] Yishay Mansour, Noam Nisan, and Prasoon Tiwari. The computational complexity of universal hashing. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 235–243, May 14–16 1990.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Nis91] Noam Nisan. Pseudorandom bits for constant depth circuits. *Combinatorica*, 11(1):63–70, 1991.

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12, 1992.

[NN90] J. Naor and M. Naor. Small-bias probability spaces: efficient constructions and applications. In *Proceedings of the 22nd Annual ACM Symposium on Theory of Computing*, pages 213–223, May 14–16 1990.

[Raz87] Alexander A. Razborov. Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Akademiya Nauk SSSR. Matematicheskie Zametki*, 41(4):598–607, 623, 1987.

[Rei05] Omer Reingold. Undirected st-connectivity in log-space. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing*, pages 376–385, May 21–24 2005.

[RVW02] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product and new constant-degree expanders. *Annals of Mathematics*, 155(1):157–187, January 2002.

[Sak96] Michael Saks. Randomization and derandomization in space-bounded computation. In *Proceedings of the 11th Annual IEEE Conference on Computational Complexity*, pages 128–149, May 24–27 1996.

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

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

[Vio04] Emanuele Viola. The complexity of constructing pseudorandom generators from hard functions. *Computational Complexity*, 13(3-4):147–188, 2004.

[Vio05] Emanuele Viola. Pseudorandom bits for constant-depth circuits with few arbitrary symmetric gates. In *Proceedings of the 20th Annual IEEE Conference on Computational Complexity*, June 12–15 2005.

[Vol99] Heribert Vollmer. *Introduction to circuit complexity*. Springer-Verlag, Berlin, 1999.

[WX05] Avi Wigderson and David Xiao. A randomness-efficient sampler for matrix-valued functions and applications. In *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, October 22–25 2005. See also Electronic Colloquium on Computational Complexity (ECCC) Technical Report TR05-107, `http://eccc.hpi-web.de/eccc/`.

[WX06] Avi Wigderson and David Xiao. Derandomizing the AW matrix-valued Chernoff bound using pessimistic estimators and applications. *Electronic Colloquium on Computational Complexity (ECCC)*, (TR06-105), August 2006.

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

[Zuc06] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, May 21–23 2006.