Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 58 (2006)

Randomness-Efficient Sampling within NC^1

Alexander Healy[*]

August 24, 2006

Abstract

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

- Randomness-efficient error-reduction for uniform probabilistic $NC^1, TC^0, AC^0[\oplus]$ and AC^0: Any function computable by uniform probabilistic circuits with error $1/3$ using r random bits is computable by uniform probabilistic circuits with error δ using $r + O(\log(1/\delta))$ random bits.
- An optimal explicit ϵ-biased generator in $AC^0[\oplus]$: 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 $AC^0[\oplus]$ 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 $BP \cdot AC^0 \subseteq$ uniform $AC^0/O(n)$.

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], [JS89], [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

[*] 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.

ISSN 1433-8092

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.

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 an efficient *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 $\text{poly}(n)$) 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., 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 graphs (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 NC^1 circuits. In fact, our sampler is computable by uniform constant-depth circuits with parity gates (i.e. $AC^0[\oplus]$), a class which is strictly weaker than NC^1 as it cannot even compute the majority of n bits [Raz87].

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 3, 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] and for the derandomized XOR Lemma of [IW97], as well as a variety of other applications.

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 Markov-Chain Monte Carlo algorithms (see [Gil94]). However, 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].[1]

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.[2] Since this bound is important to our analysis, we give a more formal statement of the bound 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 3 for a precise definition.)

Theorem 1 (Implicit 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 $\epsilon > 0$,*

$$\Pr\left[\left|\sum_{i=1}^{k} f_i(v_i) - \sum_{i=1}^{k} \mu_i\right| \geq \epsilon k\right] \leq 2e^{-\epsilon^2(1-\lambda)k/36}.$$

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.

[1] 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.

[2] [WX05] also gives a proof of a (strong) Chernoff bound using no perturbation theory but their bound does not match Gillman's.

2 Our Results

Our main result is the construction of a *sampler* that is computable by $AC^0[\oplus]$ 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[3] sampler *if: for any sequence of functions* $f_i : \{0,1\}^n \to [0,1]$ *each with mean* $\mu_i = \mathrm{E}_x[f_i(x)]$,

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

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 (with $\lambda = 1-\Omega(1)$) of size 2^n is a strong averaging sampler with seed-length $m = n + O(\log(1/\gamma)/\epsilon^2)$ and sample complexity $k = O(\log(1/\gamma)/\epsilon^2)$. Our main theorem is that uniform $AC^0[\oplus]$ 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* $k = O(\log(1/\gamma)/\epsilon^2)$*, that is computable by* uniform $AC^0[\oplus]$ *circuits of size* $\mathrm{poly}(n, 1/\epsilon, \log(1/\gamma))$.

At this point, the reader may wish to disregard the exact parameters of our construction, and instead think of our construction as computing (intuitively) a walk of length k on a constant-degree expander graph of size 2^n. Indeed, in most natural applications that employ random walks on expander graphs, one can safely substitute a sampler with the above parameters in place of the expander walk.

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

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 following new results:

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

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

for randomized logspace, and the expander walks of Gutfreund and Viola [GV04] imply randomness-efficient error reduction for the class $BP{\cdot}L$.[4] By applying our sampler construction, we obtain analogous error-reduction for a variety of classes below logspace:

Lemma 4. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by polynomial-size* uniform $BP \cdot AC^0[\oplus]$ *(respectively, $BP{\cdot}TC^0$ or $BP{\cdot}NC^1$) 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 $BP \cdot AC^0[\oplus]$ *(respectively $BP \cdot TC^0$ or $BP \cdot NC^1$) 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 uniform AC^0:

Lemma 5. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by polynomial-size* uniform $BP \cdot AC^0$ *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 $BP \cdot AC^0$ *circuits with error at most δ using* $\min\{r, \text{polylog}(n)\} + O(\log(1/\delta))$ *random bits.*

Derandomization with Linear Advice Recently, Fortnow and Klivans [FK06] have proved that $RL \subseteq L/O(n)$ – that is, one can derandomize randomized 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 $BP \cdot AC^0 \subseteq$ uniform $AC^0/O(n)$.

Ajtai and Ben-Or [ABO84] have shown that nonuniform $BP \cdot AC^0 =$ nonuniform AC^0; even for derandomizing uniform $BP \cdot AC^0$ [Ajt93], however, their technique seems to require an arbitrary polynomial amount of non-uniform advice. Theorem 6 quantifies the amount of nonuniformity that is necessary to derandomize a probabilistic AC^0 circuit, and therefore can be viewed as a refinement of their result.

A similar approach, together with a new pseudorandom generator of Viola [Vio05], yields the following:

Corollary 7. *Let $AC^0[\oplus_{\text{log}}]$ be the class of boolean functions computable by* $\text{poly}(n)$*-size AC^0 circuits having $O(\log n)$ parity gates, and similarly let $AC^0[\text{SYM}_{\text{log}}]$ be the class of boolean functions computable by* $\text{poly}(n)$*-size AC^0 circuits having $O(\log n)$ arbitrary symmetric gates (e.g., parity and majority gates). Then the following inclusions hold:*

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

[4] $BP \cdot L$ refers to randomized logspace computations that are allowed *two-way* access to the random bits, whereas the result of Bar-Yosef et al. refers to algorithms that have only *one-way* access to the random bits. See the survey of Saks [Sak96] for a discussion of the subtleties surrounding different notions of randomized space-bounded computation.

An Optimal Explicit ϵ-Biased Generator in $\mathbf{AC^0}[\oplus]$ Gutfreund and Viola [GV04] study the complexity of constructing *explicit* ϵ-biased generators (see Definition 9). They give a construction in $AC^0[\oplus]$ 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 TC^0 and a sub-optimal construction in $AC^0[\oplus]$ whose parameters are incomparable to those of [GV04]. In this work, we resolve this question entirely: using our sampler construction, we construct an *optimal* explicit ϵ-biased generator in $AC^0[\oplus]$:

Corollary 8 ([NN90] + [GV04] + Theorem 3)**.** *For every $\epsilon > 0$ and m, there is an ϵ-biased generator $G : \{0,1\}^n \to \{0,1\}^m$ with $n = O(\log m + \log(1/\epsilon))$ for which uniform $AC^0[\oplus]$ 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 such ϵ-biased generators require seed length $\Omega(\log m + \log(1/\epsilon))$ [AGHP92], and it can be shown that an explicit ϵ-biased generators acheiving the parameters of Corollary 8 requires AC^0 circuits of exponential size [GV04, MNT90]. Therefore, the construction of Corollary 8 is tight both in terms of seed-length and computational complexity.

3 Preliminaries

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

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 11). Second, one of the applications of our sampler is to build exponential-size ϵ-biased sets which are computable *explicitly* (see the definition below and Corollary 8).

Definition 9. *For $a, b \in \mathbb{F}_2^m$, let $\langle a, b\rangle_2$ denote the inner product of a and b modulo* 2.

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

An ϵ-biased generator *is a function $\Gamma : \{0,1\}^\ell \to \{0,1\}^m$ whose range is an ϵ-biased multi-set.*

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

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., [Gol99] and the references therein), it will be most convenient for us to work with the following spectral definition.

Definition 10. *Let G be a regular directed graph*[5] *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*

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

(When G is undirected, this is equivalent to the second-largest eigenvalue of P being at most λ in absolute value – see, e.g., [Mih89, Fil91, RTV04].)

We will often abuse language and refer to an "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 $(1 - \Omega(1))$-expander.

By a *random walk* $v_1, \ldots, v_k$ 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. Note that we are implicitly discarding the start vertex v_0 – while it is easy to see that the distribution is unchanged even if we keep v_0, we prefer this convention as it will simplify 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.

Constant-Depth Circuits We consider three classes of unbounded fan-in constant-depth circuits: circuits over the bases $\{\wedge, \vee \neg\}$ (i.e., AC^0), $\{\wedge, \vee, \mathit{Parity}, \neg\}$ (i.e., $AC^0[\oplus]$), and $\{\wedge, \vee, \mathit{Majority}, \neg\}$ (i.e., TC^0). Unless explicitly stated otherwise, all circuits are of polynomial size and *uniform* – specifically, we adopt the standard of *Dlogtime*-uniformity, 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 [BIS90]. Informally, a circuit is *Dlogtime*-uniform if, given 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, $AC^0/O(n)$ is the class of boolean functions f that are computable by a *Dlogtime*-uniform 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 $BP \cdot AC^0, BP \cdot AC^0[\oplus], BP \cdot TC^0$ and $BP \cdot NC^1$ 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 $AC^0 \subsetneq AC^0[\oplus] \subsetneq TC^0 \subseteq NC^1 \subseteq$ logspace, where the last inclusion holds under logspace

[5] A directed graph is *d-regular* if the in-degree and out-degree of every node is equal to some fixed d, and a directed graph is *regular* if it is d-regular for some d.

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, AC^0 can compute the *approximate* majority of n bits [Ajt93] – in particular, for any constant $\epsilon > 0$, there exists a family of 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.

4 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 $AC^0[\oplus]$ of size $\mathrm{poly}(n,k)$. Unfortunately, we do not know of any such family of expanders. Instead, we begin with a family of expander graphs of degree $\mathrm{poly}(n)$ where walks are computable in $AC^0[\oplus]$ – 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]. By [GV04], it is known that AC^0 circuits can compute walks of length $\log n$ on a Gabber-Galil graph of size 2^n, so in the sequel we focus on the case where $k = \Omega(\log n)$. We now describe the construction in more detail.

Our first graph, G, is a Cayley graph on the group $\mathbb{F}_2^n$. Specifically, we construct a $1/n$-biased set $S \subset \mathbb{F}_2^n$ of size $\mathrm{poly}(n)$ (see Definition 9) and let $\{v, w\}$ be an edge if and only if $v - w \in S$. The following well-known fact guarantees that G has second-largest eigenvalue at most $2/n$ (e.g., see [AR94]).

Lemma 11. *A Cayley graph on $\mathbb{F}_2^n$ with generators $S \subset \mathbb{F}_2^n$ 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 $AC^0[\oplus]$. First, we note that a $1/n$-biased set S of size $\mathrm{poly}(n)$ can be constructed in 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].[6] (Note that if we only wished to give a non-uniform construction, we could simply hard-wire such an ϵ-biased set into the circuit.)

[6] 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 $\langle \alpha^i, \beta \rangle \pmod 2$. 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 AC^0 of size $\mathrm{poly}(n)$ for this range of parameters.

Thus, given the description a walk $(v, s_1, \ldots, s_k) \in \{0,1\}^n \times \{0,1\}^{O(\log n)} \times \cdots \times \{0,1\}^{O(\log n)}$, to determine the i-th vertex visited by the walk, the circuit need only compute from each index s_j (in parallel) the appropriate vector $v_{s_j} \in S$ and then compute the sum

$$v + \sum_{j=1}^{i} v_{s_j}.$$

Since the summation is modulo 2, this is easily seen to be computable in $AC^0[\oplus]$ of size poly(n, k).

Now we turn to the problem of producing a *pseudorandom* sequence of steps s_i, with the goal of reducing the seed length of a walk on G, while at the same time preserving the sampling properties 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 will 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, visiting nodes $w_1, \ldots, w_k$
3. View $w_1, \ldots, w_k$ as indices in $[d] = [|H|]$
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 this walk

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 above construction is a strong averaging sampler. What is not clear, however, is how to compute this sampler in $AC^0[\oplus]$. 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 $AC^0[\oplus]$ (or even in 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 12.

1. *Choose a random starting vertex* $v_0 \in G$
2. *Take* $k/\log n$ *random walks 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* $[d] = [|H|]$
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 this walk*

This sampler has seed length $|v_0| + (k/\log n)(\log|H| + O(\log n)) = n + O(k)$ (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 it is computable in $AC^0[\oplus]$. Indeed, it is known how to compute walks of length $O(\log n)$ on poly-sized explicit expanders of constant degree in AC^0 [Ajt93, GV04],[7] and thus each of the five steps above is computable in constant depth.

We now show that Construction 12 is a strong averaging sampler. In particular, Theorem 3 is a consequence of the following lemma:

Lemma 13. *Let $H = \tilde{H}^2$ where $\tilde{H}$ is a constant-degree expander graph on* $\mathrm{poly}(n)$ *nodes. Then Construction 12 is a strong 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.[8] (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 $G \textcircled{z} g$ where:

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

Thus, if we start at $(v, w) \in G \textcircled{z} g$, 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 $G \textcircled{z} g$ (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 $v_1, v_2, \ldots, v_\ell$.

[7] As with the $1/n$-biased set S above, the delicate 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.

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

Thus, each of of the segments of length $k/\log n$ in our sampler construction corresponds to a random walk on $G \textcircled{z} \tilde{H}$, projected onto the V_G-coordinate. But what about the boundaries between these segments? In this case, Construction 12 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 $G \textcircled{z} K_d$, 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 $G \textcircled{z} \tilde{H}$ most of the time, and $G \textcircled{z} K_d$ 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 $\hat{f}(v,w) = f(v)$. 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 time-varying graph is a strong averaging sampler. By the remark after the proof of Theorem 1, it does not matter if the graph is varying over time: as long as it is a regular λ-expander at every point in time, Theorem 1 holds (and so the random walk is a good sampler). Thus, we are left with the task of showing that $G \textcircled{z} \tilde{H}$ and $G \textcircled{z} K_d$ are expanders. For this, we apply the following consequence of the main theorem of [RVW02]:

Lemma 14 ([RVW02], Corollary to Theorem 4.3)**.** *Let G be a regular graph of degree d whose edges are labeled with $1, \ldots, d$ 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 $G \textcircled{z} g$ is a $(\lambda_G + \lambda_g)$-expander.*

By Lemma 11, G is a $2/n$-expander, and by assumption $\tilde{H}$ is a $(1 - \Omega(1))$-expander, and so by Lemma 14, $G \textcircled{z} \tilde{H}$ is a $(1 - \Omega(1))$-expander.

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 14, $G \textcircled{z} K_d$ is a $2/n$-expander.

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 with $\lambda = 1 - \Omega(1)$. 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 $k = O(\log(1/\gamma)/\epsilon^2)$. □

5 Proofs of Other Results

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

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 $BP \cdot TC^0$ and $BP \cdot NC^1$ 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 $\leq 1/3$ fraction 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. By choosing the constants appropriately, this is at most δ and the result follows. □

Lemma 5. *Let $f : \{0,1\}^n \to \{0,1\}$ be a function computable by polynomial-size* uniform $BP \cdot AC^0$ *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 $BP \cdot AC^0$ *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 $BP \cdot AC^0$ [Nis91] (which has been shown to be computable in 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 applying the construction of Lemma 4 to obtain a $BP \cdot AC^0$ circuit that has error at most δ and uses $r' + O(\log(1/\delta))$ bits of randomness. (The circuit is in $BP \cdot AC^0$, and not just $BP \cdot AC^0[\oplus]$ 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 in AC^0.)

If, on the other hand, $\delta < 1/2^{r'}$, then we apply Lemma 4 with $\delta(n) = 2^{-r'}$ to obtain an AC^0 circuit that has error at most $2^{-r'}$ and uses $r' + O(r') \leq O(r')$ random bits. By applying $\Theta(\log(1/\delta)/r')$ such circuits in parallel (on the same input, but independent random strings), and taking the approximate majority of their $\Theta(\log(1/\delta)/r')$ outputs, 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]), which is at most δ for an appropriate setting of constants. □

Derandomization with Linear Advice

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

Proof. Apply Lemma 5 to obtain a $BP \cdot AC^0$ 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. □

Corollary 7. *Let $AC^0[\oplus_{\log}]$ be the class of boolean functions computable by* $\mathrm{poly}(n)$*-size AC^0 circuits having $O(\log n)$ parity gates, and similarly let $AC^0[\mathrm{SYM}_{\log}]$ be the class of boolean functions computable by* $\mathrm{poly}(n)$*-size AC^0 circuits having $O(\log n)$ arbitrary symmetric gates (e.g., parity and majority gates). Then the following inclusions hold:*

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

Proof sketch. The proof is similar to the proof of Corollary 5 and Theorem 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 $f \in BP \cdot AC^0[\oplus_{\log}]$ (respectively, $BP \cdot AC^0[\mathrm{SYM}_{\log}]$) can be computed by a $BP \cdot AC^0[\oplus]$ (respectively, $BP \cdot TC^0$) circuit using only $n^{o(1)}$ random bits. By applying Lemma 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. □

Optimal explicit ϵ-biased generator in $AC^0[\oplus]$

Corollary 8 ([NN90] + [GV04] + Theorem 3)**.** *For every $\epsilon > 0$ and m, there is an ϵ-biased generator $G : \{0,1\}^n \to \{0,1\}^m$ with $n = O(\log m + \log(1/\epsilon))$ for which uniform $AC^0[\oplus]$ 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 requires a 7-wise independent generator and a long walk on an expander graph. Constructions of 7-wise independent generators in $AC^0[\oplus]$ are known [GV04, HV06]. Since the use of an expander walk in [NN90] is simply as a hitting generator, our construction from Section 4 is more than adequate for this purpose. □

6 The Proof of Theorem 1

In this section we give an elementary proof of the following generalization of Gillman's *Chernoff Bound for Expander Walks* [Gil94].

Theorem 1. *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 $\epsilon > 0$,*

$$\Pr\left[\left|\sum_{i=1}^{k} f_i(v_i) - \sum_{i=1}^{k} \mu_i\right| \geq \epsilon k\right] \leq 2e^{-\epsilon^2(1-\lambda)k/36}.$$

Wigderson and Xiao [WX05] have recently established the same bound (up to constants) using techniques from perturbation theory – Gillman's proof (which treats the case $S_1 = \cdots = S_k$) 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 10). 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

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

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 $\mathbf{z}$ that lies in the orthogonal complement of $\mathbf{1}$. Thus, for any $\mathbf{z} \in \mathbb{R}^N$, we have that $P\mathbf{z}^{\|} = \mathbf{z}^{\|}$ and $\|P\mathbf{z}^{\perp}\| \leq \lambda \|\mathbf{z}^{\perp}\|$. Another useful fact is that for any $\mathbf{z} \in \mathbb{R}^N$, $P\mathbf{z}^{\perp}$ is orthogonal to $\mathbf{1}$ simply because $\langle \mathbf{1}, P\mathbf{z}^{\perp} \rangle = \mathbf{1}^T P \mathbf{z}^{\perp}$ and $\mathbf{1}^T P = \mathbf{1}$ since we assume G is regular (in particular, both in-regular and out-regular if it is a directed graph).

Proof of Theorem 1. Let $T = \sum_i f_i(v_i)$. We shall bound the quantity $\Pr\left[T - \sum_i \mu_i \geq \epsilon k\right]$ and the same bound will follow for $\Pr\left[T - \sum_i \mu_i \leq -\epsilon k\right]$ by replacing the functions $f_i(v)$ with $1 - f_i(v)$. Let $r \leq \min\{1, \log(1/\lambda)/2\}$ be a positive parameter to be specified later.

$$\Pr\left[T - \sum_{i=1}^{k} \mu_i \geq \epsilon k\right] = \Pr\left[T \geq \epsilon k + \sum_{i=1}^{k} \mu_i\right] = \Pr\left[e^{rT} \geq e^{r\left(\epsilon k + \sum_{i=1}^{k} \mu_i\right)}\right] \leq \frac{\mathrm{E}\left[e^{rT}\right]}{e^{r\left(\epsilon k + \sum_{i=1}^{k} \mu_i\right)}} \tag{1}$$

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

We now focus on bounding $\mathrm{E}\left[e^{rT}\right]$. 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

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

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

To bound this quantity, 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

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

and so our goal is to bound $\|\mathbf{z}_k^{\|}\|$. Intuitively, we accomplish this by showing 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 while P leaves such vectors unchanged, E_i stretches the vector $\mathbf{u}$ 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 $\mathrm{E}\left[e^{rT}\right] \approx e^{r\sum_i \mu_i}$. More precisely, we obtain a bound of $\mathrm{E}\left[e^{rT}\right] \leq e^{r\sum_i \mu_i + 9kr^2/(1-\lambda)}$, which bounds the probability in Equation 1 by $e^{9kr^2/(1-\lambda) - \epsilon kr}$, and the result follows by choosing r to minimize this probability (i.e., choosing $r = \epsilon(1-\lambda)/18$).

We first show that $\mathbf{z}_i^{\perp}$ remains short compared to the $\mathbf{z}_j^{\|}$'s.

Lemma 15. $\|\mathbf{z}_i^{\perp}\| \leq \frac{4r}{1-\lambda} \cdot \max\limits_{j<i}\{\|\mathbf{z}_j^{\|}\|\}$ *for* $1 \leq i \leq k$.

Proof. By the triangle inequality,

$$\|\mathbf{z}_i^{\perp}\| = \|(E_i P \mathbf{z}_{i-1})^{\perp}\| = \|(E_i P \mathbf{z}_{i-1}^{\|})^{\perp} + (E_i P \mathbf{z}_{i-1}^{\perp})^{\perp}\| \leq \|(E_i P \mathbf{z}_{i-1}^{\|})^{\perp}\| + \|(E_i P \mathbf{z}_{i-1}^{\perp})^{\perp}\|.$$

For the first term,

$$(E_i P \mathbf{z}_{i-1}^{\|})^{\perp} = (E_i \mathbf{z}_{i-1}^{\|})^{\perp} = ((E_i - I)\mathbf{z}_{i-1}^{\|})^{\perp} + (\mathbf{z}_i^{\|})^{\perp} = ((E_i - I)\mathbf{z}_{i-1}^{\|})^{\perp},$$

and since $E_i - I$ is diagonal with entries bounded by $e^r - 1$, we have

$$\|(E_i P \mathbf{z}_{i-1}^{\|})^{\perp}\| = \|((E_i - I)\mathbf{z}_{i-1}^{\|})^{\perp}\| \leq (e^r - 1) \cdot \|\mathbf{z}_{i-1}^{\|}\| \leq 2r \cdot \|\mathbf{z}_{i-1}^{\|}\|$$

(since $r \leq 1$ and therefore $e^r \leq 1 + r + r^2 \leq 1 + 2r$).

For the second term,

$$\|(E_i P \mathbf{z}_{i-1}^{\perp})^{\perp}\| \leq \|E_i P \mathbf{z}_{i-1}^{\perp}\| \leq e^r \cdot \|P \mathbf{z}_{i-1}^{\perp}\| \leq e^r \lambda \cdot \|\mathbf{z}_{i-1}^{\perp}\|,$$

and since we assume that $r \leq \log(1/\lambda)/2$, this is at most $\sqrt{\lambda} \cdot \|\mathbf{z}_{i-1}^{\perp}\|$.

Combining these two bounds, $\|\mathbf{z}_i^{\perp}\| \leq 2r \cdot \|\mathbf{z}_{i-1}^{\|}\| + \sqrt{\lambda} \cdot \|\mathbf{z}_{i-1}^{\perp}\|$.

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

$$\|\mathbf{z}_i^{\perp}\| \leq 2r \cdot \sum_{j=0}^{i-1} (\sqrt{\lambda})^j \|\mathbf{z}_{i-j-1}^{\|}\| \leq \frac{2r}{1-\sqrt{\lambda}} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\} \leq \frac{2r(1+\sqrt{\lambda})}{1-\lambda} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\} \leq \frac{4r}{1-\lambda} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}.$$

□

We now use Lemma 15 to bound $\|\mathbf{z}_i^{\|}\|$.

Lemma 16. $\|\mathbf{z}_i^{\|}\| \leq e^{r\mu_i + 9r^2/(1-\lambda)} \cdot \max\limits_{j<i}\{\|\mathbf{z}_j^{\|}\|\}$, *for* $1 \leq i \leq k$.

Proof. By the triangle inequality,

$$\|\mathbf{z}_i^{\|}\| = \|(E_i P \mathbf{z}_{i-1})^{\|}\| = \|(E_i P \mathbf{z}_{i-1}^{\|})^{\|} + (E_i P \mathbf{z}_{i-1}^{\perp})^{\|}\| \leq \|(E_i P \mathbf{z}_{i-1}^{\|})^{\|}\| + \|(E_i P \mathbf{z}_{i-1}^{\perp})^{\|}\|.$$

For the first term,

$$(E_i P \mathbf{z}_{i-1}^{\|})^{\|} = (E_i \mathbf{z}_{i-1}^{\|})^{\|} = \langle \mathbf{1}, E_i \mathbf{z}_{i-1}^{\|} \rangle \mathbf{u} = \langle \mathbf{1}, E_i \mathbf{u} \rangle \mathbf{z}_{i-1}^{\|} = \mathop{\mathrm{E}}_{v}\left[e^{rf_i(v)}\right] \cdot \mathbf{z}_{i-1}^{\|},$$

and using the fact that $e^x \leq 1 + x + x^2$ for $x \in [0,1]$, we have

$$\|(E_i P \mathbf{z}_{i-1}^{\|})^{\|}\| \leq \mathop{\mathrm{E}}_{v}\left[e^{rf_i(v)}\right] \cdot \|\mathbf{z}_{i-1}^{\|}\| \leq \mathop{\mathrm{E}}_{v}\left[1 + rf_i(v) + r^2 f_i(v)^2\right] \cdot \|\mathbf{z}_{i-1}^{\|}\| \leq (1 + r\mu_i + r^2) \cdot \|\mathbf{z}_{i-1}^{\|}\|.$$

For the second term,

$$(E_i P \mathbf{z}_{i-1}^{\perp})^{\|} = ((E_i - I)P\mathbf{z}_{i-1}^{\perp})^{\|} + (P\mathbf{z}_{i-1}^{\perp})^{\|} = ((E_i - I)P\mathbf{z}_i^{\perp})^{\|},$$

and since $E_i - I$ is diagonal with entries bounded by $e^r - 1$, we have

$$\|(E_i P\mathbf{z}_{i-1}^{\perp})^{\|}\| \leq \|((E_i - I)P\mathbf{z}_i^{\perp})^{\|}\| \leq (e^r - 1) \cdot \|P\mathbf{z}_{i-1}^{\perp}\| \leq (e^r - 1) \cdot \|\mathbf{z}_{i-1}^{\perp}\| \leq 2r \cdot \|\mathbf{z}_{i-1}^{\perp}\|$$

(since $r \leq 1$ and therefore $e^r \leq 1 + r + r^2 \leq 1 + 2r$).

Combining these two bounds, $\|\mathbf{z}_i^{\|}\| \leq (1 + r\mu_i + r^2) \cdot \|\mathbf{z}_{i-1}^{\|}\| + 2r \cdot \|\mathbf{z}_{i-1}^{\perp}\|$ and by Lemma 15,

$$\|\mathbf{z}_i^{\|}\| \leq (1 + r\mu_i + r^2) \cdot \|\mathbf{z}_{i-1}^{\|}\| + \frac{8r^2}{1-\lambda} \cdot \max_{j<i-1}\{\|\mathbf{z}_j^{\|}\|\} \leq \left(1 + r\mu_i + \frac{9r^2}{1-\lambda}\right) \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}.$$

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

$$e^{r\mu_i + 9r^2/(1-\lambda)} \cdot \max_{j<i}\{\|\mathbf{z}_j^{\|}\|\}.$$

□

Recalling that $\|\mathbf{z}_0^{\|}\| = 1/\sqrt{N}$, Lemma 16 implies that for all $j \geq 0$:

$$\|\mathbf{z}_j^{\|}\| \leq \frac{1}{\sqrt{N}} \prod_{i=1}^{j} e^{r\mu_i + 9r^2/(1-\lambda)},$$

and in particular, by Equation 3,

$$\mathrm{E}\left[e^{rT}\right] \leq \sqrt{N} \cdot \|\mathbf{z}_k^{\|}\| \leq \prod_{i=1}^{k} e^{r\mu_i + 9r^2/(1-\lambda)} = e^{r\sum_{i=1}^{k}\mu_i + 9r^2/(1-\lambda)}.$$

Thus, by Equation 1,

$$\Pr\left[T - \sum_{i=1}^{k}\mu_i \geq \epsilon k\right] \leq \frac{\mathrm{E}\left[e^{rT}\right]}{e^{r\left(\epsilon k + \sum_{i=1}^{k}\mu_i\right)}} \leq e^{9r^2/(1-\lambda) - \epsilon k r},$$

and finally, we minimize the right-hand side by setting $r = \epsilon(1-\lambda)/18$, noting that r is indeed at most $\min\{1, \log(1/\lambda)/2\}$ simply because $1 - \lambda \leq \log(1/\lambda)$ for all $\lambda \in [0,1]$.

$$\Pr\left[T - \sum_{i=1}^{k}\mu_i \geq \epsilon k\right] \leq e^{k\left(\frac{9}{1-\lambda}\cdot\frac{\epsilon^2(1-\lambda)^2}{18^2} - \frac{\epsilon^2(1-\lambda)}{18}\right)} = e^{-\frac{\epsilon^2(1-\lambda)k}{36}}.$$

□

Remark 17. *One can readily see that the same proof works 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).*

7 Open Questions

It is well-known that expander walks yield averaging-samplers that are optimal (up to constants) for $\epsilon = \Omega(1)$, but sub-optimal for smaller ϵ. Since pairwise-hashing is in $AC^0[\oplus]$ [GV04, HV06], one can implement the *median-of-averages sampler* of [BGG93] in TC^0 by using our sampler in lieu of the expander walks. (Majority gates are only necessary to compute the medians and averages – the actual samples can be computed in $AC^0[\oplus]$.) Can $AC^0[\oplus]$ compute an optimal *averaging* sampler?

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

8 Acknowledgements

Thanks to Kai-Min Chung, Danny Gutfreund, Salil Vadhan, Emanuele Viola and David Zuckerman for various discussions and for their encouragement – Emanuele also provided helpful comments on a draft of this paper. Thanks also to David Xiao for an email exchange about [WX05] 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.

[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.

[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.

[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)*, 4(020), 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.

[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.

[JS89] Mark Jerrum and Alistair Sinclair. Approximating the permanent. *SIAM Journal on Computing*, 18(6):1149–1178, 1989.

[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.

[RTV04] Omer Reingold, Luca Trevisan, and Salil Vadhan. Notions of reducibility between cryptographic primitives. In *Proceedings of the 1st Theory of Cryptography Conference*. Springer-Verlag, February 19–21 2004.

[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.

[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. Unpublished manuscript, July 2006.

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

ECCC ISSN 1433-8092
http://eccc.hpi-web.de/