On Pseudorandom Generators with Linear Stretch in NC^0 *

Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz

Computer Science Department, Technion, Haifa 32000, Israel
{abenny,yuvali,eyalk}@technion.ac.il

Abstract. We consider the question of constructing cryptographic pseudorandom generators (PRGs) in NC^0, namely ones in which each bit of the output depends on just a constant number of input bits. Previous constructions of such PRGs were limited to stretching a seed of n bits to $n + o(n)$ bits. This leaves open the existence of a PRG with a linear (let alone superlinear) stretch in NC^0. In this work we study this question and obtain the following main results:

1. We show that the existence of a linear-stretch PRG in NC^0 implies non-trivial hardness of approximation results *without relying on PCP machinery*. In particular, that Max 3SAT is hard to approximate to within some constant.
2. We construct a linear-stretch PRG in NC^0 under a specific intractability assumption related to the hardness of decoding "sparsely generated" linear codes. Such an assumption was previously conjectured by Alekhnovich [1].

We note that Alekhnovich directly obtains hardness of approximation results from the latter assumption. Thus, we do not prove hardness of approximation under new *concrete* assumptions. However, our first result is motivated by the hope to prove hardness of approximation under more general or standard cryptographic assumptions, and the second result is independently motivated by cryptographic applications.

1 Introduction

A cryptographic pseudorandom generator (PRG) [8, 24] is a deterministic function that stretches a short random seed into a longer string which cannot be distinguished from random by any polynomial-time observer. In this work, we study the existence of PRGs that are both (1) extremely parallel and (2) stretch their seed by a significant amount.

Considering the first goal alone, it was recently shown in [3] that the ultimate level of parallelism can be achieved under most standard cryptographic assumptions. Specifically, any PRG in NC^1 (the existence of which follows, for example, from the intractability of factoring, discrete logarithm, or lattice problems) can be efficiently "compiled" into a PRG in NC^0, namely one in which each output bit depends on just a constant number of input bits. However, the PRGs produced by this compiler can only stretch their seed by a sublinear amount: from n bits to $n+O(n^\epsilon)$ bits for some constant $\epsilon < 1$. Thus, these PRGs do not meet our second goal.

Considering the second goal alone, even a PRG that stretches its seed by just one bit can be used to construct a PRG that stretches its seed by any polynomial number of

* Research supported by grant 36/03 from the Israel Science Foundation.

bits. However, all known constructions of this type are inherently sequential. Thus, we cannot use known techniques for turning an NC^0 PRG with a sublinear stretch into one with a linear, let alone superlinear, stretch.

The above state of affairs leaves open the existence of a *linear-stretch* PRG (LPRG) in NC^0; namely, one that stretches a seed of n bits into $n + \Omega(n)$ output bits.[1] (In fact, there was no previous evidence for the existence of LPRGs even in the higher complexity class AC^0.) This question is the main focus of our work. The question has a very natural motivation from a cryptographic point of view. Indeed, most cryptographic applications of PRGs either require a linear stretch (for example Naor's bit commitment scheme [19]), or alternatively depend on a larger stretch for efficiency (this is the case for the standard construction of a stream cipher or stateful symmetric encryption from a PRG, see [14]). Thus, the existence of an LPRG in NC^0 would imply better parallel implementations of other cryptographic primitives.

1.1 Our Contribution

LPRG in NC^0 implies hardness of approximation. We give a very different, and somewhat unexpected, motivation for the above question. We observe that the existence of an LPRG in NC^0 *directly* implies non-trivial and useful hardness of approximation results. Specifically, we show (via a very simple argument) that an LPRG in NC^0 implies that Max 3SAT cannot be efficiently approximated to within some multiplicative constant. This continues a recent line of work, initiated by Feige [12] and followed by Alekhnovich [1], that provides simpler alternatives to the traditional PCP-based approach by relying on stronger assumptions. Unlike these previous works, which rely on very specific assumptions, our assumption is of a more general flavor and may serve to further motivate the study of cryptography in NC^0. On the down side, the conclusions we get are weaker and in particular are implied by the PCP theorem. In contrast, some inapproximability results from [12, 1] could not be obtained using PCP machinery. It is instructive to note that by applying our general argument to the sublinear-stretch PRGs in NC^0 from [3] we only get "uninteresting" inapproximability results that follow from standard padding arguments (assuming P$\neq$NP). Furthermore, we do not know how to obtain stronger inapproximability results based on a superlinear-stretch PRG in NC^0. Thus, our main question of constructing LPRGs in NC^0 captures precisely what is needed for this application.

Constructing an LPRG in NC^0. We present a construction of an LPRG in NC^0 under a specific intractability assumption related to the hardness of decoding "sparsely generated" linear codes. Such an assumption was previously conjectured by Alekhnovich in [1]. The starting point of our construction is a modified version of a PRG from [1] that has a large output locality (that is, each output bit depends on many input bits) but has a simple structure. The main technical tool we employ in order to reduce its locality is a randomness extractor in NC^0 that can use a "sufficiently short" seed for sources with a "sufficiently high" entropy. We construct the latter by combining the known

[1] Note that an NC^0 LPRG can be composed with itself a constant number of times to yield an NC^0 PRG with arbitrary constant stretch.

construction of randomness extractors from ϵ-biased generators [18, 6] with previous constructions of ϵ-biased generator in NC^0 [17]. Our LPRG can be implemented with locality 4; this LPRG is essentially optimal, as it is known that no PRG with locality 4 can have a *superlinear* stretch [17]. However, the existence of superlinear-stretch PRG with a higher (but constant) locality remains open.

By combining the two main results described above, one gets non-trivial inapproximability results under the intractability assumption from [1]. These (and stronger) results were *directly* obtained in [1] from the same assumption *without* constructing an LPRG in NC^0. Our hope is that future work will yield constructions of LPRGs in NC^0 under different, perhaps more standard, assumptions, and that the implications to hardness of approximation will be strengthened.

LPRG in NC^0 and Expanders. Finally, we observe that any LPRG in NC^0 contains a copy of a graph with some non-trivial expansion property. This connection implies that a (deterministic) construction of an LPRG in NC^0 must use some non-trivial combinatorial objects. (In particular, one cannot hope for "simple" transformations, such as those given in [3], to yield LPRGs in NC^0.) The connection with expanders also allows to rule out the existence of *exponentially*-strong PRGs with *superlinear* stretch in NC^0.

1.2 Related Work

The existence of PRGs in NC^0 has been recently studied in [10, 17, 3]. Cryan and Miltersen [10] observe that there is no PRG in NC^0_2 (i.e., where each output bit depends on at most two input bits), and prove that there is no PRG in NC^0_3 achieving a superlinear stretch; namely, one that stretches n bits to $n + \omega(n)$ bits. Mossel et al. [17] extend this impossibility to NC^0_4. Viola [23] shows that an LPRG in AC^0 cannot be obtained from a OWF via non-adaptive black-box constructions. This result can be extended to rule out such a construction even if we start with a PRG whose stretch is sublinear.

On the positive side, Mossel et al. [17] constructed (non-cryptographic) ε-biased generators with linear stretch and exponentially small bias in NC^0_5. Later, in [3] it was shown that, under standard cryptographic assumptions, there are pseudorandom generators in NC^0_4. However, these PRGs have only *sublinear-stretch*.

The first application of average-case complexity to inapproximability was suggested by Feige [12], who derived new inapproximability results under the assumption that refuting 3SAT is hard on average on some natural distribution. Alekhnovich [1] continued this line of research. He considered the problem of determining the maximal number of satisfiable equations in a linear system chosen at random, and made several conjectures regarding the average case hardness of this problem. He showed that these conjectures imply Feige's assumption as well as several new inapproximability results. While the works of Feige and Alekhnovich derived *new* inapproximability results (that were not known to hold under the assumption that $\mathrm{P} \neq \mathrm{NP}$), they did not rely on the relation with a standard cryptographic assumption or primitive, but rather used specific average case hardness assumptions tailored to their inapproximability applications. A relation between the security of a cryptographic primitive and approximation was implicitly used in [17], where an approximation algorithm for Max 2LIN was used to derive an upper bound on the stretch of a PRG whose locality is 4.

2 Preliminaries

Probability notation. We use U_n to denote a random variable uniformly distributed over $\{0,1\}^n$. If X is a probability distribution, or a random variable, we write $x \leftarrow X$ to indicate that x is a sample taken from X. The *min-entropy* of a random variable X is defined as $\mathrm{H}_\infty(X) \stackrel{\text{def}}{=} \min_x \log(\frac{1}{Pr[X=x]})$. The *statistical distance* between discrete probability distributions Y and Y', denoted $\|Y - Y'\|$, is defined as the maximum, over all functions A, of the *distinguishing advantage* $|\Pr[A(Y)=1] - \Pr[A(Y')=1]|$.

A function $\varepsilon(\cdot)$ is said to be *negligible* if $\varepsilon(n) < n^{-c}$ for any constant $c > 0$ and sufficiently large n. We will sometimes use $\mathrm{neg}(\cdot)$ to denote an unspecified negligible function. For two distribution ensembles $\{X_n\}_{n\in\mathbb{N}}$ and $\{Y_n\}_{n\in\mathbb{N}}$, we write $X_n \equiv Y_n$ if X_n and Y_n are identically distributed, and $X_n \stackrel{\text{s}}{\approx} Y_n$ if the two ensembles are *statistically indistinguishable*; namely, $\|X_n - Y_n\|$ is negligible in n. A weaker notion of closeness between distributions is that of *computational* indistinguishability: We write $X_n \stackrel{\text{c}}{\approx} Y_n$ if for every (non-uniform) polynomial-size circuit family $\{A_n\}$, the distinguishing advantage $|\Pr[A_n(X_n)=1] - \Pr[A_n(Y_n)=1]|$ is negligible. By definition, $X_n \equiv Y_n$ implies that $X_n \stackrel{\text{s}}{\approx} Y_n$ which in turn implies that $X_n \stackrel{\text{c}}{\approx} Y_n$. A distribution ensemble $\{X_n\}_{n\in\mathbb{N}}$ is said to be *pseudorandom* if $X_n \stackrel{\text{c}}{\approx} U_n$.

We will use the following definition of a pseudorandom generator.

Definition 1. (Pseudorandom generator) *A pseudorandom generator (PRG) is a deterministic function $G : \{0,1\}^* \to \{0,1\}^*$ satisfying the following two conditions:*

- ***Expansion**: There exists a* stretch function $s : \mathbb{N} \to \mathbb{N}$ *such that $s(n) > n$ for all $n \in \mathbb{N}$ and $|G(x)| = s(|x|)$ for all $x \in \{0,1\}^*$.*
- ***Pseudorandomness**: The ensembles $\{G(U_n)\}_{n\in\mathbb{N}}$ and $\{U_{s(n)}\}_{n\in\mathbb{N}}$ are computationally indistinguishable.*

When $s(n) = n + \Omega(n)$ we say that G is a linear-stretch *pseudorandom generator (LPRG). By default, we require G to be polynomial time computable.*

It will sometimes be convenient to define a PRG by an infinite family of functions $\{G_n : \{0,1\}^{m(n)} \to \{0,1\}^{s(n)}\}_{n\in\mathbb{N}}$. Such a family can be transformed into a single function that satisfies Definition 1 via padding. We will also rely on *ε-biased generators*, defined similarly to PRGs except that the pseudorandomness holds only against linear functions. Namely, for a bias function $\varepsilon : \mathbb{N} \to (0,1)$ we say that $G : \{0,1\}^n \to \{0,1\}^{s(n)}$ is an ε-biased generator if for every non-constant linear function $L : \mathrm{GF}_2^n \to \mathrm{GF}_2$ and all sufficiently large n's it holds that $|\Pr[L(G(U_n))=1] - \frac{1}{2}| < \varepsilon(n)$.

Locality. We say that $f : \{0,1\}^n \to \{0,1\}^s$ is *c-local* if each of its output bits depends on at most c input bits, and that $f : \{0,1\}^* \to \{0,1\}^*$ is c-local if for every n the restriction of f to n-bit inputs is c-local. The uniform versions of these classes contain functions that can be computed in polynomial time.

3 LPRG in NC^0 implies Hardness of Approximation

In the following we show that if there exists an LPRG in NC^0 then there is no polynomial-time approximation scheme (PTAS) for Max 3SAT; that is, Max 3SAT cannot be efficiently approximated within some multiplicative constant $r > 1$. Recall that in the Max 3SAT problem we are given a 3CNF boolean formula with s clauses over n variables, and our goal is to find an assignment that satisfies the largest possible number of clauses. The Max ℓ-CSP problem is a generalization of Max 3SAT in which instead of s clauses we get s boolean constraints $C = \{C_1, \ldots, C_s\}$ of arity ℓ. Again, our goal is to find an assignment that satisfies the largest possible number of constraints. (Recall that a constraint C of arity ℓ over n variables is a pair $(f : \{0,1\}^k \to \{0,1\}, (i_1, \ldots, i_k))$. A constraint C is satisfied by an assignment $(\sigma_1, \ldots, \sigma_n)$ if $f(\sigma_{i_1}, \ldots, \sigma_{i_k}) = 1$.)

The following standard lemma shows that in order to prove that Max 3SAT is hard to approximate, it suffices to prove that Max ℓ-CSP is hard to approximate. This follows by applying Cook's reduction to transform every constraint into a 3CNF.

Lemma 1. *Assume that, for some constants $\ell \in \mathbb{N}$ and $\varepsilon > 0$, there is no polynomial time $(1+\varepsilon)$-approximation algorithm for Max ℓ-CSP. Then there is an $\varepsilon' > 0$ such that there is no polynomial time $(1+\varepsilon')$-approximation algorithm for Max 3SAT.*

A simple and useful corollary of the PCP Theorem [5, 4] is the inapproximability of Max 3SAT.

Theorem 1. *Assume that $\mathrm{P} \neq \mathrm{NP}$. Then, there is an $\varepsilon > 0$ such that there is no $(1+\varepsilon)$-approximatation algorithm for Max 3SAT.*

We now prove a similar result under the (stronger) assumption that there exists an LPRG in NC^0 without relying on the PCP Theorem.

Theorem 2. *Assume that there exists an LPRG in NC^0. Then, there is an $\varepsilon > 0$ such that there is no $(1+\varepsilon)$-approximation algorithm for Max 3SAT.*

Proof. Let $s(n) = cn$ for some constant $c > 1$, and let $s = s(n)$. Let $G : \{0,1\}^n \to \{0,1\}^{s(n)}$ be an LPRG which is computable in NC^0_ℓ. Let $0 < \varepsilon < 1/2$ be a constant that satisfies $\mathrm{H}_2(\varepsilon) < 1/2 - 1/(2c)$, where $\mathrm{H}_2(\cdot)$ is the binary entropy function. Assume towards a contradiction that there exists a PTAS for Max 3SAT. Then, by Lemma 1, there exists a PTAS for Max ℓ-CSP. Hence, there exists a polynomial-time algorithm A_ε that distinguishes satisfiable instances of ℓ-CSP from instances of ℓ-CSP for which any assignment fails to satisfy a fraction ε of the constraints. We show that, given A_ε, we can "break" the LPRG G; that is, we can construct an efficient (non-uniform) adversary that distinguishes between $G(U_n)$ and U_s. Our adversary B_n will translate a string $y \in \{0,1\}^s$ into an ℓ-CSP instance ϕ_y with s constraints such that,

1. If $y \leftarrow G(U_n)$ then ϕ_y is always satisfiable.
2. If $y \leftarrow U_s$ then, with probability $1 - \mathrm{neg}(n)$, no assignment satisfies more than $(1-\varepsilon)s$ constraints of ϕ_y.

Then, B_n will run A_ε on ϕ_y and will output $A_\varepsilon(\phi_y)$. The distinguishing advantage of B is $1 - \mathrm{neg}(n)$ in contradiction to the pseudorandomness of G.

It is left to show how to translate $y \in \{0,1\}^s$ into an ℓ-CSP instance ϕ_y. We use n boolean variables $x_1, \ldots, x_n$ that represent the bits of an hypothetical pre-image of y under G. For every $1 \leq i \leq s$ we add a constraint $G_i(x) = y_i$ where G_i is the function that computes the i-th output bit of G. Since G_i is an ℓ-local function the arity of the constraint is at most ℓ.

Suppose first that $y \leftarrow G(U_n)$. Then, there exists a string $\sigma \in \{0,1\}^n$ such that $G(\sigma) = y$ and hence ϕ_y is satisfiable. We move on to the case in which $y \leftarrow U_s$. Here, we rely on the fact that such a random y is very likely to be far from every element in the range of G. More formally, we define a set $\mathrm{BAD}_n \subseteq \{0,1\}^s$ such that $y \in \mathrm{BAD}_n$ if ϕ_y is $(1-\varepsilon)$-satisfiable; that is, if there exists an assignment $\sigma \in \{0,1\}^n$ that satisfies a fraction $(1-\varepsilon)$ of the constraints of ϕ_y. In this case, the Hamming distance between y and $\mathrm{Im}(G)$ is at most εs. Therefore, the size of BAD_n is bounded by

$$|\mathrm{Im}(G)| \cdot \binom{s}{\varepsilon s} \leq 2^n 2^{\mathrm{H}_2(\varepsilon)s} = 2^{n(1+c\mathrm{H}_2(\varepsilon))} \leq 2^{n(1+c(\frac{1}{2}-\frac{1}{2c}))}.$$

Hence,

$$\Pr_{y \leftarrow U_s}[\phi_y \text{ is } (1-\varepsilon) \text{ satisfiable}] = \mathrm{BAD}_n \cdot 2^{-s} \leq 2^{n(-c+1+c(\frac{1}{2}-\frac{1}{2c}))} = 2^{(1-c)\frac{n}{2}},$$

which completes the proof. □

Remark 1. Theorem 2 can tolerate some relaxations to the notion of LPRG. In particular, since the advantage of B_n is exponentially close to 1, we can consider an LPRG that satisfies a weaker notion of pseudorandomness in which the distinguisher's advantage is bounded by $1 - 1/p(n)$ for some polynomial $p(n)$.

Papadimitriou and Yannakakis showed in [20] that if Max 3SAT does not have a PTAS (i.e., it cannot be approximated up to an arbitrary constant), then several other problems do not have PTAS as well (e.g., Max Cut, Max 2SAT, Vertex Cover). In fact, [20] defined the class Max SNP, and showed that Max 3SAT is complete for this class in the sense that any problem in Max SNP does not have a PTAS unless Max 3SAT has a PTAS. Hence, we get the following corollary (again, without the PCP machinery):

Corollary 1. *Assume that there exists LPRG in* NC^0*. Then, all Max SNP problems do not have a PTAS.*

4 A Construction of LPRG in NC^0

For ease of presentation, we describe our construction in a non-uniform way. We will later discuss a uniform variant of the construction.

4.1 The Assumption

Let $m = m(n)$ be an output length parameter where $m(n) > n$, let $\ell = \ell(n)$ be a locality parameter (typically a constant), and let $0 < \mu < 1$ be a noise parameter. Let $\mathcal{M}_{m,n,\ell}$ be the set of all $m \times n$ matrices over GF_2 in which each row contains exactly ℓ ones. For a matrix $M \in \mathcal{M}_{m,n,\ell}$ we denote by $D_\mu(M)$ the distribution of the random vector

$$Mx + e,$$

where $x \leftarrow U_n$ and $e \in \{0,1\}^m$ is a random error vector in which each entry is chosen to be 1 with probability μ (independently of other entries), and arithmetic is over GF_2. The following assumption is a close variant of a conjecture suggested by Alekhnovich in [1, Conjecture 1]. [2]

Assumption 3. *For any $m(n) = O(n)$, and any constant $0 < \mu < 1$, there exists a positive integer ℓ, and an infinite family of matrices $\{M_n\}_{n\in\mathbb{N}}$, $M_n \in \mathcal{M}_{m(n),n,\ell}$, such that*

$$D_\mu(M_n) \stackrel{c}{\approx} D_{\mu+1/m(n)}(M_n)$$

(Note that since we consider non-uniform distinguishers, we can assume that M_n is public and is available to the distinguisher.)

Alekhnovich [1] shows that if the distribution $D_\mu(M_n)$ satisfies the above assumption then it is pseudorandom. (In fact, the original claim proved in [1, Thm. 3.1] deals with slightly different distributions. However, the proof can be adapted to our setting.)

Lemma 2. *For any polynomial $m(n)$ and constant $0 < \mu < 1$, and any infinite family, $\{M_n\}_{n\in\mathbb{N}}$, of $m(n) \times n$ matrices over GF_2, if $D_\mu(M_n) \stackrel{c}{\approx} D_{\mu+1/m(n)}(M_n)$, then $D_\mu(M_n) \stackrel{c}{\approx} U_{m(n)}$.*

Proof sketch. The proof follows by combining the following easy claims:

1. $D_{\mu+1/m(n)}(M_n) \equiv D_\mu(M_n) + r_n$ where $r_n \in \{0,1\}^{m(n)}$ is a random vector in which each entry is chosen to be 1 with probability $c/m(n)$ (independently of other entries) for some constant $c > 1$.
2. Let $r_n^{t(n)}$ be the distribution resulting from summing $t(n)$ independent samples from r_n. Then, for some polynomial $t(n)$ it holds that $r_n^{t(n)} \stackrel{s}{\approx} U_{m(n)}$.
3. Let $\{A_n\}$ be a polynomial-time samplable distribution ensemble over $\mathrm{GF}_2^{m(n)}$. For a polynomial $t(n)$, let $A_n^{t(n)}$ be the sum (over GF_2) of $t(n)$ independent samples from A_n. Suppose that $D_n \stackrel{c}{\approx} D_n + A_n$ for some distribution ensemble $\{D_n\}$. Then, for every polynomial $t(n)$ we have $D_n \stackrel{c}{\approx} D_n + A_n^{t(n)}$.

[2] Our assumption is essentially the same as Alekhnovich's. The main difference between the two assumptions is that the noise vector e in [1] is a random vector of weight $\lceil \mu m \rceil$, as opposed to our noise vector whose entries are chosen to be 1 independently with probability μ. It can be shown that our assumption is implied by Alekhnovich's assumption (since our iid noise vectors can be viewed as a convex combination of noise vectors of fixed weight).

By the first claim and the Lemma's hypothesis, we have $D_\mu(M_n) \overset{c}{\approx} D_\mu(M_n) + r_n$. Hence, for some polynomial $t(n)$,

$$D_\mu(M_n) \overset{c}{\approx} D_n + r_n^{t(n)} \overset{s}{\approx} D_n + U_{m(n)} \equiv U_{m(n)},$$

where the first transition is due to the third claim and the second transition is due to the second claim. □

By combining Assumption 3 and Lemma 2, we get the following proposition:

Proposition 1. *Suppose that Assumption 3 holds. Then, for any $m(n) = O(n)$, and any constant $0 < \mu < 1$, there exists a constant $\ell \in \mathbb{N}$, and an infinite family of matrices $\{M_n\}_{n\in\mathbb{N}}$ where $M_n \in \mathcal{M}_{m(n),n,\ell}$ such that $D_\mu(M_n) \overset{c}{\approx} U_{m(n)}$.*

Remark 2. If the restriction on the density of the matrices M_n is dropped, the above proposition can be based on the conjectured (average case) hardness of decoding a random linear code (cf., [7, 15]). In fact, under the latter assumption we have that $D_\mu(M_n) \overset{c}{\approx} U_{m(n)}$ for *most* choices of M_n's.

4.2 The Construction

From here on, we let $\mu = 2^{-t}$ for some $t \in \mathbb{N}$. Then we can sample each bit of the error vector e by taking the product of t independent random bits. In this case, we can define an NC^0 function whose output distribution is pseudorandom. Namely,

$$f_n(x, \hat{e}) = M_n x + E(\hat{e})$$

where

$$x \in \{0,1\}^n, \qquad \hat{e} \in \{0,1\}^{t\cdot m(n)}, \qquad E(\hat{e}) = \left(\prod_{j=1}^{t} \hat{e}_{t\cdot(i-1)+j} \right)_{i=1}^{m(n)} . \quad (1)$$

Since $f_n(U_n, U_{t\cdot m(n)}) \equiv D_\mu(M_n)$, the distribution $f_n(U_n, U_{t\cdot m(n)})$ is pseudorandom under Assumption 3 (when the parameters are chosen appropriately). Moreover, the locality of f_n is $\ell + t = O(1)$. However, f_n is not a pseudorandom generator as it uses $n + t \cdot m(n)$ input bits while it outputs only $m(n)$ bits. To overcome this obstacle, we note that most of the entropy of $\hat{e}$ was not "used". Hence, we can apply an *extractor* to regain the lost entropy. Of course, in order to get a PRG in NC^0 the extractor should also be computed in NC^0. Moreover, to get a linear stretch we should extract all the $t \cdot m(n)$ random bits from $\hat{e}$ by investing less than n additional random bits. In the following, we show that such extractors can be implemented by using *ε-biased generators*.

First, we show that the distribution of $\hat{e}$ given $E(\hat{e})$ contains (with high probability) a lot of entropy. In the following we let $m = m(n)$.

Lemma 3. *Let $\hat{e} \leftarrow U_{t\cdot m}$ and $E(\hat{e})$ be defined as in Eq. 1. Denote by $[\hat{e}|E(\hat{e})]$ the distribution of $\hat{e}$ given the outcome of $E(\hat{e})$. Then, except with probability $e^{-(2^{-t}m)/3}$, it holds that*

$$\mathrm{H}_\infty([\hat{e}|E(\hat{e})]) \geq m(1 - 2^{-t+1}) \log(2^t - 1) \geq t \cdot m(1 - \delta(t)), \quad (2)$$

where $\delta(t) = 2^{-\Omega(t)}$.

Proof. We view $E(\hat{e})$ as a sequence of m independent Bernoulli trials, each with a probability 2^{-t} of success. Recall that $\hat{e}$ is composed of m blocks of length t, and that the i-th bit of $E(\hat{e})$ equals the product of the bits in the i-th block of $\hat{e}$. Hence, whenever $E(\hat{e})_i = 1$ all the bits of the i-th block of $\hat{e}$ equal to 1, and when $E(\hat{e})_i = 0$ the i-th block of $\hat{e}$ is uniformly distributed over $\{0,1\}^t \setminus \{1^t\}$. Consider the case in which at most $2 \cdot 2^{-t}m$ components of $E(\hat{e})$ are ones. By a Chernoff bound, the probability of this event is at least $1 - e^{-(2^{-t}m)/3}$. In this case, $\hat{e}$ is uniformly distributed over a set of size at least $(2^t - 1)^{m(1-2^{-t+1})}$. Hence, $\mathrm{H}_\infty([\hat{e}|E(\hat{e})]) \geq m(1 - 2^{-t+1}) \log(2^t - 1) \geq tm(1 - \delta(t))$, for $\delta(t) = 2^{-\Omega(t)}$. □

ε-biased generators can be used to extract random bits from distributions that contain sufficient randomness. Extractors based on ε-biased generators were previously used in [6, 11]. Formally,

Lemma 4 ([18, 2, 16]). *Let $g : \{0,1\}^s \to \{0,1\}^n$ be an ε-biased generator, and let X_n be a random variable taking values in $\{0,1\}^n$ whose min-entropy is at least p. Then,*

$$\|(g(U_s) + X_n) - U_n\| \leq \varepsilon \cdot 2^{(n-p-1)/2} \ .$$

It can be shown that for some fixed exponentially small bias $\varepsilon(n) = 2^{-\Omega(n)}$ and every constant c there exists an ε-biased generator in NC^0 that stretches n bits into cn bits. (The locality of this generator depends on c). Hence, whenever p exceed some linear threshold we can extract n bits from X_n in NC^0 by investing only n/c random bits for any arbitrary c. (Details are deferred to the full version.) However, in our case p is very close to n and so we can rely on a weaker ε-biased generator with an arbitrary linear stretch c and bias $\varepsilon = 2^{-n/poly(c)}$. Recently, Mossel et al. [17] constructed such an ε-biased generator in NC^0_5.

Lemma 5 ([17], Thm. 14). *For every constant c, there exists an ε-biased generator $g : \{0,1\}^n \to \{0,1\}^{cn}$ in NC^0_5 whose bias is at most $2^{-bn/c^4}$ (where b is some universal constant that does not depend on c).*

We remark that the above construction can be implemented in *uniform* NC^0 by using the results of [9, Theorem 7.1]. [3]

We can now describe our LPRG.

Construction 4. *Let t and ℓ be positive integers, and $c, k > 1$ be real numbers that will be used as stretch factors. Let $m = kn$ and let $\{M_n \in \mathcal{M}_{n,m,\ell}\}$ be an infinite family of matrices. Let $g : \{0,1\}^n \to \{0,1\}^{cn}$ be the ε-biased generator promised by Lemma 5. We define the function*

$$G_n(x, \hat{e}, r) = (M_n x + E(\hat{e}), g(r) + \hat{e}),$$

where $x \in \{0,1\}^n, \hat{e} \in \{0,1\}^{t \cdot m}, r \in \{0,1\}^{t \cdot m/c}, E(\hat{e}) = \left(\prod_{j=1}^{t} \hat{e}_{t \cdot (i-1)+j}\right)_{i=1}^{m}$.

[3] Theorem 7.1 of [9] gives an explicit family of asymmetric constant-degree bipartite expanders, which can replace the probabilistic construction given in [17, Lemma 12]. We note that the locality of the resulting generator depends on c. See full version for details.

Observe that G_n is an NC^0 function. We show that if the parameters are chosen properly then G_n is an LPRG.

Lemma 6. *Under Assumption 3, there exist constants $t, \ell \in \mathbb{N}$, constants $c, k > 1$, and a family of matrices $\{M_n \in \mathcal{M}_{n,m,\ell}\}$ such that the function G_n defined in Construction 4 is an LPRG.*

Proof. Set $k > 1$ to be some arbitrary constant and let $m = kn$. Let $c = 2t/(1 - 1/k)$ and choose t to be a constant satisfying:

$$\Delta \stackrel{\text{def}}{=} \frac{bt}{c^5} - \delta(t) > 0, \tag{3}$$

where $\delta(\cdot)$ is the negligible function from Eq. 2 and b is the bias constant of Lemma 5. There exists a (large) constant t satisfying the above since $\delta(t) = 2^{-\Omega(t)}$ while $bt/c^5 = \Theta(1/t^4)$. Let $\ell \in \mathbb{N}$ be a constant and $\{M_n \in \mathcal{M}_{n,m,\ell}\}$ be an infinite family of matrices satisfying Assumption 3.

First, we show that G_n has linear stretch. The input length of G_n is $n+tm+tm/c = n(tk + k/2 + 1/2)$. The output length is $m(t + 1) = n(tk + k)$. Hence, since $k > 1$, the constant $tk + k/2 + 1/2$ is smaller than the constant $tk + k$, and so the function G_n has a linear stretch.

Let $x, \hat{e}$ and r be uniformly distributed over $\{0,1\}^n, \{0,1\}^{t\cdot m}$ and $\{0,1\}^{t\cdot m/c}$ respectively. We prove that the distribution $G_{M_n}(x, \hat{e}, r)$ is pseudorandom. By Lemmas 3, 4 and 5 it holds that

$$\begin{aligned}
\|(E(\hat{e}), \hat{e} + g(r)) - (E(\hat{e}), U_{t\cdot m})\| &\leq e^{-(2^{-t}m)/3} + 2^{-b(tm/c)/c^4} 2^{(tm-(t-\delta(t))m-1)/2} \\
&\leq e^{-(2^{-t}m)/3} + 2^{(\delta(t)-bt/c^5)m} \\
&\leq e^{-(2^{-t}m)/3} + 2^{-\Delta m} = \mathrm{neg}(m) = \mathrm{neg}(n),
\end{aligned}$$

where the last inequality is due to Eq. 3. Therefore, by Proposition 1, we get that

$$(M_n x + E(\hat{e}), g(r) + \hat{e}) \stackrel{s}{\approx} (M_n x + E(\hat{e}), U_{t\cdot m}) \equiv (D_{2^{-t}}(M_n), U_{t\cdot m}) \stackrel{c}{\approx} (U_m, U_{t\cdot m}) \ .$$

□

By the above Lemma we get a construction of LPRG in NC^0 from Assumption 3. In fact, in [3] it is shown that such an LPRG can be transformed into an LPRG whose locality is 4. Hence, we have:

Theorem 5. *Under Assumption 3, there exists an LPRG in NC^0_4.*

Mossel et al. [17] showed that a PRG in NC^0_4 cannot achieve a superlinear stretch. Hence, Theorem 5 is essentially optimal with respect to stretch.

Remarks on Theorem 5.

1. (Uniformity) Our construction uses a family of matrices $\{M_n\}$ satisfying Assumption 3 as a non-uniform advice. We can eliminate this advice and construct an LPRG in *uniform* NC^0_4 by slightly modifying Assumption 3. In particular, we follow Alekhnovich (cf. [1, Remark 1]) and conjecture that any family $\{M_n\}$ of good

expanders satisfy Assumption 3. Hence, our construction can be implemented by using an explicit family of asymmetric constant-degree bipartite expanders such as the one given in [9, Theorem 7.1].

2. (The stretch of the construction) Our techniques do not yield a *superlinear* stretch PRG in NC^0. To see this, consider a variant of Assumption 3 in which we allow $m(n)$ to be superlinear and let $\mu(n)$ be subconstant. (These modifications are necessary to obtain a superlinear PRG.) In this case, the noise distribution cannot be sampled in NC^0 (since $\mu(n)$ is subconstant). This problem can be bypassed by extending Assumption 3 to alternative noise models in which the noise is not iid. However, it is not clear how such a modification affects the hardness assumption.

5 The Necessity of Expansion

As pointed out in the previous section, our construction of LPRG makes use of expander graphs. This is also the case in several constructions of "hard functions" with low locality (e.g., [13, 17, 1]). We now show that this is not coincidental at least in the case of PRGs. Namely, we show that the structure of any LPRG in NC^0 contains a copy of a graph with some expansion property. (In fact, this holds even in the case of ε-biased generators.) Then, we use known lower bounds for expander graphs to rule out the possibility of exponentially strong PRG with superlinear stretch in NC^0.

Let $g : \{0,1\}^n \to \{0,1\}^s$ be a PRG. We claim that every set S of output bits whose size is $O(\log n)$ touches at least $|S|$ input bits. Otherwise, there exists a small set S of output bits and a string $y \in \{0,1\}^{|S|}$ such that $\Pr[g_S(U_n) = y] = 0$ (where $g_S(\cdot)$ is the restriction of g to the output bits of S). Hence, an efficient adversary can distinguish between $g_S(U_n)$ and $U_{|S|}$ with advantage $2^{-O(logn)} = 1/\mathrm{poly}(n)$, in contradiction to the pseudorandomness of g. More generally, if g is ε-strong (i.e., cannot be broken by any efficient adversary with probability ε), then every set of $t \leq \log(1/\varepsilon)$ output bits touches at least t input bits. This claim extends to the case of ε-biased generators by using the Vazirani XOR Lemma [22].

In graph theoretic terms, we have a bipartite graph $G = ((\mathrm{In} = [n], \mathrm{Out} = [s]), E)$ that enjoys some output expansion property. This property is trivial when the output degree of G is high (as in standard constructions of PRGs) or when s is not much larger than n (as in the NC^0 constructions of [3]). However, when the locality is constant and the stretch is linear, G is a sparse bipartite graph having n input vertices, $s = n + \Omega(n)$ output vertices, and a constant output degree. In the standard cryptographic setting, when $\varepsilon(n)$ is negligible, we get expansion for sets of size $O(\log(n))$. That is, G expands (output) sets of size smaller than $\omega(\log n)$. When $\varepsilon < 2^{-\Omega(n)}$ (as in the ε-biased construction of [17]), we get expansion for sets of size at most $\Omega(n)$.

Radhakrishnan and Ta-Shma [21] obtained some lower bounds for similar graphs. In particular, by using [21, Thm. 1.5] it can be shown that if $g : \{0,1\}^n \to \{0,1\}^s$ is an NC^0_ℓ function that enjoys the above expansion property for sets of size $\leq t$, then $\ell \geq \Omega(\log(s/t)/\log(n/t))$. We therefore conclude that there is no $2^{-\Omega(n)}$-strong PRG (resp. $2^{-\Omega(n)}$-biased generator) with superlinear stretch in NC^0.

Acknowledgments. We thank Eli Ben-Sasson and Amir Shpilka for helpful discussions.

References

1. M. Alekhnovich. More on average case vs approximation complexity. In *Proc. 44th FOCS*, pages 298–307, 2003.
2. N. Alon and Y. Roichman. Random cayley graphs and expanders. *Random Struct. Algorithms*, 5(2):271–285, 1994.
3. B. Applebaum, Y. Ishai, and E. Kushilevitz. Cryptography in NC^0. *SIAM J. Comput.* To appear. Preliminary version in FOCS 04.
4. S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and hardness of approximation problems. *J. of the ACM*, 45(3):501–555, 1998.
5. S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of np. *J. of the ACM*, 45(1):70–122, 1998.
6. E. Ben-Sasson, M. Sudan, S. Vadhan, and A. Wigderson. Randomness-efficient low-degree tests and short pcps via epsilon-biased sets. In *Proc. 35th STOC*, pages 612–621, 2003.
7. A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology: Proc. of CRYPTO '93*, volume 773 of *LNCS*, pages 278–291, 1994.
8. M. Blum and S. Micali. How to generate cryptographically strong sequences of pseudo-random bits. *SIAM J. Comput.*, 13:850–864, 1984.
9. M. Capalbo, O. Reingold, S. Vadhan, and A. Wigderson. Randomness conductors and constant-degree lossless expanders. In *Proc. 34th STOC*, pages 659–668, 2002.
10. M. Cryan and P. B. Miltersen. On pseudorandom generators in NC^0. In *Proc. 26th MFCS*, 2001.
11. Y. Dodis and A. Smith. Correcting errors without leaking partial information. In *Proc. 37th STOC*, pages 654–663, 2005.
12. U. Feige. Relations between average case complexity and approximation complexity. In *Proc. of 34th STOC*, pages 534–543, 2002.
13. O. Goldreich. Candidate one-way functions based on expander graphs. *ECCC*, 7(090), 2000.
14. O. Goldreich. *Foundations of Cryptography: Basic Tools*. Cambridge University Press, 2001.
15. O. Goldreich, H. Krawczyk, and M. Luby. On the existence of pseudorandom generators. *SIAM J. Comput.*, 22(6):1163–1175, 1993.
16. O. Goldreich and A. Wigderson. Tiny families of functions with random properties: A quality-size trade-off for hashing. *Random Struct. Algorithms*, 11(4):315–343, 1997.
17. E. Mossel, A. Shpilka, and L. Trevisan. On ϵ-biased generators in NC^0. In *Proc. 44th FOCS*, pages 136–145, 2003.
18. J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993.
19. M. Naor. Bit commitment using pseudorandomness. *J. of Cryptology*, 4:151–158, 1991.
20. C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. of Computer and Systems Sciences*, 43:425–440, 1991.
21. J. Radhakrishnan and A. Ta-Shma. Tight bounds for depth-two superconcentrators. *SIAM J. Discrete Math.*, 13(1):2–24, 2000.
22. U. Vazirani. *Randomness, Adversaries and Computation*. Ph.d. thesis, UC Berkeley, 1986.
23. E. Viola. On constructing parallel pseudorandom generators from one-way functions. In *Proc. 20th CCC*, pages 183– 197, 2005.
24. A. C. Yao. Theory and application of trapdoor functions. In *Proc. 23rd FOCS*, pages 80–91, 1982.