Pseudorandomness for Width 2 Branching Programs

Andrej Bogdanov* Zeev Dvir † Elad Verbin‡ Amir Yehudayoff§

Abstract

Recently Bogdanov and Viola (FOCS 2007) and Lovett (ECCC-07) constructed pseudorandom generators that fool degree k polynomials over $\mathbb{F}_2$ for an arbitrary constant k. We show that such generators can also be used to fool branching programs of width 2 and polynomial length that read k bits of inputs at a time. This model generalizes polynomials of degree k over $\mathbb{F}_2$ and includes some other interesting classes of functions, for instance k-DNF.

The constructions of Bogdanov and Viola and Lovett consist of adding a constant number of independent copies of a generator that fools linear functions (an ϵ-biased set). It is natural to ask, in light of our first result, whether such generators can fool branching programs of width larger than 2. Our second result is a lower bound showing that a sum of $o(\sqrt{n}/\log n)$ independent copies of any $n^{-O(1)}$-biased set does not fool branching programs of width 5. To the best of our knowledge this is the first lower bound for such constructions.

* andrejb@tsinghua.edu.cn. ITCS, Tsinghua University, FIT Building 4-608-7. Supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grants 2007CB807900, 2007CB807901.

† zeev.dvir@weizmann.ac.il. Department of Computer Science, Weizmann institute of science, Rehovot, Israel. Research supported by Binational Science Foundation (BSF) grant, by Israel Science Foundation (ISF) grant and by Minerva Foundation grant.

‡ eladv@tsinghua.edu.cn. ITCS, Tsinghua University, FIT Building 4-608-6. Supported in part by the National Natural Science Foundation of China Grant 60553001, and the National Basic Research Program of China Grants 2007CB807900, 2007CB807901.

§ amir.yehudayoff@weizmann.ac.il. Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot, 76100 Israel. Research supported by grants from the Binational Science Foundation (BSF), the Israel Science Foundation (ISF), the Minerva Foundation, and the Israel Ministry of Science (IMOS) - Eshkol Fellowship.

1 Introduction

Reingold's proof [Rei05] that SL = L has brought renewed interest to derandomizing space bounded computations. Some attempts were made to apply these new techniques towards resolving the L versus RL question [RTV06, CRV07], but so far without success. The best known deterministic simulation of randomized logspace is by Saks and Zhou [SZ99], who show that $\mathrm{RSPACE}(\log n) \subseteq \mathrm{DSPACE}(O(\log^{3/2} n))$. This construction is based on Nisan's generator [Nis92], which is to this date the best pseudorandom source (both in terms of time and space efficiency) for randomized space bounded computations.

Logarithmic space is one of the simplest models of computation that we know of, yet progress on improving the use of randomness in this model has been stuck for over a decade now. One line of attack is to try and derandomize even simpler models of space bounded computation. In the uniform setting, restricting the model to use less than logarithmic space is not particularly natural, but there is a good way to specialize the definitions in the nonuniform setting.

A nonuniform model for a computation that uses space $s(n)$ and runs in time $t(n)$ is a *branching program* of width $2^{s(n)}$ and length $t(n)$. This device can be described by a layered directed acyclic graph, where there are $t(n)$ layers and each layer contains $2^{s(n)}$ nodes – except for last layer which consists of only two nodes, "accept" (1) and "reject" (0). Each layer j is associated with a bit $x|_j$ of the input x. Each node in layer j has 2 outgoing edges labelled by possible values of the bit $x|_j$. On input x, the computation starts in the first node in the first layer, then follows the edge labelled by $x|_1$ onto the second layer, and so on until a node in the last layer is reached. The identity of this last node is the outcome of the computation.

When $s(n)$ is very small (say constant), it is interesting to generalize the above definition so that the branching program is allowed to read $k > s(n)$ bits of the input at each step. Now $x|_j$ denotes a k-bit sub-string of the input x (not necessarily consecutive bits) and each node in layer j has 2^k outgoing edges labelled by all possible values of $x|_j$. The computation is done in the same way as before. One could think of such a branching program as having a 'global' space of $s(n)$ bits and a larger 'local' space of k bits. For the rest of this paper we consider this generalized formulation of branching programs.

The nodes in layer j represent the possible states of the randomized space-bounded computation at time j, and the outgoing edges represent the possible transitions depending on the contents of the random tape x. The block $x|_j$ is the part of the random tape "viewed" by the machine at time j. In randomized space-bounded computation, we usually restrict the machine to have one-way access to the random tape. In the branching program setting, this imposes the requirement that x is the concatenation of all the blocks $x|_j$ in order, namely $x = x|_1 \dots x|_t$. That is, at each step we read the 'next' k bits of the input, without being able to go back and look at bits we have already read. We call such a branching program *read-once*. General branching programs are much more powerful than read-once branching programs For instance, the *inner product function*,

$$IP(x_1,\ldots,x_n)=\sum\nolimits_{i=1}^{n/2} x_i x_{n/2+i} \mod 2, \quad n \text{ even},$$

can be computed by a branching program of width 2 that reads 2 bits at a time but not by any read-once branching program of width $o(n)$ that reads $o(n)$ bits at a time (note that in this example the order of the variables is important).

1.1 Pseudorandom Generators

We start by giving a formal definition of a pseudorandom generator against a class of functions.

Definition 1.1. *We say a distribution D on $\{0,1\}^n$ is* ϵ-pseudorandom *against a class C of functions from $\{0,1\}^n$ to $\{0,1\}$ if for every $f \in C$,*

$$\left|\Pr_{x\sim D}[f(x)=1]-\Pr_{x\sim\{0,1\}^n}[f(x)=1]\right| \le \epsilon$$

(where $x \sim \{0,1\}^n$ means that x is uniformly distributed in $\{0,1\}^n$). A function $G : \{0,1\}^m \to \{0,1\}^n$ is an ϵ-pseudorandom generator *(PRG) against C if the distribution $G(s)$, $s \sim \{0,1\}^m$ is ϵ-pseudorandom against C. We call m the* seed length *of the generator.*

We use (k,t,n)-2BP to denote width 2 branching programs of length t that read k bits of input at a time and compute a function from $\{0,1\}^n$ to $\{0,1\}$. Our first main result is a positive one, showing that a PRG for degree k polynomials over $\mathbb{F}_2$ is also a PRG for the class of functions computed by a (k,t,n)-2BP.

Theorem 1.2. *Let G be an ϵ-PRG against degree k polynomials in n variables over $\mathbb{F}_2$. Then G is an ϵ'-PRG against the class of functions computed by a (k,t,n)-2BP, with $\epsilon' = t\cdot\epsilon$.*

Recently Lovett [Lov07] (following work by Bogdanov and Viola [BV07]) constructed an ϵ-PRG against degree k polynomials in n variables over $\mathbb{F}_2$ with seed length $2^{O(k)}\cdot\log(n/\epsilon)$ by summing together 2^k independent copies of a generator against linear functions with bias $\epsilon^{2^{O(k)}}$. Using Theorem 1.2, this automatically yields generators for (k,t,n)-2BP's with seed length $2^{O(k)}\cdot\log(n\cdot t/\epsilon)$. Observe that a width 2 branching program that reads k bits at a time in particular can compute every polynomial of degree k (with $t = O(n^k)$). However, such programs are strictly stronger than degree k polynomials; e.g., they can compute any k-DNF.

1.2 Lower Bounds

It is tempting to test the same pseudorandom generator against branching programs of larger width. In general this is impossible:

Theorem 1.3. *For every $n > 1$ and k, there exists a distribution D on $\{0,1\}^n$ that is $\exp(-\Omega(n/4^k))$-pseudorandom against degree k polynomials but is not 0.66-pseudorandom against read-once width 3 branching programs of length n that read one bit at a time.*

This theorem is a consequence of a recent correlation bound of Viola and Widgerson [VW07]. However, we do not find this lower bound completely satisfying, since even though it rules out *general* pseudorandom generators for polynomials as a mean to fooling small width branching programs, it says nothing about the constructions in [BV07, Lov07] – sums of independent copies of generators against linear functions. Our second main result is the following theorem which shows the limitations of these constructions.

Theorem 1.4. *For every n, ϵ, and k such that $k\log(1/\epsilon) < \sqrt{n/2} - 1$, there exists a distribution D such that D is ϵ-pseudorandom against linear functions over $\{0,1\}^n$, but the sum of k independent copies of D is not $1/3$-pseudorandom against width 5 branching programs of length $2^{O(\log(k\log(1/\epsilon)))^2}$ that read one bit at a time.*

It is known [MRRW77, FT05] that the seed length of an ϵ-biased generator must be at least $\Omega(\log n + \log(1/\epsilon))$. Therefore, if we want the generator to be efficient, we are restricted to using $\epsilon = \mathrm{poly}(n)$. For this setting of parameters, Theorem 1.4 tells us that for any constant k and sufficiently large n, a branching program of width 5 and length n will not be fooled by a sum of k independent ϵ-biased generators.

The branching programs that realize this lower bound are not read-once, so it does not rule out the possibility of using sums of independent generators against linear functions to fool randomized space bounded computations even of width $\mathrm{poly}(n)$. We leave it as an intriguing open question whether Lovett's generator helps against width 3 and width 4 branching programs (read-once or not). Such devices are fairly powerful: width 3 branching programs of length t, for instance, can compute all DNF of size t (even for $k = 1$); this is a class of functions that has resisted the construction of polynomial size pseudorandom sets for some time. Width 4 can compute any sparse polynomial with at most t/n terms.

1.3 Proof Technique

It has been known for some time that *read-once* width 2 branching programs that read one bit at a time can be fooled by linear generators.[1] One way to argue this is to think of the computation of the branching program B as a boolean function over $\mathbb{F}_2^n$ and show inductively over the layers of B that the sum of the absolute values of the Fourier coefficients of B is bounded from above by t. It is easy to see that linear generators of bias ϵ are ϵL-pseudorandom against any boolean function whose sum of absolute values of Fourier coefficients is at most L, and the correctness follows from there.

[1] We are not aware of a published proof but have heard the result credited to Saks and Zuckerman.

For branching programs that read more than one bit at a time this argument cannot work, as there exist width 2 branching programs that read 2 bits at a time and that are not fooled by any small bias linear generator. One such branching program computes the inner product function $IP(x_1, \ldots, x_n)$. Nevertheless, we argue along the same lines. Instead of using the Fourier transform of the branching program, we resort to "higher-order" representations of functions using low-degree polynomials. We show that every branching program B with length t and width 2 that reads k bits at a time admits a "representation of length t" in terms of degree k polynomials. By "representation of length t" we mean that B can be written as a sum *over the reals* of the form

$$B(x) = \sum\nolimits_{p:\mathbb{F}_2^n \to \mathbb{F}_2} \alpha_p \cdot p(x)$$

where p ranges over all degree k polynomials over $\mathbb{F}_2$, and α_p are real coefficients such that $\sum_p |\alpha_p| \leq t$. Unlike the Fourier transform, for degree 2 and larger this representation is not unique. Once this representation is obtained, we argue that a pseudorandom generator for degree k polynomials is also pseudorandom for B by linearity of expectation.

While our proof is not technically difficult we find the application of "higher-order" Fourier type analysis conceptually interesting and potentially relevant for other computer science applications.

1.4 Organization

In Section 2 we prove our main positive result, Theorem 1.2. Then, in Section 3 we show limitations of existing PRG's and prove Theorem 1.3 and Theorem 1.4.

2 Fooling width 2 branching programs

Recall that we use (k, t, n)-2BP to denote width 2 branching programs of length t that read k bits of input at a time and compute a function from $\{0,1\}^n$ to $\{0,1\}$. For a function $f : \{0,1\}^n \to \{0,1\}$, we denote $\hat{f} = (-1)^f$, a map from $\{0,1\}^n$ to $\{1,-1\}$. Define $\deg(f)$ to be the degree of f when viewed as a multilinear polynomial in $\mathbb{F}_2[x_1, \ldots, x_n]$.

2.1 Width 2 Branching Programs as Sum of Polynomials

The following theorem is the basis for the proof of Theorem 1.2. It shows that width 2 branching programs have a "short representation by polynomials of small degree".

Theorem 2.1. *Let $f : \{0,1\}^n \to \{0,1\}$ be computed by a (k, t, n)-2BP. Then there exist $\alpha_1, \ldots, \alpha_s \in \mathbb{R}$ and $g_1, \ldots, g_s : \{0,1\}^n \to \{0,1\}$ such that*

1. $\hat{f}(x) = \sum_{i=1}^{s} \alpha_i \cdot \hat{g}_i(x)$ *for all* $x \in \{0,1\}^n$ *(where the sum is over the reals).*

2. For all $i \in [s],\ \deg(g_i) \le k$.

3. $\sum_{i=1}^{s} |\alpha_i| \le t$.

We defer the proof of Theorem 2.1 to Section 2.2 and proceed by showing how it implies our main result.

Proof of Theorem 1.2

Let $G : \{0,1\}^m \to \{0,1\}^n$ be an ϵ-pseudorandom generator against degree k polynomials in n variables over $\mathbb{F}_2$. Let $f : \{0,1\}^n \to \{0,1\}$ be computed by a (k,t,n)-2BP. By Theorem 2.1, there exist $\alpha_1, \ldots, \alpha_s \in \mathbb{R}$ and $g_1, \ldots, g_s : \{0,1\}^n \to \{0,1\}$ such that

1. $\hat{f}(x) = \sum_{i=1}^{s} \alpha_i \cdot \hat{g}_i(x)$ for all $x \in \{0,1\}^n$.
2. For all $i \in [s],\ \deg(g_i) \le k$.
3. $\sum_{i=1}^{s} |\alpha_i| \le t$.

For the rest of the proof $x \sim \{0,1\}^n$ and $s \sim \{0,1\}^m$ denote two independent random variables. First, note that for every function $f : \{0,1\}^n \to \{0,1\}$,

$$2 \cdot \big| \Pr[f(G(s)) = 1] - \Pr[f(x) = 1] \big| = \big| \mathbf{E}[\hat{f}(G(s)) - \hat{f}(x)] \big|.$$

Thus, using the properties above, and using the linearity of expectation,

$$\begin{aligned} 2 \cdot \big| \Pr[f(G(s)) = 1] - \Pr[f(x) = 1] \big| &= \Big| \mathbf{E}\Big[\hat{f}(G(s)) - \hat{f}(x) \Big] \Big| \\ &= \Big| \sum\nolimits_{i=1}^{s} \alpha_i \cdot \mathbf{E}[\hat{g}_i(G(s)) - \hat{g}_i(x)] \Big| \\ &\le \sum\nolimits_{i=1}^{s} |\alpha_i| \cdot \Big| \mathbf{E}[\hat{g}_i(G(s)) - \hat{g}_i(x)\Big] \Big| \\ &= \sum\nolimits_{i=1}^{s} |\alpha_i| \cdot 2 \cdot \big| \Pr[g_i(G(s)) = 1] - \Pr[g_i(x) = 1] \big| \\ &\le 2 \cdot t \cdot \epsilon, \end{aligned}$$

where the last inequality holds since G is an ϵ-pesudorandom generator against degree k polynomials. The theorem now follows. □

2.2 Proof of Theorem 2.1

Let f be a boolean function computed by a branching program B of width 2 and length t that reads k bits of inputs at a time. We will prove the theorem by induction on t.

Induction base: For the case $t = 1$, the theorem holds since $f(x)$ is a boolean function in k variables and so $\deg(f) \leq k$.

Induction step: Assume that the theorem holds for every function computed by a $(k, t-1, n)$-2BP. By the definition of such branching programs, there exists $P : \{0,1\}^{k+1} \to \{0,1\}$ such that

$$f(x) = P\left(f_{t-1}(x)\, ,\, x|_{t-1}\right),$$

where f_{t-1} is the function computed at the $(t-1)$'th layer of B, and $x|_{t-1}$ is the k-bit substring of the input x associated with the $(t-1)$'th layer. Using the representation of $\hat{P}$ by Fourier transform, there exist two maps A and B from $\{0,1\}^k$ to $\mathbb{R}$ such that for every $z \in \{0,1\}$ and $y \in \{0,1\}^k$,

$$\hat{P}(z, y) = A(y) \cdot (-1)^z + B(y)\ .$$

We will now show that both A and B have a special structure. Let p_0 and p_1 be two maps from $\{0,1\}^k$ to $\{0,1\}$ defined as

$$p_0 = P(0, y) \ \text{ and } \ p_1 = P(1, y).$$

Note that since both of p_0 and p_1 depend on at most k variables, $\deg(p_0) \leq k$ and $\deg(p_1) \leq k$. In addition,

$$\hat{p_0} = A + B \ \text{ and } \ \hat{p_1} = -A + B.$$

Thus,

$$A = \frac{1}{2}(\hat{p_0} - \hat{p_1}) \ \text{ and } \ B = \frac{1}{2}(\hat{p_0} + \hat{p_1}),$$

which implies that for every $z \in \{0,1\}$ and $y \in \{0,1\}^k$,

$$\hat{P}(z, y) = \frac{1}{2}(\hat{p_0}(y) - \hat{p_1}(y)) \cdot (-1)^z + \frac{1}{2}(\hat{p_0}(y) + \hat{p_1}(y)).$$

We can now use the induction hypothesis. By the choice of P, for every $x \in \{0,1\}^n$,

$$\hat{f}(x) = \hat{P}(f_{t-1}(x), x|_{t-1}) = \frac{1}{2}(\hat{p_0}(x|_{t-1}) - \hat{p_1}(x|_{t-1})) \cdot \widehat{f_{t-1}}(x) + \frac{1}{2}(\hat{p_0}(x|_{t-1}) + \hat{p_1}(x|_{t-1})).$$

By induction, there exist $\alpha_1, \ldots, \alpha_s \in \mathbb{R}$ and $g_1, \ldots, g_s : \{0,1\}^n \to \{0,1\}$ such that

1. $\widehat{f_{t-1}}(x) = \sum_{i=1}^{s} \alpha_i \cdot \hat{g}_i(x)$ for all $x \in \{0,1\}^n$.

2. For all $i \in [s]$, $\deg(g_i) \leq k$.

3. $\sum_{i=1}^{s} |\alpha_i| \leq t-1$.

Thus, for all $x \in \{0,1\}^n$

$$\hat{f}(x) \;=\; \frac{1}{2}(\hat{p_0}(x|_{t-1}) - \hat{p_1}(x|_{t-1})) \cdot \sum_{i=1}^{s} \alpha_i \cdot \hat{g}_i(x) + \frac{1}{2}(\hat{p_0}(x|_{t-1}) + \hat{p_1}(x|_{t-1})).$$

We complete the proof by renaming the polynomials and the coefficients in the above sum. For $j = 1, \ldots, s$, set

$$\beta_j = \frac{\alpha_j}{2} \quad \text{and} \quad h_j(x) = p_0(x|_{t-1}) \oplus g_j(x)$$

and for $j = s+1, \ldots, 2s$, set

$$\beta_j = -\frac{\alpha_{j-s}}{2} \quad \text{and} \quad h_j(x) = p_1(x|_{t-1}) \oplus g_j(x)$$

(where $\oplus$ denotes summation in $\mathbb{F}_2$). Set $\beta_{2s+1} = \beta_{2s+2} = 1/2$, set $h_{2s+1}(x) = p_0(x|_{t-1})$, and set $h_{2s+2}(x) = p_1(x|_{t-1})$. Finally, set $s' = 2s+2$. Thus,

$$\hat{f}(x) = \sum_{j=1}^{s'} \beta_j \cdot \hat{h}_j(x)$$

for all $x \in \{0,1\}^n$. In addition, every h_j is of degree at most k (since addition in $\mathbb{F}_2$ does not increase the degree), and

$$\sum_{j=1}^{s'} |\beta_j| \leq 1 + 2 \cdot \sum_{i=1}^{s} \frac{|\alpha_i|}{2} \leq 1 + (t-1) = t.$$

□

3 Limitations of Existing PRG's

In this section we explore the limitations of pseudorandom generators of two kinds. First, we show that pseudorandom generators for degree k polynomials fail for read-once width 3 branching programs (Theorem 1.3). Second, we show that a sum of several copies of pseudorandom generators for linear functions fail for width 5 branching programs (Theorem 1.4).

3.1 Proof of Theorem 1.3

We derive Theorem 1.3 from a special case of a correlation bound of Viola and Widgerson. Let $\omega = e^{2\pi i/3}$ be the cube root of unity and $\text{mod}_3 : \{0,1\}^n \to \mathbb{C}$ denote the function

$$\text{mod}_3(x_1, \ldots, x_n) = \omega^{x_1 + \cdots + x_n}$$

where the summation $x_1 + \cdots + x_n$ is evaluated *over the integers*.

Theorem 3.1 (Viola and Wigderson [VW07]). *There is a constant $\alpha > 0$ such that for every n and every polynomial $p : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree k,*

$$\left|\mathbf{Pr}_{x\sim\{0,1\}^n}[p(x) = 1 \mid \mathrm{mod}_3(x) = 1] - \mathbf{Pr}_{x\sim\{0,1\}^n}[p(x) = 1 \mid \mathrm{mod}_3(x) \neq 1]\right| \le \exp(-\alpha n/4^k).$$

Assume $n > 100$. Let D be the uniform distribution on the set of all $x \in \mathbb{F}_2^n$ such that $\mathrm{mod}_3(x) = 1$.

We will first show that D is $\exp(-\Omega(n/4^k))$-pseudorandom against degree k polynomials. For every polynomial $p : \mathbb{F}_2^n \to \mathbb{F}_2$ of degree k,

$$\begin{aligned}\mathbf{E}_{x\sim\{0,1\}^n}[p(x)] &= \mathbf{E}[p(x) \mid \mathrm{mod}_3(x) = 1]\mathbf{Pr}[\mathrm{mod}_3(x) = 1] \\ &\quad + \mathbf{E}[p(x) \mid \mathrm{mod}_3(x) \neq 1]\mathbf{Pr}[\mathrm{mod}_3(x) \neq 1] \\ &= \mathbf{E}[p(x) \mid \mathrm{mod}_3(x) = 1] \\ &\quad + (\mathbf{E}[p(x) \mid \mathrm{mod}_3(x) \neq 1] - \mathbf{E}[p(x) \mid \mathrm{mod}_3(x) = 1])\mathbf{Pr}[\mathrm{mod}_3(x) \neq 1].\end{aligned}$$

Therefore, by Theorem 3.1,

$$\left|\mathbf{E}_{x\sim\{0,1\}^n}[p(x)] - \mathbf{E}_{x\sim D}[p(x)]\right| \le \left|\mathbf{E}[p(x) \mid \mathrm{mod}_3(x) \neq 1] - \mathbf{E}[p(x) \mid \mathrm{mod}_3(x) = 1]\right| \le \exp(-\alpha n/4^k).$$

so D is $\exp(-\alpha n/4^k)$-pseudorandom against all degree k polynomials.

We will now show that D is not 0.66-pseudorandom against read-once width 3 branching programs of length n that read one bit at a time. Let $f : \{0,1\}^n \to \{0,1\}$ be the function

$$f(x) = \begin{cases} 1, & \text{if } \mathrm{mod}_3(x) = 1 \\ 0, & \text{otherwise.}\end{cases}$$

Then $\mathbf{E}_{x\sim D}[f(x)] = 1$ while

$$\mathbf{E}_{x\sim\{0,1\}^n}[f(x)] = \frac{1}{3} \cdot \mathbf{E}_{x\sim\{0,1\}^n}[1 + \mathrm{mod}_3(x) + \mathrm{mod}_3(x)^2]$$

and for $a \in \{1, 2\}$

$$|\mathbf{E}_{x\sim\{0,1\}^n}[\mathrm{mod}_3(x)^a]| = |(\mathbf{E}_{x_1\sim\{0,1\}}[\omega^{ax_1}])^n| = \left|\left(\frac{1+\omega^a}{2}\right)^n\right| = 2^{-n}$$

so that

$$\left|\mathbf{E}_{x\sim\{0,1\}^n}[f(x)]\right| \le 1/3 + 2^{-n+1}/3 < 0.34$$

and D is not 0.66-pseudorandom against f. Since f can be computed by a read-once width 3 branching program that reads one bit at a time, the proof is complete. □

3.2 Proof of Theorem 1.4

Set $m = k\log(1/\epsilon) + 1$ and partition the input $x \in \mathbb{F}_2^n$ into n/m consecutive blocks $x|_1, \dots, x|_{n/m} \in \mathbb{F}_2^m$. Consider the following distribution D.

1. Choose a random linear subspace S of $\mathbb{F}_2^m$ of dimension $(m-1)/k$.
2. For $1 \le i \le n$, choose each block $x|_i$ independently and uniformly from S.

To prove Theorem 1.4, we show the following two claims.

Claim 3.2. *The distribution D is ϵ-pseudorandom against linear functions.*

Claim 3.3. *The sum D^k of k independent samples from D is not $1/3$-pseudorandom against bounded fanin circuits (with and, or, and not gates) of depth $O((\log m)^2)$.*

By Barrington's theorem [Bar89], any circuit of depth d can be simulated by a branching program of width 5 and size 4^d, so D^k is not pseudorandom against width 5, size $2^{O((\log m)^2)}$ branching programs. This proves the theorem.

Proof of Claim 3.2. Let $a(x) = \langle a, x\rangle$ be an arbitrary nonzero linear function over $\mathbb{F}_2^n$. We split a as a sum of linear functions a_i over the blocks of x as

$$a(x) = \sum\nolimits_{i=1}^{n/m} a_i(x|_i).$$

Without loss of generality, let's assume a_1 is nonzero. Conditioned on the choice of S, the values of the functions $a_i(x|_i)$ are independent:

$$\mathbf{E}_{x\sim D}[(-1)^{a(x)}] = \mathbf{E}_S\Big[\prod\nolimits_{i=1}^{m/n} \mathbf{E}_{x|_i\sim S}[(-1)^{a_i(x|_i)}]\Big].$$

Now for any fixed choice of S, the value $\mathbf{E}_{x|_i\sim S}[(-1)^{a_i(x|_i)}]$ is one if $a_i \in S^\perp$ and zero otherwise. Here

$$S^\perp = \{y : \langle y, x\rangle = 0 \text{ for all } x \in S\}.$$

Therefore

$$|\mathbf{E}_{x\sim D}[(-1)^{a(x)}]| = \mathbf{Pr}[\text{for all } i,\ a_i \in S^\perp] \le \mathbf{Pr}[a_1 \in S^\perp] = 2^{-(m-1)/k} = \epsilon$$

and so $|\mathbf{E}_{x\sim D}[a(x)] - 1/2| \le \epsilon/2 < \epsilon$. □

Proof of Claim 3.3. Let $X_1, \dots, X_k$ be independent samples from the distribution D and $X = X_1 + \cdots + X_k$. Let S_i denote the subspace of $\mathbb{F}_2^m$ associated to the sample X_i. Since each block of X_i belongs to the subspace S_i, each block of X will belong to the sum of subspaces $S = S_1 + \cdots + S_k$. The subspace S has dimension at most $m - 1$.

This suggests the following test for X: Arrange the first $2m$ blocks of X as rows in an $m \times 2m$ matrix M and compute the rank of M over $\mathbb{F}_2$. (By our choice of parameters, $2m^2 \leq n$ so this is always possible.) If the matrix has full rank output one, otherwise output zero. If X is chosen from D^k, then all the rows of M are chosen from the same subspace of dimension $m - 1$ so M will never have full rank. If X is chosen from the uniform distribution, then M is a random $m \times 2m$ matrix and, by a union bound, the probability it doesn't have full rank is at most $2^{-m} < 1/3$.

It remains to observe that the above test, which is essentially a rank computation, can be implemented by a circuit of depth $O((\log m)^2)$ via Cook's theorem [Coo85]. □

4 Acknowledgments

We thank Anup Rao for helpful conversations on this problem. This work was done while the authors took part in "China Theory Week" workshop at Tsinghua University. We would like to thank the organizers of the workshop and in particular Andy Yao for their hospitality.

References

[Bar89] D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *J. Comput. Syst. Sci.*, 38(1):150–164, 1989.

[BV07] A. Bogdanov and E. Viola. Pseudorandom bits for polynomials. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, 2007.

[Coo85] Stephen A. Cook. A taxonomy of problems with fast parallel algorithms. *Inf. Control*, 64(1-3):2–22, 1985.

[CRV07] K. M. Chung, O. Reingold, and S. Vadhan. S-t connectivity on digraphs with a known stationary distribution. In *IEEE Conference on Computational Complexity*, pages 236–249, 2007.

[FT05] Joel Friedman and Jean-Pierre Tillich. Generalized Alon–Boppana theorems and error-correcting codes. *SIAM J. Discret. Math.*, 19(3):700–718, 2005.

[Lov07] S. Lovett. Unconditional pseudorandom generators for low degree polynomials. *ECCC*, TR07-75, 2007.

[MRRW77] R. J. McEliece, E. R. Rodemich, H. Rumsey, and L. R. Welch. New upper bound on the rate of a code via the Delsarte-MacWilliams inequalities. *IEEE Trans. Inform. Theory*, 23(2):157–166, 1977.

[Nis92] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[Rei05] O. Reingold. Undirected st-connectivity in log-space. In *Proceedings of the annual ACM Symposium on Theory of Computing*, pages 376–385, 2005.

[RTV06] O. Reingold, L. Trevisan, and S. Vadhan. Pseudorandom walks on regular digraphs and the rl vs. l problem. In *Proceedings of the annual ACM Symposium on Theory of Computing*, pages 457–466, 2006.

[SZ99] M. E. Saks and S. Zhou. $bp_H space(s) \subseteq dspace(s^{3/2})$. *Comput. Syst. Sci.*, 58(2):376–403, 1999.

[VW07] E. Viola and A. Wigderson. Norms, xor lemmas, and lower bounds for gf(2) polynomials and multiparty protocols. In *IEEE Conference on Computational Complexity*, 2007.