Electronic Colloquium on Computational Complexity, Report No. 147 (2018)

Pseudorandom Generators for Read-Once Branching Programs, in any Order

Michael A. Forbes * Zander Kelley †

August 19, 2018

Abstract

A central question in derandomization is whether randomized logspace (RL) equals deterministic logspace (L). To show that $\mathsf{RL} = \mathsf{L}$, it suffices to construct explicit pseudorandom generators (PRGs) that fool polynomial-size read-once (oblivious) branching programs (roBPs). Starting with the work of Nisan [Nis92], pseudorandom generators with seed-length $O(\log^2 n)$ were constructed (see also [INW94, GR14]). Unfortunately, improving on this seed-length in general has proven challenging and seems to require new ideas.

A recent line of inquiry (e.g., [BV10, GMR+12, IMZ12, RSV13, SVW14, HLV17, LV17, CHRT17]) has suggested focusing on a particular limitation of the existing PRGs ([Nis92, INW94, GR14]), which is that they only fool roBPs when the variables are read in a particular *known* order, such as $x_1 < \cdots < x_n$. In comparison, existentially one can obtain logarithmic seed-length for fooling the set of polynomial-size roBPs that read the variables under any fixed *unknown* permutation $x_{\pi(1)} < \cdots < x_{\pi(n)}$. While recent works have established novel PRGs in this setting for subclasses of roBPs, there were no known $n^{o(1)}$ seed-length explicit PRGs for general polynomial-size roBPs in this setting.

In this work, we follow the "bounded independence plus noise" paradigm of Haramaty, Lee and Viola [HLV17, LV17], and give an improved analysis in the general roBP unknown-order setting. With this analysis we obtain an explicit PRG with seed-length $O(\log^3 n)$ for polynomial-size roBPs reading their bits in an unknown order. Plugging in a recent Fourier tail bound of Chattopadhyay, Hatami, Reingold, and Tal [CHRT17], we can obtain a $\widetilde{O}(\log^2 n)$ seed-length when the roBP is of constant width.

1 Introduction

A central goal in complexity theory is to understand the power of randomness in computation, in particular the P vs BPP problem. A particularly natural method of showing $\mathsf{P} = \mathsf{BPP}$ is to construct an explicit ε-error pseudorandom generator (PRG) with sufficiently small seed-length ℓ, ideally logarithmic. That is, a function $G : \{0,1\}^\ell \to \{0,1\}^n$ such that for any sufficiently efficiently computable f,

$$\left| \mathop{\mathbb{E}}_{y \in \{0,1\}^\ell} f(G(y)) - \mathop{\mathbb{E}}_{x \in \{0,1\}^n} f(x) \right| \leq \varepsilon \ .$$

Given such a PRG, one can then replace the randomness of a BPP algorithm with the pseudorandom output and then enumerate over all such seeds to obtain a deterministic algorithm by majority vote

*Email: miforbes@illinois.edu. Department of Computer Science, University of Illinois at Urbana-Champaign. Supported by NSF grant CCF-1755921.

†Email: awk2@illinois.edu. Department of Computer Science, University of Illinois at Urbana-Champaign. Supported by NSF grant CCF-1755921.

ISSN 1433-8092

(if ε is a sufficiently small constant). After decades of work, the hardness-vs-randomness paradigm (see for example Vadhan [Vad12]) shows that the construction of pseudorandom generators fooling general polynomial-size circuits is intimately tied to the quest for circuit lower bounds, which remain out of reach. As such, a long line of work has sought to derandomize subclasses of BPP. A particularly fruitful model to study has been randomized logspace (RL), as not only do PRGs for RL have natural applications, but they can also be unconditionally constructed, for example as done in the seminal work of Nisan [Nis92].

In particular, Nisan [Nis92] constructed a PRG fooling the non-uniform version of RL, that is, the class of polynomial-size read-once (oblivious) branching programs (roBPs). A read-once branching program can be thought of as a finite automaton that takes in binary input strings x of some fixed size n. Additionally, the transition function of the automaton is allowed to depend on the position i of each bit x_i. We say that the branching program has width w if the each layer of time the finite automaton has w states. Visually, branching programs can be represented as a layered acyclic digraph with $n+1$ layers, each containing w nodes; the transition function is then represented by assigning two outgoing edges at each interior node into the next layer. The existence of a logspace-computable PRG $G : \{0,1\}^{\ell(n)} \to \{0,1\}^n$ for branching programs of width $w = n^{O(1)}$ is sufficient to show that $\mathsf{BPL} \subseteq \mathsf{DSPACE}(\ell(n))$.

Nisan [Nis92] gave a construction of a PRG with seed-length $\ell = O(\log^2 n)$ for polynomial-width roBPs. Since then, there have been various constructions ([INW94, GR14]) recovering the same seed-length using different techniques, but there has been little quantitative progress towards the desired seed-length $\ell = O(\log n)$.[1] In fact, it remains open even to achieve a seed-length of $\ell = O(\log^2 n / \log\log n)$, even for constant-width branching programs.

The constructions of Nisan and ([INW94, GR14]) all employ a common high-level approach which can be summarized by the following "communication" argument. The first half of a branching program can communicate with the second half only via the state reached in the middle layer. Since there are only w states in this layer, the second half of the program should "learn" roughly only $\log w$ bits of information about the input bits fed to the first half. Because of this, it is safe to reuse all but roughly $\log w$ of the bits of entropy invested to generate the first half of the input string to generate the second half. This argument is then applied recursively to the left and right subprograms.

There is some feeling that this particular recursive paradigm will not yield generators with seed-lengths better than $O(\log^2 n)$ ([BV10, RSV13, SVW14]), and that new, more flexible techniques are required to make progress. A crucial feature of this paradigm is that the PRG knows the order in which its pseudorandom output will be read. In fact, it is known that Nisan's generator fails to generate pseudorandom strings that fool branching programs if they read the bits of the string in a different order than anticipated ([Tzu09]). The search for a different paradigm motivates the following challenge: construct a PRG that fools branching programs which may read their input in any order. To formalize this, we define the notion of an unknown-order roBP: a function $g : \{0,1\}^n \to \{0,1\}$ of the form $g(x) = f(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$, where π is a permutation (independent of x) and f is a roBP.

Bogdanov, Papakonstantinou, and Wan [BPW11] constructed a PRG with seed-length $(1-\Omega(1))\cdot n$ for unknown-order roBP of width $w = n^{O(1)}$. Their primary motivation for doing so was to derive the first generator with nontrivial seed-length that fools read-once formulas. Read-once formulas can be simulated by small-width read-once branching programs for *some* order π, and hence existing generators for *known*-order roBPs ([Nis92, INW94, GR14]) would not suffice. Impagliazzo, Meka,

[1] By this we mean quantitative progress in the constant-error regime. Recently in [BCG17], Braverman, Cohen, and Garg give a hitting set (a "one-sided" PRG) with a better seed-length in the small-error regime than Nisan's generator.

and Zuckerman [IMZ12] achieved a generator with seed length $\ell = (nw)^{1/2+o(1)}$ for unknown-order roBP of width w.[2]

2 Our Work

Here, we give the first PRG with poly-logarithmic seed-length for $\mathsf{poly}(n)$-width unknown-order roBPs.

Theorem 2.1. *There exists an explicit $1/\mathsf{poly}(n)$-error pseudorandom generator $G : \{0,1\}^{O(\log^3 n)} \to \{0,1\}^n$ for the class of functions computable by a $\mathsf{poly}(n)$-width read-once (oblivious) branching program in some variable order.*

As a corollary, we also derive the first PRG with poly-logarithmic seed length for read-once formulas (see [BPW11] for the reduction).

Corollary. *There exists an explicit pseudorandom generator $G : \{0,1\}^{O(\log^3 n)} \to \{0,1\}^n$ for read-once formulas with constant fan-in.* □

2.1 Our Techniques

We now briefly describe our proof technique at a high-level, with a more technical discussion given in Section 5. The main motivation comes from the "bounded independence plus noise" paradigm introduced by Haramaty, Lee, and Viola ([HLV17, LV17]). There, they study the addition (modulo 2) of a low-wise independent distribution with a pseudorandom noise distribution. The intuition is that to fool a function f, it suffices to create a distribution to dampen all non-constant Fourier coefficients. For low-degree Fourier coefficients, this can be achieved by a low-wise independent distribution. In the other extreme, high-degree Fourier coefficients are dampened by coordinate-wise independent noise. The addition of these two distributions can then inherit the best of both distributions and fool the desired function f.

However, the above outline has two challenges. First, the noise distribution (picking each coordinate independently amongst $\{0,1\}$) requires too large a seed-length. To address this, the work of Haramaty, Lee, and Viola ([HLV17, LV17]) proposed to use a pseudorandom noise distribution where a pseudorandom set of coordinates are first chosen, and then the elements within those coordinates are then substituted with *truly* random values. While this proposal as stated still requires a large seed-length, the key observation is that the number of truly random bits has shrunk from n originally to $\approx n/2$ (for if you choose a (pseudorandom) subset of $\{1,\ldots,n\}$ it has size $\approx n/2$). Thus, one can hope to then recursively apply the construction in $\approx \log n$ rounds until no random bits are further required.

The second, more serious, challenge is to show that a single step of "bounded independence plus (pseudorandom) noise" actually fools the target function f, and doing so is the main contribution of this work (Lemma 6.3). The difficulty in addressing this is that there are too many high-degree Fourier coefficients, so that while each can be individually fooled by the construction, we cannot apply a union bound while maintaining a small seed-length. Indeed, nothing so far in this discussion has used anything about the structure of the function f, which clearly must be used to obtain a small seed-length.

To meet this challenge, we avoid a naive union bound by instead grouping high-degree Fourier coefficients into a small number of groups, each of which can be dampened at once. Specifically,

[2] In fact, their generator fools the more general model of branching programs that may read the input bits any number of times and in any adaptive order.

we group high-degree Fourier coefficients into n sets, where the i-th set contains those coefficients that "become" high-degree upon reading the i-th variable (Proposition 6.1).[3] On an intuitive level, one can then appeal to the "bounded communication" aspect of a roBP to argue that in the i-th grouping those variables read after the i-th variable can be essentially ignored. We are then left with Fourier coefficients that are of *medium* degree (for if they were very-high degree they would have been put in the j-th group for some $j < i$). The number of such medium-degree Fourier coefficients is not too large (because the degree is not too large), and yet each such coefficient is dampened by the noise distribution (because the degree is not too small). This then allows us to apply a union bound to obtain that we have dampened all the Fourier coefficients in the i-th grouping, and by applying this for all i we obtain the result.

2.2 The Constant-Width Case

Although we give the first PRG with poly-logarithmic seed-length for the general case of poly-width unknown-order roBPs, such a seed-length has been qualitatively achieved in the constant-width case as a result of a recent line of work. Reingold, Steinke, and Vadhan [RSV13] gave a PRG with seed-length $O(\log^2 n)$ for unknown-order *permutation* branching programs of constant width. Steinke, Vadhan, and Wan [SVW14] gave a PRG with seed-length $\widetilde{O}(\log^3 n)$ for unknown-order width-3 branching programs. Chattopadhyay, Hatami, Reingold, and Tal [CHRT17] gave a PRG with seed-length $\widetilde{O}(\log^{w+1} n)$ for unknown-order branching programs of constant-width w. Central to each of these results is a bound on a certain key quantity: the level-k Fourier mass of a branching program (see Section 3). In each work, a bound on this quantity is established for the class of branching programs under consideration, and then this bound is used to deduce the result.

Although we also employ a Fourier analytic approach, a major contrast between our techniques and this line of work is that in general we have no need of any nontrivial bound on the Fourier mass of branching programs. However, we can still make use of one to replace otherwise naive bounds on Fourier mass in our argument. By incorporating the level-k Fourier mass bound for constant-width branching programs derived in [CHRT17] into our approach, we get the following improvement on Theorem 2.1 in the constant-width case.

Theorem 2.2. *There exists an explicit $1/\mathsf{poly}(n)$-error pseudorandom generator $G : \{0,1\}^{\widetilde{O}(\log^2 n)} \to \{0,1\}^n$ for the class of functions computable by a $O(1)$-width read-once (oblivious) branching program in some variable order.*

Thus in the constant-width case, we nearly recover the $O(\log^2 n)$ seed-length of Nisan's generator for the more challenging model of unknown-order branching programs.

3 Preliminaries

Here we describe a convenient algebraic encoding of a branching program as a product of one-bit matrix-valued functions. Recall that a branching program of width w is a w-state finite automaton where the transition map is allowed to depend on the number of bits read so far. Let us encode the w states as the set of standard basis vectors in $\mathbb{R}^w$. Then, the transition map corresponding to the i-th input bit x_i can be encoded by a pair of transition matrices $A_{i,0}, A_{i,1} \in \mathbb{R}^{w \times w}$, defined so that A_{i,x_i} applied to the current state produces the appropriate successor state. Define the

[3] Note that this grouping depends on the order of the variables. However, this grouping only occurs in the *analysis*, and that the construction itself is oblivious to the variable order.

one-bit matrix-valued functions $F_i(x_i) = A_{i,x_i}$. With this notation in place, the value of a branching program f on an input $x \in \{0,1\}^n$ is given by the $(1,1)$ entry of the product

$$F(x) := F_1(x_1)F_2(x_2)\cdots F_n(x_n) \ .$$

This entry indicates whether the string x defines a path through the program that takes the start state to the accepting state.

Let U denote the uniform distribution over $\{0,1\}^n$ and let X be an arbitrary distribution over $\{0,1\}^n$. To show that a branching program f is ε-fooled by X, it suffices to bound the error $|\mathbb{E}_X f(X) - \mathbb{E}_U f(U)| = |\mathbb{E}_X F(X)_{1,1} - \mathbb{E}_U F(U)_{1,1}|$ by ε. However, it will be more convenient to simply bound the Frobenius norm of the entire error matrix

$$\mathbb{E}_X F(X) - \mathbb{E}_U F(U) \ ,$$

by ε. Recall that the Frobenius norm of a matrix is defined by

$$\|M\| := \sqrt{\mathrm{tr}\,(M^\top M)} = \sqrt{\sum_{i,j} M_{i,j}^2} \ ,$$

so clearly such a bound is also sufficient. In this paper, $\|\cdot\|$ will always denote the Frobenius norm.

3.1 Fourier Analysis

For every vector $\alpha \in \mathbb{F}_2^n$, define the associated Fourier character $\chi_\alpha : \mathbb{F}_2^n \to \mathbb{R}$ via

$$\chi_\alpha(x) = (-1)^{\langle \alpha, x \rangle}.$$

We say that χ_α is a degree-k Fourier coefficient if $|\alpha| = k$, where $|\alpha|$ denotes the hamming weight of α. Any function $F : \mathbb{F}_2^n \to \mathbb{R}^{w \times w}$ can be expanded in the basis of Fourier characters, with coefficients from the ring $\mathbb{R}^{w \times w}$. The Fourier expansion of F is

$$F(x) = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{F}_\alpha \chi_\alpha(x),$$

where the Fourier coefficients $\widehat{F}_\alpha$ are given by

$$\widehat{F}_\alpha := \mathop{\mathbb{E}}_{x \in \mathbb{F}_2^n} F(x)\chi_\alpha(x).$$

This identity can easily be checked with the aid of a few useful properties of Fourier characters; namely

$$\chi_\alpha(x)\chi_\beta(x) = \chi_{\alpha+\beta}(x)$$

and

$$\mathop{\mathbb{E}}_{x \in \mathbb{F}_2^n} \chi_\alpha(x) = \begin{cases} 1 & \text{if } \alpha = 0 \\ 0 & \text{otherwise.} \end{cases}$$

Aside from this, we will only need a few simple facts about Fourier analysis. Firstly, we note that the expectation of a function F under the uniform distribution is conveniently encoded by its 0-th degree Fourier coefficient:

$$\mathbb{E}_U F(U) = \sum_{\alpha \in \mathbb{F}_2^n} \widehat{F}_\alpha \mathbb{E}_U \chi_\alpha(U) = \widehat{F}_0.$$

Next, we will need Parseval's identity to get a bound on the sum of squares of Fourier coefficients.

Proposition 3.1 (Parseval for matrix-valued functions)**.**

$$\sum_{\alpha\in\mathbb{F}_2^n} \|\widehat{F}_\alpha\|^2 = \mathop{\mathbb{E}}_{x\in\mathbb{F}_2^n} \|F(x)\|^2 \,.$$

Proof: Let $\langle\cdot,\cdot\rangle$ denote the Frobenius matrix inner product, which is defined by

$$\langle M, N\rangle = \mathrm{tr}\Big(M^\top N\Big) = \sum_{i,j} M_{i,j}N_{i,j}.$$

First note that

$$\begin{aligned}\|F(x)\|^2 &= \langle F(x), F(x)\rangle \\ &= \left\langle \sum_{\alpha\in\mathbb{F}_2^n} \widehat{F}_\alpha\chi_\alpha(x), \sum_{\beta\in\mathbb{F}_2^n} \widehat{F}_\beta\chi_\beta(x)\right\rangle \\ &= \sum_{\alpha\in\mathbb{F}_2^n}\sum_{\beta\in\mathbb{F}_2^n} \left\langle \widehat{F}_\alpha, \widehat{F}_\beta\right\rangle \chi_{\alpha+\beta}(x) \,.\end{aligned}$$

Thus

$$\begin{aligned}\mathbb{E}_{x\in\mathbb{F}_2^n}\|F(x)\|^2 &= \sum_{\alpha\in\mathbb{F}_2^n}\sum_{\beta\in\mathbb{F}_2^n} \left\langle \widehat{F}_\alpha, \widehat{F}_\beta\right\rangle \mathop{\mathbb{E}}_{x\in\mathbb{F}_2^n}\chi_{\alpha+\beta}(x) \\ &= \sum_{\alpha\in\mathbb{F}_2^n} \left\langle \widehat{F}_\alpha, \widehat{F}_\alpha\right\rangle \\ &= \sum_{\alpha\in\mathbb{F}_2^n} \|\widehat{F}_\alpha\|^2 \,.\end{aligned}$$

□

We remark that if F is a branching program, then upon any input x, $F(x)$ is equal to some transition matrix that has exactly w entries of value 1 and its remaining entries are all zeros. Thus for branching programs we have $\|F(x)\|^2 = w$ for any x, and the above identity gives $\sum_{\alpha\in\mathbb{F}_2^n} \|\widehat{F}_\alpha\|^2 = w$.

Finally, we define $\mathcal{L}_k(F)$, the level-k Fourier mass of a function, as in [RSV13]:

$$\mathcal{L}_k(F) := \sum_{\substack{\alpha\in\mathbb{F}_2^n \\ |\alpha|=k}} \|\widehat{F}_\alpha\| \,.$$

Recalling that $\|\widehat{F}_\alpha\| = \|\mathbb{E}_x F(x)\chi_\alpha(x)\| \le \mathbb{E}_x\|F(x)\| = w^{1/2}$, note that we have the trivial bound

$$\mathcal{L}_k(F) \le \binom{n}{k} w^{1/2} \,,$$

for branching programs.[4]

Chattopadhyay, Hatami, Reingold, and Tal [CHRT17] derive the following bound on the level-k Fourier mass of a branching program which significantly improves upon the trivial bound in the case of small w.

Theorem 3.2 (Chattopadhyay, Hatami, Reingold, and Tal [CHRT17])**.** *Suppose $F : \mathbb{F}_n^2 \to \mathbb{R}^{w\times w}$ encodes a branching program of width w. Then*

$$\mathcal{L}_k(F) \le O(\log n)^{wk} \,.$$

□

[4] Actually, one can easily derive the slightly better bound $\mathcal{L}_k(F) \le \binom{n}{k}^{1/2} w^{1/2}$ by first applying Cauchy-Schwarz followed by Parseval's identity.

We also define the level-k Fourier complexity of width-w branching programs,

$$\mathcal{L}(n, w; k) := \max_F \sum_{i=1}^{k} \mathcal{L}_i(F) \ ,$$

where the maximum is taken over all functions $F : \mathbb{F}_2^n \to \mathbb{R}^{w \times w}$ that encode a branching program.

3.2 Pseudorandom Primitives

In this section, we collect some basic pseudorandom distributions over $\mathbb{F}_2^n$ that will serve as building blocks for our generator. It is important that the defining property of each of these distributions remains unaffected by a re-ordering of bit positions. The fact that we build our generator out of permutation-invariant components is what allows it to fool branching programs that read their input in any order.

First we introduce δ-biased distributions, which fool linear functions over $\mathbb{F}_2^n$, i.e. Fourier characters.

Definition 3.3. *Let D be a distribution over $\mathbb{F}_2^n$. We say D is δ-biased if, for every nonzero $\alpha \in \mathbb{N}_2^n$, we have*

$$|\mathbb{E}_D \chi_\alpha(D)| \leq \delta \ . \qquad \lozenge$$

It is possible to sample from a δ-biased distribution using $O(\log n + \log 1/\delta)$ random bits ([NN93, AGHP92]).

Next we have k-wise independent and γ-almost k-wise independent distributions, which look locally uniform and thus fool functions that only depend on a few bits.

Definition 3.4. *Let D be a distribution over $\mathbb{F}_2^n$. We say D is k-wise independent if, for every $f : \mathbb{F}_2 \to [-1, 1]$ that depends on at most k bits, we have*

$$\mathbb{E}_D f(D) = \mathbb{E}_U f(U).$$

If D merely satisfies

$$|\mathbb{E}_D f(D) - \mathbb{E}_U f(U)| \leq \gamma$$

for every such f, we say that D is γ-almost k-wise independent. $\lozenge$

It is possible to sample from a k-wise independent distribution using $O(k \cdot \log n)$ random bits ([Vad12]) and from a γ-almost k-wise independent distribution using $O(k + \log\log n + \log 1/\gamma)$ random bits ([NN93, AGHP92]). We remark that Fourier characters $\chi_\alpha(x)$ only depend on $|\alpha|$ bits of x, so these distributions also fool low-degree Fourier characters.

4 The Generator

We adopt our construction from the "bounded independence plus noise" framework developed by Haramaty, Lee, and Viola in [HLV17, LV17]. In fact this framework is essentially equivalent to the "mild pseudorandom restriction" framework developed by Gopalan, Meka, Reingold, Trevisan and Vadhan [GMR+12] and subsequently employed by various authors (e.g., [RSV13, SVW14, CHRT17]), but we find the bounded independence plus noise perspective more convenient to work with.

We actually give two slightly different constructions. The first construction only uses k-wise independence as a core pseudorandom primitive (along with appropriate recursion), and suffices for proving Theorem 2.1. The second construction replaces the use of *exact* k-wise independence with the use small-bias spaces and *almost* k-wise independence. This allows the error analysis to

be more general (but also slightly more involved due to additional parameters). In particular, this second construction also suffices for handing general roBPs (Theorem 2.1), but now the added generality can be tuned to achieve a better seed-length for constant-width roBPS, as required to prove Theorem 2.2.

We proceed to give the first construction.

Let $D_1, D_2, \ldots, D_r$ denote r independent copies of a $2k$-wise independent distribution and let $T_1, T_2, \ldots, T_r$ denote r independent copies of a k-wise independent distribution over $\mathbb{F}_2^n$. We then define the distribution G_r recursively as follows. Let G_0 be equal to the all-ones string in $\mathbb{F}_2^n$, and set

$$G_{i+1} := D_i + T_i \wedge G_i,$$

where $\wedge$ denotes bitwise AND and $+$ denotes addition over $\mathbb{F}_2^n$ (i.e. bitwise XOR).

Lemma 4.1. *Suppose $F : \mathbb{F}_2^n \to \mathbb{R}^{w \times w}$ encodes a branching program. Then G_r, with parameters $k = \lceil 5 \lg n + 2 \lg w \rceil$ and $r = \lceil 2 \lg n + \frac{1}{2} \lg w \rceil$, fools F with error*

$$\varepsilon = \|\mathbb{E}_{G_r} F(G_r) - \mathbb{E}_U F(U)\| \leq O\left(\tfrac{1}{n}\right).$$

This proves Theorem 2.1, since in the case of width $w = n^{O(1)}$ the price of sampling such a distribution is $O(r \cdot k \cdot \log n) = O(\log^3 n)$ random bits.

We now define the second construction, G_r^*. This time, let $D_1, D_2, \ldots, D_r$ denote r independent δ-biased distributions, and let $T_1, T_2, \ldots, T_r$ denote r independent γ-almost k-wise independent distributions. We set G_0^* equal to a $320k$-wise independent distribution, and again we define

$$G_{i+1}^* := D_i + T_i \wedge G_i^*.$$

Lemma 4.2. *Suppose $F : \mathbb{F}_2^n \to \mathbb{R}^{w \times w}$ encodes a branching program. Then G_r^*, with depth $r = \lceil \lg n \rceil$, fools F with error*

$$\varepsilon = \|\mathbb{E}_{G_r^*} F(G_r^*) - \mathbb{E}_U F(U)\| \leq O\left(\left(\sqrt{\delta}\mathcal{L}(n, w; k) + \left(\tfrac{1}{2}\right)^{k/2} + \sqrt{\gamma} + \gamma 4^k\right) \cdot nwr\right).$$

Since G_r^* can be sampled at the cost of

$$O\big((\log n + \log 1/\delta + k + \log 1/\gamma) \cdot r + k \cdot \log n\big)$$

random bits, it suffices to set

- $r = \lceil \lg n \rceil$
- $k = \lceil 3 \lg(nw/\varepsilon) \rceil$
- $\gamma = (nw/\varepsilon)^{-9}$
- $\delta = (nw\mathcal{L}(n, w; k)/\varepsilon)^{-3}$

to get a generator with seed length

$$\ell = O\big((\log(nw/\epsilon) + \mathcal{L}(n, w; k)) \cdot \log n\big)$$

that $O(\varepsilon)$-fools branching programs F of width w. From this we derive the following two corollaries by invoking either the trivial bound or the bound from [CHRT17] on the level-k Fourier mass of width-w branching programs. The second of these corollaries proves Theorem 2.2.

Corollary 4.3. *There exists an explicit PRG with seed length*

$$\ell = O\big(\log(nw/\varepsilon) \cdot \log^2 n\big)$$

that ε-fools unknown-order branching programs of width w.

Corollary 4.4. *There exists an explicit PRG with seed length*

$$\ell = O\big(w \cdot \log(nw/\varepsilon) \cdot \log n \cdot \log\log n\big)$$

that ε-fools unknown-order branching programs of width w.

5 Proof Strategy

We show that G_r successfully fools branching programs with the following inductive analysis. By adding and subtracting the term $F(D_i + T_i \wedge U)$, we have

$$\begin{aligned}\|\mathbb{E}_{G_i} F(G_i) - \mathbb{E}_U F(U)\| &= \|\mathbb{E}_{D_i}\mathbb{E}_{T_i}\mathbb{E}_{G_{i-1}} F(D_i + T_i \wedge G_{i-1}) - \mathbb{E}_U F(U)\| \\ &\le \|\mathbb{E}_{D_i}\mathbb{E}_{T_i}\mathbb{E}_U F(D_i + T_i \wedge U) - \mathbb{E}_U F(U)\| \\ &\quad + \mathbb{E}_{D_i}\mathbb{E}_{T_i}\|\mathbb{E}_{G_{i-1}} F(D_i + T_i \wedge G_{i-1}) - \mathbb{E}_U F(D_i + T_i \wedge U)\|.\end{aligned}$$

Since, for any fixed vectors $d, t \in \mathbb{F}_2^n$, the function $F'(x) := F(d + t \wedge x)$ is again some branching program, we argue that F' is fooled inductively. The bulk of our proof is then spent arguing that $\mathbb{E}_{D,T,U} F(D + T \wedge U) \approx \mathbb{E}_U F(U)$.

The starting point of this argument is the observation that the "noise-like" distribution $T \wedge U$ successfully fools any function that is divisible by a high-degree Fourier character. Specifically, suppose that T is k-wise independent and that $\alpha \in \mathbb{F}_2^n$ has hamming weight $|\alpha| \ge k$. Let $g : \mathbb{F}_2^n \to [-1, 1]$ be an arbitrary function such that g and χ_α depend on disjoint sets of input bits. Then $T \wedge U$ fools the function $f := \chi_\alpha \cdot g$ with error $\varepsilon = 1/2^k$:

$$\mathbb{E}_U f(U) = \left(\mathbb{E}_U \chi_\alpha(U)\right)\left(\mathbb{E}_U g(U)\right) = 0\,,$$

and

$$\begin{aligned}|\mathbb{E}_{T,U} f(T \wedge U)| &= \left|\mathbb{E}_T\left(\mathbb{E}_U \chi_\alpha(T \wedge U)\right)\left(\mathbb{E}_U g(U)\right)\right| \\ &\le \mathbb{E}_T |\mathbb{E}_U \chi_\alpha(T \wedge U)||\mathbb{E}_U g(U)| \\ &\le \mathbb{E}_T |\mathbb{E}_U \chi_\alpha(T \wedge U)| \\ &= \mathbb{E}_T \mathbb{1}(\alpha \wedge T = 0) \\ &\le \frac{1}{2^k}\,.\end{aligned}$$

Overall, we wish to enact the following plan. If F encodes our branching program, we wish to use Fourier analysis to rewrite F as a sum of simpler terms, and then use linearity of expectation together with a triangle inequality to argue that $D + T \wedge U$ fools each term separately.[5] Recall the product structure of F,

$$F(x) = F_1(x_1)F_2(x_2)\cdots F_n(x_n),$$

[5] This approach is inspired by the similar arguments of Haramaty, Lee, and Viola employed in [HLV17]. However, the generators they produce with this idea have seed length $\ge n^{1/2}$, while we achieve generators with poly-logarithmic seed-length.

where $F_i(x_i) = A_{i,x_i}$. We can imagine taking the Fourier expansions of these one-bit factors:

$$F_i(x_i) = \tfrac{1}{2}(A_{i,0} + A_{i,1}) + \tfrac{1}{2}(A_{i,0} - A_{i,1})(-1)^{x_i} =: B_{i,0} + B_{i,1}(-1)^{x_i} \ ,$$

so that F has the form

$$F(x) = \prod_{i=1}^{n} \left(B_{i,0} + B_{i,1}(-1)^{x_i}\right) \ .$$

If we expand this product completely, we recover the Fourier expansion of F. Certainly the terms in this sum are simple enough, but the problem is that there are too many of them; we cannot afford a 2^n-fold triangle inequality. Instead, we expand this product more "slowly": only until we see that some term has collected a degree-k Fourier character as a factor. By the above observation, this term can be killed immediately, and we go on expanding the remaining terms. At the end, we are left with only low-degree Fourier coefficients, which can all be fooled by the distribution D. By doing this carefully, we get by with only n applications of the triangle inequality.

6 Proof of Lemma 4.1

Suppose

$$F(x) = F_1(x_1)F_2(x_2)\cdots F_n(x_n)$$

encodes a branching program. We define the subprograms of F,

$$F^{\leq i}(x_1, \ldots, x_i) := F_1(x_1)F_2(x_2)\cdots F_i(x_i)$$

and

$$F^{>i}(x_{i+1}, \ldots, x_n) := F_{i+1}(x_{i+1})F_{i+2}(x_{i+2})\cdots F_n(x_n) \ ,$$

so that $F = F^{\leq i} \cdot F^{>i}$ for any i (we define $F^{>n}$ as the empty product, so that it is an identity matrix I). With this notation in place, we can re-express F in the following convenient form.

Proposition 6.1. *Suppose $F : \mathbb{F}_2^n \to \mathbb{R}^{w \times w}$ encodes a branching program. Then for any $k \geq 1$, F can be written as*

$$F = \widehat{F}_0 + L + \sum_{i=1}^{n} H_i \cdot F^{>i},$$

where

$$L = \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ 0 < |\alpha| < k}} \widehat{F}_\alpha \chi_\alpha,$$

and

$$H_i = \sum_{\substack{\alpha \in \mathbb{F}_2^i \\ |\alpha| = k \\ \alpha_i = 1}} \widehat{F}_{\bar{\alpha}}^{\leq i} \chi_\alpha.$$

Proof: We verify that the expression has the same Fourier expansion as F.

$$\begin{aligned}
\sum_{i=1}^{n} H_i \cdot F^{>i} &= \sum_{i=1}^{n} \left(\sum_{\substack{\alpha \in \mathbb{F}_2^i \\ |\alpha|=k \\ \alpha_i=1}} \widehat{F}_{\alpha}^{\leq i} \chi_\alpha \right) \left(\sum_{\beta \in \mathbb{F}_2^{n-i}} \widehat{F}_{\beta}^{>i} \chi_\beta \right) \\
&= \sum_{i=1}^{n} \sum_{\substack{\alpha \in \mathbb{F}_2^i \\ |\alpha|=k \\ \alpha_i=1}} \sum_{\beta \in \mathbb{F}_2^{n-i}} \widehat{F}_{\alpha\beta} \chi_{\alpha\beta} \\
&= \sum_{i=1}^{n} \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ \alpha_1+\alpha_2+\cdots+\alpha_i=k \\ \alpha_i=1}} \widehat{F}_\alpha \chi_\alpha(x) \\
&= \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ |\alpha| \geq k}} \widehat{F}_\alpha \chi_\alpha \ . \qquad \square
\end{aligned}$$

Now we derive a useful expectation bound for functions whose Fourier expansions are only supported at degree k.

Lemma 6.2. *Let*

$$H(x) = \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ |\alpha|=k}} \widehat{H}_\alpha \chi_\alpha(x)$$

be some function whose Fourier expansion is supported only at degree k. Let D, T, and U denote respectively a $2k$-wise independent, a k-wise independent, and a uniform distribution over $\mathbb{F}_2^n$. Then we have

$$\mathbb{E}_{D,T} \| \mathbb{E}_U H(D + T \wedge U) \|^2 \leq \frac{1}{2^k} \sum_{|\alpha|=k} \| \widehat{H}_\alpha \|^2 \ .$$

Proof: Firstly, note that

$$\begin{aligned}
\mathbb{E}_U H(D + T \wedge U) &= \mathbb{E}_U \sum_{|\alpha|=k} \widehat{H}_\alpha \cdot \chi_\alpha(D + T \wedge U) \\
&= \mathbb{E}_U \sum_{|\alpha|=k} \widehat{H}_\alpha \cdot \chi_\alpha(D) \cdot \chi_\alpha(T \wedge U) \\
&= \sum_{|\alpha|=k} \widehat{H}_\alpha \cdot \chi_\alpha(D) \cdot \mathbb{E}_U \chi_\alpha(T \wedge U) \\
&= \sum_{|\alpha|=k} \widehat{H}_\alpha \cdot \chi_\alpha(D) \cdot \mathbb{1}(\alpha \wedge T = 0) \ .
\end{aligned}$$

Letting $\langle \cdot, \cdot \rangle$ denote the Frobenius matrix inner product, we have

$$\begin{aligned}
\| \mathbb{E}_U H(D + T \wedge U) \|^2 &= \langle \mathbb{E}_U H(D + T \wedge U), \mathbb{E}_U H(D + T \wedge U) \rangle \\
&= \left\langle \sum_{|\alpha|=k} \widehat{H}_\alpha \chi_\alpha(D) \mathbb{1}(\alpha \wedge T = 0) \, , \sum_{|\beta|=k} \widehat{H}_\beta \chi_\beta(D) \mathbb{1}(\beta \wedge T = 0) \right\rangle \\
&= \sum_{|\alpha|=k} \sum_{|\beta|=k} \left\langle \widehat{H}_\alpha, \widehat{H}_\beta \right\rangle \cdot \chi_{\alpha+\beta}(D) \cdot \mathbb{1}((\alpha \vee \beta) \wedge T = 0)
\end{aligned}$$

and

$$\begin{aligned}
\mathbb{E}_D\|\mathbb{E}_U H(D+T\wedge U)\|^2 &= \mathbb{E}_D \sum_{|\alpha|=k}\sum_{|\beta|=k} \left\langle \widehat{H}_\alpha, \widehat{H}_\beta \right\rangle \chi_{\alpha+\beta}(D)\mathbb{1}((\alpha\vee\beta)\wedge T=0) \\
&= \sum_{|\alpha|=k}\sum_{|\beta|=k} \left\langle \widehat{H}_\alpha, \widehat{H}_\beta \right\rangle \Big(\mathbb{E}_D\chi_{\alpha+\beta}(D)\Big)\mathbb{1}((\alpha\vee\beta)\wedge T=0) \\
&= \sum_{|\alpha|=k} \left\langle \widehat{H}_\alpha, \widehat{H}_\alpha \right\rangle \mathbb{1}(\alpha\wedge T=0) \\
&= \sum_{|\alpha|=k} \|\widehat{H}_\alpha\|^2 \mathbb{1}(\alpha\wedge T=0)\ .
\end{aligned}$$

Finally,

$$\begin{aligned}
\mathbb{E}_{D,T}\|\mathbb{E}_U H(D+T\wedge U)\|^2 &= \mathbb{E}_T\mathbb{E}_D\|\mathbb{E}_U H(D+T\wedge U)\|^2 \\
&= \mathbb{E}_T \sum_{|\alpha|=k} \|\widehat{H}_\alpha\|^2 \mathbb{1}(\alpha\wedge T=0) \\
&= \sum_{|\alpha|=k} \|\widehat{H}_\alpha\|^2 \cdot \mathbb{E}_T\mathbb{1}(\alpha\wedge T=0) \\
&= \sum_{|\alpha|=k} \|\widehat{H}_\alpha\|^2 \cdot \left(\tfrac{1}{2}\right)^k\ . \qquad \square
\end{aligned}$$

We now have the tools in place to prove our main technical lemma.

Lemma 6.3. *Suppose $F : \mathbb{F}_2^n \to \mathbb{R}^{w\times w}$ encodes a branching program. Let D, T, and U denote respectively a $2k$-wise independent, a k-wise independent, and a uniform distribution over $\mathbb{F}_2^n$. Then $D+T\wedge U$ fools F with error*

$$\varepsilon = \|\mathbb{E}_{D,T,U}F(D+T\wedge U) - \mathbb{E}_U F(U)\| \le \frac{nw}{2^{k/2}}.$$

Proof: To analyze $\|\mathbb{E}_{D,T,U}F(D+T\wedge U) - \mathbb{E}_U F(U)\|$, we use the preceding expansion of F together with linearity of expectation and the triangle inequality. Recalling that $\mathbb{E}_U F(U) = \widehat{F}_0$, this gives

$$\begin{aligned}
&\|\mathbb{E}_{D,T,U}F(D+T\wedge U) - \mathbb{E}_U F(U)\| \le \\
&\qquad \|\mathbb{E}_{D,T,U}L(D+T\wedge U)\| + \sum_{i=1}^n \|\mathbb{E}_{D,T,U}H_i(D+T\wedge U)F^{>i}(D+T\wedge U)\|\ .
\end{aligned}$$

The low-degree term is dealt with easily by D:

$$\begin{aligned}
\mathbb{E}_{D,T,U}L(D+T\wedge U) &= \mathbb{E}_D\mathbb{E}_T\mathbb{E}_U \sum_{\substack{\alpha\in\mathbb{F}_2^n \\ 0<|\alpha|<k}} \widehat{F}_\alpha\chi_\alpha(D+T\wedge U) \\
&= \mathbb{E}_D\mathbb{E}_T\mathbb{E}_U \sum_{\substack{\alpha\in\mathbb{F}_2^n \\ 0<|\alpha|<k}} \widehat{F}_\alpha\chi_\alpha(D)\chi_\alpha(T\wedge U) \\
&= \mathbb{E}_T\mathbb{E}_U \sum_{\substack{\alpha\in\mathbb{F}_2^n \\ 0<|\alpha|<k}} \widehat{F}_\alpha \cdot \Big(\mathbb{E}_D\chi_\alpha(D)\Big)\cdot\chi_\alpha(T\wedge U) \\
&= 0\ .
\end{aligned}$$

Now for each i we have

$$\begin{aligned}
\|\mathbb{E}_{D,T,U} H_i(D+T\wedge U)F^{>i}(D+T\wedge U)\| &= \Big\|\mathbb{E}_{D,T}\big(\mathbb{E}_U H_i(D+T\wedge U)\big)\big(\mathbb{E}_U F^{>i}(D+T\wedge U)\big)\Big\| \\
&\le \mathbb{E}_{D,T}\|\mathbb{E}_U H_i(D+T\wedge U)\|\|\mathbb{E}_U F^{>i}(D+T\wedge U)\| \\
&\le \mathbb{E}_{D,T}\|\mathbb{E}_U H_i(D+T\wedge U)\| w^{1/2} \\
&\le \Big(\mathbb{E}_{D,T}\|\mathbb{E}_U H_i(D+T\wedge U)\|^2\Big)^{1/2} w^{1/2} \\
&\le \Big(\tfrac{1}{2}\Big)^{k/2}\Big(\sum_{\alpha\in\mathbb{F}_2^n}\|\widehat{F}_\alpha^{\le i}\|^2\Big)^{1/2} w^{1/2} \\
&= \Big(\tfrac{1}{2}\Big)^{k/2} w,
\end{aligned}$$

where we get the final equality by applying the Parseval identity to $F^{\le i}$. □

6.1 Proof of Lemma 4.1

Proof: Recall the induction framework outlined in Section 5:

$$\begin{aligned}
\|\mathbb{E}_{G_i}F(G_i) - \mathbb{E}_U F(U)\| &= \|\mathbb{E}_{D_i}\mathbb{E}_{T_i}\mathbb{E}_{G_{i-1}}F(D_i+T_i\wedge G_{i-1}) - \mathbb{E}_U F(U)\| \\
&\le \|\mathbb{E}_{D_i}\mathbb{E}_{T_i}\mathbb{E}_U F(D_i+T_i\wedge U) - \mathbb{E}_U F(U)\| \\
&\quad + \mathbb{E}_{D_i}\mathbb{E}_{T_i}\|\mathbb{E}_{G_{i-1}}F(D_i+T_i\wedge G_{i-1}) - \mathbb{E}_U F(D_i+T_i\wedge U)\|.
\end{aligned}$$

We have seen how to carry out the inductive step; it remains to establish the base case. To do this, we wish to think of $F(G_r)$ as a function of G_0 only, with $D_1, D_2, \ldots, D_r$ and $T_1, T_2, \ldots, T_r$ fixed.

Specifically, we do the following. Define the strings $g_0 := x$ and $g_i := D_i + T_i \wedge g_{i-1}$, and define the function $f(x) := F(g_r)$. Note that with this setup we have $F(G_r) = f(G_0)$, and so

$$\begin{aligned}
\|\mathbb{E}_{G_r}F(G_r) - \mathbb{E}_U F(U)\| &= \|\mathbb{E}_{D_r}\mathbb{E}_{T_r}\mathbb{E}_{G_{r-1}}F(D_r+T_r\wedge G_{r-1}) - \mathbb{E}_U F(U)\| \\
&\le \frac{nw}{2^{k/2}} + \mathbb{E}_{D_r}\mathbb{E}_{T_r}\|\mathbb{E}_{G_{r-1}}F(D_r+T_r\wedge G_{r-1}) - \mathbb{E}_U F(D_r+T_r\wedge U)\| \\
&\le \frac{nw}{2^{k/2}} + \frac{nw}{2^{k/2}} + \cdots + \frac{nw}{2^{k/2}} + \mathop{\mathbb{E}}_{\substack{D_1,D_2,\ldots,D_r \\ T_1,T_2,\ldots,T_r}}\|\mathbb{E}_{G_0}f(G_0) - \mathbb{E}_U f(U)\|.
\end{aligned}$$

Now we must show that the function f is fooled by G_0 for most values of D_i and T_i. Luckily, for r large enough, f is often a constant function and therefore fooled by any distribution.

In particular, let $T_i[j]$ denote the j-th bit of T_i and define the indicator random variables

$$Y_j = \bigwedge_{i=1}^{r} T_i[j].$$

Note that $f(x)$ depends on the j-th bit of x only if $Y_j = 1$, and so $f(x)$ is constant if $\sum_{j=1}^n Y_j = 0$. Also note that $\Pr(Y_j = 1) = 2^{-r}$. By applying a Markov inequality, we have

$$\mathbb{E}_{\substack{D_1,D_2,\ldots,D_r\\T_1,T_2,\ldots,T_r}}\|\mathbb{E}_{G_0}f(G_0)-\mathbb{E}_U f(U)\| \le \Pr\left(\sum_{j=1}^{n} Y_j \ge 1\right) \max_{\substack{D_1,D_2,\ldots,D_r\\T_1,T_2,\ldots,T_r}}\|\mathbb{E}_{G_0}f(G_0)-\mathbb{E}_U f(U)\|$$
$$\le \mathbb{E}\left[\sum_{j=1}^{n} Y_j\right]\cdot 2w^{1/2}$$
$$= \frac{2nw^{1/2}}{2^r}\,.$$

If we set $k \ge 5\lg n + 2\lg w$ and $r \ge 2\lg n + \frac{1}{2}\lg w$, G_r fools F with error

$$\varepsilon = \|\mathbb{E}_{G_r}F(G_r)-\mathbb{E}_U F(U)\| \le r\cdot\frac{nw}{2^{k/2}} + \frac{2nw^{1/2}}{2^r} \le \frac{3}{n}\,. \qquad \square$$

7 Proof of Lemma 4.2

The proof of Lemma 4.2 follows the same story as the previous section with details differing in two places. First, we derive an analogue of Lemma 6.2 which is slightly messier due to our now weaker pseudorandom primitives. Secondly, in order to get the best possible seed-length in the small error regime, this time we analyze the base case with a bit more care. In particular, we upgrade the Markov argument to a Chernoff bound for γ-almost k-wise independent variables.

Lemma 7.1. *Let*

$$H(x) = \sum_{\substack{\alpha\in\mathbb{F}_2^n\\|\alpha|=k}} \widehat{H}_\alpha\chi_\alpha(x)$$

be some function whose Fourier expansion is supported only at degree k. Let D, T, and U denote respectively a δ-biased, a γ-almost k-wise independent, and a uniform distribution over $\mathbb{F}_2^n$. Then we have

$$\mathbb{E}_{D,T}\|\mathbb{E}_U H(D+T\wedge U)\|^2 \le \left(2^{-k}+\gamma\right)\left(\delta\cdot\Big(\sum_{|\alpha|=k}\|\widehat{H}_\alpha\|\Big)^2 + \sum_{|\alpha|=k}\|\widehat{H}_\alpha\|^2\right)\,.$$

Proof: As before, we have

$$\|\mathbb{E}_U H(D+T\wedge U)\|^2 = \sum_{|\alpha|=k}\sum_{|\beta|=k}\left\langle\widehat{H}_\alpha,\widehat{H}_\beta\right\rangle\chi_{\alpha+\beta}(D)\mathbb{1}((\alpha\vee\beta)\wedge T=0)\,.$$

We analyze the terms with $\alpha=\beta$ and $\alpha\neq\beta$ separately. For the cross terms we have

$$\begin{aligned}
&\mathbb{E}_D\mathbb{E}_T\sum_{|\alpha|=k}\sum_{\beta\neq\alpha}\left\langle\widehat{H}_\alpha,\widehat{H}_\beta\right\rangle\chi_{\alpha+\beta}(D)\mathbb{1}((\alpha\vee\beta)\wedge T=0)\\
&= \sum_{|\alpha|=k}\sum_{\beta\neq\alpha}\left\langle\widehat{H}_\alpha,\widehat{H}_\beta\right\rangle\Big(\mathbb{E}_D\chi_{\alpha+\beta}(D)\Big)\Big(\mathbb{E}_T\mathbb{1}((\alpha\vee\beta)\wedge T=0)\Big)\\
&\le \sum_{|\alpha|=k}\sum_{\beta\neq\alpha}\|\widehat{H}_\alpha\|\|\widehat{H}_\beta\|\cdot\delta\cdot\left(2^{-k}+\gamma\right)\\
&\le \delta\cdot\left(2^{-k}+\gamma\right)\Big(\sum_{|\alpha|=k}\|\widehat{H}_\alpha\|\Big)^2\,.
\end{aligned}$$

For the like terms we have

$$\begin{aligned}
&\mathbb{E}_T\mathbb{E}_D \sum_{|\alpha|=k} \left\langle \widehat{H}_\alpha, \widehat{H}_\alpha \right\rangle \chi_0(D)\mathbb{1}(T \wedge \alpha = 0) \\
&\quad = \sum_{|\alpha|=k} \|\widehat{H}_\alpha\|^2 \cdot \mathbb{E}_T \mathbb{1}(T \wedge \alpha = 0) \\
&\quad \leq \sum_{|\alpha|=k} \|\widehat{H}_\alpha\|^2 \cdot \left(2^{-k} + \gamma\right). \qquad \square
\end{aligned}$$

We now derive the following analogue of Lemma 6.3.

Lemma 7.2. *Suppose $F : \mathbb{F}_2^n \to \mathbb{R}^{w \times w}$ encodes a branching program. Let D, T, and U denote respectively a δ-biased, a γ-almost k-wise independent, and a uniform distribution over $\mathbb{F}_2^n$, and let Then $D + T \wedge U$ fools F with error*

$$\varepsilon = \|\mathbb{E}_{D,T,U} F(D + T \wedge U) - \mathbb{E}_U F(U)\| \leq \left(\sqrt{\delta}\mathcal{L}(n, w; k) + \left(\tfrac{1}{2}\right)^{k/2} + \sqrt{\gamma}\right) \cdot nw.$$

Proof: Again we use Proposition 6.1 to split F into high and low degree components:

$$\begin{aligned}
&\|\mathbb{E}_{D,T,U} F(D + T \wedge U) - \mathbb{E}_U F(U)\| \\
&\qquad \leq \|\mathbb{E}_{D,T,U} L(D + T \wedge U)\| + \sum_{i=1}^{n} \|\mathbb{E}_{D,T,U} H_i(D + T \wedge U) F^{>i}(D + T \wedge U)\|.
\end{aligned}$$

For the low-degree component we have

$$\begin{aligned}
\|\mathbb{E}_{D,T,U} F(D + T \wedge U)\| &= \|\mathbb{E}_D\mathbb{E}_T\mathbb{E}_U \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ 0<|\alpha|<k}} \widehat{F}_\alpha \chi_\alpha(D + T \wedge U)\| \\
&= \|\mathbb{E}_T\mathbb{E}_U \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ 0<|\alpha|<k}} \widehat{F}_\alpha \Big(\mathbb{E}_D \chi_\alpha(D)\Big) \chi_\alpha(T \wedge U)\| \\
&\leq \sum_{\substack{\alpha \in \mathbb{F}_2^n \\ 0<|\alpha|<k}} \|\widehat{F}_\alpha\| \cdot \delta \\
&= \delta \sum_{i=1}^{k-1} \mathcal{L}_i(F).
\end{aligned}$$

Now we proceed as before.

$$\begin{aligned}
\|\mathbb{E}_{D,T,U}F(D+T\wedge U)-\mathbb{E}_U F(U)\| & \\
&\le \delta\sum_{i=1}^{k-1}\mathcal{L}_i(F)+\sum_{i=1}^{n}\|\mathbb{E}_{D,T,U}H_i(D+T\wedge U)F^{>i}(D+T\wedge U)\| \\
&\le \delta\sum_{i=1}^{k-1}\mathcal{L}_i(F)+\sum_{i=1}^{n}\left(\mathbb{E}_{D,T}\|\mathbb{E}_U H_i(D+T\wedge U)\|^2\right)^{1/2}w^{1/2} \\
&\le \delta\sum_{i=1}^{k-1}\mathcal{L}_i(F)+\sum_{i=1}^{n}\left(\delta\mathcal{L}_k(F^{\le i})^2+\left(\tfrac{1}{2}\right)^k+\gamma\right)^{1/2}w \\
&\le \left(\sqrt{\delta}\mathcal{L}(n,w;k)+\left(\tfrac{1}{2}\right)^{k/2}+\sqrt{\gamma}\right)\cdot nw\ . \qquad \square
\end{aligned}$$

7.1 Proof of Lemma 4.2

Proof: We proceed as in the proof of Lemma 4.1, except this time we derive a sharper bound on the quantity

$$\mathop{\mathbb{E}}_{\substack{D_1,D_2,\ldots,D_r\\T_1,T_2,\ldots,T_r}}\|\mathbb{E}_{G_0^*}f(G_0^*)-\mathbb{E}_U f(U)\|.$$

Again, define the random variable

$$Y=T_1\wedge T_2\wedge\cdots\wedge T_r.$$

Recall that G_0^* is a $320k$-wise independent distribution, so if $|Y|\le 320k$ then

$$\|\mathbb{E}_{G_0^*}f(G_0^*)-\mathbb{E}_U f(U)\|=0.$$

Therefore

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{D_1,D_2,\ldots,D_r\\T_1,T_2,\ldots,T_r}}\|\mathbb{E}_{G_0^*}f(G_0^*)-\mathbb{E}_U f(U)\| &\le \Pr(|Y|\ge 320k)\max_{\substack{D_1,D_2,\ldots,D_r\\T_1,T_2,\ldots,T_r}}\|\mathbb{E}_{G_0^*}f(G_0^*)-\mathbb{E}_U f(U)\| \\
&\le \Pr(|Y|\ge 320k)\cdot 2w^{1/2}.
\end{aligned}$$

We appeal to the following extension of the Chernoff bound for k-wise independent variables.

Lemma 7.3. (see [SVW14], Lemma A.1) *Suppose $X_1,X_2,\ldots X_t$ are γ-almost k-wise independent variables with $X_i\in\{0,1\}$. Then*

$$\Pr\left(\frac{1}{t}\sum_{i=1}^{t}X_i\ge\frac{1}{2}+\frac{a}{2}\right)\le\left(\frac{40k}{a^2t}\right)^{\lfloor k/2\rfloor}+2\gamma\left(\frac{2}{a}\right)^k.$$

As a result, if T is a γ-almost k-wise independent distribution and α is any fixed bitmask with hamming weight $|\alpha|\ge 320k$, we have

$$\Pr\Big(|\alpha\wedge T|\ge\tfrac{3}{4}|\alpha|\Big)\le\left(\tfrac{1}{2}\right)^{\lfloor k/2\rfloor}+2\gamma\cdot 4^k.$$

Noting that $\left(\frac{3}{4}\right)^r n\le 1$, a simple union bound argument shows that

$$\Pr(|T_1\wedge T_2\wedge\cdots\wedge T_r|\ge 320k)\le r\cdot\left(\left(\tfrac{1}{2}\right)^{\lfloor k/2\rfloor}+2\gamma\cdot 4^k\right).$$

To conclude, we have

$$\begin{aligned}\|\mathbb{E}_{G_r^*} F(G_r^*) - \mathbb{E}_U F(U)\| &\leq r \cdot \left(\sqrt{\delta}\mathcal{L}(n,w;k) + \left(\tfrac{1}{2}\right)^{k/2} + \sqrt{\gamma}\right) \cdot nw + r \cdot \left(\left(\tfrac{1}{2}\right)^{\lfloor k/2 \rfloor} + 2\gamma \cdot 4^k\right) \cdot 2w^{1/2} \\ &\leq O\left(\left(\sqrt{\delta}\mathcal{L}(n,w;k) + \left(\tfrac{1}{2}\right)^{k/2} + \sqrt{\gamma} + \gamma 4^k\right) \cdot nwr\right) . \qquad \square\end{aligned}$$

References

[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.

[BCG17] Mark Braverman, Gil Cohen, and Sumegha Garg. Hitting sets with near-optimal error for read-once branching programs. 2017.

[BPW11] Andrej Bogdanov, Periklis A Papakonstaninou, and Andrew Wan. Pseudorandomness for read-once formulas. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 240–246. IEEE, 2011.

[BV10] Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 30–39. IEEE, 2010.

[CHRT17] Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *Electronic Colloquium on Computational Complexity (ECCC), pages TR17–171*, 2017.

[GMR+12] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 120–129. IEEE, 2012.

[GR14] Anat Ganor and Ran Raz. Space Pseudorandom Generators by Communication Complexity Lower Bounds. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2014)*, volume 28 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 692–703, Dagstuhl, Germany, 2014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[HLV17] Elad Haramaty, Chin Ho Lee, and Emanuele Viola. Bounded independence plus noise fools products. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 79. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017.

[IMZ12] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 111–119. IEEE, 2012.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 356–364. ACM, 1994.

[LV17] Chin Ho Lee and Emanuele Viola. More on bounded independence plus noise: Pseudorandom generators for read-once polynomials. 2017.

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

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM journal on computing*, 22(4):838–856, 1993.

[RSV13] Omer Reingold, Thomas Steinke, and Salil Vadhan. Pseudorandomness for regular branching programs via fourier analysis. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 655–670. Springer, 2013.

[SVW14] Thomas Steinke, Salil Vadhan, and Andrew Wan. Pseudorandomness and fourier growth bounds for width 3 branching programs. *arXiv preprint arXiv:1405.7028*, 2014.

[Tzu09] Yoav Tzur. *Notions of weak pseudorandomness and GF*(2^n)*-polynomials*. PhD thesis, Master's thesis, Weizmann Institute of Science, Rehovot, Israel, 2009.

[Vad12] Salil P. Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il