Pseudorandom Generators for Width-3 Branching Programs

Raghu Meka*
UCLA
Los Angeles, CA, USA
raghum@cs.ucla.edu

Omer Reingold†
Stanford University
Stanford, CA, USA
reingold@stanford.edu

Avishay Tal‡
Stanford University
Stanford, CA, USA
avishay.tal@gmail.com

ABSTRACT

We construct pseudorandom generators of seed length $\widetilde{O}(\log(n) \cdot \log(1/\epsilon))$ that ϵ-fool ordered read-once branching programs (ROBPs) of width 3 and length n. For unordered ROBPs, we construct pseudorandom generators with seed length $\widetilde{O}(\log(n) \cdot \mathrm{poly}(1/\epsilon))$. This is the first improvement for pseudorandom generators fooling width 3 ROBPs since the work of Nisan [Combinatorica, 1992].

Our constructions are based on the "iterated milder restrictions" approach of Gopalan et al. [FOCS, 2012] (which further extends the Ajtai-Wigderson framework [FOCS, 1985]), combined with the INW-generator [STOC, 1994] at the last step (as analyzed by Braverman et al. [SICOMP, 2014]). For the unordered case, we combine iterated milder restrictions with the generator of Chattopadhyay et al. [CCC, 2018].

Two conceptual ideas that play an important role in our analysis are: (1) A relabeling technique allowing us to analyze a relabeled version of the given branching program, which turns out to be much easier. (2) Treating the number of colliding layers in a branching program as a progress measure and showing that it reduces significantly under pseudorandom restrictions.

In addition, we achieve nearly optimal seed-length $\widetilde{O}(\log(n/\epsilon))$ for the classes of: (1) read-once polynomials on n variables, (2) locally-monotone ROBPs of length n and width 3 (generalizing read-once CNFs and DNFs), and (3) constant-width ROBPs of length n having a layer of width 2 in every consecutive $\mathrm{poly}\log(n)$ layers.

CCS CONCEPTS

• **Theory of computation → Pseudorandomness and derandomization**.

KEYWORDS

epsilon-biased generator, pseudorandom generators for small-space computation, pseudorandom generators for space-bounded computation, random restrictions, read once branching programs

*Supported by NSF grant CCF-1553605.
†Supported in part by NSF grant CCF-1763311.
‡Supported by a Motwani Postdoctoral Fellowship and by NSF grant CCF-1763311.

STOC '19, June 23–26, 2019, Phoenix, AZ, USA

ACM ISBN 978-1-4503-6705-9/19/06…$15.00
https://doi.org/10.1145/3313276.3316319

ACM Reference Format:
Raghu Meka, Omer Reingold, and Avishay Tal. 2019. Pseudorandom Generators for Width-3 Branching Programs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on the Theory of Computing (STOC '19), June 23–26, 2019, Phoenix, AZ, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3313276.3316319

1 INTRODUCTION

A central challenge in complexity theory is to understand the trade-off between *space* and *randomness* as resources and in particular, whether BPL = L. One of the main techniques we have for approaching this question is to design pseudorandom generators that fool tests computable in small space. The latter question can be elegantly captured in the language of designing pseudorandom generators for read-once branching programs; we define these objects next.

Definition 1.1. *For $w, n \in \mathbb{N}$, a* read-once branching program (ROBP) *of* width w *and* length n *is a layered directed graph B with $n+1$ layers where all but the first layer have at most w nodes, the first layer has a single vertex designated the* start *vertex, and the vertices in the last layer are either labeled* accept *or* reject. *Each vertex in the first n layers has exactly two outgoing edges to vertices in the next layer with one labeled* 1 *and the other labeled* -1.

Given a ROBP as above, it defines a function $B : \{\pm 1\}^n \to \{\pm 1\}$ naturally where on input $x \in \{\pm 1\}^n$ starting from the start vertex, you follow the edges labeled by x_i for $1 \le i \le n$ and output -1 if the last vertex reached is accepting and 1 *otherwise.*

Derandomizing space-bounded computations is fundamentally related to designing pseudorandom generators (and *hitting set generators*) for ROBPs as above.

Definition 1.2. *Given a class of functions $\mathcal{F} = \{f : \{\pm 1\}^n \to \mathbb{R}\}$, a function $G : \{\pm 1\}^r \to \{\pm 1\}^n$ is a* pseudorandom generator (PRG) *with* error ε *(or* ε-fools*) $\mathcal{F}$ if for every $f \in \mathcal{F}$,*

$$\left| \mathop{\mathbf{Pr}}_{x \in_u \{\pm 1\}^n} [f(x)] - \mathop{\mathbf{Pr}}_{y \in_u \{\pm 1\}^r} [f(G(y))] \right| \le \varepsilon.$$

We say the generator is log-space explicit *if G can be computed in space logarithmic in the output length n and refer to r as the* seed-length *of the generator.*

It is well-known by now that if there exists a log-space explicit PRG (or even a hitting set generator) with constant error that fools ROBPs of width n and length n with seed-length $O(\log n)$, then BPL = L. In this vein, a seminal result of Nisan [Nis92] gave a log-space explicit PRG that ε-fools ROBPs of width w and length n with seed-length $r = O((\log n) \cdot \log(wn/\varepsilon))$. Despite significant attention, improving Nisan's PRG has been a fundamental bottleneck in pseudorandomness. For width $w = 2$, it is known that small-bias

spaces fool width two ROBPs ([SZ95, BDVY13]), leading to a PRG with seed-length $O(\log(n/\varepsilon))$. However, even for the case of ε a constant and width $w = 3$, the best provable PRG had seed-length $O(\log^2 n)$–no better than what Nisan's PRG gives for polynomial width ROBPs. Nearly optimal hitting-sets generators for width-3 ROBPs were given in [SZ11, GMR+12] while [BRRY14, KNP11, De11] obtained PRGs with nearly optimal seed-length for special-classes of constant-width ROBPs. In this work, we obtain the first improvement over Nisan's PRG for width-3 ROBPs:

Theorem 1 (Main Theorem)**.** *For any $\varepsilon > 0$, there exists a log-space explicit PRG that ε-fools width-3 ROBPs with seed-length*[1] $\widetilde{O}(\log(n/\varepsilon)) + O(\log(1/\varepsilon) \cdot \log(n))$.

We in fact also obtain PRG's with nearly optimal dependence for constant error for the bigger class of unordered width-3 ROBPs, which are functions computable by ROBPs under some unknown permutation (see Section 3.3 for the formal definition). In this regime, we improve the results of [SVW17] that gave a PRG with seed-length $\widetilde{O}(\log^3 n)$.

Theorem 2. *For any $\varepsilon > 0$, there exists a log-space explicit PRG that ε-fools unordered width-3 ROBPs with seed-length $\widetilde{O}(\log(n/\varepsilon)) + O(\mathrm{poly}(1/\varepsilon) \cdot \log(n))$.*

A special class of unordered width-4 ROBPs that have received recent attention are read-once polynomials (see [Tre10, LV17]) for which we give a PRG with nearly optimal seed-length both in terms of the error and input length (up to poly(log log) factors):

Theorem 3. *There exists a log-space explicit ε-PRG for the class of read-once polynomials on n variables with seed-length $\widetilde{O}(\log(n/\varepsilon))$.*

In comparison, the best previous PRG for read-once polynomials had seed-length $\widetilde{O}(\log(n/\varepsilon)) \cdot \log(1/\varepsilon)$ [LV17], thus in particular needed $\widetilde{O}(\log^2 n)$ seed-length to fool read-once polynomials with polynomially small error.

Our results rely on several new conceptual ideas as well as technical ingredients, including PRGs fooling other interesting intermediate classes of ROBPs, that we believe could be useful for other applications especially in the context of obtaining PRGs for constant-width ROBPs. Our results rely on the framework of *iterative mild random restrictions* introduced in [GMR+12] and further developed in [RSV13, SVW17, GY14, GKM15, HLV17, LV17], the latter two also present an elegant alternate view of the technique as *bounded-independence plus noise*. We describe this framework, our proof techniques next.

1.1 The Ajtai-Wigderson Framework

The Ajtai-Wigderson [AW85] framework, that was revived and refined for ROBPs in the work of Gopalan, Meka, Reingold, Trevisan, Vadhan [GMR+12], provides a "recipe" for constructing PRGs for classes of functions that simplify under (pseudo)random restrictions. Roughly speaking, in order to fool a class of functions C it suffices to fool C under pseudo-random restrictions keeping each variable alive with probability p. Equivalently, it suffices to pseudorandomly assign p-fraction of the coordinates while approximately preserving the acceptance probability (on average) of every function $f \in C$.

[1] Henceforth, $\widetilde{O}(t)$ is used to denote $O(t \cdot \mathrm{poly}\log(t))$.

Suppose we have such a pseudorandom partial assignment, and assume that the class of functions C is closed under restrictions. Then, iteratively applying a pseudorandom partial assignment on the remaining coordinates until we assigned all of them gives us a pseudorandom generator for C. We expect to assign all the coordinates after $O(p^{-1} \cdot \log n)$ iterations, thus if each iteration requires at most s random bits, we get a PRG with seed-length $O(s \cdot p^{-1} \cdot \log n)$. Naively, it seems impossible to achieve nearly-logarithmic seed length using this approach, however this was obtained in the work of [GMR+12] as explained next.

Achieving Near-Logarithmic Seed-Length. In the work of [GMR+12] the Ajtai-Wigderson approach was used to construct ε-PRGs for read-once CNFs (and read-once DNFs) with seed length $\widetilde{O}(\log(n/\varepsilon))$. In order to achieve nearly-logarithmic seed-length [GMR+12] showed that one can assign a constant fraction of the coordinates while preserving the acceptance probability up to error $\mathrm{poly}(\varepsilon/n)$ using only $s = \widetilde{O}(\log(n/\varepsilon))$ bits of randomness. Plugging into the estimates above would give naively seed-length $\widetilde{O}(\log(n/\varepsilon) \cdot \log(n))$. In order to avoid the additional factor of $\log(n)$, they prove that after pseudorandomly assigning all but $1/\mathrm{poly}\log(n)$ of the coordinates, the function simplifies significantly so that it can be fooled using additional $O(\log(n/\varepsilon))$-random bits.

We describe the approach more precisely. A p-pseudorandom restriction against a class of functions C specifies a set $T \subseteq [n]$ of roughly $p \cdot n$ of the coordinates, and an assignment $x \in \{\pm 1\}^T$ to these coordinates, such that for any $f \in C$:

$$\mathop{\mathbf{E}}_{T,x} \mathop{\mathbf{E}}_{y \in \{\pm 1\}^{[n]\setminus T}} [f(x \circ y)] = \mathop{\mathbf{E}}_{z \in \{\pm 1\}^n} [f(z)] \pm \varepsilon$$

where $(x \circ y)$ denotes the string whose T-coordinates are taken from x and other coordinates are taken from y. The main observation of [GMR+12] is that given T, it suffices that x would fool the Bias-function, defined as

$$\mathrm{Bias}_T f(x) \triangleq \mathop{\mathbf{E}}_{y \in \{\pm 1\}^{[n]\setminus T}} [f(x \circ y)].$$

This is due to the fact that

$$\left| \mathop{\mathbf{E}}_{z \in \{\pm 1\}^n} [f(z)] - \mathop{\mathbf{E}}_{x} \mathop{\mathbf{E}}_{y \in \{\pm 1\}^{[n]\setminus T}} [f(x \circ y)] \right|$$
$$= \left| \mathop{\mathbf{E}}_{z \in \{\pm 1\}^T} [\mathrm{Bias}_T f(z)] - \mathop{\mathbf{E}}_{x} [\mathrm{Bias}_T f(x)] \right|.$$

The observation that it suffices to fool the bias-function instead of just fooling the restricted functions, enabled [GMR+12] to use "mild" restrictions with $p = \Omega(1)$ for the class of CNFs/DNFs. They show that in this case, the average of the restricted functions (i.e., the bias-function) is much easier to fool than a typical restricted function.

2 PROOF OVERVIEW

Similarly to [GMR+12], in order to achieve a PRG with nearly-logarithmic seed-length fooling width-3 ROBPs, we show that:

(1) We can pseudorandomly assign half the input coordinates while preserving the acceptance probability (on average) of every width-3 ROBP up to error ε, using seed-length $\widetilde{O}(\log(n/\varepsilon))$.

(2) After pseudorandomly assigning all but $1/\mathrm{poly}\log(n)$ of the coordinates any width-3 ROBP simplifies enough so that it can be fooled using additional $\widetilde{O}(\log(n)\log(1/\varepsilon))$ random bits.

Both steps are involved and explained in greater detail in the next two sections.

2.1 Pseudorandomly Assigning Half of the Coordinates

In Section 4 we prove the following theorem showing that we can pseudorandomly assign $1/\mathrm{poly}\log\log(n/\varepsilon)$ of the coordinates while changing the acceptance probability by at most ε.

Theorem 4. *Let $n \in \mathbb{N}, \varepsilon > 0$. There exists a log-space explicit pseudorandom restriction assigning $p = 1/O(\log\log(n/\varepsilon))^6$ fraction of the variables using $O(\log(n/\varepsilon)\log\log(n/\varepsilon))$ random bits, that maintains the acceptance probability of any unordered width-3 length-n ROBP up to error ε.*

Given Theorem 4, we can assign half of the coordinates by iteratively applying the pseudorandom restriction $O(1/p)$ times. This ultimately uses $O(\log(n/\varepsilon)(\log\log(n/\varepsilon))^7) = \widetilde{O}(\log(n/\varepsilon))$ random bits to assign half of the coordinates, as promised.

We describe the techniques that go into the proof of Theorem 4. The proof proceeds in two steps. The first step (described in Section 4) reduces the task of generating a pseudorandom restriction for width-3 ROBPs to the task of generating a pseudorandom restriction for the XOR of short (logarithmic-length) width-3 ROBPs. The second step (described in Section 5 of the full version [MRT18]) is a pseudorandom restriction for the latter class of Boolean functions.

2.1.1 Reducing width-3 ROBPs to the XOR of short width-3 ROBPs. Next, we explain how we reduce fooling width-3 ROBPs to fooling the XOR of short width-3 ROBPs. Let B be a ROBP of length-n and width-3. We pick a set $T_0 \subseteq [n]$ of size $\approx n/2$ using an almost $O(\log(n/\varepsilon))$-wise independent distribution. We wish to show that for most choices of T_0, we can pseudorandomly assign pn of the coordinates in T_0, while fooling the Bias-function $\mathrm{Bias}_{T_0}B$. Our *main observation* is that for most choices for T_0, the bias-function $\mathrm{Bias}_{T_0}B$ is the average of simpler width-3 ROBPs.

Recall that every layer of edges in a ROBP contains two sets of edges, one corresponding to the transition made when the input bit equals 1 and similarly one corresponding to the input bit equaling -1. Observe that if the two sets of edges are the same, then the layer is redundant and the value of the input bit does not affect whether the ROBP accept or not. We thus assume without loss of generality that there are no redundant layers. We say that a layer of edges is a colliding layer if there are two edges marked by the same label (i.e. both labeled 1 or both labeled -1) that enter the same vertex in the next layer.

First, suppose (ideally) that all layers in a width-3 ROBP are colliding. Then, under the pseudorandom restriction, with high probability, in every $O(\log(n/\varepsilon))$ consecutive layers we will have a layer of edges whose corresponding variable is fixed to a value for which the edges in the layer collide, leaving at most 2 vertices reachable in the next layer of vertices. Using a result of Bogdanov, Dvir, Verbin, Yehudayoff [BDVY13] such restricted ROBPs can be written as linear combinations of functions of the following form: XOR of width-3 ROBPs of length $O(\log(n/\varepsilon))$ defined over disjoint sets of variables. It thus suffices to fool this XOR of short width-3 ROBPs in order to fool the restricted ROBP, as we do in Section 5 of the full version [MRT18].

The assumption that all layers in a width-3 ROBP are colliding is not necessarily true. In fact, it can be the case that in every layer of edges both the 1-edges and the (-1)-edges form a permutation on the state space with no collisions. Indeed, such ROBPs are known in the literature as permutation-ROBPs. (For example, the $\mathrm{MOD}_3(x_1,\ldots,x_n)$ function indicating whether $(\sum_i x_i \equiv 0 \mod 3)$ can be computed by width-3 permutation ROBP.) Nonetheless, as mentioned earlier, it suffices to fool the bias-function and this task is easier than fooling each restricted function.

Relabeling Under The Bias Function: In the following, we consider relabeling of a ROBP. Recall that in a ROBP every vertex has a pair of outgoing edges: one labeled 1 and the other labeled -1. A relabeling of a ROBP B is any ROBP B' that can be achieved from B by swapping the labels for some of these pairs of edges.

Our *key observation* is that the bias function $\mathrm{Bias}_T B$ of a program B does not depend on the labels of the edges associated with the variables outside T. This is due to the fact that the value of $\mathrm{Bias}_T B(x)$ on a given partial input $x \in \{\pm 1\}^T$ is the probability of acceptance of B on a random assignment to the variables in $[n] \setminus T$, and this value remains the same under any relabeling of the edges associated with the variables in $[n] \setminus T$. Moreover, a simple fact shows that any non-redundant layer of edges can be relabeled so that it is colliding. Thus, for any ROBP B and any fixed T, we can relabel the edges associated with variables with $[n] \setminus T$ so that they are colliding, yielding another width-3 ROBP, denoted B^T. We get that $\mathrm{Bias}_T B = \mathrm{Bias}_T B^T$, and B^T is a ROBP in which all layers in $[n] \setminus T$ are colliding. We can thus apply the previous argument and conclude that $\mathrm{Bias}_T B^T$ is the average of width-3 ROBPs whose vast majority have a layer of vertices of width-2 in every $O(\log(n/\varepsilon))$ consecutive layers. These ROBPs are then fooled by the pseudorandom partial assignment described in the next section.

To sum up, since the bias-function is the average over all restricted functions of B, it also equals the average over all restricted functions of B^T, and these restricted functions are simple enough for us to fool.

Relabeling was previously used in [BV10, Ste13, CGR14] to show that the best ROBPs distinguishing between certain distributions and the uniform distribution must be "locally-monotone" (see Section 3.3 for the formal definition). In general, it is unclear how to argue locally monotone programs are the hardest ROBPs to fool. Nevertheless, in [CHRT18], relabeling helped bounding the sum of absolute values of Fourier coefficients of small width ROBPs. In comparison, we use a relabeling technique to note that in the iterated random restrictions framework (when trying to fool the bias-function), one might as well treat the restricted layers as if they were locally monotone.

2.1.2 Pseudorandom restrictions for the XOR of short width-3 ROBPs. Our main result in Section 5 of the full version [MRT18] is the following:

Theorem 5. *Let $n, w, b \in \mathbb{N}$, $\varepsilon > 0$. There exists a log-space explicit pseudorandom restriction assigning $p = 1/O(\log(b \cdot \log(n/\varepsilon)))^{2w}$ fraction of n variables using $O(w \cdot \log(n/\varepsilon) \cdot (\log\log(n/\varepsilon) + \log(b)))$ random bits, that maintains the acceptance probability of any XOR of ROBPs of width-w and length-b (defined on disjoint sets of variables) up to error ε.*

Recall that in the previous section, we reduced the case of width-3 ROBPS to this case with $w = 3$ and $b = O(\log(n/\varepsilon))$. Our proof for Theorem 5 follows previous strategies by [GMR+12, GY14, GKM15, LV17]. Indeed, the functions we are trying to fool are a special case of product-functions that were recently studied in [HLV17, LV17]. Product-functions are functions of the form $f(x) = f_1(x) \cdot f_2(x) \cdots f_m(x)$ where each f_i depends on a set B_i of at most b variables, and $\{B_1, \ldots, B_m\}$ are pairwise-disjoint.

PRGs for product-functions were constructed in previous work, however none achieve the parameters we need. Haramaty, Lee and Viola [HLV17] and Lee and Viola [LV17] constructed PRGs with seed length $\widetilde{O}(b + \sqrt{mb\log(1/\varepsilon)})$ and $\widetilde{O}((b + \log(m/\varepsilon)) \cdot \log(1/\varepsilon))$ respectively for such functions. While the latter is nearly optimal for constant ε, we require ε to be smaller than $1/m$, since the reduction in the previous section from [BDVY13] incurs a multiplicative factor of m on the error. Gopalan, Meka and Kane [GKM15] achieve nearly optimal seed-length $\widetilde{O}(\log(n/\varepsilon))$ but only for the case where the blocks $B_1, \ldots, B_m$ are known.

The main reason we are able to achieve better seed-length is due to the fact that we further assume that the functions $f_1, \ldots, f_m$ are computed by constant-width ROBPs. We rely on the previous work of Chattopadhyay, Hatami, Reingold, Tal [CHRT18] who constructed PRGs for constant-width length-n ROBPs with seed-length $\operatorname{poly}\log(n)$. We observe that under an unusual setting of parameters, namely when applying this result to constant-width ROBPs of length $\operatorname{poly}\log(n)$, one gets seed-length $\widetilde{O}(\log(n/\varepsilon))$. This enables us to fool the XOR of any subset of $\operatorname{poly}\log(n)$ of the functions $f_1, \ldots, f_m$ using nearly-logarithmic seed-length. Relying on the proof strategy laid by Gopalan and Yehudayoff [GY14], we bootstrap this into a pseudorandom restriction fooling the XOR of $f_1, \ldots, f_m$.

Due to lack of space we have not included the proof of Theorem 5 in this extended abstract. Please see Section 5 in the full version [MRT18].

2.2 Simplification under Pseudorandom Restrictions

Recall that our proof strategy is similar to that of [GMR+12]:

(1) For $i = 0, \ldots, O(\log\log n)$: assign half of the remaining coordinates pseudorandomly using $\widetilde{O}(\log(n/\varepsilon))$ random bits, while changing the acceptance probability by at most ε.
(2) Pseudorandomly assign the remaining coordinates using $\widetilde{O}(\log(n/\varepsilon))$ random bits.

The first step was overviewed in the previous section. In order to carry on the second step, we wish to find some progress measure, that would decrease in each iteration of the first step. For the case of CNFs the CNF-width (i.e., the maximal number of literals in a clause) was a good progress measure for [GMR+12]. They showed that without loss of generality the CNF-width is $O(\log(n/\varepsilon))$ initially, and that it decreases by a constant-factor in each iteration of Step 1.

Our analogous progress measure is the number of colliding layers. We recall that in a ROBP, some layers of edges form permutations on the state space, while others are colliding.

We show that after the first application of step 1, with high probability the restricted ROBP can be written as a composition of m subprograms $D_1, \ldots, D_m$ where each D_i has at most 2 vertices in the first and last layers and at most $\ell_0 = O(\log(n/\varepsilon))$ colliding layers. Intuitively, this happens since every colliding layer reduces the width to 2 with constant probability and thus with high probability in any $O(\log(n/\varepsilon))$ consecutive colliding layers at least one would be set to the value that reduces the width to 2. This motivates the following definition.

Definition 2.1. *We call a ROBP B a (w, ℓ, m)-ROBP if B can be written as $D_1 \circ \ldots \circ D_m$, with each D_i being a width w ROBP with the first and last layers having at most two vertices and each D_i having at most ℓ colliding layers.*

We wish to show that the parameter ℓ (that bounds the maximal number of colliding layer in a subprogram D_i with width-2 in the first and last layers) reduces by a constant factor under any iteration of step 1. That is, to show that after iteration i of step 1 we get with high probability a $(3, \ell_i, m_i)$-ROBP where $\ell_i = \ell_0/c^i$ for some constant $c < 1$. As long as $m_i \le \exp(O(\ell_i))$, an inductive argument works since the colliding layers in each individual D_j reduces by a factor c with probability $1 - \exp(-\Omega(\ell_i))$ and we can afford a union bound over all m_i subprograms. However, we cannot afford such a union bound if $m_i \gg \exp(\ell_i)$. To handle this, we prove the following structural result: any $(3, \ell_i, m_i)$-ROBP can be well-approximated by $(3, \ell_i, C^{\ell_i})$-ROBPs for some constant C. Furthermore, we show that the error indicator of the approximator can be written as the AND of C^{ℓ_i} many $(3, \ell_i, 1)$-ROBPs, and that its expectation under the uniform distribution is doubly-exponentially small in ℓ_i. This allows us to show that the error indicator is small under the pseudo-random assignments as well, and we can safely replace a $(3, \ell_i, m_i)$-ROBP with its $(3, \ell_i, C^{\ell_i})$-ROBP approximator.

Applying the restriction and the structure result $O(\log\log n)$ times, we end up with a $(3, \ell', C^{\ell'})$ ROBP where $\ell' = O(\log(1/\varepsilon))$. As a last step, we show that $(3, \ell', C^{\ell'})$-ROBPs are fooled by the INW generator [INW94] with seed-length $\widetilde{O}(\log(n)\log(1/\varepsilon))$. This follows from the results of [BRRY14]. For the unordered case, we use the generator from the recent work of [CHHL18] for the last step, with seed-length $\widetilde{O}(\log(n) \cdot \operatorname{poly}(1/\varepsilon))$ (using a structural result by [SVW17]).

2.3 The Proof of Theorem 3

Theorem 3 is a special case of the following theorem

Theorem 6. *Let $n, w, b \in \mathbb{N}$, $\varepsilon > 0$. There exists a log-space explicit pseudorandom generator that ε-fools any XOR of ROBPs of width-w and length-b (defined on disjoint sets of variables), using seed-length $O(\log(b) + \log\log(n/\varepsilon))^{2w+2} \cdot \log(n/\varepsilon)$.*

We consider b as the progress measure, and wish to show that this parameter reduces under pseudorandom restrictions. This is analogous to the the number of colliding layers ℓ in the previous

section. However, here, in some cases, we cannot guarantee that the application of the pseudorandom restriction from Theorem 5 would decrease b. The problematic cases are when we have the XOR of more than $\exp(b)$ functions on b variables each. We show that in such cases, an "aggressive" pseudorandom restriction, assigning $1-\exp(-b)$ fraction of the variables, simplifies the function significantly, while maintaining its acceptance probability. Combining applications of mild-restrictions and aggressive-restrictions in a "decision tree of random restrictions" results in an assignment that fools the function. However, this does not give a PRG as the decisions made along the tree depend adaptively on the function we try to fool, and PRGs cannot depend on the function they try to fool. We fix this by taking the XOR of several pseudorandom assignments, one per each path in this decision tree in order to construct a PRG that fools this class of functions.

Due to lack of space we have not included this result in this extended abstract. Please see Section 6 in the full version [MRT18] for the proof.

2.4 Comparison with Forbes-Kelley

Independently and concurrently, Forbes and Kelley [FK18] constructed PRGs that ε-fool unordered width-w length-n ROBPs with seed length $O(\log(nw/\varepsilon)\log^2 n)$. For bounded width w, which is the main focus of our work, their seed-length improves to $\widetilde{O}(w\log(n/\varepsilon)\log n)$. Their result is a significant improvement to the prior state of the art PRGs for unordered ROBPs by [IMZ12] and [CHRT18]. In particular, they nearly match Nisan's parameters for constant width [Nis92].

We point out that our results are incomparable to those of Forbes-Kelley [FK18]. Our main contribution is that we surpass Nisan's parameters for width-3 and get nearly optimal seed length $\widetilde{O}(\log n)$ for constant error (in fact, our results hold even for the case of unordered ROBPs). In addition, we get nearly optimal dependency on all parameters for read-once polynomials.

The work of Forbes and Kelley also rely on the two-step approach outlined above: (1) pseudorandomly assign half the input coordinates while preserving the acceptance probability, and (2) repeat recursively until all input coordinates are assigned. For constant width ROBPs they are able to perform step (1) with nearly logarithmic seed-length $\widetilde{O}(\log(n/\varepsilon))$. This step can be seen as a stronger alternative to our Theorem 4, since they can handle any constant-width. However, [FK18] have no analogs to our simplification under pseudorandom restriction results (Sections 5), which necessitates using step (1) $O(\log n)$ times and prevents them from breaking the $O(\log^2 n)$ barrier.

3 PRELIMINARIES

Denote by U_n the uniform distribution over $\{\pm1\}^n$, and by U_S for $S\subseteq[n]$ the uniform distribution over $\{\pm1\}^S$. Denote by log the logarithm in base 2. For any function $f:\{\pm1\}^n\to\mathbb{R}$, we shorthand by $\mathbf{E}[f]=\mathbf{E}_{x\sim U_n}[f(x)]$ and by $\mathbf{Var}[f]=\mathbf{E}[f^2]-\mathbf{E}[f]^2$. For an event E we denote by 1_E its indicator function.

3.1 Restrictions

For a set $T\subseteq[n]$ and two strings $x\in\{\pm1\}^T$, $y\in\{\pm1\}^{[n]\setminus T}$ we denote by $\mathrm{Sel}_T(x,y)$ the string with

$$\mathrm{Sel}_T(x,y)_i=\begin{cases}x_i, & i\in T\\ y_i, & \text{otherwise.}\end{cases}$$

Definition 3.1 (Restriction). *Let $f:\{\pm1\}^n\to\mathbb{R}$ be a function. A* restriction *is a pair (T,y) where $T\subseteq[n]$ and $y\in\{\pm1\}^{[n]\setminus T}$. We denote by $f_{T|y}:\{\pm1\}^n\to\mathbb{R}$ the function f restricted according to (T,y), defined by $f_{T|y}(x)=f(\mathrm{Sel}_T(x,y))$.*

Definition 3.2 (Random Valued Restriction). *Let $n\in\mathbb{N}$. A random variable (T,y), distributed over restrictions of $\{\pm1\}^n$ is called* random-valued *if conditioned on T, the variable y is uniformly distributed over $\{\pm1\}^{[n]\setminus T}$.*

Definition 3.3 (p-Random Restriction). *A p-*random restriction *is a random-valued restriction over pairs (T,y) sampled in the following way: For every $i\in[n]$, independently, pick i to T with probability p; Sample y uniformly from $\{\pm1\}^{[n]\setminus T}$. We denote this distribution of restrictions by $\mathcal{R}_p$.*

Definition 3.4 (The Bias-Function). *Let $f:\{\pm1\}^n\to\mathbb{R}$. Let $T\subseteq[n]$. We denote by $\mathrm{Bias}_T(f):\{\pm1\}^n\to\mathbb{R}$ the function defined by $(\mathrm{Bias}_T(f))(x)=\mathbf{E}_{y\sim U_{[n]\setminus T}}[f_{T|y}(x)]$. When T is clear from the context, we shorthand $\mathrm{Bias}_T(f)$ as $\widetilde{f}$.*

3.2 Small-biased Distributions

We say that a distribution $\mathcal{D}$ over $\{\pm1\}^n$ is δ-biased[2] if for any non-empty $S\subseteq[n]$ it holds that $|\mathbf{E}_{x\sim\mathcal{D}}[\prod_{i\in S}x_i]|\le\delta$. [NN93, AGHP92, ABN+92, BT13, Ta-17] show that δ-biased distributions can be explicitly sampled using $O(\log(n/\delta))$ random bits.

Let $p\in(0,1]$. We say that a distribution $\mathcal{D}_p$ over subsets of $[n]$ is δ-biased with marginals p if for any non-empty $S\subseteq[n]$ it holds that $\mathbf{Pr}_{T\sim\mathcal{D}_p}[S\subseteq T]=p^{|S|}\pm\delta$.

Claim 3.5. *Let $p=2^{-a}$ for some integer $a>0$, let $\mathcal{D}$ be an ε-biased distribution over $\{\pm1\}^{na}$. Define $\mathcal{D}_p$ to be a distribution over subsets of $[n]$ as follows: Sample $x\sim\mathcal{D}$. Output $T=\{i\in[n]:\bigwedge_{j\in[a]}(x_{(i-1)a+j}=1)\}$. Then $\mathcal{D}_p$ is ε-biased with marginals p.*

3.3 Branching Programs

A read-once branching program (ROBP) B of length n and width w is a directed layered graph with $n+1$ layers of vertices denoted $V_1,\ldots,V_{n+1}$. Each V_i consists of $w_i\le w$ vertices $\{v_{i,1},\ldots,v_{i,w_i}\}$, and between every two consecutive layers V_i and V_{i+1} there exists a set of directed edges (from V_i to V_{i+1}), denoted E_i, such that any vertex in V_i has precisely two out-going edges in E_i, one marked by 1 and one marked by -1. The vertices in V_{n+1} are marked with either 'accept' and 'reject'.

A branching program B and an input $x\in\{\pm1\}^n$ naturally describes a computation path in the layered graph: we start at node $v_1=v_{1,1}$ in V_1. For $i=1,\ldots,n$, we traverse the edge going out from v_i marked by x_i to get to a node $v_{i+1}\in V_{i+1}$. The resulting computation path is $v_1\to v_2\to\ldots\to v_{n+1}$. We say that B accepts

[2] Note that the terms bias-function and small-biased distributions are unrelated.

x iff the computation path defined by B and x reaches an accepting node. Naturally B describes a Boolean function $B : \{\pm 1\}^n \to \{\pm 1\}$ whose value is -1 on input x iff B accepts x.

Unordered branching programs are defined similarly, expect that there exists a permutation $\pi \in S_n$ such that in step i the computation path follows the edge marked by x_{π_i}, for $i \in [n]$. We also consider unordered branching programs on $[n]$ of shorter length $n' \leq n$. In such case, the program stops after reading n' input bits.[3]

For two programs B_1 and B_2 defined over disjoint sets of variables and having the end width of B_1 equal the start width of B_2, we denote by $B_1 \circ B_2$ the concatenation of B_1 and B_2, defined in the natural way.

Locally Monotone Branching Programs. Let B be a width-w length-n ROBP. For any vertex v in the ROBP, denote by β_v the probability to accept a uniformly random input starting from the vertex v. Since renaming the vertices in each layer does not affect the functionality of B, we may assume without loss of generality that the vertices in V_i are ordered according to β_v. That is, for every $i \in [n+1]$ we have $\beta_{v_{i,1}} \leq \beta_{v_{i,2}} \leq \ldots \leq \beta_{v_{i,w_i}}$. In case of equalities, we break ties arbitrarily but commit to a strict ordering of the nodes in each layer. B is called locally monotone if for any vertex v in B the vertex reached from v using the 1-edge has larger or equal index than the vertex reached from v using the (-1)-edge.

For $i \in [n]$, denote by $E_{i,1}$ the set of edges in E_i marked by 1 and similarly define $E_{i,-1}$. We say that E_i is a identity layer if $E_{i,1} = E_{i,-1}$ (in which case x_i does not affect the output of of B). We say that E_i is a permutation layer if both $E_{i,1}$ and $E_{i,-1}$ form a matching between V_i and V_{i+1} (i.e., $|V_i| = |V_{i+1}|$ and for $b \in \{-1, 1\}$ no two edges in $E_{i,b}$ enter the same vertex in V_{i+1}). The following is a key lemma from the work of [BV10].

Lemma 3.6 (Collision Lemma [BV10])**.** *In a locally monotone branching program, every permutation layer is an identity layer.*

To see it, note that if we think of the vertices in each layer $\{v_{i,1}, \ldots, v_{i,w_i}\}$ as written from top to bottom according to β_v, then in a locally monotone program for any vertex v the 1-edge leads to the same vertex or to a vertex below the one that follows the (-1)-edge. Thus, assuming both $E_{i,-1}$ and $E_{i,1}$ form a matching, the only way this could happen is if they both form the same matching.

The following is an immediate corollary of results from [CHRT18] and [SVW17] (see the full version [MRT18] for more details).

4 FROM WIDTH-3 ROBPS TO THE XOR OF SHORT ROBPS

In the full version [MRT18, Section 5], we prove the following theorem.

Theorem 5. *Let $n, w, b \in \mathbb{N}$, $\varepsilon > 0$. There exists a log-space explicit pseudorandom restriction assigning $p = 1/O(\log(b \cdot \log(n/\varepsilon)))^{2w}$ fraction of n variables using $O(w \cdot \log(n/\varepsilon) \cdot (\log\log(n/\varepsilon) + \log(b)))$ random bits, that maintains the acceptance probability of any XOR of ROBPs of width-w and length-b (defined on disjoint sets of variables) up to error ε.*

[3] Note that in the unordered case, the set of bits being read could be an arbitrary subset of $[n]$ of size n'.

The pseudorandom restriction assigns p fraction of the variables as follows:

(1) Choose a set of coordinates $T \subseteq [n]$ according to a δ_T-biased distribution with marginals p, for $\delta_T := p^{O(\log(n/\varepsilon))}$.
(2) Assign the variables in T according to a δ_x-biased distribution, for $\delta_x := (\varepsilon/n)^{O(\log b)}$.

Known constructions of small-biased distributions [NN93, AGHP92, ABN+92, BT13, Ta-17] show that it suffices to use $O(\log(n/\delta_T) + \log(n/\delta_x)) \leq O(w \cdot \log(n/\varepsilon) \cdot (\log\log(n/\varepsilon) + \log(b)))$ random bits to sample the restriction.

In this section, we show how to design pseudorandom restrictions for unordered width-3 ROBPs from pseudorandom restrictions to the XOR of many width-3 ROBPs of length $O(\log(n/\varepsilon))$. We get the following theorem.

Theorem 4. *Let $n \in \mathbb{N}, \varepsilon > 0$. There exists a log-space explicit pseudorandom restriction assigning $p = 1/O(\log\log(n/\varepsilon))^6$ fraction of the variables using $O(\log(n/\varepsilon)\log\log(n/\varepsilon))$ random bits, that maintains the acceptance probability of any unordered width-3 length-n ROBP up to error ε.*

Proof Sketch. In this section, we shall show that under pseudorandom restrictions leaving each variable alive with probability 1/2, with high probability, the bias function of a ROBP B can be written as a linear combination (up to a small error) over functions of the form $f_1 \cdot f_2 \cdot \ldots \cdot f_m$ where each f_i is a short subprogram of the original program of length $O(\log(n/\varepsilon))$, and each f_i is defined on a disjoint set of coordinates. Each function g in the linear combination will have a weight $\alpha_g \in [-1, 1]$, and the sum of absolute values of weights over all functions participating in the linear combination will be at most n. This will show that any generator that ε/n-fools the XOR of short width-3 ROBPs also ε-fools width-3 length-n ROBPs under random restrictions.

The reduction will first establish that with high probability (over the choice of the set of coordinates that are left alive) the bias function of a ROBP B can be written as the average of width-3 length-n ROBPs, whose vast majority have at most $O(\log(n/\varepsilon))$ layers between every two layers with width-2. Then, we use a result of Bogdanov, Dvir, Viola, Yehudayoff [BDVY13] that reduces branching programs with many width-2 layers to the XOR of short ROBPs.

We focus on the first part of the reduction. First, consider the case when B is locally-monotone. In this case, every layer of edges is either the identity layer or a colliding layer (Lemma 3.6). Assume without loss of generality that there are no identity layers. Then, under a pseudorandom restriction, with high probability, in every $O(\log(n/\varepsilon))$ consecutive layers we will have a layer of edges whose corresponding variable is fixed to the value on which the edges in the layer collide, leaving at most 2 vertices reachable in the next layer of vertices. Removing unreachable vertices, we get that with high probability under the random restriction, in every $O(\log(n/\varepsilon))$ consecutive layers there is a layer of vertices with width-2.

However, in the case that B is not locally-monotone (e.g., when B is a permutation ROBP) it could the case that the widths of all layers of vertices remain 3 under the random restriction. Our main observation is that since the bias function takes the average over all assignments to the restricted variables, the bias function of B does not depend on the labels of edges marked by the restricted variables.

More formally, for any $T \subseteq [n]$, if B and C are two ROBPs with the same graph structure that only differ on the labels on the edges in layers $[n] \setminus T$, then $\mathrm{Bias}_T(B) = \mathrm{Bias}_T(C)$. Thus, once T is fixed we may relabel the layers in $[n] \setminus T$ so that they are locally-monotone, yielding a new ROBP B', and then apply the bias function. Using the analysis of the locally monotone case, we get that the bias function of B' (and thus the bias function of B) is the average of B' over all restrictions fixing the coordinates in $[n] \setminus T$, and we know that most of these restricted ROBPs have width-2 in every $O(\log(n/\varepsilon))$ consecutive layers.

Essentially, the bias function allows us to imagine as if we are taking the average over restrictions of B' rather than restrictions of B, and restrictions of B' are "simpler" to fool than restrictions of B since they have many layers with width-2.

We defer the formal argument to the full version [MRT18].

5 PSEUDORANDOM GENERATORS FOR WIDTH-3 ROBPS

In this section, we construct pseudorandom generators fooling width-3 ROBPs (3ROBPs, in short) with seed-length $\widetilde{O}(\log n)$. For ordered width-3 ROBPs we can guarantee error $1/\mathrm{poly}\log(n)$ using seed-length $\widetilde{O}(\log n)$:

Theorem 1 (Main Theorem). *For any $\varepsilon > 0$, there exists a log-space explicit PRG that ε-fools width-3 ROBPs with seed-length[4] $\widetilde{O}(\log(n/\varepsilon)) + O(\log(1/\varepsilon) \cdot \log(n))$.*

Note that in comparison, even for constant $\varepsilon > 0$, the best previous generators had seed-length $O(\log^2 n)$ for ordered 3ROBPs. We also get similar improvements for unordered 3ROBPs but with worse dependence on the error ε.

Theorem 2. *For any $\varepsilon > 0$, there exists a log-space explicit PRG that ε-fools unordered width-3 ROBPs with seed-length $\widetilde{O}(\log(n/\varepsilon)) + O(\mathrm{poly}(1/\varepsilon) \cdot \log(n))$.*

5.1 Proof Overview

We heavily rely on the pseudorandom restriction from Theorem 4 that assigns $p = 1/\mathrm{poly}\log\log(n)$ of the variables while changing the acceptance probability by at most $1/\mathrm{poly}(n)$. As a first step we assign a constant fraction of the coordinates.

Assigning most of the coordinates. The first step is rather simple: we apply iteratively $O(1/p)$ times the pseudorandom restriction from Theorem 4 to get the following result.

Claim 5.1. *Let $\delta > 0$. For all constants $\alpha \in (0,1)$, there is a pseudorandom restriction $\rho = (T, y)$ using $\widetilde{O}(\log(n/\delta))$ random bits, changing the acceptance probability of 3ROBPs by at most δ. Furthermore, T is $(\delta/n)^{\omega(1)}$ biased with marginals α and y is $(\delta/n)^{\omega(1)}$ biased.*

The proof is the same as that of Claim 4.1 in the full version, and is omitted.

Let $\varepsilon > \delta > 0$. Let B be a 3ROBP of length-n. First, we claim that after applying the pseudorandom restriction ρ in Claim 5.1, with high probability (at least $1 - \mathrm{poly}(\varepsilon/n)$), $B|_\rho$ has a simpler structure in that between any two width-2 layers the subprogram has at most $O(\log(n/\varepsilon))$ colliding layers. Concretely, we use the following definitions.

Definition 5.2. *Given a ROBP B, we call a layer of edges* colliding *if either the edges marked by -1 and the edges marked by 1 collide.*

Definition 5.3. *We call a ROBP B a (w, ℓ, m)-ROBP if B can be written as $D_1 \circ \ldots \circ D_m$, with each D_i being a width w ROBP with the first and last layers having at most two vertices and each D_i having at most ℓ colliding layers.*

We show that after applying the pseudorandom restriction ρ in Claim 5.1, with high probability the restricting ROBP $B|_\rho$ is a $(3, O(\log(n/\varepsilon)), m)$-ROBPs. Now, we wish to iteratively apply Claim 5.1, making the ROBP simpler in each step. We will have one progress measures on the restricted ROBP: the maximal number of colliding layers in a subprogram (denoted ℓ). We show that the number of colliding layers reduces by a constant-factor in each iteration. To do so, we prove a structural result on $(3, \ell, m)$-ROBPs, showing that such ROBPs can be well-approximated by $(3, \ell, C^\ell)$-ROBPs for some constant C. This allows us to not worry about the number of sub-programs and use the number of colliding layers as a progress measure. Applying the restriction and the structure result $O(\log\log n)$ times, we end up with a ROBP where $\ell = O(\log(1/\varepsilon))$. We also show that ROBPs with few colliding layers are fooled by the INW generator. This follows from the results of [BRRY14].

5.2 Reducing the Length of $(3, \ell, m)$-ROBPs

Here, we show that $(3, \ell, m)$-ROBPs can be approximated by $(3, \ell, C^\ell)$-ROBPs for some constant C. A crucial point in the analysis is that we need the approximation to hold not just under the uniform distribution but also under the pseudo-random distribution. Fortunately, we are able to do so by arguing that the *error* function detecting when our approximation is wrong is itself computable by a conjunction of negations of width 3-ROBPs with few colliding layers.

Lemma 5.4 (Main Structural Result). *For any $C \geq 1$ the following holds. Any $(3, \ell, m)$-ROBP B can be written as $B' + E$ where B' is a $(3, \ell, C^\ell)$-ROBP and either $E \equiv 0$ or for any x, $|E(x)| \leq F(x) = \wedge_{i=1}^{C^\ell}(\neg F_i(x))$ where F_i are non-zero events that can be computed by $(3, \ell, 1)$-ROBPs on disjoint variables.*

We shall also show (in the next claim) that any non-zero event F_i that can be computed by $(3, \ell, 1)$-ROBP, happens with probability at least $4^{-(\ell+1)}$ under the uniform distribution. Thus, $\mathbf{Pr}_{x \sim U_n}[\wedge_{i=1}^{C^\ell}(\neg F_i(x))] \leq (1 - 4^{-(\ell+1)})^{C^\ell} \leq \exp(4^{-(\ell+1)} \cdot C^\ell)$ which is doubly-exponentially small in ℓ provided that C is a large enough constant.

For any vertex v in a ROBP, we denote by p_v the probability to reach v under a uniform random assignment to the inputs.

Claim 5.5. *In a ROBP with width w and at most ℓ colliding layers, every vertex whose $p_v > 0$ has $p_v \geq 2^{-(\ell+1)\cdot(w-1)}$.*

We remark that this bound is sharp.

Proof. We prove by induction (on the length of the program) that any program with width at most w, exactly ℓ colliding

[4] Henceforth, $\widetilde{O}(t)$ is used to denote $O(t \cdot \mathrm{poly}\log(t))$.

layers and exactly t reachable states in the last layer, has $p_v \geq 2^{-\ell\cdot(w-1)-(t-1)}$ for any reachable vertex v. Without loss of generality all nodes in the program are reachable (otherwise, we remove vertices that aren't reachable).

Consider a program B of length n with parameters (t, ℓ, w). Removing the last layer gives a program B' of length $n-1$ with parameters (t', ℓ', w). By the induction hypothesis for any v' in the last layer of B' we have $p_{v'} \geq \delta$ for $\delta := 2^{-\ell'\cdot(w-1)-(t'-1)}$.

We perform a case analysis. The following simple bound will be used in all cases. Let v be a vertex in the last layer of B. Assume that e edges enter v from vertices in the second to last layer. Then, $p_v \geq \frac{1}{2}\cdot\delta\cdot e$. In particular, since we assumed all vertices are reachable, any vertex in the last layer have $p_v \geq \delta/2$.

If $\ell' = \ell$ and $t' = t$, then the last layer of edges in B is regular, i.e., any node in the last layer in B has exactly two ingoing edges. In this case any vertex v in the last layer has $p_v \geq \frac{1}{2}\cdot\delta\cdot 2 = \delta = 2^{-\ell\cdot(w-1)-(t-1)}$.

If $\ell' = \ell$, then $t' \leq t$, since there are no collisions in the last layer of edges. Since we already handled the case $t' = t$, we may assume $t' \leq t-1$. For any vertex v in the last layer we have $p_v \geq \delta/2 \geq \frac{1}{2}\cdot 2^{-\ell'(w-1)-(t'-1)} \geq \frac{1}{2}\cdot 2^{-\ell(w-1)-(t-2)} = 2^{-\ell(w-1)-(t-1)}$.

If $\ell' < \ell$, then we consider two sub-cases: if $t = 1$ then only one vertex is reachable in the last layer and its p_v equals 1. Otherwise, $t \geq 2$ and $t' \leq w$ thus $t' \leq t + (w-2)$ and for any vertex v in the last layer we have $p_v \geq \delta/2 \geq \frac{1}{2}\cdot 2^{-\ell'(w-1)-(t'-1)} \geq \frac{1}{2}\cdot 2^{-(\ell-1)(w-1)-(t+(w-2)-1)} = 2^{-\ell(w-1)-(t-1)}$. □

We say that two vertices v and v' in a ROBP are locally-equivalent if the 1-edges exiting v and v' reach the same vertex and the (-1)-edges exiting v and v' reach the same vertex. We say that a ROBP has no-redundant vertices if any vertex in the program is reachable, and there are no locally-equivalent vertices. In the following, without loss of generality we can assume that ROBPs have no-redundant vertices, because we can eliminate unreachable vertices and merge locally-equivalent vertices.

Claim 5.6 (Colliding Layers $\implies$ Colliding)**.** *Let B be a 3ROBP with width-2 at the start and finish, at least one colliding layer and no-redundant vertices. Let $v_{1,1}$ and $v_{1,2}$ be the two start nodes. Then, there exists a string on which the two paths from $v_{1,1}$ and $v_{1,2}$ collide.*

Proof. First consider the case that B has width 2. Then, there exists a layer i and a value $b \in \{\pm 1\}$ such that the two edges marked by b in the i-th layer collide. Any string whose i-th bit equals b results in colliding paths.

For the rest of the proof assume that B has a layer with width 3. Let $V_1, \ldots, V_{n+1}$ be the layers of vertices in B. Let i denote the index of the last layer in B with width 3. Since B has width-2 at the end, $i < n+1$.

There are six edges between V_i and V_{i+1}: three edges marked with $x_i = -1$ and three edges marked with $x_i = 1$. Since $|V_{i+1}| = 2$, by the Pigeon-hole principle, there are two edges marked with $x_i = -1$ going to some vertex $v \in V_{i+1}$, and two edges marked with $x_i = 1$ going to some vertex $v' \in V_{i+1}$ (v' is not necessarily different from v). These two pairs of edges cannot be starting from the same two nodes in V_i since then the two nodes will be locally-equivalent. By renaming the nodes in V_i, we can assume that the two edges from $v_{i,1}, v_{i,2} \in V_i$ marked with -1 go to $v \in V_{i+1}$ and the two edges from $v_{i,2}, v_{i,3} \in V_i$ marked with 1 go to $v' \in V_{i+1}$.

Since $v_{i,2}$ is reachable, there is an input $(x_1, \ldots, x_{i-1})$ that leads from $v_{1,1}$ or $v_{1,2}$ to $v_{i,2}$. Without loss of generality, we assume that $v_{i,2}$ is reachable from $v_{1,1}$. Let $\widetilde{v} \in V_i$ be the vertex reached by following the same input $(x_1, \ldots, x_{i-1})$ starting from the other start vertex $v_{1,2}$. If $\widetilde{v} = v_{i,2}$, then we already found a collision. If $\widetilde{v} = v_{i,1}$ then for the choice $x_i = -1$ the two paths defined by $(x_1, \ldots, x_i)$ starting from $v_{1,1}$ and $v_{1,2}$ collide on $v \in V_{i+1}$. Similarly, if $\widetilde{v} = v_{i,3}$, then for the choice $x_i = 1$ the two paths collide on $v' \in V_{i+1}$. □

Claim 5.7 ("First Collisions" can be detected by 3ROBPs)**.** *Let B be a 3ROBP with 2 vertices at the first layer, denoted $v_{1,1}, v_{1,2}$. Suppose there are at most ℓ colliding layers in B and that there exists a string on which the two paths from $v_{1,1}$ and $v_{1,2}$ collide. Let u be the first vertex on which a collision can occur, and let E be the event that a collision happened on u. Then, E can be computed by another width-3 ROBP with at most ℓ-colliding layers.*

Proof. To simulate whether the paths starting from $v_{1,1}$ and $v_{1,2}$ collide at u, we consider the 3ROBP that keeps the **unordered** pair corresponding to the states of the two paths during the computation. In each layer until u, we have only states corresponding to $\{0,1\}, \{0,2\}$ or $\{1,2\}$. When we reach the layer of u we have two states: "accept" (corresponding to a collision on u) and "reject" (corresponding to anything else). Observe that any non-colliding layer in the original program defines a non-colliding layer in the new branching program (as a permutation over a finite set also defines a permutation over unordered pairs from this set). Thus, there are at most ℓ colliding layers in the 3ROBP computing E. □

We are now ready to prove the main structural lemma – Lemma 5.4. In the following, we consider branching programs with two initial nodes $v_{1,1}, v_{1,2}$. We interpret the value of the program on input x as its average value on the two paths starting from $v_{1,1}$ and $v_{1,2}$. That is, the program can get value $1, 0$ or -1 depending on whether the two paths from $v_{1,1}$ and $v_{1,2}$ accept or not.

Throughout this section we think of the error terms as $\{0,1\}$-indicators (instead of the usual $\{\pm 1\}$-notation for other Boolean functions). We shall use $A \wedge B$ and $\overline{A}$ to denote the standard AND and negation of these Boolean values.

Lemma 5.8. *Let $B = D_1 \circ \ldots \circ D_m$ be a ROBP where each D_i is a width-3 ROBP with at most 2 vertices on the first and last layers. Then, for any $j \in \{2, \ldots m\}$ we can write $B(x)$ as the sum of $(D_j \circ \ldots \circ D_m)(x)$ and an error term $E(x)$, that is bounded in absolute value by $\overline{\mathsf{FCol}_j(x)} \wedge \ldots \wedge \overline{\mathsf{FCol}_m(x)}$ where $\mathsf{FCol}_i(x)$ denotes the event that the two paths in D_i collide on input x at the first vertex on which it is possible to collide in D_i.*

Proof. Assume without loss of generality that no layer of vertices has width-1 except for maybe the first. For $j = 2, \ldots, m$, let $v_{j,1}$ and $v_{j,2}$ be the two nodes at the first layer of the subprogram D_j. If D_1 has two nodes at the first layer, then denote them by $v_{1,1}$ and $v_{1,2}$, otherwise denote the single node by $v_{1,1}$. Let x be an input to the branching program B. If the two paths defined by x from $\{v_{j,1}, v_{j,2}\}$ collide at some point, then the value of $B(x)$ equals the value of $(D_j \circ \ldots \circ D_m)(x)$. If the two paths do not collide, then $(D_j \circ \ldots \circ D_m)(x) = 0$, since it is the average of two paths

with different outcomes, thus $E(x) = B(x) - (D_j \circ \ldots \circ D_m)(x)$ is at most 1 in absolute value. Furthermore, in such a case, for all $i \in \{j, \ldots, m\}$ it holds that both paths in the subprogram D_i starting from $v_{i,1}$ and $v_{i,2}$ on input x do not collide, i.e., $\mathsf{FCol}_i(x) = 0$. Overall, we got that $B(x) = E(x) + (D_j \circ \ldots \circ D_m)(x)$, and $E(x) \neq 0$, it holds that $\overline{\mathsf{FCol}_j(x)} \wedge \ldots \wedge \overline{\mathsf{FCol}_m(x)} = 1$ (i.e., $|E(x)| \leq \overline{\mathsf{FCol}_j(x)} \wedge \ldots \wedge \overline{\mathsf{FCol}_m(x)}$). □

Proof of Lemma 5.4. Let B be a $(3, \ell, m)$-ROBP $B = D_1 \circ \ldots \circ D_m$. If B has no colliding layers, then there is nothing to prove since B itself is a $(3, \ell, 1)$-ROBP. If B has colliding layers, then without loss of generality each D_i has at least one colliding layer (since otherwise we can merge subprograms with no colliding layers with their successors or predecessors). If $m \leq C^\ell$, there is nothing to prove and we can take $B' = B$ and $E = 0$. Suppose that $m > C^\ell$. Let $j = m - C^\ell + 1 > 1$. Let $B' = D_j \circ \cdots \circ D_m$ and let $F(x) = \overline{\mathsf{FCol}_j(x)} \wedge \ldots \wedge \overline{\mathsf{FCol}_m(x)}$ where $\mathsf{FCol}_i(x)$ denotes the event that the two paths in D_i collide on input x at the first vertex on which it is possible to collide in D_i. Then, by the previous claim, we can write $B = B' + E$ where for any input x, $|E(x)| \leq F(x)$. We argue that this gives the desired decomposition. Indeed, By Claim 5.7, for $i \in \{j, \ldots, m\}$ the event $\mathsf{FCol}_i(x)$ can be computed by a $(3, \ell, 1)$-ROBP. Further, by Claim 5.6 each D_i has a possible collision, and thus each FCol_i is a non-zero event. □

5.3 PRGs for ROBPs with Few Colliding Layers

In this section we show that we can ε-fool ordered ROBPs with at most ℓ-colliding layers with $\widetilde{O}(\log(\ell/\varepsilon) \cdot \log(n))$ seed-length.

Theorem 5.9. *For any $\varepsilon > 0$, there is a log-space explicit PRG that ε-fools ordered width w-ROBPs with length n and at most ℓ colliding layers using seed length*

$$O((\log\log n + \log(1/\varepsilon) + \log(\ell) + w) \cdot \log n.$$

The above relies on the PRGs for regular branching programs and generalizations of them due to Braverman, Rao, Raz, and Yehudayoff [BRRY14]. In the following, we say that a read-once branching program B is δ-reachable if for all reachable vertices v in B we have $p_v(B) \geq \delta$, where

$$p_v(B) := \Pr_{x \sim U_n}[\text{reaching } v \text{ on the walk on } B \text{ defined by } x].$$

We start by quoting a result by Braverman, Rao, Raz, Yehudayoff [BRRY14].

Theorem 5.10 ([BRRY14])**.** *There is a log-space explicit PRG that ε-fools all δ-reachable ROBPs of length-n and width-w using seed length*

$$O(\log\log n + \log(1/\varepsilon) + \log(1/\delta) + \log(w)) \cdot \log n.$$

Next, we reduce the task of fooling ROBPs with at most ℓ-colliding layers to the task of fooling δ-reachable ROBPs. The reduction is similar to that in [CHRT18]. The main difference is that we simulate a ROBP with width w by a δ-reachable ROBP of width $w + 1$ by adding a new sink state that should be thought of as "immediate stop". This change seems essential in our case, and the reduction from [CHRT18] does not seem to satisfy the necessary properties here.

Lemma 5.11. *Let $\delta \leq 2^{-(w-1)}$. Let $\mathcal{D}$ be a distribution on $\{\pm1\}^n$ that ε-fools all δ-reachable ROBPs of length n and width $w + 1$. Then, $\mathcal{D}$ also fools width-w ROBPs with at most ℓ colliding layers with error at most $(\ell w + 1) \cdot \varepsilon + (2^w w\ell) \cdot \delta$.*

Proof. Let $\mathcal{D}$ be a distribution on $\{\pm1\}^n$ that ε-fools all δ-reachable ROBPs of length-n and width-w. The first observation is that $\mathcal{D}$ also fools prefixes of these programs. This reason is simple: to simulate the prefix of length-k of a δ-reachable ROBP B, one can just reroute the last $n - k$ layers of edges in B so that they would "do nothing", i.e. that they would be the identity transformation regardless of the values of $x_{k+1}, \ldots, x_n$.

Let B be a length n width-w ROBP with at most ℓ colliding layers. Next, we introduce B', a δ-reachable ROBP of length-n and width-$(w + 1)$, that would help bound the difference between

$$B(U_n) := \Pr_{x \sim U_n}[B(x) = 1] \qquad \text{and} \qquad B(\mathcal{D}) := \Pr_{x \sim \mathcal{D}}[B(x) = 1],$$

where U_n is the uniform distribution over $\{\pm1\}^n$. Let B' be the the following modified version of B. To construct B' we consider a sequence of $\ell + 1$ branching programs $B_0, \ldots, B_\ell$ where $B_0 = B$ and $B' = B_\ell$. Let $i_1, \ldots, i_\ell$ be the colliding layers in B. For $j = 1, \ldots, \ell$ we take B_j to be B_{j-1} except we may reroute some of the edges in the i_j-th layer. We explain the rerouting procedure. For $j = 1, \ldots, \ell$ we calculate the probability to reach vertices in layer V_{i_j} of B_{j-1}. If some vertex v in the i_j-th layer has probability smaller than $2^{w-1} \cdot \delta$, then we reroute the two edges going from the vertex v to go to "immediate stop". We denote by V_{small} the set of vertices for which we rerouted the outgoing edges from them.

First, we claim that any reachable vertex v in B_ℓ has $p_v \geq \delta$. Let $i_{\ell+1} = n + 1$ for convenience. We apply induction and show that for $j = 0, 1, \ldots, \ell$ any vertex reachable by B_j in layers $1, \ldots, i_{j+1}$ has $p_v \geq \delta$. The base case holds because up to layer i_1 the branching program has no colliding layers and we may apply Claim 5.5 to get that $p_v \geq 2^{-(w-1)} \geq \delta$. To apply induction assume the claim holds for B_{j-1} and show that it holds for B_j. The claim obviously holds for all vertices in layers $1, \ldots, i_j$ in B_j since we didn't change any edge in those layers going from B_{j-1} to B_j. Let v be a reachable vertex in layer i where $i_j < i \leq i_{j+1}$ in B_j. It means that there is a vertex v' with $p_{v'}(B_j) \geq 2^{w-1} \cdot \delta$ in the i_j-th layer of B_j (and also in B_{j-1}) and a path going from v' to v. Looking at the subprogram from v' to v we note that this is a subprogram with no colliding edges (only the first layer has the potential to be colliding, but in a ROBP the first layer can never be colliding as there is only one edge marked by (-1) and only one edge marked by -1). By Claim 5.5 the probability to get from v' to v is at least $2^{-(w-1)}$. Thus, the probability to reach v is at least $p_{v'}(B_j) \cdot \mathbf{Pr}[\text{reach } v | \text{reached } v'] \geq 2^{w-1} \cdot \delta \cdot 2^{-(w-1)} = \delta$.

Next, we bound $|B(U_n) - B(\mathcal{D})|$ by using the triangle inequality

$$|B(U_n) - B(\mathcal{D})| \leq |B(U_n) - B'(U_n)| + |B'(U_n) - B'(\mathcal{D})| + |B'(\mathcal{D}) - B(\mathcal{D})| \quad (1)$$

and bounding each of the three terms separately.

(1) The first term is bounded by the probability of reaching one of the nodes in V_{small} in B' when taking a uniform random walk. This follows since if the path defined by x didn't pass through V_{small} then we would end up with the same node in both B and B' (since no rerouting affected the path). Each

vertex v in V_{small} has $p_v(B') < 2^{w-1}\cdot\delta$. By union bound, the probability to pass through V_{small} is at most $|V_{\text{small}}|\cdot 2^{w-1}\cdot\delta$.

(2) The second term is at most ε since the program B' is δ-reachable.

(3) Similarly to the first term, the third term is bounded by the probability of reaching one of the nodes in V_{small} in B' when taking a walk sampled by $\mathcal{D}$.

$$
\begin{aligned}
&|B'(\mathcal{D}) - B(\mathcal{D})| \\
&\le \Pr_{x\sim\mathcal{D}}[\text{reaching } V_{\text{small}} \text{ on the walk on } B' \text{ defined by } x] \\
&\le \sum_{v\in V_{\text{small}}} \Pr_{x\sim\mathcal{D}}[\text{reaching } v \text{ on the walk on } B' \text{ defined by } x]
\end{aligned}
$$

However since $\mathcal{D}$ is pseudorandom for prefixes of B', for each $v \in V_{\text{small}}$ the probability of reaching v when walking according to $\mathcal{D}$ is ε-close to the probability of reaching v when walking according to U_n.

$$
\begin{aligned}
&|B'(\mathcal{D}) - B(\mathcal{D})| \\
&\le \sum_{v\in V_{\text{small}}} (\varepsilon + \Pr_{x\sim U_n}[\text{reaching } v \text{ on the walk on } B' \text{ defined by } x]) \\
&= \sum_{v\in V_{\text{small}}} \left(\varepsilon + p_v(B')\right) \le |V_{\text{small}}|\cdot(\varepsilon + 2^{w-1}\delta)
\end{aligned}
$$

Summing the upper bound on the three terms in Eq. (1) gives:

$$|B(U_n) - B(\mathcal{D})| \le |V_{\text{small}}|\cdot(\varepsilon + 2^w\delta) + \varepsilon \le \ell w\cdot(\varepsilon + 2^w\delta) + \varepsilon. \quad \square$$

Proof of Theorem 5.9. Take $\varepsilon' = \varepsilon/(2(\ell w+1))$ and $\delta = \varepsilon'/2^w$. Take the generator from Theorem 5.10 with parameters δ and ε'. Applying Lemma 5.11, the error of this generator on the class of ROBPs with width w length n and at most ℓ colliding layers is at most $(\ell w+1)\cdot\varepsilon' + (2^w\cdot w\cdot\ell)\cdot\delta \le \varepsilon/2 + \varepsilon/2 = \varepsilon$. By Theorem 5.10, its seed length is

$$O(\log\log n + \log(1/\varepsilon') + \log(1/\delta) + \log(w))\cdot\log(n)$$

which is at most $O(\log\log n + \log(1/\varepsilon) + \log(\ell) + w)\cdot\log(n)$. $\square$

5.4 Proof of Theorem 1

We are now ready to prove our main result on fooling 3ROBPs. Our generator is obtained by applying Claim 5.1 iteratively $O(\log\log n)$ times and then using a PRG fooling 3ROBPs with at most $O(\text{poly}(1/\varepsilon))$ colliding layers as in Theorem 5.9. The intuition is as follows.

Let B be a 3ROBP and let ρ_0 be a pseudorandom restriction as in Claim 5.1. We first show that with probability at least $1-\varepsilon/n$ over ρ_0, $B^0 = B|_{\rho_0}$ is a $(3,\ell_0,m)$-ROBP for $\ell_0 = O(\log(n/\varepsilon))$. Let $B^0 = D_1^0\circ\cdots\circ D_m^0$ where each D_i^0 has at most ℓ_0 colliding layers and begins and ends with width two layers. Let ρ_1 be an independent pseudo-random restriction as in Claim 5.1. Then $B^1 \equiv B^0|_{\rho_1} = D_1^0|_{\rho_1}\circ\cdots\circ D_m^0|_{\rho_1}$ and it is easy to check that with probability at least $1-2^{-\Omega(\ell_0)}$, each $D_i^0|_{\rho_1}$ has at most $\ell_0/2$ colliding layers. Ideally, we would like to apply a union bound over the different D_i^0 and conclude that B^1 is a $(3,\ell_0/2,m)$-ROBP. In the first step, this approach works since $m \le C^{\ell_0}$ for a large enough constant C (by the definition on ℓ_0), and we can afford a union bound. We get that with probability at least $1-2^{-\Omega(\ell_0)}$, B^1 is a $(3,\ell_0/2,m_1)$-ROBP (for some $m_1 \le m$). Continuing this process by induction, at step i we have that B^i is a $(3,\ell_0/2^i,m_i)$-ROBP. To carry the union bound in the i-th step we need $m_i \le C^{\ell_0/2^i}$, however m_i could be much larger than that. Nevertheless, we know that we can always approximate B^i with a $(3,\ell_0/2^i,C^{\ell_0/2^i})$-ROBP by Lemma 5.4. This approximation allows us to apply the union bound and conclude that the number of colliding layers in each block decreases by a factor of 2. We iterate this approach until the maximal number of colliding layers in a subprogram is at most $O(\log(1/\varepsilon))$, and then use the PRG from Theorem 5.9.

To carry the induction forward as outlined above, we need the following lemma that shows that the error terms simplify as well under the pseudorandom restrictions.

Lemma 5.12. *For any constant $C \ge 20$, there exists $\alpha \in (0,1)$ such that the following holds. Let $\ell, n \in \mathbb{N}$ be sufficiently large and $m = C^\ell \le n$. Let $F = \overline{\text{FCol}_1} \wedge \ldots \wedge \overline{\text{FCol}_m}$ where $\text{FCol}_i(x)$ are non-zero events on disjoint variables computed by $(3,\ell,1)$-ROBPs. Let ρ be a pseudorandom restriction as in Claim 5.1 with parameter α and error parameter $\delta \le 1/n^5$. Then, with probability at least $1-2C^{-\ell/2}$, we have $F|_\rho \le \overline{\text{FCol}'_i} \wedge \ldots \wedge \overline{\text{FCol}'_{\sqrt{m}}}$ where $\text{FCol}'_i(x)$ are non-zero events on disjoint variables computed by $(3,\ell/2,1)$-ROBPs.*

Due to lack of space, we defer the proof of Lemma 5.12 to the full version.

We are now ready to prove the main theorem, Theorem 1.

Proof of Theorem 1. Let $C \ge 20$. Let $\alpha \in (0,1)$ be a constant to be chosen later. Let $\ell_0 = O(\log(n/\varepsilon))$. Let k be a parameter to be chosen later and let $\ell_i = \ell_0/2^i$ for $1 \le i \le k$.

Our generator is as follows. First choose $\rho_0,\rho_1,\ldots,\rho_k$ independent pseudo-random restrictions as in Claim 5.1 with parameter α and $\delta = \varepsilon/n^{10}$. After iteratively applying the restrictions $\rho_0,\rho_1,\ldots,\rho_k$, we set the remaining bits using the generator from Theorem 5.9 for a parameter $\ell = \ell_k\cdot C^{\ell_k}$ and error parameter ε' to be chosen later. Let Y be the output distribution of the generator.

Let $B^0 = B|_{\rho_0}$. We first claim that B^0 is a $(3,\ell_0,m)$-ROBP with high probability. In the following let X be uniformly random over $\{\pm1\}^n$.

Claim 5.13. *With probability at least $1-\varepsilon/n$, $B|_{\rho_0}$ is a $(3,\ell_0,m)$-ROBP and $\mathbf{E}_{\rho_0,X}[B|_{\rho_0}(X)] = \mathbf{E}_X[B(X)] \pm \delta$.*

For $0 \le i \le k$, let $\rho^i \triangleq \rho_0\circ\cdots\rho_i$. We will show the following claim by induction on i.

Claim 5.14. *For $0 \le i \le k$, with probability at least $1-\frac{\varepsilon}{n} - 4i\cdot C^{-\ell_i}$, $B|_{\rho^i}$ can be written as $B^i + E^0 + E^1 + \cdots + E^i$ where B^i is a $(3,\ell_i,C^{\ell_i})$-ROBP and the error terms E^j for $0 \le j \le i$ satisfy: Either $E_j \equiv 0$ or $|E^j(x)| \le F^j(x)$ with $F^j(x) = \wedge_{h=1}^{m_j}(\neg F_h^j(x))$ where F_h^j are non-zero events computed by $(3,\ell_i,1)$-ROBPs on disjoint sets of variables and $m_j = C^{\ell_i}$.*

Furthermore, $\mathbf{E}_{\rho^i,X}[B|_{\rho^i}(X)] = \mathbf{E}_X[B(X)] \pm (i+1)\delta$.

A crucial point in the above is that the functions $F^0,\ldots,F^i$ bounding the error terms are conjunctions of negations of $(3,\ell_i,1)$-ROBPs and there exactly C^{ℓ_i} in each of them.

Proof. For $i = 0$, the claim follows immediately by applying Lemma 5.4 to $B|_{\rho_0}$. Now, suppose the claim is true for i. Suppose,

we can write $B|_{\rho^i} = B^i + \mathcal{E}^i$, where $\mathcal{E}^i = E^0 + E^1 + \cdots + E^i$ as in the claim. By the induction hypothesis, this happens with probability at least $1 - \frac{\varepsilon}{n} - 4i \cdot C^{-\ell_i}$.

Clearly, $B|_{\rho^{i+1}} = B^i|_{\rho_{i+1}} + \mathcal{E}^i|_{\rho_{i+1}}$. Let $B^i = D_1 \circ \cdots \circ D_{m'}$ be a decomposition where each D_j has at most ℓ_i colliding layers, starts and ends with width-2 layers and $m' \le C^{\ell_i}$.

Now, observe that as each D_j has at most ℓ_i colliding layers, the probability that at least $\ell_i/2$ of these colliding layers are unfixed under ρ_{i+1} is at most $\binom{\ell_i}{\ell_i/2} \cdot (\alpha^{\ell_i/2} + (\varepsilon/n)^{\omega(1)}) \le 2^{\ell_i}\alpha^{\ell_i/2}$ by Claim 5.1. Thus, by a union bound over $1 \le j \le m'$, with probability at least $1 - 2^{\ell_i}\alpha^{\ell_i/2} \cdot C^{\ell_i} \ge 1 - C^{-\ell_i}$ (for a suitable choice of α), over ρ_{i+1}, $B^i|_{\rho_{i+1}}$ is a $(3, \ell_i/2, C^{\ell_i})$-ROBP. Now, conditioning on this event, by Lemma 5.4, we can write $B^i|_{\rho_{i+1}}$ as $B^{i+1} + E^{i+1}$, where B^{i+1} is a $(3, \ell_{i+1}, C^{\ell_{i+1}})$-ROBP and E^{i+1} satisfies the conditions of the claim. Thus, with probability at least $1 - \frac{\varepsilon}{n} - 4i \cdot C^{-\ell_i} - C^{-\ell_i}$,

$$\begin{aligned} B|_{\rho^{i+1}} &= B^i|_{\rho_{i+1}} + \mathcal{E}^i|_{\rho_{i+1}} \\ &= B^{i+1} + \mathcal{E}^i|_{\rho_{i+1}} + E^{i+1}, \end{aligned}$$

where B^{i+1}, and E^{i+1} satisfy the conditions of the claim.

We just need to argue that $\mathcal{E}^i|_{\rho_{i+1}}$ can be written in the requisite form. To this end, note that for $0 \le j \le i$, $|E^j|_{\rho_{i+1}}| \le F^j|_{\rho_{i+1}}$. By the induction hypothesis, we either have $E^j \equiv 0$ or we can write $|E^j| \le F^j = \wedge_{h=1}^{m_j}(\neg F_h^j(x))$ where F_h^i are $(3, \ell_i, 1)$-ROBPs on disjoint sets of variables and $m_j = C^{\ell_i}$. We can now apply Lemma 5.12 to conclude that with probability at least $1 - 2C^{-\ell_i/2}$, we can write $F^j|_{\rho_{i+1}} = \wedge_{h=1}^{m'_j}(\neg H_h^j(x))$ where H_h^j are non-zero events computed by $(3, \ell_i/2, 1)$-ROBPs on disjoint sets of variables and $m'_j = C^{\ell_i/2}$. This satisfies the constraints of the claim.

Adding up the failure probabilities over the choice of ρ_{i+1}, we get the desired decomposition for $i+1$ with probability at least

$$1 - \frac{\varepsilon}{n} - 4i \cdot C^{-\ell_i} - C^{-\ell_i} - (i+1) \cdot 2C^{-\ell_i/2} \ge 1 - \frac{\varepsilon}{n} - 4(i+1)C^{-\ell_{i+1}}.$$

(since $2C^{-\ell_i} \le C^{-\ell_{i+1}}$). The furthermore part follows immediately from Claim 5.1. The claim now follows by induction. □

We are now ready to prove the theorem. By the above claim, we have that with probability at least $1 - \frac{\varepsilon}{n} - 4kC^{-\ell_k}$ over the choice of $\rho_0, \rho_1, \ldots, \rho_k$, we can write

$$B|_{\rho^k} = B^k + E^0 + \cdots + E^k,$$

where B^k is a $(3, \ell_k, C^{\ell_k})$-ROBP and $E^0, \ldots, E^k$ can be bounded by functions $F^0, \ldots, F^k$ that are conjunctions of negations of C^{ℓ_k} non-zero events computed by $(3, \ell_k, 1)$-ROBPs.

Note that each such F^j can be written as a width-4 ROBP, say H^j, by adding an additional layer to compute the conjunction and that the number of collisions in the width 4 ROBP is at most $\ell_k \cdot C^{\ell_k}$. Therefore, if we let Y be the output distribution of the generator from Theorem 5.9 with $\ell = \ell_k \cdot C^{\ell_k}$ and error parameter ε', we get that for all $0 \le j \le k$, and X uniformly random over $\{\pm 1\}^n$,

$$\mathbf{E}[B^k(X)] = \mathbf{E}[B^k(Y)] \pm \varepsilon'$$

$$\mathbf{E}[|E^j(Y)|] \le \mathbf{E}[H^j(Y)] \le \mathbf{E}[H^j(X)] + \varepsilon' \le (1 - 4^{-(\ell_k+1)})^{C^{\ell_k}} + \varepsilon'$$

where we used Claim 5.5 to bound $\mathbf{E}[H^j(X)]$. Since $C \ge 20$, $\mathbf{E}[|E^j(Y)|] \le \exp(-2^{\ell_k}) + \varepsilon'$.

Combining the above inequalities we get that with probability at least $1 - \frac{\varepsilon}{n} - 4kC^{-\ell_k}$ over the choice of $\rho_0, \rho_1, \ldots, \rho_k$

$$\left| \mathop{\mathbf{E}}_{X}[B|_{\rho^k}(X)] - \mathop{\mathbf{E}}_{Y}[B|_{\rho^k}(Y)] \right| \le \varepsilon' + (k+1) \cdot (\exp(-2^{\ell_k}) + \varepsilon').$$

Finally, as we also have that

$$\left| \mathop{\mathbf{E}}_{\rho_0, \ldots, \rho_k}[B|_{\rho^k}(X)] - \mathbf{E}[B(X)] \right| \le (k+1) \cdot \delta,$$

we get

$$\begin{aligned} &\left| \mathop{\mathbf{E}}_{\rho_0, \ldots, \rho_k}[B|_{\rho^k}(Y)] - \mathbf{E}[B(X)] \right| \\ &\le ((k+1) \cdot \delta) \; + \; \left(\frac{\varepsilon}{n} + 4kC^{-\ell_k} \right) \\ &\quad + \; \left(\varepsilon' + (k+1) \cdot (\exp(-2^{\ell_k}) + \varepsilon') \right). \end{aligned}$$

To get $\ell_k = \log\log\log(n) + \log(1/\varepsilon)$ we set $k = \log(\ell_0/\ell_k) = O(\log\log n)$. Furthermore, setting $\delta = \varepsilon/n^{10}$ and $\varepsilon' = \varepsilon/4k$, the above error bound becomes

$$\left| \mathop{\mathbf{E}}_{\rho_0, \ldots, \rho_k}[B|_{\rho^k}(Y)] - \mathbf{E}[B(X)] \right| \le \varepsilon.$$

Finally, we estimate the seed-length of our generator. Choosing the random restrictions takes $\widetilde{O}(\log(n/\delta)) = \widetilde{O}(\log(n/\varepsilon))$ random bits. Sampling Y requires seed-length

$$\begin{aligned} &O(\log\log n + \log(1/\varepsilon') + \log(\ell_k \cdot C^{\ell_k}) + 4) \cdot \log n \\ &= O(\log\log n + \log(1/\varepsilon)) \cdot \log n. \end{aligned}$$

Thus, the final seed-length is $\widetilde{O}(\log(n/\varepsilon)) + O(\log(1/\varepsilon)(\log n))$. The theorem follows. □

5.5 Proof of Claim 5.13

Claim. *With probability at least $1 - \varepsilon/n$, $B|_{\rho_0}$ is a $(3, \ell_0, m)$-ROBP and $\mathbf{E}_{\rho_0, X}[B|_{\rho_0}(X)] = \mathbf{E}_X[B(X)] \pm \delta$.*

Proof. The second part follows from Claim 5.1. We are left to prove the first part.

Let $\rho_0 = (T, y)$ be the pseudorandom restriction, where $T \subseteq [n]$ and $y \in \{\pm 1\}^{[n]\setminus T}$. Assume there are L colliding layers in B and let $i_1, i_2, \ldots, i_L$ be their indices. For $j \in [L]$, call a layer i_j "good" under the choice of (T, y) if $i_j \in [n] \setminus T$ and the edges in the i_j-layer of B marked by y_{i_j} collide.

For $j \in \{1, \ldots, L - \ell_0 + 1\}$ let $\mathcal{E}_j$ be the event that none of layers $\{i_j, i_{j+1}, \ldots, i_{j+(\ell_0-1)}\}$ is good. Recall that T is sampled from a $(\varepsilon/n)^{\omega(1)}$-biased distribution with marginals α, and y is sampled from a $(\varepsilon/n)^{\omega(1)}$-biased distribution. For $\mathcal{E}_j$ to happen, we must have a partition $S_1 \cup S_2 = \{j, j+1, \ldots, j+\ell_0-1\}$ such that all layers $i_{j'}$ for $j' \in S_1$ are in T and all layers $i_{j''}$ for $j'' \in S_2$ are in $[n] \setminus T$ but the edged marked by $y_{i_{j''}}$ in the $i_{j''}$-th layer do not collide. For any fixed j and fixed partition $S_1 \cup S_2 = \{j, j+1, \ldots, j+\ell_0-1\}$, the above event happens with probability at most

$$\begin{aligned} (\alpha^{|S_1|} + (\varepsilon/n)^{\omega(1)}) \cdot (2^{-|S_2|} + (\varepsilon/n)^{\omega(1)}) &\le 2 \cdot \alpha^{|S_1|} \cdot 2^{-|S_2|} \\ &= 2 \cdot \alpha^{|S_1|} \cdot 2^{-(\ell_0 - |S_1|)} \end{aligned}$$

(using $|S_1| + |S_2| = \ell_0 = O(\log(n/\varepsilon))$). Overall,

$$\begin{aligned}\mathbf{Pr}[\mathcal{E}_j] &\le \sum_{S_1 \subseteq \{j,\ldots,j+\ell_0-1\}} (2\cdot\alpha^{|S_1|}\cdot 2^{-(\ell_0-|S_1|)}) \\ &= 2\cdot(\tfrac{1}{2}+\alpha)^{\ell_0} \le \varepsilon/n^2\end{aligned}$$

assuming $\alpha > 0$ is a sufficiently small constant and $\ell_0 = c\log(n/\varepsilon)$ for a sufficiently large constant $c > 0$. By the union bound,

$$\mathbf{Pr}[\mathcal{E}_1 \vee \mathcal{E}_2 \vee \ldots \vee \mathcal{E}_{L-\ell_0+1}] \le (L-\ell_0+1)\cdot\varepsilon/n^2 \le \varepsilon/n.$$

Under the event that all $\mathcal{E}_j$ are false, we get that $B|_\rho$ can be written as $D_1 \circ \ldots \circ D_m$ where each D_i is a width-3 ROBP with at most ℓ_0 colliding layers and at most 2 vertices on the first and last layer. □

5.6 Pseudorandom Generator for Unordered 3ROBPs

In this section, using the recent generator of Chattopadhyay, Hatami, Hosseini, Lovett [CHHL18], and a Fourier bound by Steinke, Vadhan and Wan [SVW17], we show that we can also handle unordered 3ROBPs, thus proving Theorem 2.

Lemma 5.15 (Lemma 3.14 [SVW17]). *Let $\ell \in \mathbb{N}$ and let B be a width-w ROBP with at most ℓ colliding layers. Then, for all $k = 1,\ldots,n$ it holds that $L_{1,k}(f) \le O(w^3\cdot\ell)^k$.*

Theorem 5.16 (Theorem 4.5 [CHHL18]). *Let $\mathcal{F}$ be a family of n-variate Boolean functions closed under restrictions. Assume that for all $f \in \mathcal{F}$ for all $k = 1,\ldots,n$, $L_{1,k}(f) \le a\cdot b^k$. Then, for any $\varepsilon > 0$, there exists a log-space explicit PRG which fools $\mathcal{F}$ with error ε, whose seed length is $O(\log(n/\varepsilon)\cdot(\log\log(n)+\log(a/\varepsilon))\cdot b^2)$.*

Corollary 5.17. *There is a log-space explicit PRG that ε-fools unordered ROBPs with width w, length n and at most ℓ colliding layers using seed length*

$$O(\log(n/\varepsilon)\cdot(\log\log(n)+\log(1/\varepsilon))\cdot w^6\ell^2)$$

Proof of Theorem 2. The proof is essentially the same as that of Theorem 1, where instead of using the generator from Theorem 5.9 to set the bits after the pseudorandom restrictions, we use the generator from the above corollary. The final seed-length has a worse dependence on ε as we need to set $\ell = C^{\log(1/\varepsilon)+\log\log\log(n)} = \mathrm{poly}(1/\varepsilon)\cdot\mathrm{poly}\log\log(n)$ in Cor. 5.17. □

ACKNOWLEDGEMENTS

We would like to thank Oded Goldreich and Salil Vadhan for very helpful comments on an earlier version of this manuscript.

REFERENCES

[ABN+92] N. Alon, J. Bruck, J. Naor, M. Naor, and R. M. Roth. Construction of asymptotically good low-rate error-correcting codes through pseudo-random graphs. *IEEE Trans. Information Theory*, 38(2):509–516, 1992.

[AGHP92] N. Alon, O. Goldreich, J. Håstad, and R. Peralta. Simple construction of almost k-wise independent random variables. *Random Structures and Algorithms*, 3(3):289–304, 1992.

[AW85] M. Ajtai and A. Wigderson. Deterministic simulation of probabilistic constant depth circuits. In *FOCS*, pages 11–19, 1985.

[BDVY13] A. Bogdanov, Z. Dvir, E. Verbin, and A. Yehudayoff. Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9:283–293, 2013.

[BRRY14] M. Braverman, A. Rao, R. Raz, and A. Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014.

[BT13] A. Ben-Aroya and A. Ta-Shma. Constructing small-bias sets from algebraic-geometric codes. *Theory of Computing*, 9:253–272, 2013.

[BV10] J. Brody and E. Verbin. The coin problem and pseudorandomness for branching programs. In *Proceedings of the 51st annual FOCS*, pages 30–39, 2010.

[CGR14] G. Cohen, A. Ganor, and R. Raz. Two sides of the coin problem. In *APPROX-RANDOM*, pages 618–629, 2014.

[CHHL18] E. Chattopadhyay, P. Hatami, K. Hosseini, and S. Lovett. Pseudorandom generators from polarizing random walks. In *CCC*, volume 102 of *LIPIcs*, pages 1:1–1:21. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018.

[CHRT18] E. Chattopadhyay, P. Hatami, O. Reingold, and A. Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *STOC*, pages 363–375. ACM, 2018.

[De11] A. De. Pseudorandomness for permutation and regular branching programs. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011*, pages 221–231, 2011.

[FK18] M. A. Forbes and Z. Kelley. Pseudorandom generators for read-once branching programs, in any order. In *FOCS*, 2018.

[GKM15] P. Gopalan, D. M. Kane, and R. Meka. Pseudorandomness via the discrete fourier transform. In *FOCS*, pages 903–922, 2015.

[GMR+12] P. Gopalan, R. Meka, O. Reingold, L. Trevisan, and S. P. Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *FOCS*, pages 120–129, 2012.

[GY14] P. Gopalan and A. Yehudayoff. Inequalities and tail bounds for elementary symmetric polynomial. *CoRR*, abs/1402.3543, 2014.

[HLV17] E. Haramaty, C. H. Lee, and E. Viola. Bounded independence plus noise fools products. In *32nd Computational Complexity Conference, CCC 2017*, pages 14:1–14:30, 2017.

[IMZ12] R. Impagliazzo, R. Meka, and D. Zuckerman. Pseudorandomness from shrinkage. In *FOCS*, pages 111–119, 2012.

[INW94] R. Impagliazzo, N. Nisan, and A. Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the 26th annual STOC*, pages 356–364, 1994.

[KNP11] M. Koucký, P. Nimbhorkar, and P. Pudlák. Pseudorandom generators for group products: extended abstract. In *STOC*, pages 263–272, 2011.

[LV17] C. H. Lee and E. Viola. More on bounded independence plus noise: Pseudorandom generators for read-once polynomials. *ECCC*, 24:167, 2017.

[MRT18] R. Meka, O. Reingold, and A. Tal. Pseudorandom generators for width-3 branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 25:112, 2018.

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

[NN93] J. Naor and M. Naor. Small-bias probability spaces: Efficient constructions and applications. *SIAM J. on Computing*, 22(4):838–856, 1993.

[RSV13] O. Reingold, T. Steinke, and S. Vadhan. Pseudorandomness for regular branching programs via Fourier analysis. In *APPROX-RANDOM*, pages 655–670. 2013.

[Ste13] J. P. Steinberger. The distinguishability of product distributions by read-once branching programs. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013*, pages 248–254, 2013.

[SVW17] T. Steinke, S. P. Vadhan, and A. Wan. Pseudorandomness and fourier-growth bounds for width-3 branching programs. *Theory of Computing*, 13(1):1–50, 2017.

[SZ95] M. Saks and D. Zuckerman. Personal Communication, 1995.

[SZ11] J. Sima and S. Zák. Almost k-wise independent sets establish hitting sets for width-3 1-branching programs. In *CSR*, pages 120–133, 2011.

[Ta-17] A. Ta-Shma. Explicit, almost optimal, epsilon-balanced codes. In *Proceedings of the 49th Annual ACM SIGACT STOC 2017*, pages 238–251, 2017.

[Tre10] L. Trevisan. Open problems in unconditional derandomization. *Presentation at China Theory Week*, 2010.