Electronic Colloquium on Computational Complexity, Report No. 67 (2025)

Better Weighted Pseudorandom Generators Against Low Weight Read-Once Branching Programs

Ben Chen* Amnon Ta-Shma†

May 21, 2025

Abstract

In this work, we combine the work of Chen *et al.* and Hoza to obtain a WPRG against *regular* ROBPs with seed length $O(\log t \cdot (\log w + \sqrt{\log \frac{1}{\varepsilon}} + \log\log t) + \log \frac{1}{\varepsilon})$, improving upon previous construction which also include some additional lower order terms.

*Department of Computer Science, Tel Aviv University. The research leading to these results has received funding from the Israel Science Foundation (grant number 443/22). Email: ben1chen@gmail.com.

†Department of Computer Science, Tel Aviv University. The research leading to these results has received funding from the Israel Science Foundation (grant number 443/22). Email: amnon@tauex.tau.ac.il.

ISSN 1433-8092

1 Introduction

Derandomizing space-bounded probabilistic computation is a major challenge of theoretical computer science. The problem comes in two flavors:

- Derandomization, where the goal is to replace the probabilistic algorithm with another deterministic, space-bounded algorithm solving the same problem in about the same space, and,
- Pseudo-randomness, where derandomization is achieved by retaining the probabilistic algorithm, but replacing its random coins by an explicit, shallow distribution with a small support.

We call the first type (of general derandomization) *white-box* derandomization, because the derandomization is specific to the specific algorithm it deranomizes. We call the second type (using pseudo-randomness) *black-box* derandomization, because the derandomization is oblivious to the specific algorithm. We refer the reader to the excellent survey paper [H$^+$22] for further reading.

In this paper we focus on black-box derandomization. The class of algorithms we try to fool is the set of (w, t, Σ) read-once branching programs (ROBPs) which can be thought of as a (t+1)-layered directed graph, where each vertex in the graph has Σ outgoing edges connecting to the next layer, according to the machine action on the given coin toss. $(w, t, \Sigma) - \mathsf{ROBP}$s are the non-uniform analog of space-bounded machines using $\log w$ space and t time, getting at each time step a uniform toss of a $|\Sigma|$-sided dice. For a formal definition see Definition 2.1. Our goal is to replace the uniform distribution used by the ROBP with another distribution having a small support.

A prominent example is the INW PRG [INW94] which is a function $P : \{0,1\}^s \to \Sigma^t$, such that for every $(w, t, \Sigma) - \mathsf{ROBP}$ B, running B on a uniform string $\sigma \in \Sigma^t$ behaves similarly to running B on a uniform string taken from the image of the PRG. In symbols, $||\mathbf{E}_{x\in\{0,1\}^s} B(INW(x)) - \mathbf{E}_{\sigma\in\Sigma^t} B(\sigma)|| \leq \varepsilon$ The INW PRG has seed length $\log|\Sigma| + O(\log t \cdot \log(\frac{wt}{\varepsilon}))$. Remarkably, after more than thirty years of intensive research, and a flurry of results, there is no better PRG known today for general $(w, t, \Sigma) - \mathsf{ROBP}$s.

Braverman, Cohen and Garg [BCG19] generalized the PRG notion by assigning integral (possibly negative) weights to the outputs of the PRG. We say a weighted PRG (WPRG) derandomizes a $(w, t, \Sigma) - \mathsf{ROBP}$ B if the weighted average of running B on the image of the WPRG is close to the true average (see Definition 2.3 for a formal definition). Note that derandomization by a WPRG is still black-box derandomization as the WPRG is oblivious to the specific machine in the class it tries to fool. [BCG19] showed a WPRG against $(w, t, \Sigma) - \mathsf{ROBP}$ with seed length $\widetilde{O}(\log t \log wt + \log \frac{1}{\varepsilon})$. The [BCG19] result is technical and complicated. Following [AKM$^+$20], [CDR$^+$21, PV21] obtained the same result in a cleaner way using Richardson's iteration. Table 1 summarizes the results for general ROBP.

There has been a flurry of work trying to improve and/or generalize these bounds, for PRG or WPRG against restricted ROBPs. A partial list of these results include:

Seed Size	Reference	Remarks
$O(\log t \cdot \log \frac{wt}{\varepsilon})$	[Nis90, INW94]	PRG
$\widetilde{O}(\log t \cdot \log(wt) + \log \frac{1}{\varepsilon})$	[BCG19]	WPRG
$\widetilde{O}(\log t \cdot \log(wt)) + O(\log \frac{1}{\varepsilon})$	[CL20]	WPRG
$O(\log t \cdot \log(wt)) + \widetilde{O}(\log \frac{1}{\varepsilon})$	[CDR+21, PV21]	WPRG
$O(\log t \cdot \log(wt) + \log \frac{1}{\varepsilon})$	[Hoz21]	WPRG
$O(\frac{\log t \cdot \log(wt)}{\log \log w} + \log \frac{w}{\varepsilon} \log_t \log \frac{1}{\varepsilon})$	[CDR+21, PV21]	WPRG when $t \ll w$
$O(\frac{\log t \cdot \log(wt)}{\log \log w} + \log w \log \log \log w + \log \frac{1}{\varepsilon})$	[CW25]	WPRG when $t \ll w$

Table 1: PRG and WPRG for general ROBP

- Work on fooling small width ROBPs (e.g. width-3 ROBP [MRT19, CHL+23]),
- Work on restricted ROBPs such as regular [BRRY14, CHL+23, CL23], permutation [HPV21, PV21, CHL+23, CW25] or monotone [DMR+21] branching programs,

See Definition 2.8 for a formal definition of regular ROBPs. For the other classes, we refer the interested reader to the survey papers [H+22, HH+24]. The result in this paper focuses on constructing a WPRG against small-width, regular ROBPs.

Braverman *et al.*[BRRY14] showed a PRG against *regular* $(w, t, \Sigma) - \mathsf{ROBP}$ with seed length $O(\log t \cdot \log \frac{w \log t}{\varepsilon})$. In a seminal result, and using completely different techniques, Ahmadinejad *et al.* gave a *white-box* algorithm completely derandomizing the class of regular ROBPs. Specifically, they show that for *regular* $(w, t, \Sigma) - \mathsf{ROBP}$ B, one can approximate $\mathbf{E}_{\sigma \in \Sigma^t} B(\sigma)$ in space $\widetilde{O}(\log wt \cdot \log \log \frac{1}{\varepsilon})$. Chen *et al.*[CHL+23] vastly simplified the [AKM+20] technique, and we discuss this soon. In addition, they gave a WPRG against *regular* $(w, t, \Sigma)-$ ROBP with seed length $O(\log t \cdot (\log w + \sqrt{\log \frac{1}{\varepsilon}} + \log \log t)) + O(\log \frac{w}{\varepsilon} \cdot \log \log \frac{1}{\varepsilon})$, where $\log \log t$ and $\log \log \frac{1}{\varepsilon}$ are lower order terms. Compared with the previous WPRGs for *general* ROBPs, they remove the $O(\log^2 t)$ term at the expense of adding a $\log t \sqrt{\log \frac{1}{\varepsilon}}$ term (up to lower order terms). This is an improvement when w is small relative to t and $\varepsilon > 2^{-\log^2 t}$.

In this paper we shave off the lower order term multiplying the $\log \frac{w}{\varepsilon}$ term. Specifically:

Theorem 1.1. *(informal) There exists an explicit* WPRG *that* ε*–fools* regular $(w, t, \Sigma) -$ ROBP *with seed length* $O(\log t \cdot (\log w + \sqrt{\log \frac{1}{\varepsilon}} + \log \log t)) + O(\log \frac{1}{\varepsilon})$.

In many cases this completely removes the lower order terms. For example, when $t \leq 2^w$, or ε is mildly small, i.e., $\varepsilon \leq 2^{-(\log \log t)^2}$. We summarize these results in Table 2.

We next discuss the previous techniques and our contribution. The common theme underlying many of these results, is the attempt to replace $\log t \log \frac{1}{\varepsilon}$ by $\log t + \log \frac{1}{\varepsilon}$, an

Seed Size	Reference	Remarks
$\widetilde{O}(\log(wt)\cdot\log\log\frac{1}{\varepsilon})$	[AKM+20]	White-box (space)
$O(\log t\cdot\log\frac{w}{\varepsilon})$	[BRRY14]	PRG
$O(\log t\cdot\log wt+\log\frac{1}{\varepsilon})$	[Hoz21]	WPRG, general ROBP
$O(\log t(\log w+\sqrt{\log\frac{1}{\varepsilon}}+\log\log t)+\log\frac{1}{\varepsilon}\log\log\frac{1}{\varepsilon})$	[CHL+23]	WPRG
$O(\log t(\log w+\sqrt{\log\frac{1}{\varepsilon}}+\log\log t)+\log\frac{1}{\varepsilon})$	This paper	WPRG

Table 2: PRG and WPRG for *regular* ROBP

endeavor often termed "liberating" the error. Braverman *et al.* [BCG19] were the first to show such a result, and for that they constructed the first WPRGs, crucially exploiting negative weights. In a breakthrough result, Ahmadinejad *et al.* [AKM+20] obtained an almost optimal white-box derandomization, by using a host of new techniques, including Richardson's iteration for error reduction. Cohen *et al.* [CDR+21] and Pyne *et al.* [PV21] noticed that Richardson's iteration give a clean way to obtain the [BCG19] result. We now explain this technique, as we build upon it.

Richardson's iteration is an amazingly efficient way to reduce the approximation error. Suppose we have at hand a PRG ε_{prg}-fooling $(w,t,\Sigma)-$ ROBP, for some large error ε_0. We will show how to construct from it a WPRG ε-fooling the same class for a much smaller ε. We first describe a *white-box* algorithm, and then discuss how to turn it to a *black-box* WPRG.

Suppose we want to fool a (w,t,Σ) ROBP B. Let W_i denote the random walk matrix of B at time $i\in[t-1]$ (for a formal definition see Definition 2.1). Instead of working with t linear operators W_i, we define the "clocked" transformation W of dimension $w(t+1)\times w(t+1)$, that has $(t+1)\times(t+1)$ blocks, each of dimension $w\times w$, and where the blocks are indexed by $i,j\in[t]$ and the (i,j)'th block is $W[i,j]=W_i$ when $j=i+1$ and zero otherwise.

The "clocked Laplacian" of B is $L=I-W$. Since $W^{t+1}=0$, L is invertible and $L^{-1}=I+W+...+W^t$. One can see that the blocks of L^{-1} are

$$L^{-1}[i,j]=\begin{cases}0_w & \text{If } j>i\\ I_w & \text{If } i=j\\ \Pi_{i\to j}W_i & \text{If } 0\le i<j\le t,\end{cases}$$

where $\Pi_{i\to j}W_i$ is taken to be $W_j\cdot\ldots\cdot W_i$, and notice that this is a non-commutative product. Thus, we see that the *exact* operator L^{-1} carries all the information that we need, and, in particular,

$$L^{-1}[0,t]=W_{t-1}\ldots W_0 \tag{1}$$

is the operator capturing the execution of the ROBPB.

By definition $L^{-1}[i,j] = \Pi_{i\to j} W_i = \mathbf{E}_{\sigma\in\Sigma^{j-i}} B_{i\to j}(\sigma)$ where $B_{i\to j}$ is the $(w, j-i, \Sigma)$ ROBP that acts as B on the layers between i and j (see Definition 2.1 for a formal definition). Now assume we have a crude PRG P fooling ROBPs. We replace each block $\mathbf{E}_{\sigma\in\Sigma^{j-i}} B_{i\to j}(\sigma)$ with the pseudo-random block $\mathbf{E}_{x\in\{0,1\}^s} B_{i\to j}(P_{i\to j}(x))$ (again, see Definition 2.2, for a rigorous definition). Approximating each block independently we get an approximation $\overset{I}{\widetilde{L^{-1}}}$ to L^{-1}:

$$\overset{I}{\widetilde{L^{-1}}}[i,j] = \begin{cases} 0_w & \text{If } j > i \\ I_w & \text{If } i = j \\ \mathbf{E}_{x\in\{0,1\}^s} B_{i\to j}(P_{i\to j}(x)) & \text{If } 0 \le i < j \le t \end{cases}$$

We denote this approximation by $\overset{I}{\widetilde{L^{-1}}}$ because each block is approximated separately and independently. It is not difficult to see that

$$||\overset{I}{\widetilde{L^{-1}}} - L^{-1}|| \le (t+1)\varepsilon_{prg} \tag{2}$$

Richardson's approximation takes any ε_0 approximation A_0 to L^{-1}, (with independent approximations as above, $\varepsilon_0 = (t+1)\varepsilon_{prg}$), in some sub-multiplicative norm, and gives a better approximation to L^{-1} with low-error ε. Specifically, the new approximation is $\sum_{j=0}^{m}(I - A_0 L)^i \cdot A_0$, where $m = \log_{1/\varepsilon_0}(1/\varepsilon)$. Doing the calculation, one can see that this approximation can be done in space $\widetilde{O}(\log wt \log\log \frac{1}{\varepsilon})$ [AKM+20, CHL+23].

Cohen *et al.* [CDR+21] and Pyne *et al.* [PV21] noticed that if A_0 is the expectation of B over the image of a PRG P, then one can convert the above white-box algorithm to a black-box WPRG. This is explained in Section 2.2. Roughly speaking, opening $(I - A_0 L)^i$ to multiplications, we get a weighted PRG that incorporates up to m independent calls to the PRG P.

Now, having m independent calls to P is too expensive. To solve this issue:

- [CDR+21, PV21] use a low-error INW generator to create m dependent seeds that are used in the m applications of P. The crucial point here is that even though the INW generator has very low error, it is very short $m \ll \log \frac{1}{\varepsilon}$, and therefore the penalty incurred by this is off by only lower order terms such as $\log\log(1/\varepsilon)$.
- Hoza [Hoz21] solved the problem in a completely different way. Instead of using the INW PRG to generate m dependent seeds, Hoza keeps the different applications of P *independent*. However, in each application, instead of averaging over all the seeds of P, Hoza uses sampler, where on the right hand side we have all possible seeds to the PRG P. Hoza samples a single x on the left hand side of the sampler, and in each of the independent applications of P, it averages only over the neighbors of x in the sampler.

More formally, fixing x on the left hand side of the sampler, we get an operator $\widetilde{L^{-1}}^{H_x}$ approximating L^{-1}, where

$$\widetilde{L^{-1}}^{H_x}[i,j] = \begin{cases} 0_w & \text{If } j > i \\ I_w & \text{If } i = j \\ \mathbf{E}_{y \in Y} B_{i \to j}(P_{i \to j}(Samp(x,y))) & \text{If } 0 \leq i < j \leq t \end{cases}$$

Hoza showed that for many fixed $x \in X$, the approximation is ε–good. The bad x's might be very bad, but good samplers have a tiny fraction δ of bad vertices.

Both [CDR+21, PV21] and [Hoz21] take the error ε_{prg} of the crude PRG P to be $\varepsilon_{prg} = poly(\frac{1}{t})$[1], so that after using Equation (2), $\varepsilon_0 < 1$. We next address this point.

Chen *et al.* [CHL+23] simplified [AKM+20], and also combined their ideas with previous ideas to give improved WPRGs against restricted ROBPs. The basic idea is to replace the *independent* approximation of each block $L^{-1}[i,j]$, by *dependent* approximations. Specifically, they introduced the shortcut graph and independently approximated only blocks that correspond to edges in the short-cut graph. They put an edge between i and j iff $j - i$ is a power of 2 dividing i. They then complete the other blocks from these approximations. For $i < j$, they first find the shortest path from i to j on the shortcut graph, and they let the (i,j)'th block be the product of the approximations along the edges of this path. We denote the resulting approximation by $\widetilde{L^{-1}}^{SC}$, where

$$\widetilde{L^{-1}}^{SC}[i,j] = \begin{cases} 0_w & \text{If } j > i \\ I_w & \text{If } i = j \\ \mathbf{E}_{s_1,...,s_\ell \in \{0,1\}^s} B_{i \to j}(P^{SC}_{i \to j}(s_1,...,s_\ell)) & \text{If } 0 \leq i < j \leq t \end{cases}$$

where

$$P^{SC}_{i \to j}(s_1,...,s_\ell) = P_{i_0 \to i_k}(s_1) \circ ... \circ P_{i_{k-1} \to i_k}(s_k), \tag{3}$$

and $i = i_0, ..., i_k = j$ is the shortest path from i to j in the shortcut graph (see Definition 2.6 for a formal treatment). Chen *et al.* proved that the true inverse of $\widetilde{L^{-1}}^{SC}$ is a meaningful object (that corresponds to a weighted graph, with possibly negative weights). They use the sparsity of the shortcut graph to prove:

[1]Specifically [Hoz21] additionally requires the error to be smaller than w due to constraints arising from seed calculations.

Lemma 1.2. *[CHL+23](informal) If for every $i < j$ the* PRG $P_{i\to j}$ *ε_{prg}-fools $B_{i\to j}$ then*

$$||I - \widetilde{L^{-1}}^{SC} L||_\infty \leq O(MW(B) \cdot w^2 \cdot \log t) \cdot \varepsilon_{prg},$$

where $MW(B)$ is the maximal weight of B and its subprograms $B_{i\to j}$

We also note that Braverman *et al.* showed that $MW(B)$ is at most $poly(w)$ for regular ROBPs.

We now see that with the new bound in Lemma 1.2, and when working against *regular* ROBPs, we only need to choose $\varepsilon_{\mathsf{PRG}} \leq poly(\frac{1}{w \log t})$, which is much larger than $1/t$ when $w \ll t$. We can then try to reduce the error to ε using Richardson's iteration. Opening Richardson's iteration, reduces to m independent applications of operators that come from the shortcut graph, which reduces to $O(m \log t)$ applications of the original PRG P. To get the results stated before, [CHL+23] replace the $O(m \log t)$ independent applications of P, with *dependent* applications, where the seeds are the output of a short (length $O(m \log t)$) low-error INW generator.

We suggest a different way of implementing the Richardson's reduction. Chen *et al.* reduce the new operator to a concatenation of several applications of the original PRG P and they use *dependent* seeds for the different applications. Instead, inspired by Hoza's work on WPRG for general ROBPs, we would have liked to run *independent* applications of P^{SC} (defined in Equation (3)), but over a smaller sample space. I.e., we want to use a sampler, such that most vertices x on the left hand side, define a good subsample of the the seeds of the PRG P^{SC}, and then, opening Richardson's approximation, we run m independent applications of P^{SC}, but averaging each (independently) only over the neighbors of x.

Working out the parameters we find out that this approach is too expensive. The reason for that is that the PRG P^{SC} has $O(\log t)$ independent calls to the original PRG P, and these independent calls are too expansive. Our final solution is to use *dependent* applications for the $O(\log t)$ applications of P in P^{SC}, using, say, the INW generator, and to use *independent* calls to P^{SC} using Hoza's sampler trick. Specifically, we define the derandomized short-cut operator

$$\widetilde{L^{-1}}^{SC+INW}[i,j] = \begin{cases} 0_w & \text{If } j > i \\ I_w & \text{If } i = j \\ \mathbf{E}_{s_1 \in S_{INW}} B_{i\to j}(P^{SC}_{i\to j}(INW(s_1))) & \text{If } 0 \leq i < j \leq t \end{cases}$$

We then use independent applications of this operator on the simpler subset. Specifically, this gives the operator:

$$\widetilde{L^{-1}}^{SC+INW+H_x}[i,j] = \begin{cases} 0_w & \text{If } j > i \\ I_w & \text{If } i = j \\ \mathbf{E}_{y\in Y} B_{i\to j}(P^{SC}_{i\to j}(INW(Samp(x,y)))) & \text{If } 0 \le i < j \le t \end{cases}$$

Saying it differently, Chen *et al.* take dependent samples for all the $O(m \log t)$ applications of the original PRG P, whereas we take m independent samples (but averaged over a smaller subset using the sampler) of a PRG that takes $O(\log t)$ dependent samples of P along the short-cut graph. This gives the results stated before.

The WPRG we have seen before works not only for regular ROBPs, but also for general ROBPs with small weight (where weight here means Definition 2.7). Meka *et al.* [MRT19] showed that a random restriction (with a small bias distribution) of a width 3 ROBP is (with a good probability) a small weight ROBP. Thus, Chen *et al.* plug their WPRG into the [MRT19] construction. However plugging our improved WPRG does not improve the bottom line result because the restrictions contain more dominant terms than our improvement.

The paper is organized as follows. We begin with definitions, notations and background in Section 2. In Section 3 we present the construction of our low weight WPRGand prove its correctness.

2 Preliminaries and Background

$[k]$ denotes the set $\{0,\ldots,k\}$. U_t denotes the uniform distribution over $\{0,1\}^t$. Let M be a $w \times w$ matrix. The infinite induced norm $||M||_\infty$ is $||M||_\infty = \max_{||x||_\infty=1}||Mx||_\infty = max_{1\le r\le w}|M_r|_1$, where M_r is the r'th row of M. I_w denotes the $w \times w$ identity matrix and 0_w the $w \times w$ zero matrix.

For a $k \times k$ matrix M and $i,j \in [k-1]$, $M[i,j]$ is the value of M at the i'th row and j'th column. For consistency the rows and columns of a $w \times w$ matrix are indexed by $[w-1]$, i.e., the indices run from 0 to $w-1$. For every $f : [w-1] \to [w-1]$ there is a corresponding $w \times w$ boolean matrix M_f such that $M_f[i,j] = 1$ iff $f(j) = i$. We denote the set of such matrices by $SBM_{w\times w}$ (stochastic, boolean matrices).

Definition 2.1 (ROBP)**.** *Let Σ be an arbitrary subset, $w,t \in \mathbb{N}$. B is a width w length t read once branching program (ROBP) on alphabet Σ if it is a sequence of t functions $(B_0, B_1, ..., B_{t-1})$, with $B_i : \Sigma \to SBM_{w\times w}$. The* evaluation *of B on input $\sigma_0, ...\sigma_{t-1} \in \Sigma^t$ is the linear operator $B(\sigma_0, ..., \sigma_{t-1}) \stackrel{\text{def}}{=} B_{t-1}(\sigma_{t-1}) \cdot ... \cdot B_0(\sigma_0)$. We also say B is a $(w,t,\Sigma) - $ROBP. For $i < j \in [t]$ we let $B_{i\to j}$ denote the sequence $(B_i, ..., B_{j-1})$, which is a $(w, j-i, \Sigma) -$ROBP. For any $i \in [t-1]$ the random walk matrix in time i is $W_i = E_{\sigma\in\Sigma}B_i(\sigma)$.*

Definition 2.2 (PRG)**.** *Let Σ be an arbitrary subset and $s,t \in \mathbb{N}$. A (s,t,Σ) pseudo random generator is a function $PRG : \{0,1\}^s \to \Sigma^t$. For $\epsilon > 0$ and a norm $||\cdot||$, we say PRG*

$(\epsilon, ||\cdot||)$-*fools* $(w,t,\Sigma) - \mathsf{ROBP}$ *if for every* $(w,t,\Sigma) - \mathsf{ROBP}$ B *we have:*

$$||\mathbf{E}_{\sigma\in\Sigma^t} B(\sigma) - \mathbf{E}_{x\in\{0,1\}^s} B(PRG(x))|| \leq \epsilon$$

Definition 2.3 (WPRG)**.** *Let* Σ *be an arbitrary subset and* $s,t,k \in \mathbb{N}$*. A weighted* (s,t,Σ,k) *pseudo random generator is a tuple* (Γ, P)*, where* $P : \{0,1\}^s \to \Sigma^t$ *and* $\Gamma : \{0,1\}^s \to [-k,k]$*. For* $\epsilon > 0$ *and a norm* $||\cdot||$*, we say* (Γ, P) $(\epsilon, ||\cdot||)$*-fools* $(w,t,\Sigma) - \mathsf{ROBP}$ *if for every* $(w,t,\Sigma) - \mathsf{ROBP}$ B *we have:*

$$||\mathbf{E}_{\sigma\in\Sigma^t} B(\sigma) - \mathbf{E}_{x\in\{0,1\}^s} \Gamma(s) B(P(s)))|| \leq \epsilon$$

Definition 2.4 (Sampler)**.** *A function* $Samp : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *is an* (ε,δ)*-sampler if for every function* $f : \{0,1\}^m \to [0,1]$*,*

$$\Pr_{x\in\{0,1\}^n} [|\mathbf{E}_{u\in U_m} f(u) - \mathbf{E}_{y\in U_d} f(Samp(x,y))| \leq \varepsilon]] \geq 1-\delta$$

Theorem 2.5. *[RVW00, Gol11, CL20] For every* $\delta, \varepsilon > 0$ *and integer* m*, there exists an* (ε,δ)*-sampler* $f : \{0,1\}^n \times \{0,1\}^d \to \{0,1\}^m$ *s.t.* $d = O(\log\log\frac{1}{\delta} + \log\frac{1}{\varepsilon})$ *and* $n = m + O(\log\frac{1}{\delta} + \log\frac{1}{\varepsilon})$ *that can be computed in space* $O(m + \log\frac{1}{\delta\varepsilon})$*.*

Definition 2.6 ([CHL+23] Shortcut Graph)**.** *For any* $n \in \mathbb{N}$*, the shortcut graph* SC_N *on* $N = 2^n$ *is a graph* (V,E)*. Where* $V = [N]$ *and*

$$E = \{(i, i+2^q) \in V \times V | 2^q \text{ divides } i\}$$

2.1 Weight and Regular ROBPs

Let B be a $(w,t,\Sigma) - \mathsf{ROBP}$, and $\vec{q} \in \{0,1\}^w$ a vector indicating which of the vertices on the final layer is accepting. Let $u \in [w-1]$ and $i \in [t]$. The accept rate of (i,u) in relation to $B, \vec{q}$ is

$$Accpt(B,\vec{q},i,u) = \begin{cases} \vec{q}_u & \text{If } i = t, \\ (\vec{q}^T E_{\sigma\in\Sigma^{t-i}} B_{i\to t}(\sigma))_u & \text{Else} \end{cases}$$

Definition 2.7 (Weight of ROBP)**.** *[BRRY14, CHL+23] Let* B *be a* $(w,t,\Sigma) - \mathsf{ROBP}$*. The* weight *of* B*, denoted as* $Weight(B)$*, is*

$$Weight(B) = max_{\vec{q}\in\{0,1\}^w} \sum_{i=0}^{t-1} \sum_{u\in[w-1],\sigma\in\Sigma} |Accpt(B,q,i,u) - Accpt(B,q,i+1,\Gamma(i,u,\sigma))|$$

where $\Gamma(i,u,\sigma)=v$ *is the unique* $v\in[w-1]$ *s.t.* $B_i(\sigma)_{v,u}=1$. *We define the maximal weight for* B *as* $MW(B)$ *by*

$$MW(B)=\max_{i<j\in[t]} Weight(B_{i\to j})$$

Definition 2.8 (Regular branching program)**.** *Let* B *be as above. We say* B *is a* regular ROBP *if for every* $i\in[t-1]$ *the random walk matrix* W_i *is doubly stochastic.*

Theorem 2.9. *[BRRY14] If* B *is a* regular $(w,t,\{0,1\})-$ROBP*, then* $MW(B)\leq w(w-1)$.

2.2 Black Box Richardson's Iteration

Fix B a $(w,t,\Sigma)-$ ROBP. Let W be the clocked random walk matrix, L the "clocked Laplacian" matrix, L^{-1} its inverse and $A_0=\widetilde{L^{-1}}$ its approximation as defined in Section 1. We assume we have a family of PRGs $\{P_{i\to j}\}_{i,j\in[t]}$ such that

$$A_0[i,j]=\mathbf{E}_s B_{i\to j}(P_{i\to j}(s))$$

and A_0 approximates L^{-1}, i.e., $||I-A_0L||\leq\varepsilon$, where ε is some crude approximation. We now use,

Lemma 2.10 (Richardson's iteration)**.** *Let* $m\in\mathbb{N}$ *and* A_0 *a matrix. Denote*

$$A_m=\sum_{i\in[m]}(I-A_0L)^iA_0 \tag{4}$$

For any sub-multiplicate norm, if $||I-A_0L||\leq\varepsilon$ *then* $||I-A_mL||\leq\varepsilon^m$.

Opening up the powers in Equation (4) to multiplications, Cohen *et al.* [CDR+21] and Pyne *et al.* [PV21] showed that if A_0 is defined by a PRG, then $A_m[0,t]$ can also be evaluated in a black-box manner, alas with weights, i.e., there exists a WPRG (Γ,G) that implements A_m, i.e., $A_m[0,t]=\mathbf{E}_{s\in S_G}\Gamma(s)B(G(s))$.

To explain how this is done we need some definitions. A *partition* of $[t]$ into i parts is $\pi\in[t]^{i+1}$ where $0\leq\pi_0\leq...\leq\pi_i=t$. A *labeling* of a partition π with i parts is a function $\ell:[i-1]\to\{0,1\}$. A *labeled partition* of size i (i,π,ℓ) is a partition π with i parts along with a labeling function $\ell:[i-1]\to\{0,1\}$ of π. Fix an integer m. We let $\Pi(t,m)$ denote the set of all labeled partitions (i,π,ℓ) of $[t]$ of size $i\in[m]$.

Definition 2.11 (Independent seeds WPRG)**.** *[CDR+21, PV21] Keeping notation as above we assume* $A_0[i,j]=\mathbf{E}_s B_{i\to j}(P_{i\to j}(s))$. *We define* (Γ,G) *where*

$$\Gamma:\Pi(t,m)\to\mathbb{R}$$
$$G:\Pi(t,m)\times\{0,1\}^{s\cdot(m+1)}\times\Sigma^m\to\Sigma^t$$

defined by:

$$\Gamma(i,\pi,\ell) = (-1)^{\sum_{k\in[i]}\ell_k}\cdot|\Pi(t,m)|$$
$$G((i,\pi,\ell),y_0,...,y_m,\sigma_1,...,\sigma_m) = P_{0\to\pi_0}(y_0)\circ P^{\ell_0}_{\pi_0\to\pi_1}(\sigma_1,y_1)\circ...\circ P^{\ell_{i-1}}_{\pi_{i-1}\to\pi_i}(\sigma_i,y_i)$$

where $P^b_{i\to\ j}$ *is defined by*

- $P^0_{i\to j}(\sigma,s) = P_{i\to j}(s)$
- $P^1_{i\to j}(\sigma,s) = \sigma\circ P_{i\to j}(s)$

Lemma 2.12. *[CDR+21, PV21] As* $A_0[i,j] = \mathbf{E}_s B_{i\to j}(P_{i\to j}(s))$ *we have that* $A_m[0,t] = \mathbf{E}_{s\in S_G}\Gamma(s)B_{0\to t}(G(s))$

2.3 Moderate Approximation in Low Weights ROBP

As described in Section 1 Chen *et al.* used the shortcut graph (given in Definition 2.6) to define their approximation $\widetilde{L^{-1}}^{SC}$ to the inverse of the clocked laplacian. They then defined $\widetilde{L}$ to be the true inverse of $\widetilde{L^{-1}}^{SC}$, i.e., $\widetilde{L} = (\widetilde{L^{-1}}^{SC})^{-1}$. They proved that:

Lemma 2.13. *[CHL+23, Lemma 6.4] If for every* $i<j\in[t]$ *the* PRG $P_{i\to j}$ ε_{prg}*-fools* $B_{i\to j}$ *then*

$$||L^{-1}(\widetilde{L}-L)||_\infty \leq 1.5\cdot MW(B)\cdot w^2\cdot\log t\cdot\varepsilon_{prg},$$

They also proved:

Corollary 2.14. *[CHL+23, Corollary 4.11] Let* L *and* $\widetilde{L}$ *be lower uni-triangular matrices of the same dimension,* $\delta\in[0,1)$*. If* $||L^{-1}(\widetilde{L}-L)||_\infty\leq\delta$ *then* $||(\widetilde{L})^{-1}(\widetilde{L}-L)||_\infty\leq\frac{\delta}{1-\delta}$

Together this implies:

Lemma 2.15. *[CHL+23] If for every* $i<j\in[t]$ *the* PRG $P_{i\to j}$ ε_{prg}*-fools* $B_{i\to j}$ *then*

$$||I-\widetilde{L^{-1}}^{SC}L||_\infty\leq 3\cdot MW(B)\cdot w^2\cdot\log t\cdot\varepsilon_{prg}.$$

3 Our Construction

As described in the introduction, our paper combines ideas from [CHL+23] and [Hoz21]. The construction begins in the same way as [CHL+23]. We use the shortcut graph to define an initial approximation $\widetilde{L^{-1}}^{SC}[i,j]$ for each block $L^{-1}[i,j]$. The approximation in [CHL+23] is

defined as $O(\log t)$ independent approximations of our original PRG. To use Hoza's sampler idea, we start by replacing the independent approximations with dependent approximations using the INW PRG. We denote this approximation by $\widetilde{L^{-1}}^{SC+INW}$. We then apply Hoza's sampler idea to get our final approximation which we denote by $\widetilde{L^{-1}}^{SC+INW+H_x}$. We now give the details.

Let $w, t, W \in \mathbb{N}, \varepsilon > 0$ and Σ an arbitrary alphabet. We construct a WPRG that ε-fools $(w, t, \Sigma) - \mathsf{ROBP}$ with max weight at most W.

We define three new parameters $\varepsilon_0, \ell_{max}, m$ where:

- $\varepsilon_0 = \frac{1}{w} \cdot 2^{-\sqrt{\log \frac{1}{\varepsilon}}}$ - our initial error for Richardson's iteration,
- $\ell_{max} = 2 \log t$ - the maximal length of a shortest path in the graph SC_t of Definition 2.6,
- $m = \frac{\log \frac{2}{\varepsilon}}{\log \frac{1}{\varepsilon_0}} + 1$ -the maximal degree of the Richardson's iteration.

We use the following building blocks

- [BRRY14] : $P_{i \to j} : S_{BRRY} \to \Sigma^{j-i}$ is a PRG ε_{BRRY}-fooling $(w, j-i, \Sigma) - \mathsf{ROBP}$ with max weight W, where $\varepsilon_{BRRY} = \frac{\varepsilon_0}{6Ww^2 \log t}$. It has seed
$$\log|S_{BRRY}| = \log|\Sigma| + O(\log t \log \frac{w \cdot W \log t}{\varepsilon_0})$$
- [INW94]: $INW : S_{INW} \to S_{BRRY}^{\ell_{max}}$ is a PRG ε_{INW}-fooling $(w, \ell_{max}, S_{BRRY}) - \mathsf{ROBP}$, where $\varepsilon_{INW} = \frac{\varepsilon_0}{8w(t+1)}$. It has seed
$$\log|S_{INW}| = \log|S_{BRRY}| + O(\log \ell_{max} \log \frac{wt}{\varepsilon_0})$$
- [RVW00, Gol11, CL20] (also stated in Theorem 2.5): $Samp : X \times Y \to S_{INW}$ is a $(\delta_s = \frac{\varepsilon}{2w^2(2t)^{m+3}}, \varepsilon_s = \frac{\varepsilon_0}{8w(t+1)})$-sampler. Its parameters are bounded by
$$\log|X| = \log|S_{INW}| + O(\log \frac{1}{\varepsilon_s \cdot \delta_s}) = \log|S_{INW}| + O(\log \frac{w}{\varepsilon} + m \log t)$$
$$\log|Y| = O(\log \frac{\log \frac{1}{\delta}}{\varepsilon_s}) = O(\log \frac{mwt}{\varepsilon_0}) = O(\log \frac{wt}{\varepsilon_0})$$

As in [CHL+23], we define
$$P_{i \to j}^{SC_t}(s_1, ..., s_k) = P_{i_0 \to i_1}(s_1) \circ ... \circ P_{i_{k-1} \to i_k}(s_k)$$
where $i = i_0, ..., i_k = j$ is the shortest path from i to j in the shortcut graph SC_t.

Next, for any fixed $x \in X$, we define a family of PRG $\{P^{(x)}_{i\to j} : Y \to \Sigma^{j-i}\}_{i<j\le t}$

$$P^{(x)}_{i\to j}(y) = P^{SC_t}_{i\to j}(INW(Samp(x,y))), \text{ and,}$$
$$P^{(x),b}_{i\to j}(\sigma, y) = \begin{cases} P^{(x)}_{i\to j}(y) & b = 0, \\ \sigma \circ P^{(x)}_{i+1\to j}(y) & b = 1. \end{cases}$$

We are now ready to define our WPRG.

Definition 3.1. *We define* $W_{final} = (\Gamma, G)$ *where*

$$\Gamma : \Pi(t,m) \to R$$
$$G : \Pi(t,m) \times X \times Y^{m+1} \times \Sigma^m$$

by

$$\Gamma(i,\pi,t) = (-1)^{\sum_{k\in[i]} t_k} |\Pi(t,m)|$$
$$G((i,\pi,t), x, y_0, ..., y_m, \sigma_1, ..., \sigma_m) = P^{(x)}_{0\to\pi_1}(y_0) \circ P^{(x),t_1}_{\pi_1\to\pi_2}(\sigma_1, y_1) \circ ... \circ P^{(x),t_i}_{\pi_i\to\pi_{i+1}}(\sigma_i, y_i)$$

Theorem 3.2 (Main theorem)**.** *W_{final} is an explicit* WPRG *$(\varepsilon, ||\cdot||_{\ell_1})$-fooling $(w,t,\Sigma)-$*ROBP *with max weight at most W. Its seed length s is*

$$s = O\left(\log(w \cdot W \cdot \log t) \cdot \log t + \log(t \cdot |\Sigma|) \cdot \sqrt{\log \frac{1}{\varepsilon}} + \log \frac{1}{\varepsilon}\right).$$

An immediate corollary of Theorem 2.9 gives a WPRG for *regular* ROBP:

Theorem 3.3 (WPRG for *regular* ROBP)**.** $\Sigma = \{0,1\}$*. W_{final} is an explicit* WPRG *$(\varepsilon, ||\cdot||_{\ell_1})$-fooling* regular *$(w,t,\Sigma) -$* ROBP *with seed*

$$s = O\left(\log t \cdot \left(\log(w \cdot \log t) + \sqrt{\log \frac{1}{\varepsilon}}\right) + \log \frac{1}{\varepsilon}\right).$$

Tracing parameters:

Lemma 3.4 (Seed length)**.** *The* WPRG *W_{final} has seed length*

$$O\left(\log(w \cdot W \cdot \log t) \cdot \log t + \log(t \cdot |\Sigma|) \cdot \sqrt{\log \frac{1}{\varepsilon}} + \log \frac{1}{\varepsilon}\right).$$

and its weights are bounded by $t \cdot (2t)^{1+\sqrt{\log \frac{1}{\varepsilon}}}$.

Proof. If one of the parameters w, W is bigger than t, then [Hoz21] is a simpler WPRG achieving the stated parameters, so w.l.o.g assume $w, W \leq t$.

The seed length s of W_{final} is

$$\begin{aligned} s &= O(m(\log|Y| + \log|\Sigma|) + \log|X| + \log|\Pi(t,m)|) \\ &= O\left(\frac{\log\frac{1}{\varepsilon}}{\log\frac{1}{\varepsilon_0}}\left(\log\frac{t}{\varepsilon_0} + \log|\Sigma|\right) + \log t \cdot \log\frac{w \cdot W \cdot \log t}{\varepsilon_0} + \log\frac{w}{\varepsilon}\right) \end{aligned}$$

Plugging $\varepsilon_0 = \frac{1}{w} \cdot 2^{-\sqrt{\log\frac{1}{\varepsilon}}}$ we get the desired seed length.

The weights of the WPRG are all the same size $|\Pi(t,m)|$. They are bounded by the term $|\Pi(t,m)| = \sum_{i=0}^{m}\binom{t}{i}2^i \leq 2^m \cdot t^{m+1} = t(2t)^m \leq t \cdot (2t)^{1+\sqrt{\log\frac{1}{\varepsilon}}}$. □

Finally we claim correctness:

Lemma 3.5 (Correctness)**.** *The* WPRG W_{final} $(\varepsilon, ||\cdot||_{\ell_1})$*-fools* $(w,t,\Sigma) - \mathsf{ROBP}$ *with max weight at most* W*.*

We prove the lemma in the remaining subsections.

3.1 Correctness

Following [CHL+23, Hoz21], we prove the following two lemmas, which immediately imply Lemma 3.5. Fix any B a $(w,t,\Sigma) - \mathsf{ROBP}$ with $MW(B) \leq W$.

Definition 3.6 (Good and bad x)**.** *Call* $x \in X$ good *if for every* $i,j \in [t], u,v \in [w-1]$ *we have that*

$$\left|\mathbf{E}_{s\in S_{INW}} B_{i\to j}(P^{SC}_{i\to j}(INW(s)))_{v,u} - \mathbf{E}_{y\in Y} B_{i\to j}(P^{SC}_{i\to j}(INW(Samp(x,y))))_{v,u}\right| \leq \varepsilon_s,$$

where ε_s *is the sampler accuracy. We say* x *is* bad *if it is not good.*

Now,

Lemma 3.7 (Bad x)**.** $\mathbf{Pr}_{x\in X}[x \text{ is bad}] \leq \delta_s w^2(t+1)^2$*, where* δ_s *is the sampler confidence. In addition, for every* $x \in X$*,* $||\mathbf{E}_{s\in S_G}\Gamma(s) \cdot B(G^{(x)}(s)) - \mathbf{E}_{\sigma\in\Sigma^t}B(\sigma)||_{\ell_1} \leq t(2t)^m + 1$.

Lemma 3.8 (Good x)**.** *For every good* $x \in X$*,* $||\mathbf{E}_{s\in S_G}\Gamma(s) \cdot B(G^{(x)}(s)) - \mathbf{E}_{\sigma\in\Sigma^t}B(\sigma)||_{\ell_1} \leq \frac{\varepsilon}{2}$.

Assuming the two lemmas we conclude:

Proof of Lemma 3.5. Let B a $(w,t,\Sigma) - \mathsf{ROBP}$ with $MW(B) \leq W$. Then,

$$\begin{aligned} ||\mathbf{E}_x\mathbf{E}_s\Gamma(s)B(G^{(x)}(s)) - \mathbf{E}_{\sigma\in\Sigma^t}B(\sigma)||_{\ell_1} &\leq \mathop{\mathbf{Pr}}_{x\in X}[x \text{ is bad}] \cdot (t(2t)^m + 1) + \frac{\varepsilon}{2} \\ &\leq \delta_s w^2(2t)^{m+3} + \frac{\varepsilon}{2} \leq \varepsilon \end{aligned}$$

□

3.2 Bad x

Proof of Lemma 3.7. For $i, j \in [t]$ and $u, v \in [w-1]$ define

$$f_{i,j,u,v}(s) = B_{i\to j}(P^{SC}_{i\to j}(INW(s)))_{v,u}$$

Say $x \in X$ is bad for i, j, u, v if

$$|\mathbf{E}_{s\in S_{INW}} f_{i,j,u,v}(s) - \mathbf{E}_{y\in Y} f_{i,j,u,v}(Samp(x,y))| > \varepsilon_s$$

Since Samp is an $(\varepsilon_s, \delta_s)-$sampler, the probability x is bad for i, j, u, v is at most δ_s. By the union bound

$$\mathop{\mathbf{Pr}}_{x\in X}[x \text{ is good}] \geq 1 - \delta_s \cdot w^2(t+1)^2.$$

Finally, fix any $x \in X$. As $|\Gamma(s)| = |\Pi(t,m)|$ for every $s \in S_G$, we get:

$$\begin{aligned}||\mathbf{E}_{s\in S_G}\Gamma(s)B(G^{(x)}(s)) - \mathbf{E}_{\sigma\in\Sigma^t}B(\sigma)||_{\ell_1} &\leq ||\mathbf{E}_{s\in S_G}\Gamma(s)B(G^{(x)}(s))||_{\ell_1} + ||\mathbf{E}_{\sigma\in\Sigma^t}B(\sigma)||_{\ell_1}\\ &\leq |\Pi(t,m)|\cdot \max_{s\in S_G}||B(G^{(x)}(s))||_{\ell_1} + \max_{\sigma\in\Sigma^t}||B(\sigma)||_{\ell_1}\\ &\leq |\Pi(t,m)| + 1 \leq t(2t)^m + 1.\end{aligned}$$

□

3.3 Good x

Notice that by definition

$$\mathbf{E}_{s\in S_{INW}} f_{i,j,u,v}(s) = \left(\widetilde{L^{-1}}^{SC+INW}_{[i,j]}\right)_{v,u}$$

$$\mathbf{E}_{y\in Y} f_{i,j,u,v}(Samp(x,y)) = \left(\widetilde{L^{-1}}^{SC+INW+H_x}_{[i,j]}\right)_{v,u}$$

and so

Lemma 3.9. *For every good x,* $||\widetilde{L^{-1}}^{SC+INW} - \widetilde{L^{-1}}^{SC+INW+H_x}||_\infty \leq \varepsilon_s \cdot w(t+1) \leq \frac{\varepsilon_0}{8}$.

Furthermore,

Lemma 3.10. *For every good x,* $||I - \widetilde{L^{-1}}^{SC+INW+H_x} L||_\infty \leq \varepsilon_0$.

Proof. Using the triangle inequality and $||L||_\infty \leq 2$ we have:

$$||I - \widetilde{L^{-1}}^{SC+INW+H_x} L||_\infty \leq ||I - \widetilde{L^{-1}}^{SC} L||_\infty + ||\widetilde{L^{-1}}^{SC} - \widetilde{L^{-1}}^{SC+INW+H_x}||_\infty \cdot ||L||_\infty \leq$$
$$\leq ||I - \widetilde{L^{-1}}^{SC} L||_\infty + 2(||\widetilde{L^{-1}}^{SC+INW} - \widetilde{L^{-1}}^{SC}||_\infty + ||\widetilde{L^{-1}}^{SC+INW} - \widetilde{L^{-1}}^{SC+INW+H_x}||_\infty).$$

Now,

- we recall that B is a $(w, t, \Sigma) - \mathsf{ROBP}$ with $MW(B) \leq W$. Hence, by Lemma 2.15,

$$||I - \widetilde{L^{-1}}^{SC} L||_\infty \leq 3 \cdot MW(B) \cdot w^2 \cdot \log t \cdot \varepsilon_{BRRY} \leq \frac{\varepsilon_0}{2}$$

- Also, since INW ε_{INW}-fools $(w, k, S_{BRRY}) - \mathsf{ROBP}$ for all $k \leq \ell$,

$$||\widetilde{L^{-1}}^{SC+INW} - \widetilde{L^{-1}}^{SC}||_\infty \leq \varepsilon_{INW} \cdot w(t+1) \leq \frac{\varepsilon_0}{8}$$

- Finally, $||\widetilde{L^{-1}}^{SC+INW} - \widetilde{L^{-1}}^{SC+INW+H_x}||_\infty \leq \frac{\varepsilon_0}{8}$ by Lemma 3.9.

Together, this completes the proof of the lemma. □

Now, let

$$A_0^x = \widetilde{L^{-1}}^{SC+INW+H_x},$$

and recall that

$$A_0^x[i,j] = \mathbf{E}_{y \in Y} B_{i \to j}(P_{i \to j}^{SC}(INW(Samp(x,y)))) \tag{5}$$

Let A_m^x denote the matrix defined as the result of m Richardson's iterations. Namely,

$$A_m^x = \sum_{i=0}^{m} (I - \widetilde{L^{-1}}^{SC+INW+H_x} L)^i \cdot \widetilde{L^{-1}}^{SC+INW+H_x}$$

By Lemma 2.10 together with Lemma 3.10 and our choice of $m = 1 + \frac{\log \frac{2}{\varepsilon}}{\log \frac{1}{\varepsilon_0}}$, we conclude that

$$||I - A_m^x L||_\infty \leq \frac{\varepsilon_0 \cdot \varepsilon}{2} \leq \frac{\varepsilon}{2w}.$$

We are finally ready to prove Lemma 3.8, following [CHL+23, Section 6.2].

Proof. (of Lemma 3.8) For any basis vector $e = e_{i,u}$ (indexed by $i \in [t]$ and $u \in [w-1]$), we have that $||L^{-1}e||_\infty = 1$, because it contains columns of stochastic matrices applied on e_u. Hence,

$$\begin{aligned} ||L^{-1}e - A^x_m e||_\infty &= ||L^{-1}e - A^x_m L L^{-1}e||_\infty \\ &= ||(I - A^x_m L)L^{-1}e||_\infty \\ &\leq \frac{\varepsilon}{2w}||L^{-1}e||_\infty \leq \frac{\varepsilon}{2w}. \end{aligned}$$

In particular, this is true for all $e = e_{0,u}$, $u \in [w-1]$. We therefore conclude that for any $u, v \in [w-1]$,

$$|(A^x_m[0,t] - L^{-1}[0,t])[v,u]| \leq \frac{\varepsilon}{2w}$$

Recalling that

$$L^{-1}[0,t] = \mathbf{E}_{\sigma \in \Sigma^t} B(\sigma) \tag{6}$$

$$A^x_m[0,t] = \mathbf{E}_{s \in S_G} \Gamma(s) B_{0 \to t}(G^{(x)}(s)) \tag{7}$$

where Equation (6) is by Equation (1), and Equation (7) is by Lemma 2.12, where the $P = \{P_{i \to j}\}$ that we plug into the lemma is $\{P^{SC}_{i \to j}(INW(Samp(x,y)))\}$, where y is the seed and x is fixed (see Equation (5)).

Therefore, for any $u, v \in [w-1]$:

$$|(\mathbf{E}_s \Gamma(s) B(G^{(x)}(s)) - \mathbf{E}_{\sigma \in \Sigma^t} B(\sigma))[v,u]| \leq \frac{\varepsilon}{2w}$$

We conclude:

$$||\mathbf{E}_s \Gamma(s) B(G^{(x)}(s)) - \mathbf{E}_{\sigma \in \Sigma^t} B(\sigma)||_{\ell_1} \leq \frac{\varepsilon}{2}$$

□

References

[AKM+20] AmirMahdi Ahmadinejad, Jonathan Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil Vadhan. High-precision estimation of random walks in small space. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1295–1306. IEEE, 2020.

[BCG19] Mark Braverman, Gil Cohen, and Sumegha Garg. Pseudorandom pseudo-distributions with near-optimal error for read-once branching programs. *SIAM Journal on Computing*, 49(5):STOC18–242, 2019.

[BRRY14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM Journal on Computing*, 43(3):973–986, 2014.

[CDR+21] Gil Cohen, Dean Doron, Oren Renard, Ori Sberlo, and Amnon Ta-Shma. Error reduction for weighted prgs against read once branching programs. *Leibniz international proceedings in informatics*, 200(22), 2021.

[CHL+23] Lijie Chen, William M Hoza, Xin Lyu, Avishay Tal, and Hongxun Wu. Weighted pseudorandom generators via inverse analysis of random walks and shortcutting. In *2023 IEEE 64th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1224–1239. IEEE, 2023.

[CL20] Eshan Chattopadhyay and Jyun-Jie Liao. Optimal error pseudodistributions for read-once branching programs. *arXiv preprint arXiv:2002.07208*, 2020.

[CL23] Eshan Chattopadhyay and Jyun-Jie Liao. Recursive error reduction for regular branching programs. *arXiv preprint arXiv:2309.04551*, 2023.

[CW25] Kuan Cheng and Ruiyang Wu. Weighted pseudorandom generators for read-once branching programs via weighted pseudorandom reductions. *arXiv preprint arXiv:2502.08272*, 2025.

[DMR+21] Dean Doron, Raghu Meka, Omer Reingold, Avishay Tal, and Salil Vadhan. Pseudorandom generators for read-once monotone branching programs. *Leibniz international proceedings in informatics*, 207(58), 2021.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation: In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, pages 302–332. Springer, 2011.

[H+22] William M Hoza et al. Recent progress on derandomizing space-bounded computation. *Bulletin of EATCS*, 138(3), 2022.

[HH+24] Pooya Hatami, William Hoza, et al. Paradigms for unconditional pseudorandom generators. *Foundations and Trends® in Theoretical Computer Science*, 16(1-2):1–210, 2024.

[Hoz21] William M Hoza. Better pseudodistributions and derandomization for space-bounded computation. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021.

[HPV21] William M Hoza, Edward Pyne, and Salil Vadhan. Pseudorandom generators for unbounded-width permutation branching programs. In *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, pages 7–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021.

[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, 1994.

[MRT19] Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom generators for width-3 branching programs. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 626–637, 2019.

[Nis90] Noam Nisan. Pseudorandom generators for space-bounded computations. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, pages 204–212, 1990.

[PV21] Edward Pyne and Salil Vadhan. Pseudodistributions that beat all pseudorandom generators. In *36th Computational Complexity Conference (CCC 2021)*, pages 33–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021.

[RVW00] Omer Reingold, Salil Vadhan, and Avi Wigderson. Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 3–13. IEEE, 2000.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il