Deterministic Approximation of Random Walks via Queries in Graphs of Unbounded Size

Edward Pyne* Salil Vadhan†

Abstract

Consider the following computational problem: given a regular digraph $G = (V, E)$, two vertices $u, v \in V$, and a walk length $t \in \mathbb{N}$, estimate the probability that a random walk of length t from u ends at v to within $\pm\varepsilon$. A randomized algorithm can solve this problem by carrying out $O(1/\varepsilon^2)$ random walks of length t from u and outputting the fraction that end at v.

In this paper, we study *deterministic* algorithms for this problem that are also restricted to carrying out walks of length t from u and seeing which ones end at v. Specifically, if G is d-regular, the algorithm is given oracle access to a function $f : [d]^t \to \{0, 1\}$ where $f(x)$ is 1 if the walk from u specified by the edge labels in x ends at v. We assume that G is *consistently labelled*, meaning that the edges of label i for each $i \in [d]$ form a permutation on V.

We show that there exists a deterministic algorithm that makes $\mathrm{poly}(dt/\varepsilon)$ nonadaptive queries to f, regardless of the number of vertices in the graph G. Crucially, and in contrast to the randomized algorithm, our algorithm does not simply output the average value of its queries. Indeed, Hoza, Pyne, and Vadhan (ITCS 2021) showed that any deterministic algorithm of the latter form that works for graphs of unbounded size must have query complexity at least $\exp(\tilde{\Omega}(\log(t)\log(1/\varepsilon)))$.

In the language of pseudorandomness, our result is a separation between the query complexity of "deterministic samplers" and "deterministic averaging samplers" for the class of "permutation branching programs of unbounded width". Our separation is stronger than the prior separation of Pyne and Vadhan (CCC 2021), and has a much simpler proof (not using spectral graph theory or the Impagliazzo–Nisan–Wigderson pseudorandom generator). On the other hand, the algorithm of Pyne and Vadhan is explicit and computable in small space, whereas ours is not explicit (unless we assume the existence of an optimal explicit pseudorandom generator for permutation branching programs of bounded width).

1 Introduction

Consider the following computational problem: given a regular digraph $G = (V, E)$, two vertices $u, v \in V$, and a walk length $t \in \mathbb{N}$, estimate the probability that a random walk of length t from u ends at v to within $\pm\varepsilon$. A randomized algorithm can solve this problem by carrying out $O(1/\varepsilon^2)$ random walks of length t from u and outputting the fraction that end at v.

In this paper, we study *deterministic* algorithms for this problem that are also restricted to carrying out walks of length t from u and seeing which ones end at v. Specifically, if G is d-regular, the algorithm is given oracle access to a function $f : [d]^t \to \{0, 1\}$ where $f(x)$ is 1 if the walk from u specified by the edge labels in x ends at v. We assume that G is *consistently labelled*, meaning that the edges of label i for each $i \in [d]$ form a permutation on V. (It can be shown that every d-regular digraph has a consistent labelling, so this requirement does not constrain the graph structure, only the labelling.) In the case where we have "white-box" access to the graph, Ahmadinejad, Kelner, Murtagh, Peebles, Sidford, and Vadhan [AKM+20] gave a deterministic algorithm for this problem with space complexity $\tilde{O}(\log(|V| \cdot td/\varepsilon))$.

We show that there exists a deterministic algorithm that makes $\mathrm{poly}(dt/\varepsilon)$ nonadaptive queries to f, regardless of the number of vertices in the graph G. Crucially, and in contrast to the randomized algorithm, our algorithm does not simply output the average value of its queries. Indeed, Hoza, Pyne, and Vadhan [HPV21] showed that any deterministic algorithm of the latter form that works for graphs of unbounded size must have query complexity at least $\exp(\tilde{\Omega}(\log(t)\log(1/\varepsilon)))$.

*Harvard University, epyne@college.harvard.edu. Supported by NSF grant CCF-1763299.

†Harvard Univeristy, salil_vadhan@harvard.edu. Supported by NSF grant CCF-1763299 and a Simons Investigator Award.

Below we present these results in the language of pseudorandomness, as a separation between the query complexity of "deterministic samplers" and "deterministic averaging samplers" for the class of "permutation branching programs of unbounded width". Our separation is stronger than the prior separation of Pyne and Vadhan [PV21], and has a much simpler proof (not using spectral graph theory or the Impagliazzo–Nisan–Wigderson pseudorandom generator). On the other hand, the algorithm of Pyne and Vadhan is explicit and computable in small space, whereas ours is not explicit (unless we assume the existence of an optimal explicit pseudorandom generator for permutation branching programs of bounded width).

1.1 Ordered Branching Programs Motivated by the goal of derandomizing space-bounded computation, i.e. proving $\mathbf{BPL} = \mathbf{L}$, there has been extensive work on estimating the acceptance probabilities of *ordered branching programs*, which capture how a randomized small-space algorithm uses its random bits.

DEFINITION 1.1. *An **ordered branching program (OBP)** B of **length** n and **width** w computes a function $B : \{0,1\}^n \to \{0,1\}$. On an input $\sigma \in \{0,1\}^n$, the branching program computes as follows. It has $n+1$ layers $V_0, \ldots, V_n$, each with vertices labeled $\{1, \ldots, w\}$. It starts at a fixed start state $v_0 \in V_0$. Then for $r = 1, \ldots, n$, it reads the next symbol σ_r and updates its state according to a transition function $B_r : V_{r-1} \times \{0,1\} \to V_r$ by taking $v_r = B_r(v_{r-1}, \sigma_r)$. For $v \in V_i$ and $u \in V_j$ for $j > i$, we write $B[v, x] = u$ if the program transitions to state u starting from state v on input $x \in \{0,1\}^{j-i}$.*

Moreover, there is an accept state $v_{acc} \in V_n$. For $x \in \{0,1\}^n$, we define $B(x) = 1$ iff $B[v_0, x] = v_{acc}$. That is, B accepts the inputs x that lead it from the start state v_0 in the first layer to the accept state in the last layer v_{acc}.

Given a graph $G = (V, E)$ with w vertices, outdegree 2, $n \in \mathbb{N}$, and two vertices $u, v \in V$, we can define an associated ordered branching program $B^{G,u,v,n}$ which simulates walks of length n in G. Specifically, we set $v_0 = u$, $v_{acc} = v$, and $B_r^{G,u,v,n}(v_{r-1}, \sigma)$ to be the σ'th neighbor of vertex v_{r-1} for every $r = 1, \ldots, n$, $v_{r-1} \in V_{r-1} = \{1, \ldots, w\}$, and $\sigma \in \{0,1\}$. Then $B^{G,u,v,n}(x) = 1$ iff carrying out a walk of length n according to the edge labels in x leads from u to v in G. In particular, the probability that $B^{G,u,v,n}$ accepts a uniformly random input x is exactly the probability that a random walk of length n from u ends at v, exactly the computational problem we wish to solve. Compared to a general ordered branching program, $B^{G,u,v,n}$ has the same transition function at every layer.

The definition of branching programs naturally generalizes to alphabet sizes $d > 2$, allowing for simulation of random walks on graphs of degree d. We present our results just for the $d = 2$ case for simplicity, but they extend to larger d as well.

Motivated by the derandomization of space-bounded computation, there has been three decades of work on deterministically estimating the acceptance probability of ordered branching programs in small space (for instance [BNS92, Nis92, INW94, SZ99, BCG18, MRT19, Hoz21] and many others). In the case where we have white-box access to the branching program, the algorithm of Saks and Zhou [SZ99], as recently improved by Hoza [Hoz21], achieves space complexity $o(\log^{3/2} n)$ in the case $w = \mathrm{poly}(n)$ and $\varepsilon = 1/\mathrm{poly}(n)$.

Our focus, however, is on "black-box" derandomization, where we only have oracle access to the function $B : \{0,1\}^n \to \{0,1\}$. In this setting, we consider two questions:

1. Is there a small set $Q \subseteq \{0,1\}^n$ such that knowing the value of an arbitrary branching program B on all points of Q allows us to estimate $\Pr[B(U_n) = 1]$ up to additive error ε? We call the size of the smallest such Q the **query complexity** of two-sided derandomization.

2. If so, can we explicitly construct this set, and compute the estimate of the probability, in space $O(\log |Q|)$?

Such algorithms can be seen as deterministic analogues of *samplers* for a restricted class of functions. (See the survey [Gol11] for a general treatment of samplers.) These are defined as follows, following the language of Cheng and Hoza [CH20]:

DEFINITION 1.2. *Let $\mathcal{F}$ be a class of functions $f : \{0,1\}^n \to \{0,1\}$. A **deterministic ε-sampler** for $\mathcal{F}$ is an algorithm* SAMP *that, given oracle access to an arbitrary $f \in \mathcal{F}$, makes queries to f and outputs $\mathrm{SAMP}^f() \in \mathbb{R}$ such that*

$$|\,\mathrm{SAMP}^f() - \Pr[f(U_n) = 1]| \leq \varepsilon.$$

We say the the ***query complexity*** *S is the maximum over $f \in \mathcal{F}$ of the number of distinct queries made. We say the sampler is* ***explicit*** *if, given n, ε, and parameters defining the family $\mathcal{F}$ and $\varepsilon > 0$,* SAMP *can be computed by a uniform algorithm with space complexity $O(\log S)$.*

We say that SAMP *is* ***nonadaptive*** *if it makes nonadaptive queries to its oracle f. A special case of nonadaptive samplers are* ***deterministic averaging samplers*** *whereby the output of the sampler is the average of f over its set Q of oracle queries.*

It can be shown that deterministic averaging ε-samplers for a class $\mathcal{F}$ are equivalent to ε-pseudorandom generators (PRGs) for the class $\mathcal{F}$, where the seed length of the PRG is equal to the logarithm of the query complexity of the sampler.

We also consider deterministic hitters (i.e. hitting set generators), a weaker one-sided analogue of deterministic samplers.

DEFINITION 1.3. *Let $\mathcal{F}$ be a class of functions $f : \{0,1\}^n \to \{0,1\}$. A* ***deterministic ε-hitter*** *for $\mathcal{F}$ is a set $H \subseteq \{0,1\}^n$ such that for every $f \in \mathcal{F}$ where $\Pr_{x \leftarrow U_n}[f(x) = 1] > \varepsilon$, there is $x \in H$ such that $f(x) = 1$. We say that H is* ***explicit*** *if the elements of H can be enumerated in space $O(\log |H|)$.*

It can be shown that a deterministic ε-sampler SAMP for a class $\mathcal{F}$ implies a deterministic 2ε-hitter H for $\mathcal{F}$, where H is explicit if SAMP is. (Let H be the queries made by SAMP^{f_0} on the all-zeroes function f_0.)

A standard application of the probabilistic method (see Appendix A) shows that there do exist deterministic averaging samplers (i.e. PRGs) with polynomial query complexity for ordered branching programs of polynomial size. Specifically, for ordered branching programs of length n and width w, there exists a deterministic averaging ε-sampler with query complexity $\text{poly}(nw/\varepsilon)$, and this is optimal. However, constructing an *explicit* deterministic sampler with matching query complexity has been a longstanding open problem. The classic deterministic averaging samplers (i.e. PRGs) of Nisan [Nis92] and Impagliazzo–Nisan–Wigderson [INW94] have query complexity $\exp(\Theta(\log(n)\log(nw/\varepsilon)))$, and this has not been improved except where $w \leq 3$ or $\varepsilon = n^{-\omega(1)}$ or $w = n^{-\omega(1)}$.

However, in other models the picture is not so clear. There has been extensive work on *permutation branching programs*, which are a subset of ordered branching programs that posses additional structure.

DEFINITION 1.4. *An* ***(ordered) permutation branching program*** *of length n, and width w is an ordered branching program where for all $t \in [n]$ and $\sigma \in \{0,1\}$, $B_t(\cdot, \sigma)$ is a permutation on $[w]$.*

Similarly to how general ordered branching programs can simulate walks on general directed graphs of outdegree 2, permutation branching programs can simulate walks on 2-*regular* and *consistently labelled* directed graphs G. Indeed, for every such graph G, the ordered branching program $B^{G,u,v,n}$ defined earlier will be a permutation branching program.

There are several constructions of deterministic averaging samplers for permutation branching programs that beat the classic Nisan and INW analyses in the constant width regime [BRRY10,KNP11,De11,Ste12]. We consider the opposite regime, that of *unbounded width permutation branching programs with a single accept state.* This model was introduced by Hoza, Pyne and Vadhan [HPV21], and corresponds to derandomizing walks with *no* constraint on the size of the graph, merely requiring it to be consistently labeled.

In the case of *one-sided* derandomization, they established that deterministic hitters for bounded-width permutation branching programs are also deterministic hitters for unbounded width permutation BPs:

PROPOSITION 1.1. ([HPV21] PROPOSITION 7.1) *Given $n \in \mathbb{N}$ and $\delta > 0$, there is a value $w = O(n^2/\delta)$ such that if $H \subseteq \{0,1\}^n$ is a deterministic δ-hitter for permutation branching programs of length n and width w, then H is a deterministic 2δ-hitter for permutation branching programs of length n and unbounded width (with a single accept state).*

This result, together with an accompanying lower bound (see Claim 2.1), established that optimal hitters for bounded-width permutation branching programs imply optimal hitters for the unbounded width case, but says nothing about two-sided derandomization.

In the two-sided regime, they constructed an explicit deterministic averaging ε-sampler for unbounded width permutation branching programs with query complexity $\exp(\tilde{O}(\log(n)\log(1/\varepsilon)))$. Moreover, they showed an

unconditional lower bound on the query complexity of deterministic averaging samplers of $\exp(\tilde{\Omega}(\log(n)\log(1/\varepsilon)))$. They also showed a random set Q of points in $\{0,1\}^n$ fails to produce a deterministic averaging sampler whp unless $|Q| = \exp(\Omega(n))$, so in contrast to the case of general ordered branching programs, they obtained an explicit deterministic sampler with exponentially smaller query complexity than is obtained via the probabilistic method. However, there remained a gap between their upper bound of $\exp(\tilde{O}(\log(n)\log(1/\varepsilon)))$ and the lower bound of $(n/\varepsilon)^{\Omega(1)} = \exp(\Omega(\log(n/\varepsilon)))$ on the query complexity of general (possibly non-averaging) deterministic ε-samplers for this model.

Next, Pyne and Vadhan [PV21] constructed an explicit deterministic ε-sampler (in fact a weighted pseudorandom generator, which outputs a fixed linear combination of the queried points) for the model with query complexity $\exp(\tilde{O}(\log(n)\sqrt{\log(n/\varepsilon)} + \log(1/\varepsilon)))$. This deterministic sampler obtains smaller query complexity than every deterministic *averaging* sampler when $\varepsilon = n^{-\Omega(1)}$. Thus, they obtained an unconditional separation between the query complexity of deterministic averaging and general deterministic samplers for the model. However, this left a gap between the lower bound on query complexity of $\exp(\Omega(\log(n/\varepsilon)))$ and the upper bound of $\exp(\min\{\tilde{O}(\log(n)\sqrt{\log(n/\varepsilon)} + \log(1/\varepsilon)), \tilde{O}(\log(n)\log(1/\varepsilon))\})$ required for two-sided derandomization of the model. In addition, their construction was highly involved, and relied on sophisticated results in spectral graph theory [CKK+18, AKM+20], as well as the connection between the INW generator on permutation branching programs and the derandomized square of Rozenman and Vadhan [RV05].

1.2 Our Contribution Our main result is to resolve the query complexity of derandomizing unbounded-width permutation branching programs.

THEOREM 1.1. *There is a non-explicit deterministic nonadaptive ε-sampler for permutation branching programs of length n and unbounded width (with a single accept state) that has query complexity* $\mathrm{poly}(n/\varepsilon)$.

Thus, we establish the optimal query complexity for deterministic algorithms estimating the fraction of fixed-length walks from u that end at v for arbitrary u, v in an arbitrarily sized consistently-labeled graph. Furthermore, we obtain a deterministic sampler that achieves query complexity $\mathrm{poly}(n)$ for $\varepsilon = 1/\mathrm{poly}(n)$, whereas every deterministic *averaging* sampler with these parameters has query complexity $\exp(\Omega(\log^2 n))$ [HPV21]. This gives a simple unconditional separation between averaging samplers and general nonadaptive samplers in the no-randomness regime with respect to a natural computational model.

We prove this result via a reduction from the unbounded-width case to the bounded-width case. We show that an optimal family of samplers for bounded-width permutation branching programs can be used to construct an optimal sampler for unbounded-width ones. Since optimal non-explicit samplers for the bounded-width case exist via the probabilistic method (See Appendix A), this immediately establishes our result.

We now state the reduction. For the remainder of the paper, rather than working with branching programs with a single accept state $v_{acc} \in V_n$, we allow branching programs to have a set $V_{acc} \subseteq V_n$ of accept vertices, where $B(x) = 1$ if $B[v_0, x] \in V_{acc}$. We let $a = |V_{acc}|$ be the number of accept vertices.

THEOREM 1.2. *Let $\mathcal{S} = \{\mathrm{SAMP}_{n,w,\varepsilon}\}$ be a family of deterministic ε-samplers $\mathrm{SAMP}_{n,w,\varepsilon}$ for permutation branching programs of length n and width w such that $\mathrm{SAMP}_{n,w,\varepsilon}$ has query complexity $\mathrm{poly}(nw/\varepsilon)$. From $\mathcal{S}$, we can construct a deterministic ε-sampler $\mathrm{SAMP}'_{n,a,\varepsilon}$ for permutation branching programs of length n and unbounded width with a accept vertices that has query complexity $\mathrm{poly}(na/\varepsilon)$. Moreover, if the samplers $\mathrm{SAMP}_{n,w,\varepsilon}$ in $\mathcal{S}$ are explicit then so is $\mathrm{SAMP}'_{n,a,\varepsilon}$, and if the samplers $\mathrm{SAMP}_{n,w,\varepsilon}$ in $\mathcal{S}$ are non-adaptive then so is $\mathrm{SAMP}'_{n,a,\varepsilon}$.*

Note that this reduction preserves explicitness and (non-)adaptiveness, so optimal *explicit* deterministic samplers (for instance, optimal explicit PRGs) for the bounded width case imply explicit deterministic samplers for unbounded-width permutation BPs that have optimal space complexity $O(\log(na/\varepsilon))$. Put differently, optimal black-box two-sided derandomization of permutation branching programs with k vertices in *all* layers is no harder than derandomization of permutation branching programs with k accept vertices in the *final* layer and no restriction on width.

We summarize the current known derandomizations for unbounded-width permutation branching programs. An entry of "Conditional" means an optimal explicit construction for permutation branching programs of bounded width would imply an explicit construction.

Object	Query Complexity	Explicit?	Reference
Det. Hitter	$\exp(O(\log(na/\varepsilon)))$	Conditional	[HPV21]
PRG	$\exp(\widetilde{O}(\log(n)\log(a/\varepsilon)))$	Yes	[HPV21]
WPRG	$\exp(\tilde{O}(\log(n)\sqrt{\log(na/\varepsilon)}+\log(a/\varepsilon)))$	Yes	[PV21]
Det. Sampler	$\exp(O(\log(na/\varepsilon)))$	Conditional	This work.

1.3 Proof Overview Our construction is very simple, and in contrast to prior work on the model [HPV21, PV21] the proof uses neither special properties of the INW PRG [INW94], nor results from spectral graph theory.

The key idea behind Theorem 1.2 is that compositions of layers of permutation branching programs themselves define permutations. More concretely, fixing a permutation branching program B, an input $x \in \{0,1\}^{n-i}$ and an accept state $v_f \in V_{acc}$ in the final layer, there is at most *one* state v in layer i such that $B[v,x] = v_f$. With this observation, we can use a sparse set of strings $T \subseteq \{0,1\}^n$ to restrict the branching program. For every layer V_i, we remove all states $v \in V_i$ where for all $x \in T$, $B[v, x_{1..n-i}] \notin V_{acc}$. Since each element of T can cause at most $a = |V_{acc}|$ vertices in every layer to be included in the restricted program, which we denote B_T, we limit the width of B_T to at most $|T| \cdot a$. Furthermore, by adding $n|T|$ dummy states we have that B_T can be computed by a permutation branching program.

We next show that there is a sparse set T such that the restriction induced by T is a good approximation of the original program. We take T to be the set of points queried by a deterministic hitter for permutation branching programs of unbounded width. To obtain T from our hypothesis, we use a result of HPV, which proves that samplers for the bounded-width case are *hitters* for the unbounded-width case (Proposition 1.1). We show that the states not included in the restricted program are unimportant, in that removing all of them simultaneously only changes the acceptance probability $\Pr[B(U_n) = 1]$ by at most $\varepsilon/2$.

Then to estimate the acceptance probability of the restricted program, we use a *second* sampler that is good against branching programs of width $|T| \cdot a$, and return the output of the sampler on the restricted program B_T. Unfortunately, even if T is explicit it is unclear how to learn B_T given only oracle access to the original program B. To avoid having to do so, we construct a way to compute $B_T(x)$ for arbitrary $x \in \{0,1\}^n$ given only oracle access to B; we apply this procedure whenever the second sampler queries B_T. Thus we obtain a good estimate of $\Pr[B_T(U_n) = 1]$, which is itself a good estimate of $\Pr[B(U_n) = 1]$, so we conclude.

1.4 Organization In Section 2 we recall that a deterministic ε-sampler for a class of functions implies a deterministic 2ε-hitter for that class, and use this to establish the optimal space and query complexity of samplers for unbounded-width permutation BPs. In Section 3 we prove that hitters can be used to restrict unbounded-width permutation BPs to bounded width, and that this restriction can be done in a black-box manner, and this restriction is a good approximation of the original program. Then in Section 4, we combine these two results and prove the main theorem.

2 Samplers Imply Hitters

We first recall that an arbitrary deterministic ε-sampler for a model that includes the all-zeroes function (which includes functions computed by permutation branching programs of width at least 2) induces a deterministic 2ε-hitter for the model. We use this to establish tight lower bounds on space and query complexity for deterministic samplers for permutation branching programs of unbounded width.

PROPOSITION 2.1. *Let $\mathcal{F}$ be a class of functions $f : \{0,1\}^n \to \{0,1\}$ that includes the constant function $f_0(x) = 0$. Then if $\mathcal{A}$ is an deterministic ε-sampler for $\mathcal{F}$, the set of queries Q made by $\mathcal{A}$ on the all-zeroes function f_0 is a deterministic 2ε-hitter for $\mathcal{F}$, and moreover Q is explicit if $\mathcal{A}$ is.*

Proof. Assuming for contradiction this is not the case, there is $f \in \mathcal{F}$ such that $f(Q) = 0$ but $\Pr[f(U_n) = 1] > 2\varepsilon$. But since the sampler must output a single estimate $\mathrm{SAMP}^f() = \mathrm{SAMP}^{f_0}()$ for f and the all-0 program f_0 (since the value of both functions on all queried points are identical), it must fail to estimate the acceptance probability of one to within ε, a contradiction. ∎

We then recall the optimal seed length for deterministic hitters for permutation branching programs of unbounded width.

CLAIM 2.1. ([HPV21] CLAIM 7.3) *Given $n, a \in \mathbb{N}$ and $\varepsilon \in (1/4, 0)$ such that $1/2 > \varepsilon/a \geq 2^{-n}$, let $H \subseteq \{0,1\}^n$ be a deterministic ε-hitter for permutation branching programs of unbounded width and at most a accept vertices. Then $|H| = (na/\varepsilon)^{\Omega(1)} = \exp(\Omega(\log(na/\varepsilon)))$.*

We recall the proof in Appendix A. From this, we can derive the optimal space and query complexity of a sampler.

COROLLARY 2.1. *Given $1/8 > \varepsilon \geq 2^{-n}$ and $n, a \in \mathbb{N}$, let $\mathcal{A}$ be a deterministic ε-sampler for permutation branching programs of length n and unbounded width with at most a accept vertices, where $\varepsilon/a \geq 2^{-n}$. Then $\mathcal{A}$ has query complexity $(na/\varepsilon)^{\Omega(1)}$. Moreover, if $\mathcal{A}$ is explicit it has space complexity $s = \Omega(\log(na/\varepsilon))$.*

Proof. We apply Proposition 2.1 to $\mathcal{A}$ and obtain a deterministic 2ε-hitter $Q \subseteq \{0,1\}^n$ for permutation branching programs of length n with at most a accept vertices. By Claim 2.1 we obtain $|Q| = (na/\varepsilon)^{\Omega(1)}$ which establishes the claimed bound on query complexity. Furthermore if $\mathcal{A}$ is explicit and has space complexity s, it must run in time $2^{O(s)}$ since it is required to halt, and thus its query complexity $|Q|$ is at most $2^{O(s)}$. Combined with the lower bound on $|Q|$, we have that $s = \Omega(na/\varepsilon)$. ∎

3 Hitters Induce Bounded-Width Approximators

We next show that, given a permutation branching program B and a sufficiently good deterministic hitter, the set of states $v \in V_i$ for which there is a hitter output whose prefix reaches an accept state starting from v forms an approximator of the original program. To show this, we define the program "cut out" by a deterministic hitter. Then we show that we can evaluate this approximator program on any input given only oracle access to B.

DEFINITION 3.1. *Given a set $H \subseteq \{0,1\}^n$ and a permutation branching program B of length n with a accept vertices V_{acc} and vertices $V_0, \ldots, V_n$, let the hit states in layer i be*

$$K_i = \{v \in V_i : \exists x \in H \text{ s.t. } B[v, x_{1..n-i}] \in V_a\}.$$

*WLOG pad all such sets to have size $K = \max_{i=0}^{n} |K_i|$, where all transitions from padding states in K_i do not lead to K_{i+1}. The **induced hit program** B_H is the length n permutation branching program with states in layer i given by $\{K_i\} \cup \{\{0, \ldots, n\} \times [K]\}$, where we identify states in $\{0, \ldots, n\} \times [K]$ by (j, v). For $v \in K_i$, define the transition function*

$$(B_H)_i(v, b) = \begin{cases} B_i(v, b) & B_i(v, b) \in K_{i+1} \\ (i, v) & \text{otherwise.} \end{cases}$$

Then greedily define transitions for $\{(i, v) : v \in [K]\}$ to maintain the permutation property. For all states (j, v) for $j \neq i$, let $(B_H)_i((j, v), b) = (j, v)$.

We next show that the width of the induced hit program is bounded by the size of the domain of H (and thus its seed length). This will allow us to derandomize the induced hit program as a standard bounded-width permutation branching program.

LEMMA 3.1. *Given $H \subseteq \{0,1\}^n$ and a permutation branching program B of length n with a accept vertices, the width of the induced hit program B_H is at most $|H| \cdot (n+2) \cdot a$.*

To prove this, we require a proposition essentially showing that composing multiple layers of a permutation branching program produces a permutation branching program of higher degree. This is the only element of the proof that uses the fact that B is a permutation, rather than regular, branching program.

PROPOSITION 3.1. *For every permutation branching program B, for every distinct $v, v' \in V_i$ and $\sigma \in \{0,1\}^k$ so that $i + k \leq n$, $B[v, \sigma] \neq B[v', \sigma]$.*

Proof. We prove this by induction on k. The base case of $k = 0$ is vacuously true. Assuming it holds for k, let B be an arbitrary permutation branching program and $v, v' \in V_i$ arbitrary distinct states. Let $\sigma \in \{0,1\}^{k+1}$ be arbitrary. From the permutation property it must be the case that $u_1 = B[v, \sigma_1] \neq B[v', \sigma_1] = u_2$, so $B[v, \sigma] = B[u_1, \sigma_{2..k}] \neq B[u_2, \sigma_{2..k}] = B[v', \sigma]$ where the inequality follows from the inductive step, and since σ, B and v, v' were arbitrary we conclude. ∎

We can then prove Lemma 3.1.

Proof. It suffices to show that the number of included states of the original program satisfies $|K_i| \leq |H| \cdot a$ for all $i \in \{0, \ldots, n-1\}$, since the width of B_H is bounded by $(n+2) \cdot K = (n+2) \cdot \max_{i=0}^{n} |K_i|$. For every fixed accept state $u \in V_{acc}$, there are at most $|H|$ states $v \in V_i$ such that there exists $x \in H$ such that $B[v, x_{1..n-i}] = u$ by Proposition 3.1, so we conclude via a union bound. ∎

The induced hit program is well defined for every $H \subseteq \{0,1\}^n$. However, we wish to show that the program induced by a sufficiently good deterministic hitter is a close approximation of the original permutation branching program.

LEMMA 3.2. *Let $H \subseteq \{0,1\}^n$ be a deterministic δ/na-hitter for permutation branching programs of length n and unbounded width with a single accept state. Then for every permutation branching program B of length n and unbounded width with at most a accept vertices, the induced hit program B_H satisfies*

$$|\Pr[B_H(U_n) = 1] - \Pr[B(U_n) = 1]| \leq \delta.$$

Proof. Let V_{neg} be the set of states of B not included in the induced hit program B_H. For every $v \in V_{\text{neg}}$ in layer $n-k$, using the fact that H is a deterministic δ/na-hitter for branching programs of length n, and hence for length $n-k \leq n$ since branching programs can ignore bits, we obtain

$$\Pr[B[v, U_k] \in V_{acc}] = \sum_{u \in V_{acc}} \Pr[B[v, U_k] = u] \leq a \cdot \frac{\delta}{na}.$$

If $v_0 \in V_{\text{neg}}$ then B_H is the all zeroes program and the above implies $\Pr[B(U_n) = 1] \leq \delta$ so we are done. Thus assume that $v_0 \notin V_{\text{neg}}$. For arbitrary $x \in \{0,1\}^n$ such that $B(x) \neq B_H(x)$, it must be the case that $B(x) = 1$ while $B_H(x) = 0$, i.e. B passes through some element of V_{neg} in its computation on x and $B[v_0, x] \in V_{acc}$. Therefore,

$$\begin{aligned}
|\Pr[B_H(U_n) = 1] - \Pr[B(U_n) = 1]| &\leq \Pr_{x \leftarrow U_n}[B(x) \neq B_H(x)] \\
&= \Pr_{x \leftarrow U_n}\left[(B[v_0, x] \in V_{acc}) \bigwedge \left(\bigvee_{i=1}^{n} B[v_0, x_{1..i}] \in V_{\text{neg}}\right)\right] \\
&\leq \sum_{i=1}^{n} \sum_{v \in V_i \cap V_{\text{neg}}} \Pr_{x \leftarrow U_n}\left[(B[v_0, x_{1..i}] = v) \wedge (B[v, x_{i+1..n}] \in V_{acc})\right] \\
&= \sum_{i=1}^{n} \sum_{v \in V_{\text{neg}} \cap V_i} \Pr[B[v_0, U_i] = v] \cdot \Pr[B[v, U_{n-i}] \in V_{acc}] \\
&\leq \sum_{i=1}^{n} \left(\sum_{v \in V_{\text{neg}} \cap V_i} \Pr[B[v_0, U_i] = v]\right) \cdot \frac{\delta}{n} \\
&\leq \sum_{i=1}^{n} 1 \cdot \frac{\delta}{n} = \delta.
\end{aligned}$$

∎

Finally, we show that given H, we can evaluate an arbitrary input on the induced hit program.

LEMMA 3.3. *Given a permutation branching program B of length n and a set $H \subseteq \{0,1\}^n$, for every $x \in \{0,1\}^n$ we have*

$$B_H(x) = \bigwedge_{i=0}^{n} \left(\bigvee_{y \in H} B(x_{1..i} || y_{i+1..n})\right)$$

where $||$ denotes string concatenation and $x_{1..0}$ and $y_{n+1..n}$ are the empty string.

Proof. Fix arbitrary $x \in \{0,1\}^n$ and let $v_i = B[v_0, x_{1..i}]$ for all $i \in \{0,\dots,n\}$. First suppose the RHS evaluates to 1. For $i \in \{0,\dots,n\}$ we have $1 = \bigvee_{y\in H} B(x_{1..i}||y_{i+1..n})$, so there is some $y \in H$ such that $B[v_i, y_{i+1..n}] \in V_{acc}$, which is precisely the condition for including v_i in the induced hit program B_H, and this holds for every i, so $B_H(x) = 1$. Now suppose the RHS evaluates to 0. Fixing the least i such that $\bigvee_{y\in H} B(x_{1..i}||y_{i+1..n}) = 0$, we have that v_i is not included in B_H and so $B[v_{i-1}, x_i] = u \notin K_i$. Since u is always subsequently wired to itself and marked as reject in the final layer we have $B_H(x) = 0$. ∎

Note that this implies that we can evaluate $B_H(x)$ given x and oracle access to B, and this procedure is explicit if H is.

4 Putting it All Together

We can now go from samplers for bounded-width permutation branching programs to samplers for unbounded-width permutation branching programs. We follow the outline in the proof sketch in Section 1.4. First, we use an optimal sampler for the bounded-width case to generate an optimal deterministic hitter for the bounded-width case, which implies an optimal deterministic hitter H for the unbounded-width case. Then we use a second sampler and evaluate it on the induced hit program B_H, and by choosing the parameters for the second sampler appropriately obtain an accurate estimate of $\Pr[B_H(U_n) = 1]$ and thus $\Pr[B(U_n) = 1]$.

THEOREM 4.1. *Let $\mathcal{S} = \{\mathrm{SAMP}_{n,w,\varepsilon}\}$ be a family of deterministic ε-samplers $\mathrm{SAMP}_{n,w,\varepsilon}$ for permutation branching programs of length n and width w such that $\mathrm{SAMP}_{n,w,\varepsilon}$ has query complexity $\mathrm{poly}(nw/\varepsilon)$. From $\mathcal{S}$, we can construct a deterministic ε-sampler $\mathrm{SAMP}'_{n,a,\varepsilon}$ for permutation branching programs of length n and unbounded width with a accept vertices that has query complexity $\mathrm{poly}(na/\varepsilon)$. Moreover, if the samplers $\mathrm{SAMP}_{n,w,\varepsilon}$ in $\mathcal{S}$ are explicit then so is $\mathrm{SAMP}'_{n,a,\varepsilon}$, and if the samplers $\mathrm{SAMP}_{n,w,\varepsilon}$ in $\mathcal{S}$ are non-adaptive then so is $\mathrm{SAMP}'_{n,a,\varepsilon}$.*

Proof. By assumption, we have a deterministic ε-hitter for permutation branching programs of length n and width $w = O(n^3 a/\varepsilon)$, where this w is that obtained from Proposition 1.1 with $n = n$ and $\delta = \varepsilon/(8na)$. Applying Proposition 2.1, we obtain a deterministic $\varepsilon/(4na)$-hitter $H \subseteq \{0,1\}^n$ for permutation BPs of length n and width $w = O(n^3 a/\varepsilon)$, with $|H| = (na/\varepsilon)^{O(1)} = \exp(O(\log(na/\varepsilon)))$. Applying Proposition 1.1, we have that H is a deterministic $\varepsilon/2na$-hitter for permutation branching programs of unbounded width with a single accept state.

Now let $\mathcal{A} = \mathrm{SAMP}_{n,w',\varepsilon/2}$ be an $\varepsilon/2$-sampler for permutation branching programs of length n and width $w' = |H| \cdot (n+2) \cdot a = \mathrm{poly}(na/\varepsilon)$. By assumption, $\mathcal{A}$ has query complexity $\mathrm{poly}(na/\varepsilon)$.

Finally, given an arbitrary permutation branching program B with at most a accept vertices, define $\mathrm{SAMP}'^{B}_{n,a,\varepsilon}() = \mathcal{A}^{B_H}()$, where whenever $\mathcal{A}$ queries the value of B_H on $x \in \{0,1\}^n$, we apply Lemma 3.3, so $\mathrm{SAMP}'^{B}_{n,a,\varepsilon}$ queries

$$\{(x_{1..i}||y_{1..n-i}) : i \in \{0,\dots,n\}, y \in H\}.$$

Thus $\mathrm{SAMP}'_{n,a,\varepsilon}$ has query complexity that equals the query complexity of $\mathcal{A}$ times $(n+1)\cdot|H|$, for a total query complexity of $\mathrm{poly}(na/\varepsilon)$. If $\mathcal{S} = \{\mathrm{SAMP}_{n,w,\varepsilon}\}$ is non-adaptive (i.e. the queries made by $\mathcal{A}$ do not depend on B_H and thus B) then $\mathrm{SAMP}'_{n,a,\varepsilon}$ is. Finally, if $\mathcal{S} = \{\mathrm{SAMP}_{n,w,\varepsilon}\}$ is explicit then $\mathrm{SAMP}'_{n,a,\varepsilon}$ is by definition.

Finally, we prove $\mathrm{SAMP}'_{n,a,\varepsilon}$ is an ε-sampler. We have that B_H is a permutation branching program of length n and width at most $|H| \cdot (n+2) \cdot a$ by Lemma 3.1, so by our choice of parameters $\mathcal{A}$ is an $\varepsilon/2$-sampler for B_H, i.e. $|\mathcal{A}^{B_H}() - \Pr[B_H(U_n) = 1]| \le \varepsilon/2$. Then we conclude by the triangle inequality:

$$\begin{aligned}
&|\,\mathrm{SAMP}'^{B}_{n,a,\varepsilon}() - \Pr[B(U_n) = 1]| \\
&\quad = |\mathcal{A}^{B_H}() - \Pr[B(U_n) = 1]| && \text{(Lemma 3.3)} \\
&\quad \le |\mathcal{A}^{B_H}() - \Pr[B_H(U_n) = 1]| + |\Pr[B_H(U_n) = 1] - \Pr[B(U_n) = 1]| \\
&\quad \le |\mathcal{A}^{B_H}() - \Pr[B_H(U_n) = 1]| + \varepsilon/2 && \text{(Lemma 3.2)} \\
&\quad \le \varepsilon
\end{aligned}$$

and since B was arbitrary we obtain the result. ∎

References

[AKM+20] AmirMahdi Ahmadinejad, Jonathan A. Kelner, Jack Murtagh, John Peebles, Aaron Sidford, and Salil P. Vadhan. High-precision estimation of random walks in small space. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 1295–1306. IEEE, 2020.

[BCG18] Mark Braverman, Gil Cohen, and Sumegha Garg. Hitting sets with near-optimal error for read-once branching programs. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 353–362. ACM, 2018.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992. Twenty-first Symposium on the Theory of Computing (Seattle, WA, 1989).

[BRRY10] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. In *FOCS*, pages 40–47. IEEE Computer Society, 2010.

[CH20] Kuan Cheng and William M. Hoza. Hitting sets give two-sided derandomization of small space. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 10:1–10:25. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[CKK+18] Michael B Cohen, Jonathan Kelner, Rasmus Kyng, John Peebles, Richard Peng, Anup B Rao, and Aaron Sidford. Solving directed laplacian systems in nearly-linear time through sparse lu factorizations. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 898–909. IEEE, 2018.

[De11] Anindya De. Pseudorandomness for permutation and regular branching programs. In *IEEE Conference on Computational Complexity*, pages 221–231. IEEE Computer Society, 2011.

[Gol11] Oded Goldreich. A sample of samplers: A computational perspective on sampling. In Oded Goldreich, editor, *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*, volume 6650 of *Lecture Notes in Computer Science*, pages 302–332. Springer, 2011.

[Hoz21] William M. Hoza. Better pseudodistributions and derandomization for space-bounded computation. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 28:1–28:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[HPV21] William M. Hoza, Edward Pyne, and Salil P. Vadhan. Pseudorandom generators for unbounded-width permutation branching programs. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 7:1–7:20. 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 the Theory of Computing*, pages 356–364, Montréal, Québec, Canada, 23–25 May 1994.

[KNP11] Michal Koucký, Prajakta Nimbhorkar, and Pavel Pudlák. Pseudorandom generators for group products: extended abstract. In Lance Fortnow and Salil P. Vadhan, editors, *STOC*, pages 263–272. ACM, 2011.

[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. ACM, 2019.

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

[PV21] Edward Pyne and Salil Vadhan. Pseudodistributions That Beat All Pseudorandom Generators (Extended Abstract). In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[RV05] Eyal Rozenman and Salil Vadhan. Derandomized squaring of graphs. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM '05)*, number 3624 in Lecture Notes in Computer Science, pages 436–447, Berkeley, CA, August 2005. Springer.

[Ste12] Thomas Steinke. Pseudorandomness for permutation branching programs without the group theory. Technical Report TR12-083, Electronic Colloquium on Computational Complexity (ECCC), July 2012.

[SZ99] Michael Saks and Shiyu Zhou. $\mathrm{BP_HSPACE}(S) \subseteq \mathrm{DSPACE}(S^{3/2})$. *Journal of Computer and System Sciences*, 58(2):376–403, 1999.

A The Optimal Seed Length For Deterministic Hitters

We prove the optimal seed length for deterministic hitters for unbounded-width permutation branching programs. The proof is identical to that given in Hoza et al. [HPV21] except we explicitly consider the number of accept vertices in the final layer.

CLAIM 2.1. ([HPV21] CLAIM 7.3) *Given $n, a \in \mathbb{N}$ and $\varepsilon \in (1/4, 0)$ such that $1/2 > \varepsilon/a \geq 2^{-n}$, let $H \subseteq \{0,1\}^n$ be a deterministic ε-hitter for permutation branching programs of unbounded width and at most a accept vertices. Then $|H| = (na/\varepsilon)^{\Omega(1)} = \exp(\Omega(\log(na/\varepsilon)))$.*

Proof. We prove the bounds on n and ε/a separately:

1. If $|H| \leq n-1$, there is some nonzero vector $z \in \mathbb{F}_2^n$ such that for every x,
$$\bigoplus_{i=1}^{n} z_i \cdot H(x)_i = 0.$$
The function $B(x) = \mathbb{I}[\oplus_{i=1}^n z_i \cdot x_i = 1]$ can be computed by a permutation branching program (of width 2) with a single accept state, and $\Pr[B(U_n) = 1] \geq 1/2$, but $B(x) = 0$ for all $x \in H$, a contradiction.

2. If $|H| \leq a/4\varepsilon$, there are at most $a/4\varepsilon$ distinct length $l = \lceil \log(\varepsilon/a) \rceil - 1$ prefixes of elements of H. Letting B be a permutation branching program that reaches the ith state in the final layer on input x if
$$\sum_{j=1}^{l} 2^{j-1} \cdot x_j = i,$$
we have that the final state of $B(U_n)$ is distributed uniformly over $\{0, \ldots, 2^l - 1\}$. Then choosing a distinct states in the final layer that are not reached by every output of H (where we use that $2^l - a/4\varepsilon \geq a$) and marking them as accept, we obtain that $\Pr[B(U_n) = 1] \geq a(2\varepsilon/a) = 2\varepsilon$ whereas $B(x) = 0$ for all $x \in H$, a contradiction.

□

We also note the optimal seed length for hitters for general ordered branching programs (which include permutation branching programs of bounded width). Together with the result converting deterministic samplers into hitters (Proposition 2.1), this establishes the query complexity of two-sided derandomization for the model is $(nw/\varepsilon)^{\Omega(1)}$.

CLAIM A.1. *Given $n, w \in \mathbb{N}$ and $1/2 > \varepsilon \geq 2^{-n}$, let $H \subseteq \{0,1\}^n$ be a deterministic ε-hitter for ordered branching programs of width w and length n. Then $|H| = (nw/\varepsilon)^{\Omega(1)}$.*

Proof. We prove the bounds on w and $1/\varepsilon$ separately. The bound $|H| = n^{\Omega(1)}$ is identical to the same bound in the prior proof.

1. If $|H| \leq 1/2\varepsilon$, there is some string σ of length $l = \lceil \log(1/\varepsilon) \rceil - 1$ such that $x_{1..l} \neq \sigma$ for every $x \in H$. Then let B be a width-2 ordered branching program where
$$B(x) = 1 \iff x_{1..l} = \sigma$$
(note that we use the program is non-regular to keep the width at 2), we have $\Pr[B(U_n) = 1] \geq 2\varepsilon$ but $B(x) = 0$ for all $x \in H$, a contradiction.

2. If $|H| \leq w/2$, there are at most $w/2$ distinct length $l = \lceil \log(w) \rceil$ prefixes of elements of H. Letting B be a branching program that reaches the ith state in the final layer on input x if
$$\sum_{j=1}^{l} 2^{j-1} \cdot x_j = i,$$
we have that the final state of $B(U_n)$ is distributed uniformly over $\{0, \ldots, 2^l - 1\}$. Then choosing $w/2$ distinct states in the final layer that are not reached by every $x \in H$ and marking them as accept states, we obtain that $\Pr[B(U_n) = 1] \geq 1/2$ whereas $B(x) = 0$ for all $x \in H$, a contradiction.

□

We recall the existence of non-explicit deterministic averaging samplers for ordered branching programs of bounded width. Since ordered branching programs are a superset of permutation branching programs, this likewise implies the existence of optimal non-explicit averaging samplers for bounded-width permutation branching programs. Together with Theorem 1.2 this implies Theorem 1.1.

PROPOSITION A.1. *There is a family* $\mathcal{S} = \{\mathrm{SAMP}_{n,w,\varepsilon}\}$ *of (non-explicit) deterministic* ε*-samplers* $\mathrm{SAMP}_{n,w,\varepsilon}$ *for ordered branching programs of length* n *and width* w *such that* $\mathrm{SAMP}_{n,w,\varepsilon}$ *has query complexity* $\mathrm{poly}(nw/\varepsilon)$.

Proof. First note that a branching program of length n and width w has a description using $k = \mathrm{poly}(nw)$ bits, so there are at most 2^k such programs. Now fix $n, w \in \mathbb{N}$ and $\varepsilon > 0$ and consider a random set Q of size $|Q| = 3k/\varepsilon$ where Q_i is a random independently chosen element of $\{0,1\}^n$ for all i. Fixing an arbitrary ordered branching program B of length n and width w, let $\mu = \Pr[B(U_n) = 1]$ be its accept probability and WLOG assume $\mu \geq 1/2$, since an additive estimate of the accept probability implies an equivalent estimate of the reject probability. Then let $Y_i = B(Q_i)$ be the random variable that is 1 when B accepts on Q_i. We have $Y_i \in [0,1]$ and they are independent for all i. Applying a Chernoff bound over the randomness of the strings in Q, we obtain for all $\delta \in (0,1)$

$$\Pr\left[\left|\frac{1}{|Q|}\sum_i Y_i - \mu\right| \geq \delta\mu\right] \leq 2\exp(-|Q|^2\delta^2\mu/3).$$

Then choosing $\delta = \varepsilon$ we obtain

$$\Pr\left[\left|\frac{1}{|Q|}\sum_i Y_i - \mu\right| \geq \varepsilon\right] \leq 2\exp(-3k^2).$$

By a union bound, the probability that a random set Q of the chosen size fails to be a deterministic averaging sampler for at least one of the 2^k length n, width w branching programs is at most $2\exp(k - 3k^2) < 1$. Thus there exists some Q_{good} that is good for all such programs, so Q_{good} generates a deterministic averaging ε-sampler $\mathrm{SAMP}_{n,w,\varepsilon}$ for ordered branching programs of length n and width w. By construction, $|Q_{good}| = \mathrm{poly}(nw/\varepsilon)$. □