Electronic Colloquium on Computational Complexity, Report No. 123 (2012)

Better Pseudorandom Generators from Milder Pseudorandom Restrictions.

Parikshit Gopalan
MSR-SVC

Raghu Meka
IAS Princeton[*]

Omer Reingold
MSR-SVC

Luca Trevisan
Stanford University

Salil Vadhan[†]
Harvard University

Abstract

We present an iterative approach to constructing pseudorandom generators, based on the repeated application of mild pseudorandom restrictions. We use this template to construct pseudorandom generators for combinatorial rectangles and read-once CNFs and a hitting set generator for width-3 branching programs, all of which achieve near-optimal seed-length even in the low-error regime: We get seed-length $\tilde{O}(\log(n/\varepsilon))$ for error ε. Previously, only constructions with seed-length $O(\log^{3/2} n)$ or $O(\log^2 n)$ were known for these classes with error $\varepsilon = 1/\mathrm{poly}(n)$.

The (pseudo)random restrictions we use are milder than those typically used for proving circuit lower bounds in that we only set a constant fraction of the bits at a time. While such restrictions do not simplify the functions drastically, we show that they can be derandomized using small-bias spaces.

[*] Supported in part by NSF grant DMS-0835373. Work done in part while the author was an intern at Microsoft Research Silicon Valley.

[†] School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138. salil@seas.harvard.edu. Supported in part by NSF grant CCF-1116616. Work done in part while on leave as a Visiting Researcher at Microsoft Research Silicon Valley and a Visiting Scholar at Stanford University.

ISSN 1433-8092

1 Introduction

1.1 Pseudorandom Generators

The theory of pseudorandomness has given compelling evidence that very strong pseudorandom generators exist. For example, assuming that there are computational problems solvable in exponential time that require exponential-sized circuits, Impagliazzo and Wigderson [IW97] have shown that for every n, c and $\varepsilon > 0$, there exist efficient pseudorandom generators (PRGs) mapping a random seed of length $O(\log(n^c/\varepsilon))$ to n pseudorandom bits that cannot be distinguished from n uniformly random bits with probability more than ε, by any Boolean circuit of size n^c. These PRGs, which fool arbitrary efficient computations (represented by polynomial-sized Boolean circuits), have remarkable consequences for derandomization: every randomized algorithm can be made deterministic with only a polynomial slowdown, and thus $\mathrm{P} = \mathrm{BPP}$.

These results, however, remain conditional on a circuit complexity assumption whose proof seems far off at present. Since PRGs that fool a class of Boolean circuits also imply lower bounds for that class, we cannot hope to remove the assumption. Thus unconditional generators are only possible for restricted models of computation for which we have lower bounds.

Bounded-depth circuits and *bounded-space algorithms* are two models of computations for which we know how to construct PRGs with $O(\log^{O(1)}(n/\varepsilon))$ seed length [Nis91, Nis92]. Known PRG constructions for these classes have found several striking applications including the design of streaming algorithms [Ind06], algorithmic derandomization [Siv02], randomness extractors [Tre01], hashing [CRSW11], hardness amplification [HVV06], almost k-wise independent permutations [KNR05], and cryptographic PRGs [HHR06]. Arguably, constructing PRGs with the optimal $O(\log(n/\varepsilon))$ seed length for these classes are two of the outstanding open problems in derandomization.

Nisan [Nis92] devised a PRG of seed length $O(\log^2 n)$ that fools polynomial-width branching programs, the non-uniform model of computation that captures logspace randomized algorithms: a space-s algorithm is modeled by a branching program[1] of width 2^s. Nisan's generator has been used by Saks and Zhou [SZ99] to prove that every randomized logspace algorithms can be simulated in space $O(\log^{3/2} n)$, Nisan's generator remains the best known generator for polynomial-width branching programs (and logspace randomized algorithms) and, despite much progress in this area [INW94, NZ96, RR99, Rei08, RTV06, BRRY10, BV10b, KNP11, De11], there are very few cases where we can improve on Nisan's twenty year old bound of $O(\log^2 n)$ [Nis92]. For constant-width *regular* branching programs, Braverman et al. [BRRY10] have given a pseudorandom generator with seed length $\tilde{O}((\log n) \cdot (\log(1/\varepsilon)))$, which is $\tilde{O}(\log n)$ for $\varepsilon = 1/\mathrm{polylog}(n)$, but is no better than Nisan's generator when $\varepsilon = 1/\mathrm{poly}(n)$. Only for constant-width *permutation* branching programs and for width-2 branching programs has seed length $O(\log(n/\varepsilon))$ been achieved, by Koucký, Nimbhorkar, Pudlák [KNP11] and Saks and Zuckerman [SZ95], respectively. Remarkably, even for width-3 branching programs we do not know of any efficiently computable PRG with seed length $o(\log^2 n)$. Recently, Sima and Zak [SZ11] have constructed *hitting set generators* (HSGs, which are a weaker form of pseudorandom generators) for width-3 branching programs with optimal seed length $O(\log n)$, for a large error parameter $\varepsilon > 5/6$.

In a different work, Nisan [Nis91] also gave a gives a PRG that ε-fools AC_0 circuits of depth d and size s using seed length $O(\log^{2d+6}(s/\varepsilon))$. For the special case of depth-2 circuits, that is, CNFs and DNFs, the work of Bazzi [Baz09], simplified by Razborov [Raz09], provides a PRG of seed length $O(\log n \cdot \log^2(s/\varepsilon))$, which has been improved to $\tilde{O}(\log^2(s/\varepsilon))$ by De et al. [DETT10]. For

[1] Space-bounded randomized algorithms are modeled by *oblivious, read-once* branching programs, which read the input bits in a specified order and read each input bit only once. In this paper, all the references to "branching programs" refer to "oblivious read-once branching programs."

the restricted case of *read-k* DNFs and CNFs, De et al. (for k =1), and Klivans et al. [KLW10] (for k constant) improve the seed length to $O(\log \varepsilon^{-1} \cdot \log s)$, which is optimal for constant ε, but it is essentially no better than the bound for general CNFs and DNFs when ε is polynomial in $1/n$.

The model of *combinatorial rectangles* is closely related to both bounded-width branching programs and read-once CNFs and are interesting combinatorial objects with a variety of applications of their own [ASWZ96]. The problem of constructing PRGs for combinatorial rectangles is closely related to the construction of small sample spaces that approximate the uniform distribution on many multivalued random variables [EGL+98]: they can be seen as an alternate generalization of the versatile notion of almost k-wise independent distributions on $\{0,1\}^n$ to larger domains $[m]^n$. Versions of this problem where each coordinate is a real interval were first studied in number theory and analysis [ASWZ96]. Subsequently there has been much work on this problem [EGL+98, LLSZ97, ASWZ96, Lu02, Vio11]. A PRG with seed length $O(\log n + \log^{3/2}(1/\varepsilon))$ [Lu02] is known for combinatorial rectangles; such a generator achieves the optimal seed length $O(\log n)$ when $\varepsilon \geq 2^{-O(\log^{2/3} n)}$, but not for $\varepsilon = 1/\text{poly}(n)$. It is known how to construct HSGs (which are a weakening of PRGs) with seed length $O(\log(n/\varepsilon))$ [LLSZ97].

Indeed, there are few models of computations for which we know how to construct PRGs with the optimal seed length $O(\log(n/\varepsilon))$ or even $\log^{1+o(1)}(n/\varepsilon)$. The most prominent examples are bounded-degree polynomials over finite fields [NN93, AGHP92, BV10a, Lov08, Vio08], with parities (which are fooled by small-bias distributions [NN93]) as a special case, and models that can be reduced to these cases, such as width-2 branching programs [SZ95, BDVY09].

In summary, there are several interesting models of computation for which a *polylogarithmic* dependence on n and $1/\varepsilon$ is known, and the dependence on one parameter is logarithmic on its own (e.g. seed length $O(\log n \log(1/\varepsilon))$), but a logarithmic bound in both parameters together has been elusive. Finally, we remark that not having a logarithmic dependence on the error ε is often a symptom of a more fundamental bottleneck. For instance, HSGs with constant error for width 4 branching programs imply HSGs with polynomially small error for width 3 branching programs, so achieving the latter is a natural first step towards the former. A polynomial-time computable PRG for CNFs with seed length $O(\log n/\varepsilon)$ would imply the existence of a problem in exponential time that requires depth-3 circuits of size $2^{\Omega(n)}$ and that cannot be solved by general circuits of size $O(n)$ and depth $O(\log n)$, which is a long-standing open problem in circuit complexity [Val77].

1.2 Our Results

In this paper, we construct the first generators with seed length $\tilde{O}(\log(n/\varepsilon))$ (where $\tilde{O}(\ )$ hides polylogarithmic factors in its argument) for several well-studied classes of functions mentioned above.

- PRGs for combinatorial rectangles. Previously, it was known how to construct HSGs with seed length $O(\log(n/\varepsilon))$ [LLSZ97], but the best seed length for PRGs was $O(\log n + \log^{3/2}(1/\varepsilon))$ [Lu02].
- PRGs for read-once CNF and DNF formulas. Previously, De, Etesami, Trevisan, and Tulsiani [DETT10] and Klivans, Lee and Wan [KLW10] had constructed PRGs with seed length $O(\log n \cdot \log(1/\varepsilon))$.
- HSGs for width 3 branching programs. Previously, Sima and Zak [SZ11] had constructed hitting set generators for width 3 branching programs with seed length $O(\log n)$ in case the error parameter ε is very large (greater than 5/6).

As a corollary of our PRG for combinatorial rectangles we get improved hardness amplification in NP by combining our results with those of Lu Tsai and Wu [LTW07] - we refer to Section 5 for details[2].

1.3 Techniques

Our generators are all based on a general new technique — the iterative application of "mild" (pseudo)random restrictions.

To motivate our technique, we first recall Håstad's switching lemma [Ajt83, FSS84, Hås86]: if we randomly assign a $1 - 1/O(k)$ fraction of the variables of a k-CNF, then the residual formula on the $n/O(k)$ unassigned variables is likely to become a constant. Ajtai and Wigderson [AW85] proposed the following natural approach to constructing PRGs for CNFs: construct a small pseudorandom family of restrictions that: 1) makes any given CNF collapse to a constant function with high probability; and 2) ensures that the CNF collapses to each constant function with the *right* probability as determined by the bias of the formula. Known derandomizations of the switching lemma are far from optimal in terms of the number of random bits needed [AW85, AAI$^+$01, GMR12]. We will show that, for read-once CNFs, such a pseudorandom restriction can be generated using $\tilde{O}(\log(m/\varepsilon))$ random bits.

We apply restrictions that only set a constant fraction of the variables at a time. The novel insight in our construction is that although we cannot set all the bits at one go from a small-bias distribution, we can set a constant fraction of bits from such a distribution and prove that the bias of the formula is preserved (on average). Hence we use only $\tilde{O}(\log(m/\varepsilon))$ truly random bits per phase. While such mild random restrictions do not drastically simplify the formulas, we show that in each phase a suitable measure of progress improves (e.g. most clauses will either be satisfied or will have reduced width), implying that the formula collapses to a constant after $O(\log\log(m/\varepsilon))$ steps; and so the total randomness will be $\tilde{O}(\log(m/\varepsilon))$. The idea of setting a few variables at a time is inspired by a recent PRG for hashing balls into bins due to Celis, Reingold, Segev, and Wieder [CRSW11].

We illustrate our technique below with a toy example.

A Toy Example. Consider a read-once CNF formula f of width w with $m = 2^{w+1}$ clauses in which the variables appear in order (aka the Tribes function of [BL85]). That is,

$$f(x) = f_1(x_1, \ldots, x_w) \wedge f_2(x_{w+1}, \ldots, x_{2w}) \wedge \cdots \wedge f_m(x_{(m-1)w+1}, \ldots, x_{mw})$$

where each f_i is the OR function. f has constant bias and can be computed both by a combinatorial rectangle and a width-3 branching program. De et al. showed that fooling this function with error ε using small-bias spaces requires seed-length $\Omega(w \log(1/\varepsilon)/\log\log(1/\varepsilon))$.

Assume we partition the input bits into two parts: x which contains the first $w/2$ variables of each clause and y which contains the rest. Let $x \circ y$ denote the concatenation of the two strings. We would like to show that for $\mathcal{D}$ a small-bias distribution and $\mathcal{U}$ the uniform distribution,

$$\left| \mathop{\mathbb{E}}_{x \sim \mathcal{D}, y \sim \mathcal{U}} [f(x \circ y)] - \mathop{\mathbb{E}}_{x \sim \mathcal{U}, y \sim \mathcal{U}} [f(x \circ y)] \right| \leq \varepsilon \tag{1.1}$$

A naive approach might be to view setting $y \sim \mathcal{U}$ as applying a random restriction with probability $1/2$. If this simplified the function f to the extent that it can be fooled by small-bias spaces, we would be done. Unfortunately, this is too much to hope for; it is not hard to see that

[2] We thank an anonymous referee for pointing out this application.

such a random restriction is very likely to give another Tribes-like function with width $w/2$, which is not much easier to fool using small bias than f itself.

Rather, we need to shift our attention to the *bias function* of f. For each partial assignment x, we define the bias function $F(x)$ as

$$F(x) = \mathop{\mathbb{E}}_{y \sim \mathcal{U}}[f(x \circ y)]. \tag{1.2}$$

We can now rewrite Equation (1.1) as

$$\left| \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[F(x)] - \mathop{\mathbb{E}}_{x \sim \mathcal{U}}[F(x)] \right| \leq \varepsilon \tag{1.3}$$

Our key insight is that for restrictions as above, the function F is in fact easy to fool using a small-biased space. This is despite the fact that $F(x)$ is an average of functions $f(x \circ y)$ (by Equation (1.2)), most of which are Tribes-like and hence are not easy to fool.

Let us give some intuition for why this happens. Since $f(x \circ y) = \prod_{i=1}^{m} f_i(x \circ y)$,

$$F(x) = \mathop{\mathbb{E}}_{y \sim \mathcal{U}}[f(x \circ y)] = \prod_{i=1}^{m} \mathop{\mathbb{E}}_{y \sim \mathcal{U}}[f_i(x \circ y)] = \prod_{i=1}^{m} F_i(x),$$

where $F_i(x)$ is the *bias function* of the i^{th} clause. But note that over a random choice of y, $f_i(x)$ is set to 1 with probability $1 - 2^{-w/2}$ and is a clause of width $w/2$ otherwise. Hence

$$F_i(x) = \mathop{\mathbb{E}}_{y \sim \mathcal{U}}[f_i(x \circ y)] = 1 - \frac{1}{2^{w/2}} + \frac{\bigvee_{j=1}^{w/2} x_{w(i-1)+j}}{2^{w/2}}.$$

As a consequence, over a random choice of x, we now have

$$F_i(x) = \begin{cases} 1 & \text{w.p. } 1 - 2^{-w/2} \\ 1 - 2^{-w/2} & \text{w.p. } 2^{-w/2} \end{cases}$$

Thus each $F_i(x)$ is a random variable with $\mathbb{E}_x[F_i(x)] = 1 - 2^{-w}$ and $\mathsf{Var}_x[F_i(x)] \approx 2^{-3w/2}$. In contrast, when we assign all the variables in the clauses at once, each $f_i(x)$ behaves like a Bernoulli random variable with bias $1 - 2^{-w}$. While it also has $\mathbb{E}_x[f_i(x)] = 1 - 2^{-w}$, the variance is much larger: $\mathsf{Var}_x[f_i(x)] \approx 2^{-w}$. The qualitative difference between $2^{-3w/2}$ and 2^{-w} is that in the former case, the sum of the variances over all 2^{w+1} clauses is small ($2^{-w/2}$), but in the latter it is more than 1. We leverage the small total variance to show that small-bias fools F, even though it does not fool f itself. Indeed, setting any constant fraction $\alpha < 1$ of variables in each clause would work.

We now sketch our proof that small-bias spaces fool F. Let $g_i(x) = F_i(x) - (1 - 2^{-w})$ be F_i shifted to have mean 0, so that $\mathbb{E}_x[g_i(x)^2] = \mathsf{Var}[F_i(x)]$. We can write

$$F(x) = \prod_{i=1}^{m} \left(1 - 2^{-w} + g_i(x)\right) = \sum_{k=1}^{m} c_k S_k(g_1(x), \dots, g_m(x)) \tag{1.4}$$

where S_k denotes the k^{th} elementary symmetric polynomial and $c_k \in [0, 1]$.[3]

[3] In the toy example we are currently studying, an alternative and simpler approach is to write $F_i(x) = (1 - 2^{-w/2})^{1-h_i(x)}$, where $h_i(x) = \vee_{j=1}^{w/2} x_{w(i-1)+j}$ is the indicator for whether x already satisfies the i'th clause on its own. Then $F(x) = \prod_i F_i(x)$ expands as a power series in $\sum_i (1 - h_i(x) - 2^{-w/2})$, and higher moment bounds can be used to analyze what happens when we truncate this expansion. However, this expansion is rather specific to the highly symmetric Tribes function, whereas we are able to apply the expansion in terms of symmetric polynomials much more generally.

Under the uniform distribution, one can show that

$$\mathop{\mathbb{E}}_{x\sim\mathcal{U}}\left[\,|S_k(g_1(x),\ldots,g_m(x))|\,\right] \leq \left(\sum_{i=1}^{m} \mathop{\mathbb{E}}_{x\sim\mathcal{U}}\left[g_i(x)^2\right]\right)^{k/2} \leq 2^{-wk/4}.$$

Thus for $k \geq O((\log n)/w)$, we expect each term in the summation in Equation (1.4) to be $1/\mathrm{poly}(n)$. So we can truncate at $d = O((\log n)/w)$ terms and retain a good approximation under the uniform distribution.

Our analysis of the small-bias case is inspired by the *gradually increasing independence* paradigm of Celis et al. [CRSW11], developed in the context of hashing. Every monomial in the g_i's of degree at most d depends on at most $wd = O(\log n)$ variables. A small-bias space provides an almost $O(\log n)$-wise independent distribution on the variables of x, so the $g_i(x)$'s will be almost d-wise independent. This ensures that polynomials in $g_1(x),\ldots,g_m(x)$ of degree at most d (such as $S_1,\ldots,S_d$) will behave like they do under the uniform distribution. But we also need to argue that the S_k's for $k > d$ have a small contribution to $\mathbb{E}_{x\sim\mathcal{D}}\left[F(x)\right]$.

Towards this end, we prove the following inequality for any real numbers $z_1,\ldots,z_m$:

$$\text{If } \ |S_1(z_1,\ldots,z_m)| \leq \frac{\mu}{2} \text{ and } \ |S_2(z_1,\ldots,z_m)| \leq \frac{\mu^2}{2}, \text{ then } |S_k(z_1,\ldots,z_m)| \leq \mu^k.$$

The proof uses the Newton–Girard formulas (see [CLO07]) which relate the symmetric polynomials and power sums. This lets us repeat the same truncation argument, provided that $S_1(g_1(x),\ldots,g_m(x))$ and $S_2(g_1(x),\ldots,g_m(x))$ are tightly concentrated even under small-bias distributions. We prove this concentration holds via suitable higher moment inequalities.[4]

This lets us show that small bias fools $F(x)$. By iterating this argument $\log w$ times, we get a PRG for f with polynomially small error and seed-length $O((\log n)(\log w)) = O((\log n)(\log\log n))$.

Read-Once CNFs. The case of general read-once CNFs presents several additional challenges. Since we no longer know how the variables are grouped into clauses, we (pseudo)randomly choose a subset of variables to assign using ε-biased spaces, and argue that for most clauses, we will not assign few variables. Clauses could now have very different sizes, and our approximation argument relied on tuning the amount of independence (or where we truncate) to the width of the clause. We handle this via an XOR lemma for ε-biased spaces, which lets us break the formula into $O(\log\log n)$ formulae, each having clauses of nearly equal size and argue about them separately.

Combinatorial Rectangles. A combinatorial rectangle $f : [W]^m \to \{0,1\}$ is a function of the form $f(x_1,\ldots,x_m) = \wedge_{i=1}^m f_i(x_i)$ for some Boolean functions $f_1,\ldots,f_m$. Thus, here we know which parts of the input correspond to which clauses (like the toy example above), but our clauses are arbitrary functions rather than ORs. To handle this, we use a more powerful family of gradual restrictions. Rather than setting $w/2$ bits of each co-ordinate, we instead (pseudo)randomly restrict the domain of each x_i to a set of size $W^{1/2}$. More precisely, we use a small-bias space to pseudorandomly choose hash functions $h_1,\ldots,h_m : [W^{1/2}] \to [W]$ and replace f with the restricted function $f'(z_1,\ldots,z_m) = \wedge_{i=1}^m (f_i \circ h_i)(z_i)$.

[4] These inequalities actually require higher moment bounds for the g_i's. We ignore this issue in this description for clarity, and because we suspect that this requirement should not be necessary.

Width 3 Branching Programs. For width 3 branching programs, inspired by Sima and Zak [SZ11] we reduce the task of constructing HSGs for width 3 to that of constructing HSGs for read-once CNF formulas where we also allow some clauses to be parities. Our PRG construction for read-once CNFs directly extends to also handle such formulas with parities (intuitively because small-bias spaces treat parities just like individual variables). The first step of our reduction actually works for any width d, and shows how to reduce the the task of constructing HSGs for width d to constructing hitting set generators for width d branching programs with sudden death, where the states in the *bottom level* are all assumed to be Reject states.

Organization. Section 2 gives some preliminaries on pseudorandomness. Section 3 develops our main new technical tools for constructing sandwiching approximators for symmetric functions. We prove an XOR Lemma for ε-biased spaces in Section 4.

Section 5 describes our PRG construction for combinatorial rectangles. The reduction from hitting sets for width 3 branching programs to hitting sets for CNFs with parity is in Section 6. The generator for read-once CNFs and for CNFs with parity are presented in Section 7 and Section 8 respectively.

2 Preliminaries

We briefly review some notation and definitions. We use $x \sim \mathcal{D}$ to denote sampling x from a distribution $\mathcal{D}$. For a set S, $x \sim S$ denotes sampling uniformly from S. By abuse of notation, for a function $G : \{0,1\}^s \to \{0,1\}^n$ we let G denote the distribution over $\{0,1\}^n$ of $G(y)$ when $y \sim \{0,1\}^s$. For a function $f : \{0,1\}^n \to \mathbb{R}$, we denote $\mathbb{E}[f] = \mathbb{E}_{x\sim\{0,1\}^n}[f(x)]$.

Hitting Set Generators and Pseudorandom Generators.

Definition 2.1 (Hitting Set Generators)**.** *A generator $G : \{0,1\}^r \to \{0,1\}^n$ is an (ε, δ)-*hitting set generator *(*HSG*) for a class $\mathcal{C}$ of Boolean functions if for every $f \in \mathcal{C}$ such that $\mathbb{E}[f] \geq \varepsilon$, we have $\mathbb{E}_{x\sim G} f(x) \geq \delta$. We refer to r as the seed-length of the generator and say G is explicit if there is an efficient algorithm to compute G that runs in time* $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$*.*

Typically, our hitting set generators will be (ε, δ) generators for some $\delta = \mathrm{poly}(\varepsilon, 1/n)$. Given two functions $g, h : \{0,1\}^n \to \{0,1\}$ we say $g \leq h$ if $g(x) \leq h(x)$ for all $x \in \{0,1\}^n$. To prove that G hits h, it suffices to show G hits some function $g \leq h$.

Definition 2.2 (Pseudorandom Generators)**.** *A generator $G : \{0,1\}^r \to \{0,1\}^n$ is an ε-*pseudorandom generator *(*PRG*) for a class $\mathcal{C}$ of Boolean functions if for every $f \in \mathcal{C}$, $|\mathbb{E}[f] - \mathbb{E}_G[f(y)]| \leq \varepsilon$. We refer to r as the seed-length of the generator and say G is explicit if there is an efficient algorithm to compute G that runs in time* $\mathrm{poly}(n, 1/\varepsilon)$*. We say G ε-fools $\mathcal{C}$ and refer to ε as the error.*

We shall make extensive use of *small-bias spaces*, introduced in the seminal work of Naor and Naor [NN93]. Usually these are defined as distributions over $\{0,1\}^n$, but it is more convenient for us to work with $\{\pm 1\}^n$.

Definition 2.3. *A distribution $\mathcal{D}$ on $\{\pm 1\}^n$ is said to be ε-biased if for every nonempty subset $I \subseteq [n]$, $|\mathbb{E}_{x\sim\{\pm 1\}^n}[\prod_{i\in I} x_i]| \leq \varepsilon$.*

There exist explicit constructions of ε-biased spaces which can be sampled from with $O(\log n + \log(1/\varepsilon))$ random bits [NN93]. These give efficient pseudorandom generators for the class of parity functions.

Definition 2.4. *Let $0 < \alpha, \delta < 1/2$. We say a distribution on $\mathcal{D}$ on $2^{[n]}$ is δ-almost independent with bias α if $I \leftarrow \mathcal{D}$ satisfies the following conditions:*

- *For every $i \in [n]$, $\mathbb{P}[i \in I] = \alpha$.*
- *For any distinct indices $i_1, \ldots, i_k \in [n]$ and $b_1, \ldots, b_k \in \{0,1\}^k$,*

$$\mathbb{P}\left[\wedge_{j=1}^{k}(1(i_j \in I) = b_j)\right] = \prod_{j=1}^{k} \mathbb{P}[1(i_j \in I) = b_j] \pm \delta.$$

There exist explicit constructions of distributions in $\mathcal{D}$ as above which only need $O(\log n + \log(1/\alpha\delta))$ random bits [NN93]. We will write $I \leftarrow \mathcal{D}(\alpha, \delta)$ for short whenever I is sampled from a δ-almost independent distribution with bias α as above.

Sandwiching Approximators. One of the central tools we use is to construct *sandwiching polynomial approximations* for various classes of functions. The approximating polynomials (P_ℓ, P_u) we construct for a function f will have two properties: 1) low-complexity as measured by the "L_1-norm" of P_ℓ, P_u and 2) they "sandwich" f, $P_u \leq f \leq P_u$. The first property will be important to argue that small-bias spaces *fool* the approximating polynomials and the second property will allow us to lift this property to the function being approximated. We formalize these notions below. For notational convenience, we shall view functions and polynomials as defined over $\{\pm 1\}^n$.

Definition 2.5. *Let $P : \{\pm 1\}^n \to \mathbb{R}$ be a polynomial defined as $P(x) = \sum_{I \subseteq [n]} c_I \prod_{i \in I} x_i$. Then, the L_1-norm of P is defined by $\mathsf{L}_1[P] = \sum_{I \subseteq [n]} |c_I|$. We say $f : \{\pm 1\}^n \to \mathbb{R}$ has δ-*sandwiching approximations *of L_1 norm t if there exist functions $f_u, f_\ell : \{\pm 1\}^n \to \mathbb{R}$ such that*

$$f_\ell(x) \leq f(x) \leq f_u(x) \; \forall x, \quad \mathbb{E}[f_u(x)] - \mathbb{E}[f_\ell(x)] \leq \delta, \quad \mathsf{L}_1(f_\ell), \mathsf{L}_1(f_u) \leq t.$$

We refer to f_ℓ and f_u as the lower and upper sandwiching approximations to f respectively.

It is easy to see that the existence of such approximations implies that f is $\delta + t\varepsilon$ fooled by any ε-biased distribution. In fact, as was implicit in the work of Bazzi [Baz09] and formalized in the work of De et. al. [DETT10], being fooled by small-bias spaces is essentially equivalent to the existence of good sandwiching approximators.

Lemma 2.6. *[DETT10] Let $f : \{\pm 1\}^n \to \mathbb{R}$ be a function. Then, the following hold for every $0 < \varepsilon < \delta$:*

- *If f has δ-sandwiching approximations of L_1-norm at most δ/ε, then for every ε-biased distribution $\mathcal{D}$ on $\{\pm 1\}^n$, $|\mathbb{E}_{x \sim \mathcal{D}}[f(x)] - \mathbb{E}[f]| \leq \delta$.*
- *If for every ε-biased distribution $\mathcal{D}$, $|\mathbb{E}_{x \sim \mathcal{D}}[f(x)] - \mathbb{E}[f]| \leq \delta$, then, f has (2δ)-sandwiching approximations of L_1-norm at most $|\mathbb{E}[f]| + \delta + (\delta/\varepsilon)$[5].*

[5] De et al. actually show a bound of δ/ε on the L_1 norm of the sandwiching approximators excluding their constant term. But it is easy to see that the constant term of the approximators is bounded by $|\mathbb{E}[f]| + \delta$.

Pseudorandom Generators for CNFs. A Conjunctive normal form formula (CNF) is a conjunction of disjunctions of literals. Throughout we view CNFs as functions on $\{\pm 1\}^n$, where we identify -1 with `false` and 1 with `true`. We say a CNF $f = C_1 \wedge C_2 \wedge \cdots \wedge C_m$ is a read-once CNF (RCNF), if no variable appears (by itself or as is its negation) more than once. We call m the size of f and the maximum number of variables in $C_1, \ldots, C_m$ the width of f. We shall also use the following results of [DETT10], [KLW10] which say that RCNFs with small number of clauses have very good sandwiching approximators.

Theorem 2.7. *Let $f : \{\pm 1\}^n \to \{0, 1\}$ be a RCNF with at most m clauses. Then, for every $\varepsilon > 0$, f has ε-sandwiching polynomials with L_1-norm at most $m^{O(\log(1/\varepsilon))}$.*

Theorem 2.8. *Let $f : \{\pm 1\}^n \to \{0, 1\}$ be a CNF with at most m clauses and width at most w. Then, for every $\varepsilon > 0$, f has ε-sandwiching polynomials with L_1-norm at most $(m/\varepsilon)^{O(w \log w)}$.*

3 Sandwiching Approximators for Symmetric Functions

For $k \geq 1$, let $S_k : \mathbb{R}^m \to \mathbb{R}$ denote the k^{th} elementary symmetric polynomial defined by

$$S_k(z_1, \ldots, z_m) = \sum_{I \subseteq [m], |I| = k} \prod_{i \in I} z_i.$$

Our main result on sandwiching approximators for symmetric functions is the following:

Theorem 3.1. *Let $g_1, \ldots, g_m : \{\pm 1\}^n \to \mathbb{R}$ be functions on disjoint sets of input variables and $\sigma_1, \sigma_2, \ldots, \sigma_m$ be positive numbers such that for all $i \in [m]$,*

$$\mathbb{E}[g_i] = 0, \quad \mathsf{L}_1[g_i] \leq t, \quad \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n}[(g_i)^{2k}] \leq (2k)^{2k} \sigma_i^{2k} \;\; \textit{for } k \geq 1.$$

Let $\sigma^2 = (\sum_i \sigma_i^2)/m$ and $\delta \in (0, 1)$ and $\varepsilon, k > 0$ be such that

$$m\sigma^2 \leq \frac{1}{\log(1/\delta)^{25}}, \quad k = \left\lceil \frac{5 \log(1/\delta)}{\log(1/m\sigma^2)} \right\rceil, \quad \varepsilon = \frac{\delta^4}{(mt + 1)^{2k}}. \tag{3.1}$$

Let $P(x) = \sum_{i=0}^m c_i S_i(g_1(x), \ldots, g_m(x))$ be a symmetric multilinear function of the g_is that computes a bounded function $P : \{\pm 1\}^n \to [-B, B]$, with $|c_i| \leq C$ for all $i \in [m]$. Then,

1. *For every ε-biased distribution $\mathcal{D}$, we have*

$$\left| \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n}[P(x)] - \mathop{\mathbb{E}}_{x \sim \mathcal{D}}[P(x)] \right| \leq O(B + C)\delta.$$

2. *P has $O(B + C)\delta$ sandwiching approximations of L_1 norm $O((B + C)(mt + 1)^{2k} \delta^{-3})$.*

As an illustration of this theorem, we state the following immediate corollary which formalizes the argument for the toy example in the introduction.

Theorem 3.2. *Let $\kappa > 0$ be a constant. Let $g_1, \ldots, g_m : \{\pm 1\}^n \to [-\sigma, \sigma]$ be functions on disjoint sets of input variables with $\mathbb{E}[g_i] = 0$, $\mathsf{L}_1[g_i] = O(1)$ and $\sigma \leq 1/m^{-1/2-\kappa}$. Let $P : \{\pm 1\}^n \to [-1, 1]$ be a symmetric polynomial in g_i's of the form $P(x) = \sum_{i=0}^m c_i S_i(g_1, \ldots, g_m)$, with $|c_i| \leq 1$. Then, for every $\delta \in (0, 1)$, with $\log(1/\sigma) \geq \Omega_\kappa(\log(1/\delta))$, P has δ-sandwiching polynomials of L_1-norm at most $\mathrm{poly}(1/\delta)$.*

To derive Theorem 3.2 from Theorem 3.1, observe that in the notation from Section 1.3, $m = 2^{w+1}$, $\sigma^2 \approx 2^{-3w/2}$ and all the other conditions hold.

In the rest of this section, we prove the first statement of Theorem 3.1. The second statement follows from the first by Lemma 2.6. We first sketch the steps involved in the proof.

Let k, ε be as in the theorem and let $\mathcal{D}$ be a ε-biased distribution. Let $P_{\leq k} \equiv \sum_{i=0}^{k} c_i S_i(g_1, \ldots, g_m)$. We will prove the theorem by showing that P cannot distinguish the uniform distribution from $\mathcal{D}$ by a series of inequalities:

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\{\pm 1\}^n}[P(g_1(x), \ldots, g_m(x))] &\approx_\delta \mathop{\mathbb{E}}_{x\sim\{\pm 1\}^n}[P_{\leq k}(g_1(x), \ldots, g_m(x))]\\ &\approx_\delta \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\leq k}(g_1(x), \ldots, g_m(x))]\\ &\approx_\delta \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(g_1(x), \ldots, g_m(x))].\end{aligned}$$

Of these, the second inequality will follow from the fact that $\mathsf{L}_1[P_{\leq k}] = \mathrm{poly}(1/\delta)$ (this is not too hard). The first inequality can be seen as a special case of the last inequality as the uniform distribution is an ε-biased distribution for any ε. Much of our effort will be in showing the last inequality.

To do this, we first show that there is an event $\mathcal{E}$ that happens with high probability under any ε-biased distribution, and conditioned on which $P_{\leq k}$ is a very good approximation for P. We then prove the last inequality by conditioning on the event $\mathcal{E}$ and using Cauchy-Schwarz to bound the error when $\mathcal{E}$ does not occur. The event $\mathcal{E}$ will correspond to $|S_1(g_1, \ldots, g_m)|$, $|S_2(g_1, \ldots, g_m)|$ being small, which we show happens with high probability using classical moment bounds. Finally, we show that $P_{\leq k}$ approximates P well if $\mathcal{E}$ happens by using the Newton-Girard Identities for symmetric polynomials (see Lemma 3.6).

3.1 Proof of Theorem 3.1

Our first task will be to show that under the assumptions of the theorem, $|\sum_i g_i(x)|$ and $|\sum_i g_i(x)^2|$ are small with high probability. We do so by first bounding the k'th moments of these variables and applying Markov's inequality. For this we will use Rosenthal's inequalities ([Ros72], [JSZ85], [Pin94]) which state the following:

Lemma 3.3. *For independent random variables $Z_1, \ldots, Z_m$ such that $\mathbb{E}[Z_i] = 0$, and all $k \in \mathbb{N}$,*

$$\mathbb{E}\left[\left(\sum_{i=1}^{m} Z_i\right)^{2k}\right] \leq (2k)^{2k} \max\left(\sum_{i=1}^{m} \mathbb{E}[Z_i^{2k}], \left(\sum_{i=1}^{m} \mathbb{E}[Z_i^2]\right)^k\right). \tag{3.2}$$

For independent non-negative random variables $Z_1, \ldots, Z_m$, and all $k \in \mathbb{N}$,

$$\mathbb{E}\left[\left(\sum_{i=1}^{m} Z_i\right)^{k}\right] \leq k^{k} \max\left(\sum_{i=1}^{m} \mathbb{E}[Z_i^{k}], \left(\sum_{i=1}^{m} \mathbb{E}[Z_i]\right)^k\right). \tag{3.3}$$

Lemma 3.4. *For all integers $k \geq 2$,*

$$\mathop{\mathbb{E}}_{x\sim\{\pm 1\}^n}\left[\left(\sum_{i=1}^{m} g_i(x)\right)^{2k}\right] \leq (2k)^{4k}\left(\sum_{i=1}^{m} \sigma_i^2\right)^k \tag{3.4}$$

$$\mathop{\mathbb{E}}_{x\sim\{\pm 1\}^n}\left[\left(\sum_{i=1}^{m} (g_i(x))^2\right)^{k}\right] \leq (2k)^{3k}\left(\sum_{i=1}^{m} \sigma_i^2\right)^k \tag{3.5}$$

Proof. Let $Z_i = g_i(x)$, $x \sim \{\pm 1\}^n$. Then, Z_i's are independent mean-zero variables. Now, by Rosenthal's inequality, Equation (3.2),

$$\begin{aligned}
\mathbb{E}\left[\left(\sum_i Z_i\right)^{2k}\right] &\leq (2k)^{2k} \max\left(\sum_i \mathbb{E}[Z_i^{2k}], \left(\sum_i \mathbb{E}[Z_i^2]\right)^k\right) \\
&\leq (2k)^{2k} \max\left(\sum_i (2k)^{2k}\sigma_i^{2k}, \left(\sum_i 4\sigma_i^2\right)^k\right) \\
&\leq (2k)^{4k} \max\left(\sum_i \sigma_i^{2k}, \left(\sum_i \sigma_i^2\right)^k\right) \\
&= (2k)^{4k}\left(\sum_i \sigma_i^2\right)^k .
\end{aligned}$$

The second bound follows similarly by applying Rosenthal's inequality,Equation (3.3), to the non-negative random variables $Z_i^2 = g_i^2$:

$$\mathbb{E}\left[\left(\sum_i Z_i^2\right)^k\right] \leq k^k \max\left(\sum_i \mathbb{E}[Z_i^{2k}], \left(\sum_i \mathbb{E}[Z_i^2]\right)^k\right) \leq (2k)^{3k}\left(\sum_{i=1}^m \sigma_i^2\right)^k .$$

□

A consequence of Lemma 3.4 is the following:

Corollary 3.5. *For all $k \geq 2$, under any ε-biased distribution $\mathcal{D}$,*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{i=1}^m g_1\right)^{2k}\right] \leq (2k)^{4k}(m\sigma^2)^k + \varepsilon(mt)^{2k} \tag{3.6}$$

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{i=1}^m g_1^2\right)^{k}\right] \leq (2k)^{3k}(m\sigma^2)^k + \varepsilon(mt^2)^{k} \tag{3.7}$$

Proof. Note that for any function $h : \{\pm 1\}^n \to \mathbb{R}$, $\mathsf{L}_1[h^k] \leq (\mathsf{L}_1[h])^k$. Therefore, applying this inequality to $h \equiv \sum_i g_i$, we get $\mathsf{L}_1[\,(\sum_i g_i)^{2k}\,] \leq (mt)^{2k}$. The first inequality now follows from Lemma 3.4 and Lemma 2.6. The second inequality follows similarly. □

Next we show that $|\sum_i g_i|$, $\sum_i g_i^2$ being small implies the smallness in absolute value of $S_k(g_1, \ldots, g_m)$ for every $k \geq 2$. Note that there is no probability involved in this statement.

Lemma 3.6. *Let $z_1, \ldots, z_m$ be real numbers that satisfy*

$$\left|\sum_{i=1}^m z_i\right| \leq \mu, \quad \sum_{i=1}^m z_i^2 \leq \mu^2.$$

Then for every $k \geq 2$ we have

$$|S_k(z_1, \ldots, z_m)| \leq \mu^k.$$

Proof. To prove this lemma, we first bound the power sums $E_k(z_1,\ldots,z_m)$ which are defined as

$$E_k(z_1,\ldots,z_m)=\sum_{i=1}^m z_i^k.$$

Note that $E_1 = S_1$. We start by bounding E_k for $k \geq 2$ using the $\mathbb{L}_k$ norm inequalities

$$|E_k(z_1,\ldots,z_m)|^{\frac{1}{k}} \leq \left(\sum_{i=1}^m |z_i|^k\right)^{\frac{1}{k}} \leq \left(\sum_{i=1}^m z_i^2\right)^{\frac{1}{2}} = E_2(z_1,\ldots,z_m)^{\frac{1}{2}}$$

Hence we have $|E_k(z_1,\ldots,z_m)| \leq \mu^k$.

The relation between the power sums and elementary symmetric polynomials is given by the Newton-Girard identities (see [CLO07], Chapter 7.1 for instance) discovered in the 17th century.

$$S_k(z_1,\ldots,z_m)=\frac{1}{k}\sum_{i=1}^k(-1)^{i-1}S_{k-i}(z_1,\ldots,z_m)E_i(z_1,\ldots,z_m). \tag{3.8}$$

We use these to show by induction on k that $|S_k| \leq \mu^k$. For $k = 2$, we have

$$S_2(z_1,\ldots,z_m)=\frac{1}{2}(S_1(z_1,\ldots,z_m)^2 - E_2(z_1,\ldots,z_m)) \leq \frac{1}{2}(\mu^2+\mu^2) \leq \mu^2.$$

Assume we have proved the bound up to $k-1$. Using the Newton-Girard formula,

$$|S_k(z_1,\ldots,z_m)| \leq \frac{1}{k}\sum_{i=1}^k |S_{k-i}(z_1,\ldots,z_m)||E_i(z_1,\ldots,z_m)| \leq \frac{1}{k}\sum_{i=1}^k \mu^{k-i}\mu^i \leq \mu^k.$$

□

Let

$$P_{\leq k}(x)=\sum_{i=0}^k c_i S_i(g_1,\ldots,g_m)$$

denote the truncation of P to degree k. We use the following bounds for $P_{\leq k}$.

Lemma 3.7. *Let $P, m, t, C, \mathcal{D}$ be as in Theorem 3.1. If $m\sigma^2 \leq \frac{1}{2}$, then for every $k \in \mathbb{N}$,*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[P_{\leq k}(x)^2\right] \leq 2C^2 + \varepsilon\cdot(mt+1)^{2k}\cdot C^2. \tag{3.9}$$

Proof. We observe that the symmetric polynomials $S_0 = 1,\ldots,S_k$ on $g_1,\ldots,g_m$ are mutually orthogonal under the uniform distribution, i.e., for $i \neq j$,

$$\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[S_i(g_1(x),\ldots,g_m(x))\cdot S_j(g_1(x),\ldots,g_m(x))] = 0.$$

For brevity, we shall omit writing out the argument x in the following. For $i \geq 1$, we have

$$\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}\left[S_i(g_1,\ldots,g_m)^2\right] = \sum_{|S|=i}\prod_{j\in S}\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}\left[g_j^2\right] \leq \left(\sum_{j=1}^m \mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[g_j^2]\right)^i \leq (m\sigma^2)^i.$$

Therefore, assuming that $m\sigma^2 \le 1/2$,

$$\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}\left[P_{\le k}(g_1,\ldots,g_m)^2\right] = \sum_{i=0}^{k} c_i^2 \mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}\left[S_i(g_1,\ldots,g_m)^2\right] \le C^2\sum_{i=0}^{k}(m\sigma^2)^i \le 2C^2. \tag{3.10}$$

Since $\mathsf{L}_1[g_j] \le t$, we have

$$\begin{aligned}\mathsf{L}_1\left[S_i(g_1,\ldots,g_m)\right] &\le \binom{m}{i}t^i,\\ \mathsf{L}_1\left[P_{\le k}\right] &\le C\sum_{i=0}^{k}\binom{m}{i}t^i \le C\cdot(mt+1)^k, \text{ and}\\ \mathsf{L}_1\left[P_{\le k}^2\right] &\le \mathsf{L}_1\left[P_{\le k}\right]^2 \le C^2\cdot(mt+1)^{2k}.\end{aligned}$$

Hence

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\le k}(x)^2] \le C^2(2+\varepsilon(mt+1)^{2k}) \le 2C^2 + \varepsilon\cdot(mt+1)^{2k})\cdot C^2.$$

□

Setting Parameters. In Theorem 3.1, we choose

$$k = \left\lceil \frac{5\log(1/\delta)}{\log(1/m\sigma^2)}\right\rceil$$

which guarantees $\delta^5/2 \le (m\sigma^2)^k \le \delta^5$. By Equation (3.1) we have

$$m\sigma^2 \le \frac{1}{\log(1/\delta)^{25}},$$

from which it follows that

$$k \le \frac{\log(1/\delta)}{5\log\log(1/\delta)}, \text{ and} \tag{3.11}$$

$$(2k)^{4k} \le \frac{1}{\delta}. \tag{3.12}$$

Finally, for all ε small enough so that $\varepsilon\cdot(mt+1)^{2k} \le \delta^4$, the following bounds will hold under the assumptions of Theorem 3.1, by Corollary 3.5 and Lemma 3.7,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[P_{\le k}(x)^2\right] \le 4C^2, \tag{3.13}$$

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{i=1}^{m} g_i(x)\right)^{2k}\right] \le (2k)^{4k}(m\sigma^2)^k + \varepsilon(mt)^{2k} \le 2\delta^4 \tag{3.14}$$

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{i=1}^{m} g_i(x)^2\right)^{k}\right] \le (2k)^{3k}(m\sigma^2)^k + \varepsilon(mt)^{2k} \le 2\delta^4. \tag{3.15}$$

We now proceed to prove Statement (1) in Theorem 3.1, which we restate below with specific constants.

Lemma 3.8. *With the notation from Theorem 3.1, we have*

$$\left|\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(x)]\right| \leq (4B + 13C)\cdot\delta. \tag{3.16}$$

Proof. We will show that under any ε-biased distribution $\mathcal{D}$,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x) - P_{\leq k}(x)|] \leq (2B + 6C)\delta. \tag{3.17}$$

Note that $\mathcal{U}$ is ε-biased for $\varepsilon = 0$, so the above bound applies to it. We derive Equation (3.16) from Equation (3.17) as follows:

$$\left|\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(x)]\right| \leq \left|\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P(x)] - \mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P_{\leq k}(x)]\right| + \left|\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P_{\leq k}(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\leq k}(x)]\right| + \left|\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\leq k}(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(x)]\right|. \tag{3.18}$$

The first and last terms are bounded using Equation (3.17). We bound the middle term by

$$|\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P_{\leq k}(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\leq k}(x)]| \leq \varepsilon\cdot\mathsf{L}_1[P_{\leq k}(x)] \leq \varepsilon\cdot C\cdot(mt+1)^k \leq C\delta^4.$$

Equation (3.16) follows by plugging these bounds into Equation (3.18):

$$\left|\mathop{\mathbb{E}}_{x\sim\{\pm1\}^n}[P(x)] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(x)]\right| \leq 2(2B + 6C)\delta + C\delta^4 \leq (4B + 13C)\delta.$$

We now prove Equation (3.17). Define a good event $G \subseteq \{\pm1\}^n$ containing those x for which the following bounds hold:

$$\left|\sum_{i=1}^m g_i(x)\right| \leq \delta^{\frac{1}{k}}, \quad \left|\sum_{i=1}^m (g_i(x))^2\right| \leq \delta^{\frac{2}{k}}. \tag{3.19}$$

For $x \in G$, $P_{\leq k}(x)$ gives a good approximation to $P(x)$. By Lemma 3.6, we have $|S_\ell(g_1(x), \ldots, g_m(x))| \leq \delta^{\ell/k}$ for all $\ell \geq 2$. Hence, for all $x \in G$

$$|P(x) - P_{\leq k}(x)| \leq \sum_{\ell=k+1}^m |c_\ell S_\ell(g_1(x), \ldots, g_m(x))| \leq C\sum_{\ell=k+1}^m \delta^{\ell/k} \leq C\delta\sum_{\ell\geq1}\delta^{\ell/k} \leq 2C\delta. \tag{3.20}$$

We now bound the probability of $\neg G$ using Markov's inequality applied to a k'th moment bound obtained from Equations (3.14) and (3.15):

$$\Pr_{x\sim\mathcal{D}}\left[\left|\sum_{i=1}^m g_i(x)\right| \geq \delta^{1/k}\right] = \Pr_{x\sim\mathcal{D}}\left[\left|\sum_{i=1}^m g_i(x)\right|^{2k} \geq \delta^2\right] \leq \frac{1}{\delta^2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{i=1}^m g_i(x)\right)^{2k}\right] \leq 2\delta^2,$$

$$\Pr_{x\sim\mathcal{D}}\left[\left|\sum_{i=1}^m g_i(x)^2\right| \geq \delta^{2/k}\right] = \Pr_{x\sim\mathcal{D}}\left[\left|\sum_{i=1}^m g_i(x)^2\right|^{k} \geq \delta^2\right] \leq \frac{1}{\delta^2}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\left(\sum_{i=1}^m g_i(x)^2\right)^{k}\right] \leq 2\delta^2,$$

Let $\mathbf{1}_G(x)$ and $\mathbf{1}_{\neg G}(x)$ denote the indicators of G and $\neg G$ respectively. We have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\mathbf{1}_{\neg G}(x)] \leq \Pr_{x\sim\mathcal{D}}\left[\left|\sum_{i=1}^{m} g_i(x)\right| \geq \delta^{1/k}\right] + \Pr_{x\sim\mathcal{D}}\left[\left|\sum_{i=1}^{m} g_i(x)^2\right| \geq \delta^{2/k}\right] \leq 4\delta^2. \tag{3.21}$$

Further,

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x) - P_{\leq k}(x)|] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x) - P_{\leq k}(x)| \cdot \mathbf{1}_G(x)] + \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x) - P_{\leq k}(x)| \cdot \mathbf{1}_{\neg G}(x)] \tag{3.22}$$

By Equation 3.20, we have

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x) - P_{\leq k}(x)| \cdot \mathbf{1}_G(x)] \leq \max_{x\notin G} |P(x) - P_{\leq k}(x)| \leq 2C\delta \tag{3.23}$$

To bound the second term,

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x) - P_{\leq k}(x)| \cdot \mathbf{1}_{\neg G}] &\leq \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P(x)| \cdot \mathbf{1}_{\neg G}] + \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[|P_{\leq k}(x)| \cdot \mathbf{1}_{\neg G}] \\ &\leq \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(x)^2]^{\frac{1}{2}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\mathbf{1}_{\neg G}]^{\frac{1}{2}} + \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\leq k}(x)^2]^{\frac{1}{2}} \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\mathbf{1}_{\neg G}]^{\frac{1}{2}} \\ &\leq B \cdot 2\delta + 2C \cdot 2\delta \end{aligned} \tag{3.24}$$

where we use the bounds

$$\begin{aligned}\mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P(x)^2] &\leq B^2 \quad &&(\text{Since } |P(x)| \leq B) \\ \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[P_{\leq k}(x)^2] &\leq 4C^2 \quad &&(\text{Equation (3.13)}) \\ \mathop{\mathbb{E}}_{x\sim\mathcal{D}}[\mathbf{1}_{\neg G}] &\leq 4\delta^2. \quad &&(\text{Equation (3.21)})\end{aligned}$$

Plugging Equations (3.23) and (3.24) into Equation (3.22) we get Equation (3.17). □

4 An XOR Lemma for ε-biased spaces

In this section, we prove an XOR Lemma that helps us show the existence of good sandwiching approximators for the composition of a function on few variables with functions on disjoint sets of variables, each of which have good sandwiching approximators. We call it an XOR lemma, since one can view it as a generalization of Vazirani's XOR lemma.

Theorem 4.1. *Let $f^1, \dots, f^k : \{\pm 1\}^n \to [0,1]$ be functions on disjoint input variables such that each f^i has ε-sandwiching approximations of L_1 norm t. Let $H : [0,1]^k \to [0,1]$ be a multilinear function in its inputs. Let $h : \{\pm 1\}^n \to [0,1]$ be defined as $h(x) = H(f^1(x), \dots, f^k(x))$. Then h has $(16^k\varepsilon)$-sandwiching approximations of L_1 norm $4^k(t+1)^k$.*

Proof. For $S \subseteq [k]$ define the monomial

$$M^S(x) = \prod_{i\in S} f^i(x) \prod_{j\notin S}(1 - f^j(x)).$$

Let f_u^i and f_ℓ^i denote the upper and lower sandwiching approximations to f^i. Then we have

$$\begin{aligned} f_u^i(x) \geq f^i(x), &\quad \mathop{\mathbb{E}}_{x\sim\{\pm 1\}^n}[f_u^i(x) - f^i(x)] \leq \varepsilon. \\ 1 - f_\ell^j(x) \geq 1 - f_\ell^j(x), &\quad \mathop{\mathbb{E}}_{x\sim\{\pm 1\}^n}[(1 - f_\ell^j(x)) - (1 - f^j(x))] \leq \varepsilon. \end{aligned}$$

Hence, if we define

$$M_u^S(x) = \prod_{i \in S} f_u^i(x) \prod_{j \notin S} (1 - f_\ell^j(x)),$$

then we have

$$M_u^S(x) \geq M^S(x) \ \forall \ x \in \{\pm 1\}^n,$$
$$\mathsf{L}_1[M_u^S] = \prod_{i \in S} \mathsf{L}_1[f_u^i] \prod_{j \notin S} \mathsf{L}_1[1 - f_\ell^j] \leq (t+1)^k.$$

We will show using a hybrid argument, that

$$\mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} [M_u^S(x) - M^S(x)] \leq 2^k \varepsilon.$$

For simplicity, we only do the case $S = [k]$. We define a sequence of polynomials $M_u^S = M_0, M_1 \ldots, M_k = M^S$ where

$$M_i(x) = \prod_{j=1}^{i} f^j(x) \prod_{j=i+1}^{k} f_u^j(x).$$

We now have

$$\begin{aligned}
\mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} [M_i(x) - M_{i+1}(x)] &= \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} \left[\left(f_u^{i+1}(x) - f^{i+1}(x)\right) \cdot \prod_{j=1}^{i} f^j(x) \cdot \prod_{j=i+2}^{k} f_u^j(x) \right] \\
&= \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} \left[f_u^{i+1}(x) - f^{i+1}(x)\right] \cdot \prod_{j=1}^{i} \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} \left[f^j(x)\right] \cdot \prod_{j=i+2}^{k} \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} \left[f_u^j(x)\right] \\
&\leq \varepsilon \prod_{j=1}^{i} 1 \prod_{j=i+2}^{k} (1 + \varepsilon) \leq (1 + \varepsilon)^{k-i-1} \varepsilon
\end{aligned}$$

where we use the facts that $\mathbb{E}_{x \sim \{\pm 1\}^n}[f^j] \leq 1$ and $\mathbb{E}_{x \sim \{\pm 1\}^n}[f_u^j] \leq \mathbb{E}_{x \sim \{\pm 1\}^n}[f^j] + \varepsilon \leq 1 + \varepsilon$. We now have

$$\begin{aligned}
\mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} \left[M_u^S(x) - M^S(x)\right] &\leq \sum_{i=0}^{k-1} \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n} [M_i(x) - M_{i+1}(x)] \\
&\leq \varepsilon(1 + (1 + \varepsilon) \cdots (1 + \varepsilon)^{k-1}) \leq 2^k \varepsilon.
\end{aligned}$$

To construct a lower-sandwiching approximator, we observe that

$$\sum_{S \subseteq [k]} M^S(x) = \prod_{i \in [k]} (f^i(x) + 1 - f^i(x)) = 1.$$

Hence if we define

$$M_\ell^S(x) = 1 - \sum_{T \neq S} M_u^T(x)$$

then

$$M_\ell^S(x) \le 1 - \sum_{T \neq S} M^T(x) = M^S(x),$$

$$\mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n}[M^S(x) - M_\ell^S(x)] = \sum_{T \neq S} M_u^T(x) - M^T(x) \le 4^k \varepsilon,$$

$$\mathsf{L}_1[M_\ell^S] \le 2^k(t+1)^k.$$

Finally, let $\mathbf{1}_S \in \{0,1\}^k$ denote the indicator vector of the set S. Since H is multilinear, we can write

$$H(y) = \sum_{S \subseteq [k]} H(\mathbf{1}_S) \prod_{i \in S} y_i \prod_{j \notin S}(1 - y_j)$$

where $H(\mathbf{1}_S) \in [0,1]$. Hence

$$h(x) = \sum_{S \subseteq [k]} H(\mathbf{1}_S) \prod_{i \in S} f_i(x) \prod_{j \notin S}(1 - f_j(x)) = \sum_{S \subseteq [k]} H(\mathbf{1}_S) M^S(x)$$

We define the polynomials

$$h_u(x) = \sum_{S \subseteq [k]} H(\mathbf{1}_S) M_u^S(x), \;\; h_\ell(x) = \sum_{S \subseteq [k]} H(\mathbf{1}_S) M_\ell^S(x).$$

It follows that

$$h_u(x) \ge h(x) \ge h_\ell(x)$$

$$\mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n}[h_u(x) - h_\ell(x)] \le \sum_{S \subseteq [k]} H(\mathbf{1}_S) \mathop{\mathbb{E}}_{x \sim \{\pm 1\}^n}[M_u^S(x) - M_\ell^S(x)] \le 16^k \varepsilon,$$

$$\mathsf{L}_1[h_u] \le 2^k(t+1)^k, \;\; \mathsf{L}_1[h_\ell] \le 4^k(t+1)^k.$$

□

5 A PRG for Combinatorial Rectangles

We start by defining combinatorial rectangles (CRs).

Definition 5.1. *A* combinatorial rectangle *is a function $f : (\{\pm 1\}^w)^m \to \{0,1\}$ of the form $f(x_1, \ldots, x_m) = \bigwedge_{i=1}^m f_i(x_i)$, where $f_i : \{\pm 1\}^w \to \{0,1\}$, and each $x_i \in \{\pm 1\}^w$.We refer to the f_is as the* co-ordinate functions *of f. We refer to m as the* size[6] *of f and w as the* width.

We construct an explicit PRG for CRs with seed-length $\tilde{O}(\log m + w + \log(1/\delta))$. The previous best construction due to Lu had a seed-length of $O(\log m + w + \log^{3/2}(1/\delta))$ [Lu02].

Theorem 5.2. *There is an explicit pseudorandom generator for the class of combinatorial rectangles of width w and size m with error at most δ and seed-length $O((\log w)(\log(m) + w + \log(1/\delta)) + \log(1/\delta) \log\log(1/\delta) \log\log\log(1/\delta))$.*

[6] This is usually referred to as the *dimension* in the literature; we use this terminology for the CNF analogy.

Our generator uses a recursive sampling technique and we next describe a single step of this recursive procedure. For this informal description suppose that $\delta = 1/\mathrm{poly}(m)$, $w = O(\log m)$ and let $v = 3w/4$. Fix a CR $f : (\{\pm 1\}^w)^m \to \{0,1\}$.

Consider the following two-step process for generating a uniformly element x from $(\{\pm 1\}^w)^m$.

- Choose a sequence of multi-sets $S_1, \ldots, S_m \subseteq \{\pm 1\}^w$ each of size 2^v by picking 2^v elements of $\{\pm 1\}^w$ independently and uniformly at random.
- Sample $x_i \sim S_i$ and set $x = (x_1, \ldots, x_m)$.

This results in an x that is uniformly distributed over $(\{\pm 1\}^w)^m$. We will show that the $\mathbb{E}_x[f(x)]$ will not change much, even if the sampling in the first step can is done pseudorandomly using a small-bias space for suitably small ε.

Our final generator is obtained by iterating the one-step procedure for $T = O(\log\log m))$ steps: At step t we choose multi-sets $S_1^t \subseteq S_1^{t-1}, \ldots, S_m^t \subseteq S_m^{t-1}$ each of cardinality exactly $2^{(3/4)^t w}$ using small-bias. After T steps, we are left with a rectangle of width $w = O(\log\log m)$. Such rectangles can be fooled by ε-bias spaces where $\varepsilon = 1/m^{O(\log\log m)}$. The total randomness used over all the steps is $O((\log m) \cdot (\log\log m))$.

5.1 Sandwiching Approximations for Bias Functions

In the following, let f be a CR of width w and coordinate functions $f_1, \ldots, f_m : \{\pm 1\}^w \to \{0,1\}$. We describe a restriction of f which reduces the width from w to $v = 3w/4$.

- For every $a \in \{\pm 1\}^v$, we sample string $x_a = (x_{a,1}, \ldots, x_{a,m}) \sim \{\{\pm 1\}^w\}^m$.
- For $i \in [m]$, we define restricted co-ordinate functions f_i^v on inputs y_i by $f_i^v(y_i) = f(x_{y_i,i})$.
- Define the restricted rectangle $f^v : (\{\pm 1\}^v)^m \to \{0,1\}$ on $y_1, \ldots, y_m$ by

$$f^v(y_1, \ldots, y_m) = \bigwedge_{i=1}^m f_i^v(y_i) \tag{5.1}$$

Let $\bar{x} \in \{\{\pm 1\}^w\}^{2^v \times m}$ denote the matrix whose rows are indexed by $a \in \{\pm 1\}^v$, the columns by $i \in [m]$ and (a,i)'th entry is given by $\bar{x}[a,i] = x_{a,i} \in \{\pm 1\}^w$. Every such matrix defines a *restriction* of f. We will show that if choosing $\bar{x}$ from an ε-biased space for $\varepsilon = 1/\mathrm{poly}(m)$ suitably small, and from the uniform distribution have almost the same effect on f. For $i \in [m]$, let $\bar{x}[i]$ denote the i'th column of $\bar{x}$. For each coordinate function f_i, define the *sample average* function

$$\bar{f}_i(\bar{x}) = \frac{1}{2^v} \sum_{a \in \{\pm 1\}^v} f_i(x_{a,i}) = \mathop{\mathbb{E}}_{a \sim \{\pm 1\}^v} [f_i^v(a)]\,. \tag{5.2}$$

Note that each $\bar{f}_i$ only depends on column i of $\bar{x}$. Define the *bias* function of $\bar{x}$ as

$$F(\bar{x}) = \prod_{i=1}^m \bar{f}_i(\bar{x}) = \mathop{\mathbb{E}}_{y \sim (\{\pm 1\}^v)^m} [f^v(y)]\,. \tag{5.3}$$

The main lemma of this section shows that this bias function can be fooled by small-bias spaces.

Lemma 5.3 (Main)**.** *Let F be as defined in Equation (5.3). Assume that $\delta < 1/4$ and $w \leq \log(1/\delta)$, $v = 3w/4 \geq 50\log\log(1/\delta)$. Then $F(x)$ has δ-sandwiching approximations of L_1 norm* $\mathrm{poly}(1/\delta)$.

We start by stating two simple claims.

Claim 5.4. *For the sample average functions $\bar{f}_i$ defined as in Equation (5.2), we have*

$$\mathsf{L}_1(\bar{f}_i) \le \mathsf{L}_1(f_i) \le 2^{w/2}.$$
$$\mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[\bar{f}_i(\bar{x})] = \mathop{\mathbb{E}}_{z\sim\{\pm1\}^w}[f_i(z)].$$

Proof. From Equation (5.2), it follows that

$$\mathsf{L}_1[\bar{f}_i] \le \frac{1}{2^v}\sum_{a\in\{\pm1\}^v} \mathsf{L}_1[f_i] = \mathsf{L}_1[f_i] \le 2^{w/2}$$

where the last inequality holds for any Boolean function on w input bits. The bound on the expectation follows directly from Equation (5.2). □

The justification for the name bias function comes from the following lemma.

Claim 5.5. *For f^v and F as defined in Equation (5.1) and Equation (5.3),*

$$\mathop{\mathbb{E}}_{y\sim(\{\pm1\}^v)^m}[f^v(y)] = F(\bar{x}).$$

Proof. Note that

$$f_i^v(y_i) = \sum_{a\in\{\pm1\}^v} \mathbf{1}_{y=a} f_i(x_{a,i}),$$
$$\text{hence} \quad \mathop{\mathbb{E}}_{y\sim(\{\pm1\}^v)^m}[f_i^v(y)] = \frac{1}{2^v}\sum_{a\in\{\pm1\}^v} f_i(x_{a,i}) = \bar{f}_i(\bar{x}).$$

It follows that

$$\begin{aligned}
\mathop{\mathbb{E}}_{y\sim(\{\pm1\}^v)^m}[f^v(y)] &= \mathop{\mathbb{E}}_{y\sim(\{\pm1\}^v)^m}\left[\bigwedge_{i=1}^m f_i^v(y_i)\right] \\
&= \mathop{\mathbb{E}}_{y\sim(\{\pm1\}^v)^m}\left[\prod_{i=1}^m f_i^v(y)\right] \\
&= \prod_{i=1}^m \mathop{\mathbb{E}}_{y\sim(\{\pm1\}^v)^m}[f_i^v(y)] \\
&= \prod_{i=1}^m \bar{f}_i(\bar{x}) \\
&= F(\bar{x}).
\end{aligned}$$

□

We will prove Lemma 5.3 by applying Theorem 3.1 to the functions $g_i : \{\{\pm1\}^w\}^{2^v} \to \mathbb{R}$ defined as follows: $g_i(\bar{x}) = (\bar{f}_i(\bar{x}) - p_i)/p_i,$, where $p_i = \mathbb{E}_{a\sim\{\pm1\}^w}[f_i(x)]$. (We assume $p_i \neq 0$.)

We will need the following technical lemma, which helps us show that the functions g_i satisfy the moment conditions needed to apply Theorem 3.1. For brevity, let $\mathcal{U}$ denote $(\{\pm1\}^v)^{2^v\times m}$ in the remainder of this section.

Lemma 5.6. *Let $p_i, g_i, \mathcal{U}$ be defined as above. We have $\mathbb{E}_{\bar{x}\sim\mathcal{U}}[g_i(\bar{x})^{2k}] \le (2k)^{2k}\sigma_i^{2k}$ where*

$$\sigma_i^2 = \begin{cases} \frac{(1-p_i)}{2^v p_i} & \textit{for } p_i \in [2^{-v/10}, 1/2], \\ \frac{2(1-p_i)}{2^v} & \textit{for } p_i \in [1/2, 1-2^{-v}], \\ \frac{2}{2^{2v}} & \textit{for } p_i \in [1-2^{-v}, 1]. \end{cases}$$

Proof. W start by bounding the moments of $(\bar{f}_i(\bar{x}) - p_i)$. We have

$$2^v(\bar{f}_i(\bar{x}) - p_i) = \sum_{a\in\{\pm1\}^v} (f(x_{a,i}) - p_i)$$

which is the sum of 2^v i.i.d p_i-biased random variables with mean 0. Hence we can apply Rosenthal's inequality (Equation (3.2)) to get

$$(2^v)^{2k} \mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[(\bar{f}_i(\bar{x}) - p_i)^{2k}] \le (2k)^{2k} \max\left(2^v((1-p_i)^{2k}p_i + p_i^{2k}(1-p_i)), (2^v p_i(1-p_i))^k\right)$$

Hence we have

$$\begin{aligned} \mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[g_i(\bar{x})^{2k}] &= \frac{1}{p_i^{2k}} \mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[(\bar{f}_i(\bar{x}) - p_i)^{2k}] \\ &\le \frac{(2k)^{2k}}{(2^v)^{2k}} \max\left(2^v\left(\left(\frac{1-p_i}{p_i}\right)^{2k} p_i + 1 - p_i\right), \left(2^v \cdot \frac{1-p_i}{p_i}\right)^k\right) \end{aligned}$$

We will use the following bounds

$$2^v\left(\left(\frac{1-p_i}{p_i}\right)^{2k} p_i + 1 - p_i\right) \le \begin{cases} 2^{v(1+k/5)} & \text{for } p_i \in [2^{-v/10}, 1/2]. \\ 2^{v+1}(1-p_i) & \text{for } p_i \in [1/2, 1]. \end{cases}$$

$$\left(2^v \cdot \frac{1-p_i}{p_i}\right)^k \le \left(2^{v+1}(1-p_i)\right)^k \quad \text{for } p_i \in [1/2, 1].$$

From this it follows that $\mathbb{E}_{\bar{x}\sim\mathcal{U}}[g_i(\bar{x})^{2k}] \le (2k)^{2k}\sigma_i^{2k}$ where

$$\sigma_i^2 = \begin{cases} \frac{(1-p_i)}{2^v p_i} & \text{for } p_i \in [2^{-v/10}, 1/2], \\ \frac{2(1-p_i)}{2^v} & \text{for } p_i \in [1/2, 1-2^{-v}], \\ \frac{2}{2^{2v}} & \text{for } p_i \in [1-2^{-v}, 1]. \end{cases}$$

□

Proof of Lemma 5.3. We first show the claim under the assumption that $\mathbb{E}[f] = p \ge \delta$ and later show how to get around this assumption. Define the sets

$$S_1 = \{i : p_i \in (0, 2^{-v/10}]\}, \quad S_2 = \{i : p_i \in (2^{-v/10}, 1-2^{-v}]\}, \quad S_3 = \{i : p_i \in (1-2^{-v}, 1]\}.$$

For $j \in [3]$, let $F_j(\bar{x}) = \prod_{i\in S_j} \bar{f}_i(\bar{x})$ so that $F(\bar{x}) = \prod_{j=1}^3 F_j(\bar{x})$. We will construct sandwiching approximations for each F_j and then combine them via Theorem 4.1. We assume without loss of generality that $p_i \le 1 - 2^{-w}$. Else, the i'th coordinate has bias 1 and can be ignored without changing the rest of the proof.

Sandwiching F_1. We show that $\mathsf{L}_1[F_1]$ is itself small. Observe that $\delta \le p = \prod_{i=1}^m p_i \le \prod_{i\in S_1} p_i \le 2^{-v|S_1|/10}$, which implies that $|S_1| \le 10\log(1/\delta)/v$. Thus, by Claim 5.4,

$$\mathsf{L}_1[F_1] \le \prod_{i\in S_1} \mathsf{L}_1[\bar{f}_i] \le 2^{\frac{w}{2}|S_1|} \le \left(\frac{1}{\delta}\right)^{5w/v} \le \frac{1}{\delta^{20/3}}.$$

Sandwiching F_2. Note that $F_2(\bar{x}) = \prod_{i\in S_2} \bar{f}_i(\bar{x}) = \prod_{i\in S_2} p_i \cdot (1 + g_i(\bar{x}))$. Notice that F_2 is a symmetric polynomial in the g_i's, so we will obtain sandwiching polynomials for F_2 by applying Theorem 3.1 to g_i's. As before, $\delta \le p \le \prod_{i\in S_2} p_i \le (1-2^{-v})^{|S_2|}$, so we have $|S_2| \le 2^v \log(1/\delta)$. Further we can write $\delta \le p \le \prod_{i\in S_2}(1-(1-p_i)) \le e^{-\sum_{i\in S_2}(1-p_i)}$, so that $\sum_{i\in S_2}(1-p_i) \le 2\log(1/\delta)$. By Lemma 5.6, we have $\mathbb{E}_{x\sim\{\pm1\}^n}[g_i(\bar{x})^{2k}] \le (2k)^{2k}\sigma_i^{2k}$, where $2/2^v \le \sigma_i^2 = (1-p_i)/2^{9v/10}$ for every $i \in S_2$. Hence,

$$\sum_{i\in S_2} \sigma_i^2 = \sum_{i\in S_2} \frac{1-p_i}{2^{9v/10}} \le \frac{2\log(1/\delta)}{2^{9v/10}} \le \frac{1}{\log(1/\delta)^{25}}.$$

Hence Theorem 3.1 implies the existence of $O(\delta)$ ($B = 1$ and $C = \prod_{i\in S_2} p_i \le 1$) sandwiching approximations with L_1 norm bounded by $(mt+1)^{2k}$ where

$$m = |S_2| \le 2^v \log(1/\delta) \le 2^{5v/4}, \quad t \le 2^{w/2} \le 2^v, \quad k \le \frac{5\log(1/\delta)}{\log(1/\sum_i \sigma_i^2)} \le \frac{25\log(1/\delta)}{4v}.$$

which implies the L_1 norm is bounded by $\mathrm{poly}(1/\delta)$.

Sandwiching F_3. We write

$$F_3(x) = \prod_{i\in S_3} \bar{f}_i(x) = \prod_{i\in S_3} p_i(1+g_i(x)).$$

Note that each $i \in S_3$ satisfies $1 - p_i \ge 2^{-w}$, which implies that $|S_3| \le 2^{w+1}\log(1/\delta))$. Let $\sigma_i^2 = 2/2^{2v} \ge \frac{1}{2^w}$. Then, by Lemma 5.6, $\mathbb{E}_{\bar{x}\sim\mathcal{U}}[g_i(x)^k] \le (2k)^{2k}\sigma_i^{2k}$ and we have

$$\sum_{i\in S_3} \sigma_i^2 \le \frac{2^{w+1}\log(1/\delta))}{2^{2v}} \le \frac{1}{2^{3v/5}} \le \frac{1}{\log(1/\delta)^{25}}.$$

By Theorem 3.1, F_3 has $O(\delta)$ sandwiching approximations with L_1 norm bounded by $(mt+1)^{2k}$ where

$$\begin{aligned} m &\le 2^w \log(1/p) \le 2^{3v/2}, \\ t &\le 2^{w/2} \le 2^v, \\ k &\le \frac{5\log(1/\delta)}{\log(1/\sum_i \sigma_i^2)} \le \frac{25\log(1/\delta)}{3v}. \end{aligned}$$

which implies the L_1 norm is bounded by $\mathrm{poly}(1/\delta)$.

Sandwiching F. Since each F_j has $O(\delta)$ sandwiching approximations with L_1 norm $\mathrm{poly}(1/\delta)$, by Theorem 4.1, $F = F_1F_2F_3$ has $O(\delta)$ sandwiching approximations of L_1 norm $\mathrm{poly}(1/\delta)$.

Handling all values of $\mathbb{E}[f]$. Finally, to get rid of the condition $\mathbb{E}[f] \geq \delta$, assume that $\mathbb{E}[f] \leq \delta$. If $\mathbb{E}[f] = 0$, $f = 0$ so there is nothing to prove. If $\mathbb{E}[f] > 0$, every co-ordinate f_i has at least one satisfying assignment. We repeat the following procedure until the expectation exceeds δ: pick a co-ordinate i which is not already the constant 1 function and add a new satisfying assignment to i. Such a co-ordinate i exists because $\delta < 1$. We repeat this until we get a rectangle f^t such that $\mathbb{E}[f^t] \geq \delta$. Denote the resulting sequence

$$f = f^0 \leq f^1 \cdots \leq f^t.$$

We claim that for every j,

$$\mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[f^j(x)] \leq \mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[f^{j+1}(x)] \leq 2 \mathop{\mathbb{E}}_{\bar{x}\sim\mathcal{U}}[f^j(x)].$$

The last inequality holds since at each step, we at most double the acceptance probability of the chosen co-ordinate, and hence of the overall formula. Hence we have

$$\mathbb{E}[f^t] \leq 2\,\mathbb{E}[f^{t-1}] \leq 2\delta.$$

We use the upper approximator for f^t as the upper approximator for f and 0 as the lower approximator. This gives sandwiching approximators with error at most 2δ and L_1 norm poly$(1/\delta)$.

This completes the proof of the lemma. □

5.2 A Recursive Sampler for Combinatorial Rectangles

We now use Lemma 5.3 recursively to prove Theorem 5.2. Our generator is based on a derandomized recursive sampling procedure which we describe below. The inputs are the width w and the size m of the rectangles we wish to fool and an error parameter $\delta \leq 1/2^w$.

1. Let $v_0 = w$, $v_j = \left(\frac{3}{4}\right)^j w$.
2. While $v_j \geq 50 \log\log(1/\delta)$ we sample $\bar{x}_j \in \{\{\pm 1\}^{v_{j-1}}\}^{2^{v_j}\times m}$ according to an ε_1-biased distribution for $\varepsilon \leq (1/\delta)^{c_1}$ for some large constant c_1.
3. Assume that at step t (where $t = O(\log w)$), $v_t \leq 50\log\log(1/\delta)$. Sample an input $\bar{x}_t \in (\{\pm 1\}^{v_{t-1}})^m$ from an ε_2-biased distribution where, for some large constant c_2,

$$\varepsilon_2 \leq (1/\delta)^{c_2(\log\log(1/\delta)\log\log\log(1/\delta))}.$$

We next describe how we use $\mathbf{x} = (\bar{x}_1, \ldots, \bar{x}_t)$ to output an element of $(\{\pm 1\}^w)^m$. For $k \in \{1, \ldots, t-1\}$ we denote by s_k the recursive sampling function which takes strings $\bar{x}_j \in \{\{\pm 1\}^{v_{j-1}}\}^{2^{v_j}\times m}$ for $j \in \{k+1, \ldots, t-1\}$ and $\bar{x}_t \in (\{\pm 1\}^{v_t})^m$ and produces an output string $s_k(\bar{x}_{k+1}, \ldots, \bar{x}_t) \in (\{\pm 1\}^{v_k})^m$. Set $s_{t-1}(\bar{x}_t) \equiv \bar{x}_t$. Fix $k < t-1$ and let $z = s_{k+1}(\bar{x}_{k+2}, \ldots, \bar{x}_t)$ be already defined. To define s_k, we will use z to look up entries from the matrix $\bar{x}_{k+1}$, so that the i'th coordinate of s_k will be the entry of $\bar{x}_{k+1}$ in the z_i'th row and i'th column:

$$s_k(\mathbf{x}) \equiv s_k(\bar{x}_{k+1}, \ldots, \bar{x}_t) = ((\bar{x}_{k+1})_{z_1,1}, (\bar{x}_{k+1})_{z_2,2}, \ldots, (\bar{x}_{k+1})_{z_m,m}) \in (\{\pm 1\}^{v_k})^m.$$

The above definition, though intuitive is a bit cumbersome to work with. It will be far easier for analysis to fix the input combinatorial rectangle $f : (\{\pm 1\}^w)^m \to \{0, 1\}$ and study the effect of the samplers s_k on f. Let $f^0 = f$. Each matrix $\bar{x}_j$ gives a restriction of f^{j-1}: it defines restricted co-ordinate functions $f_i^j : \{\pm 1\}^{v_j} \to \{0,1\}$ and a corresponding restricted rectangle $f^j : \{\{\pm 1\}^{v_j}\}^m \to \{0,1\}$. We only use the following property of the s_js:

$$f(s_0(\mathbf{x})) = f^1(s_1(\mathbf{x})) \cdots f^{t-1}(s_{t-1}(\mathbf{x})). \tag{5.4}$$

To analyze the last step, we use the following corollary that follows from [DETT10].

Corollary 5.7. *Every combinatorial rectangle $f : \{\{\pm 1\}^v\}^m \to \{\pm 1\}$ is δ-fooled by ε-bias spaces for $\varepsilon = (m2^v/\delta)^{-O(v \log v)}$.*

Proof. Each co-ordinate function f_i can be expressed as a CNF formula with 2^v clauses of width v. Hence we can write f as a CNF formula with $m2^v$ clauses of width v. Now apply Theorem 2.8. □

For brevity, in the following let

$$\mathcal{U} = (\{\pm 1\}^{v_0})^{2^{v_1} \times m} \times (\{\pm 1\}^{v_1})^{2^{v_2} \times m} \times \cdots \times (\{\pm 1\}^{v_{t-2}})^{2^{v_{t-1}} \times m} \times (\{\pm 1\}^{v_{t-1}})^m,$$

be the domain of $\mathbf{x}$ as defined in the generator construction.

Let $\mathcal{D}^j$ denote the distribution on $\mathcal{U}$ where $\bar{x}_i$ are sampled from an ε-biased distribution for $i < j$ and uniformly for $i \geq j$. Then, $s_0(\mathcal{D}^0)$ is the uniform distribution on $\{\{\pm 1\}^w\}^m$ whereas $s_0(\mathcal{D}^t)$ is the output of our Recursive Sampler.

Lemma 5.8. *Let $f : \{\{\pm 1\}^w\}^m \to \{0,1\}$ be a combinatorial rectangle with width w and size m. For distributions $\mathcal{D}^0$ and $\mathcal{D}^t$ defined above, we have*

$$\left| \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^0}[f(s_0(\mathbf{x}))] - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^t}[f(s_0(\mathbf{x}))] \right| \leq \delta.$$

Proof. Let $\delta' = \delta/t$. We will show by a hybrid argument that for all $j \in \{1, \ldots, t\}$

$$\left| \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^{j-1}}[f(s_0(\mathbf{x}))] - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^j}[f(s_0(\mathbf{x}))] \right| \leq \delta'. \tag{5.5}$$

In both $\mathcal{D}^{j-1}$ and $\mathcal{D}^j$, $\bar{x}_i$ is drawn from an ε-biased distribution for $i < j$, and from the uniform distribution for $i > j$. The only difference is $\bar{x}_j$ which is sampled uniformly in $\mathcal{D}^{j-1}$ and from an ε-biased distribution in $\mathcal{D}^j$.

We couple the two distributions by drawing $\bar{x}_i$ for $i < j$ according to an ε-biased distribution. By Equation (5.4), we get

$$\mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^{j-1}}[f(s_0(\mathbf{x}))] = \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^{j-1}}[f^{j-1}(s_{j-1}(\mathbf{x}))], \quad \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^j}[f(s_0(\mathbf{x}))] = \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^j}[f^{j-1}(s_{j-1}(\mathbf{x}))]$$

and our goal is now to show that

$$\left| \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^{j-1}}[f^{j-1}(s_{j-1}(\mathbf{x}))] - \mathop{\mathbb{E}}_{\mathbf{x} \sim \mathcal{D}^j}[f^{j-1}(s_{j-1}(\mathbf{x}))] \right| \leq \delta'. \tag{5.6}$$

Define the bias function F^{j-1} of the rectangle f^{j-1} as in Equation (5.3). The string $\bar{x}_j$ defines a restricted rectangle $f^j : \{\{\pm 1\}^{v_j}\}^m \to \{0,1\}$. Applying Claim 5.5 we get

$$\mathop{\mathbb{E}}_{z \sim (\{\pm 1\}^{v_j})^m}[f^j(z)] = F^{j-1}(\bar{x}_j).$$

In both distributions $\mathcal{D}^{j-1}$ and $\mathcal{D}^j$, $\bar{x}_{j+1}, \ldots, \bar{x}_t$ are distributed uniformly at random, hence $s_j(\mathcal{D}^{j-1}) = s_j(\mathcal{D}^j) \sim (\{\pm 1\}^{v_j})^m$ are uniformly distributed, and this variable is independent of $\bar{x}_j$. So we have

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}^{j-1}}[f^{j-1}(s_{j-1}(x))] = \mathop{\mathbb{E}}_{\bar{x}_j \sim \mathcal{D}^{j-1}}\left[\mathop{\mathbb{E}}_{(\bar{x}_{j+1} \ldots, \bar{x}_t) \sim \mathcal{D}^{j-1}}[f^j(s_j(\bar{x}_{j+1}, \ldots, \bar{x}_t))]\right] = \mathop{\mathbb{E}}_{\bar{x}_j \sim \mathcal{D}^{j-1}}[F^{j-1}(\bar{x}_j)],$$

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}^{j}}[f^{j-1}(s_{j-1}(x))] = \mathop{\mathbb{E}}_{\bar{x}_j \sim \mathcal{D}^{j}}\left[\mathop{\mathbb{E}}_{(\bar{x}_{j+1} \ldots, \bar{x}_t) \sim \mathcal{D}^{j}}[f^j(s_j(\bar{x}_{j+1}, \ldots, \bar{x}_t))]\right] = \mathop{\mathbb{E}}_{\bar{x}_j \sim \mathcal{D}^{j}}[F^{j-1}(\bar{x}_j)]$$

Thus it suffices to show that

$$\left| \mathop{\mathbb{E}}_{\bar{x}_j \sim \mathcal{D}^{j-1}}[F^{j-1}(\bar{x}_j)] - \mathop{\mathbb{E}}_{\bar{x}_j \sim \mathcal{D}^{j}}[F^{j-1}(\bar{x}_j)] \right| \leq \delta'$$

By Lemma 5.3, this holds true for $j \leq t-1$ provided that $\varepsilon_1 \leq \mathrm{poly}(1/\delta')$.

For $j = t$, note that this is equivalent to showing that ε_2-bias fools the rectangle f^t. By Corollary 5.7, f^t is δ' fooled by ε_2-biased spaces where

$$\varepsilon_2 = \left(\frac{m2^{v_t}}{\delta'}\right)^{-O(v_t \log v_t)} = \left(\frac{1}{\delta'}\right)^{O(\log\log(1/\delta') \log\log\log(1/\delta'))}.$$

Plugging these back into Equation (5.5), the error is bounded by $t \cdot \delta' \leq \delta$. □

To complete the proof of Theorem 5.2, we observe that the total seed-length is

$$\begin{aligned} s &= O\left((\log w)(\log(m2^w/\varepsilon_1) + \log(m2^w/\varepsilon_2)\right) \\ &= O\left(\log w\,(\log m + w + \log(1/\delta)) + \log(1/\delta)\log\log(1/\delta)\log\log\log(1/\delta)\right). \end{aligned}$$

We next state an application of our PRG to hardness amplification in NP. Say that a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is (ε, s)-hard if any circuit of size s cannot compute f on more than a $1/2 - \varepsilon$ fraction of inputs. The hardness amplification problem then asks if we can use a mildly hard function in a black-box manner to construct a much harder function. Following the works of O'Donnell [O'D04] and Healy, Vadhan and Viola [HVV04], Lu, Tsai and Wu [LTW07] showed how to construct $(2^{-\Omega(n^{2/3})}, 2^{-\Omega(n^{2/3})})$-hard functions in NP from $(1/\mathrm{poly}(n), 2^{\Omega(n)})$-hard functions in NP. Their improvement comes from using the PRG for combinatorial rectangles of Lu [Lu02] to partly derandomize the constructions of Healy, Vadhan and Viola. By using our PRG for combinatorial rectangles, Theorem 5.2, instead of Lu's generator in the arguments of Lu, Tsai and Wu immediately leads to the following improved hardness amplification within NP.

Corollary 5.9. *If there is a balanced function in* NP *that is* $(1/\mathrm{poly}(n), 2^{\Omega(n)})$*-hard, then there exists a function in* NP *that is* $(1/2^{n/\mathrm{poly}(\log n)}, 2^{n/\mathrm{poly}(\log n)})$*-hard.*

6 HSGs for Read-Once Branching Programs

In thsi section, we reduce the problem of constructing an HSG for width 3 branching programs to the problem of HSG construction for CNF formulas which are allowed to have parity functions as clauses. We start with some definitions.

A *read-once branching program* (ROBP) B of *width* d has a vertex set V partitioned into $n+1$ layers $V_0 \cup \cdots \cup V_n$ where

1. $V_0 = \{(0,0)\}$.

2. $V_t = \{(t,i)\}_{i\in[d]}$ for $t \in \{1,\ldots,n-1\}$.

3. $V_n = \{(n,1),(n,d)\}$.

The vertex $(0,0)$ is referred to as the Start state, while $(n,1)$ and (n,d) are referred to as Acc and Rej, respectively. Each vertex in $v \in V_t$ has two out-edges labeled 0 and 1, which lead to vertices $N_0(v)$ and $N_1(v)$ respectively in V_{t+1}. We refer to the set of states $\{(t,1)\}_{t=1}^n$ as the *top level* and $\{(t,d)\}_{t=1}^n$ as the *bottom level.*

A string $x \in \{0,1\}^n$ defines a path in $V_0 \times \cdots \times V_n$ beginning at Start and following the edge labeled x_i from V_i. Let $\mathsf{Path}(x) = \mathsf{Path}_0(x),\ldots,\mathsf{Path}_n(x)$ denote this sequence of states, i.e., $\mathsf{Path}_1(x) = (0,0)$, and $\mathsf{Path}_{i+1}(x) = N_{x_i}(\mathsf{Path}_i(x))$. The string x is *accepted* if $\mathsf{Path}_n(x) = \mathsf{Acc}$. Thus the branching program naturally computes a function $f : \{0,1\}^n \to \{0,1\}$. Let $\mathbb{E}[f] = \mathbb{E}_{x\sim\{0,1\}^n}[f(x)] = Pr_x[f(x)=1]$.

Let $\mathsf{BP(d,n)}$ denote the set of all $f : \{0,1\}^n \to \{0,1\}$ that can be computed by width d ROBPs. Our hitting set generator for $\mathsf{BP(3,n)}$ uses a reduction to the problem of hitting CNF formulas where clauses can be disjunctions of variables or parity functions.

Definition 6.1. *Let* $\mathsf{CNF}^{\oplus}(n)$ *denote the class of read once formulas* $f : \{0,1\}^n \to \{0,1\}$ *of the form* $f = \wedge_{i=1}^m T_i$ *where each* T_i *is either a disjunction of literals or a parity function of literals and the* T_i*s are on disjoint variables.*

Theorem 6.2. *For every* $f \in \mathsf{BP(3,n)}$ *there is an integer* k *and* $g \in \mathsf{CNF}^{\oplus}(n-k)$ *such that* $0^k \circ g^{-1}(1) \subseteq f^{-1}(1)$ *and* $\mathbb{E}[g] \geq (\mathbb{E}[f]/n)^{O(1)}$.

Given this reduction, we get a HSG for $\mathsf{BP(3,n)}$ by using the PRG for $\mathsf{CNF}^{\oplus}$ that we construct in Theorem 8.2:

Theorem 6.3. *For every* $\varepsilon > 0$*, there exists an explicit* $(\varepsilon,(\varepsilon/n)^{O(1)})$-$\mathsf{HSG}$ $G : \{0,1\}^r \to \{0,1\}^n$ *for* $\mathsf{BP(3,n)}$ *with a seed-length of* $O((\log(n/\varepsilon)) \cdot (\log\log(n/\varepsilon))^3)$.

We remark that using similar techniques, we can also achieve a seed-length of $O((\log n)(\log(1/\varepsilon)))$ which is better than the above bound for large values of ε. We defer the details of this to the full version.

The reduction in Theorem 6.2 is carried out in three steps.

- The first step (for the sake of HSGs) reduces arbitrary width 3 programs to "sudden death" width 3 programs, where the last state in every layer is a Rej state. (This step in fact works for all widths.)

- The second step reduces "sudden death" width 3 programs to intersections of width 2 programs.

- The third step reduces intersections of width 2 programs to $\mathsf{CNF}^{\oplus}$ formulae.

6.1 Reduction to Branching Programs with Sudden Death

Definition 6.4. *A width* d *BP with* sudden death *is a BP where the bottom level states are all* Rej *states. Formally this means* $N_0((t,d)) = N_1((t,d)) = (t+1,d)$ *for all* $t = 1,\ldots,n-1$. *Let* $\mathsf{BP^{Rej}(d,n)}$ *denote the set of functions computable by such programs.*

We reduce the problem of constructing hitting sets for width d BPs to for ones with sudden death.

Theorem 6.5. *For every $f \in \mathsf{BP}(\mathsf{d}, \mathsf{n})$ there is an integer k and a $g : \{0,1\}^{n-k} \to \{0,1\}$, $g \in \mathsf{BP}^{\mathsf{Rej}}(\mathsf{d}, \mathsf{n})$ such that $0^k \circ g^{-1}(1) \subseteq f^{-1}(1)$ and $\mathbb{E}[g] \geq \mathbb{E}[f]^2/2n$.*

We first setup some notation. For a vertex $v \in V$ let $p(v)$ denote the probability of reaching Acc starting from v over a uniformly random choice of $x_{i+1}, \ldots, x_n$. We call a state $v \in V$ such that $p(v) = 0$ a Rej state. We order states in V_t so that

$$p((t,1)) \geq p((t,2)) \cdots \geq p((t,d)).$$

By definition,

$$p(v) = \frac{1}{2}(p(N_0(v)) + p(N_1(v))).$$

It follows that

$$\mathbb{E}[f] = p((0,0)) \leq p((1,1)) \leq \cdots \leq p((n,1)) = 1,$$
$$\mathbb{E}[f] \geq p((1,d)) \geq p((2,d)) \geq \cdots \geq p((n,d)) = 0$$

Observe that, if $v \in V_j$ is such that $p(v) \leq \mu$, then $p((i,d)) \leq \mu$ for all $i \geq j$.

Lemma 6.6. *Let $B \in \mathsf{BP}(\mathsf{d}, \mathsf{n})$. Let R be a set of states such that $p(v) \leq \mu$ $\forall v \in R$ and let j be the first layer such that $R \cap V_j \neq \emptyset$. Let B' be obtained from B by converting all states in R into* Rej *states by redirecting the edges out of $v \in R \cap V_i$, $i \geq j$, to $((i+1,d))$. Let $p'(v)$ denote the accepting probabilities of vertices in B'. Then for all $v \in V$, we have $p'(v) \geq p(v) - \mu$.*

Proof. If $p(v) \leq \mu$ the claim is trivial, so fix v such that $p(v) > \mu$. Let $R(x)$ denote the event that we visit a vertex in R if we follow x from v in B and let $u(x)$ denote the first vertex in R that is visited by this path. Let $\mathsf{Acc}(x)$ denote the event that B accepts. We have

$$\begin{aligned}\Pr_x[R(x) \wedge \mathsf{Acc}(x)] &= \sum_{r \in R} \Pr_x[u(x) = r \wedge \mathsf{Acc}(x)] \\ &= \sum_{r \in R} \Pr_x[u(x) = r] \cdot \Pr_x[\mathsf{Acc}(x) | u(x) = r] \leq \sum_{r \in R} \Pr_x[u(x) = r] \cdot \mu \leq \mu,\end{aligned}$$

where we use $\Pr_x[\mathsf{Acc}(x) | u(x) = r] = p(r) \leq \mu$ for all $r \in R$. But then

$$\Pr_x[\mathsf{Acc}(x) \wedge \overline{R(x)}] = \Pr_x[\mathsf{Acc}(x)] - \Pr_x[\mathsf{Acc}(x) \wedge R(x)] \geq p(v) - \mu.$$

Finally, note that if we accept x without ever reaching R in B, then x is also accepted by B'. Hence $p'(v) \geq p(v) - \mu$. □

Proof of Theorem 6.5. Let B be a branching program computing a function f so that $\mathbb{E}[f] \geq \varepsilon$. Let i denote the first layer where $p((i,d)) \leq \varepsilon/2$. Note that $i \leq n$ since $p((n,d)) = 0$. Every state v up to layer $i-1$ satisfies $p(v) \geq \varepsilon/2$. Further, for every $j \geq i$, $p((i+1,d)) \leq \varepsilon/2$. Fix $k = i-2$ and let v be the state in level $i-1$ reached from Start on the string 0^k. Consider the branching program B' of length $n' = n-k$ where we make v the new start state and keep the rest of the program unchanged. The vertex set of B' is $V' = \{v\} \cup_{j=i}^{n+1} V_j$ and it computes $f' : \{0,1\}^{n'} \to \{0,1\}$ such that

$$\mathop{\mathbb{E}}_{y \in \{0,1\}^{n'}}[f'(y)] = p(v) \geq \varepsilon/2.$$

Thus, a random walk starting at v reaches the top level with probability at least $\varepsilon/2$ (since this is a necessary condition for B' to accept). For $j \in \{i, \ldots, n-1\}$, let $q(j)$ denote the probability that we reach the top level for the first time at layer j. So

$$\sum_{j=i}^{n+1} q(j) \geq \varepsilon/2.$$

Hence there exists j so that $q(j) \geq \varepsilon/2n$.

We now make the following modifications to B' to get a program B'' which is a width d program with sudden death:

- For $t \in \{i, \ldots, j-1\}$ we convert the states $(t, 1)$ into Rej states.
- For $t \in \{j, \ldots, n+1\}$ we convert the states (t, d) into Rej states.

We don't need to add an additional layer for making these modifications since we are turning one state in each layer to a Rej state.

It is clear that B'' computes a function $f'' \leq f'$. Our goal is to show that B'' accepts a large subset of inputs accepted by B'. Indeed, we claim that

$$\mathop{\mathbb{E}}_{y \in \{0,1\}^{n'}}[f''(y)] \geq \frac{\varepsilon^2}{4n}.$$

We observe that the probability that a random walk starting at v reaches the top level for the first time in layer j is the same in B'' as in B', hence it equals $q(j) \geq \varepsilon/2n$. Further, using Lemma 6.6 (to the sub-program of B' starting at $(j, 1)$) we claim that

$$p''(j, 1) \geq p'(j, 1) - \varepsilon/2 \geq \varepsilon/2$$

where we use the fact that $p'(j, 1) = p(j, 1) \geq p(1, 1) \geq \varepsilon$. Note that the probability that B'' accepts is at least $q(j)p''(j, 1) \geq \varepsilon^2/4n$, which comes from strings which reach state $(j, 1)$ and then reach Acc.

The theorem now follows by setting $g \equiv f''$. By definition, $f'' \in \mathsf{BP}^{\mathsf{Rej}}(\mathsf{d}, \mathsf{n} - \mathsf{k})$ and

$$\mathsf{0}^k \circ (f'')^{-1}(1) \subseteq \mathsf{0}^k \circ (f')^{-1}(1) \subseteq f^{-1}(1).$$

$\square$

6.2 From $\mathsf{BP}^{\mathsf{Rej}}(3)$ to Intersections of $\mathsf{BP}(2)$

We now reduce width 3 programs with sudden death to intersections of width 2 programs.

Theorem 6.7. *Let $f : \{0,1\}^n \to \{0,1\}$ be in $\mathsf{BP}^{\mathsf{Rej}}(3, \mathsf{n})$. Then, there exists a function $g : \{0,1\}^n \to \{0,1\}$ that is an intersection of functions in $\mathsf{BP}(2, \mathsf{n})$ such that $g \leq f$ and if $p = \mathbb{E}[f]$, then $\mathbb{E}[g] \geq (p/2)^{13}$.*

Throughout this section, we are given $B \in \mathsf{BP}^{\mathsf{Rej}}(\mathsf{d}, \mathsf{n})$ computing $f : \{0,1\}^n \to \{0,1\}$. Let Bad denote the set of non-reject states that have an out-edge leading to a Rej state (which are all states such that $p(v) = 0$). Further for each $x \in \{0,1\}^n$, let $\mathsf{Bad}(x)$ denote the number of Bad states visited by x.

Lemma 6.8. *We have*

$$\Pr_{x \sim \{0,1\}^n}[\mathsf{Bad}(x) \geq t] \leq 2^{-t+1}.$$

Proof. Suppose that $t \geq 1$. For $i \in [n]$, let Y_i denote the number of vertices in Bad visited by $\mathsf{Path}(x)$ in the first i layers. Then, $Y_n = \mathsf{Bad}(x)$. We claim that,

$$\mathbb{P}[Y_i = Y_{i+1} = Y_{i+2} = \cdots = Y_n \,|\, Y_i, \mathsf{Path}_i(x) \in \mathsf{Bad}] \geq 1/2. \tag{6.1}$$

This is because, if $\mathsf{Path}_i(x) \in \mathsf{Bad}$, then with probability at least $1/2$, $\mathsf{Path}_{i+1}(x)$ is a Rej state, in which case $\mathsf{Path}_j(x)$ is a Rej state for every $j \geq i+1$.

Further, if $Y_n \geq t$, then there must be an index $i < n$, where $Y_i \geq t-1$, $\mathsf{Path}_i(x) \in \mathsf{Bad}$ and $Y_n > Y_i$ (for instance i can be the least j such that $Y_j = t-1$). Therefore,

$$\begin{aligned}\mathbb{P}[Y_n \geq t] &= \mathbb{P}[\,(\exists i < n,\, Y_i \geq t-1,\, \mathsf{Path}_i(x) \in \mathsf{Bad}) \,\wedge\, (Y_n > Y_i)] \\ &= \mathbb{P}[\,(\exists i < n,\, Y_i \geq t-1,\, \mathsf{Path}_i(x) \in \mathsf{Bad})\,] \cdot \mathbb{P}[Y_n > Y_i \,|\, Y_i, \mathsf{Path}_i(x) \in \mathsf{Bad}] \\ &\leq \frac{1}{2} \cdot \mathbb{P}[\,(\exists i < n,\, Y_i \geq t-1,\, \mathsf{Path}_i(x) \in \mathsf{Bad})\,] \leq \frac{1}{2} \cdot \mathbb{P}[Y_n \geq t-1],\end{aligned}$$

where the last two inequalities follow from Equation (6.1) and the fact that Y_i's are non-decreasing. The claim now follows by induction. □

Corollary 6.9. *Let* $\Pr_{x\sim\{0,1\}^n}[f(x) = 1] = p$. *Then*

$$\mathop{\mathbb{E}}_{x \sim f^{-1}(1)}[\mathsf{Bad}(x)] = \mathop{\mathbb{E}}_{x \sim \{0,1\}^n}[\mathsf{Bad}(x) | f(x) = 1] \leq 2\log(2/p).$$

Proof. We have

$$\Pr_x[\mathsf{Bad}(x) \geq t | f(x) = 1] = \frac{\Pr_x[(\mathsf{Bad}(x) \geq t) \text{ and } (f(x) = 1)]}{\Pr_x[f(x) = 1]} \leq \frac{1}{2^{t-1}p}.$$

Let $t^* = \log(2/p)$. We then bound

$$\begin{aligned}\mathop{\mathbb{E}}_x[\mathsf{Bad}(x)|f(x) = 1] &= \sum_{t \geq 0} t \cdot \Pr_x[\mathsf{Bad}(x) = t | f(x) = 1] \\ &\leq t^* + \sum_{t > t^*} t \cdot \mathop{\mathbb{P}}_x[\mathsf{Bad}(x) = t | f(x) = 1] \\ &\leq t^* + \sum_{t > t^*} \frac{t}{2^{t-1}p} \\ &= t^* + \frac{2}{p} \cdot \left(\frac{(t^*+1)}{2^{t^*}} + \frac{1}{2^{t^*}} \right) = 2\log(1/p) + 2 \leq 2\log(2/p).\end{aligned}$$

□

The rest of our argument is specific to $d = 3$. We restrict our attention to the accepting strings $x \in f^{-1}(1)$. For each vertex $v \in V$ let $q(v) = \Pr_{x \sim f^{-1}(1)}[v \in \mathsf{Path}(x)]$. Each layer t has three states $(t, 1), (t, 2)$ and $(t, 3) \in \mathsf{Rej}$. We assume that $q((t, 1)) \geq q((t, 2)) \geq q(t, 3) = 0$ (since accepting strings never visit a Rej state). We first bound the probability mass on states in the set Bad.

Lemma 6.10. *We have*

$$\sum_{v \in \mathsf{Bad}} q(v) = \mathop{\mathbb{E}}_{x \sim f^{-1}(1)}[\mathsf{Bad}(x)].$$

Proof. We have

$$\sum_{v \in \mathsf{Bad}} q(v) = \sum_{v \in \mathsf{Bad}} \Pr_{x \sim f^{-1}(1)}[x \text{ visits } v] = \mathop{\mathbb{E}}_{x \sim f^{-1}(1)}[\mathsf{Bad}(x)],$$

by linearity of expectations. □

We partition the set Bad based on the value of $q(v)$:

$$\mathsf{Bad}^s = \left\{ v \in \mathsf{Bad} : q(v) < \frac{1}{4} \right\}, \mathsf{Bad}^l = \left\{ v \in \mathsf{Bad} : q(v) \geq \frac{1}{4} \right\}.$$

By Lemma 6.10 and Corollary 6.9 it follows that $|\mathsf{Bad}^\ell| \leq 8 \log(2/p)$.

Lemma 6.11. *We have*

$$\Pr_{x \sim f^{-1}(1)}[\mathsf{Path}(x) \cap \mathsf{Bad}^s = \emptyset] \geq (p/2)^4.$$

Proof. Since for all t, $q((t,1)) \geq 1/2$ we have $(t,1) \notin \mathsf{Bad}^s$. Sort the vertices in Bad^s according to layer, so that $\mathsf{Bad}^s = \{(t_1,2), \ldots, (t_w,2)\}$. We have

$$\Pr_{x \sim f^{-1}(1)}[\mathsf{Path}(x) \cap \mathsf{Bad}^s = \emptyset] = \prod_{i=1}^{w} \Pr_{x \sim f^{-1}(1)}[(t_i,2) \notin \mathsf{Path}(x) | (t_1,2), \ldots, (t_{i-1},2) \notin \mathsf{Path}(x)].$$

Note that if $(t_{i-1},2) \notin \mathsf{Path}(x)$ then $(t_{i-1},1) \in \mathsf{Path}(x)$. Hence conditioning on not visiting $(t_1,2), \ldots, (t_{i-1},2)$ is the same as conditioning on visiting $(t_1,1), \ldots, (t_{i-1},1)$. Further, conditioning on visiting $(t_1,1), \ldots, (t_{i-1},1)$ is the same as conditioning on $(t_{i-1},1)$. Therefore,

$$\begin{aligned}\Pr_{x \sim f^{-1}(1)}[(t_i,2) \in \mathsf{Path}(x) | (t_1,1), \ldots, (t_{i-1},1) \in \mathsf{Path}(x)] &= \Pr_{x \sim f^{-1}(1)}[(t_i,2) \in \mathsf{Path}(x) | (t_{i-1},1) \in \mathsf{Path}(x)] \\ &\leq \frac{\Pr_{x \sim f^{-1}(1)}[(t_i,2) \in \mathsf{Path}(x)]}{\Pr_{x \sim f^{-1}(1)}[(t_{i-1},1) \in \mathsf{Path}(x)]} \\ &\leq \frac{q(t_i,2)}{q(t_{i-1},1)} \leq \frac{4}{3} \cdot q((t_i,2)),\end{aligned}$$

because $q(t_{i-1},1) = 1 - q(t_{i-1},2) \geq 3/4$. Hence we have

$$\begin{aligned}\Pr_{x \sim f^{-1}(1)}[\mathsf{Path}(x) \cap \mathsf{Bad}^s = \emptyset] &= \prod_{i=1}^{w} \Pr[(t_i,2) \notin \mathsf{Path}(x) | (t_{i-1},1) \in \mathsf{Path}(x)] \\ &= \prod_{i=1}^{w} (1 - \frac{4q((t_i,2))}{3}) \geq e^{-2(\sum_{i=1}^{w} q((t_i,2)))} \geq (p/2)^4\end{aligned}$$

where we used the fact that for $z \leq 1/4$, $(1 - 4z/3) \geq e^{-2z}$ and $\sum_{v \in \mathsf{Bad}^s} q(v) \leq 2\log(2/p)$. □

We are now ready to prove Theorem 6.7.

Proof of Theorem 6.7. Observe that by the above claim, we can replace vertices in Bad^s by Rej vertices, and get a new program B' such that $B' \leq B$ and $\mathbb{E}[B'] \geq p \cdot (p/2)^4 \geq (p/2)^5$. Lastly, we handle the vertices in Bad^l, which currently have transitions to Rej. Assume that these vertices are $v_1, \ldots, v_j$ and that they read variables $x_{i_1}, \ldots, x_{i_j}$. There exists a fixing $a_{i_1}, \ldots, a_{i_j}$ of these variables such that the probability of acceptance of B' over the remaining variables is at least

$(p/2)^5$. Let $B'(a)$ denote the program obtained by hardwiring these values in B'. Now consider the program $B'' = B'(a) \wedge (x_{i_1} = a_{i_1}) \wedge \cdots \wedge (x_{i_j} = a_{i_j})$, then $B'' \leq B'$ and

$$\mathbb{E}[B''] \geq (p/2)^5 \cdot \frac{1}{2^{|\mathsf{Bad}^l|}} \geq (p/2)^{13},$$

since $|\mathsf{Bad}^l| \leq 8\log(2/p)$.

We only need to argue that $B'(a)$ and hence B'' is an intersection of width 2 branching programs. Note that $B'(a)$ is a width 2 program but with Rej states for every vertex in $\mathsf{Bad}^s = \{(t_1, 2), \ldots, (t_w, 2)\}$. But we can view $B'(a)$ as an intersection of branching programs B'_i for $i \in \{1, \ldots, w-1\}$, where B'_i has start state $(t_i, 1)$ and accept state $(t_{i+1}, 1)$. This completes the proof of the claim. □

6.3 Reducing intersections of $\mathsf{BP}(2)$ to $\mathsf{CNF}^{\oplus}$

We now perform the final step in our sequence of reductions to prove Theorem 6.2.

Theorem 6.12. *Let $f : \{0,1\}^n \to \{0,1\}$ be an intersection of width 2 BPs on disjoint sets of inputs, i.e., $f = f_1 \wedge f_2 \wedge \cdots \wedge f_m$, where each $f_i \in \mathsf{BP(2, n)}$. Then, there exists a $\mathsf{CNF}^{\oplus}$ $g : \{0,1\}^n \to \{0,1\}$ such that, $g \leq f$ and $\mathbb{E}[g] \geq \mathbb{E}[f]^{O(1)}$.*

We use the following characterization of width 2 branching programs as decision lists due to Saks and Zuckerman [SZ95] and Bshouty, Tamon and Wilson [BTW98]. For a set $S \subseteq [n]$, let $\mathrm{And}(S)$ denote all functions of the form $\wedge_j y_j$ where $j \in S$ and $y_j \in \{x_j, \bar{x_j}\}$. We define $\mathrm{Or}(S)$ and $\mathrm{XOR}(S)$ similarly. Note that all these classes contain the constant functions.

Theorem 6.13 ([SZ95, BTW98])**.** *Let $f \in \mathsf{BP}(2)$ be computed by a read-once, width 2 branching program that reads variables x_S for $S \subseteq [n]$. Then f is computable by a decision list $\mathcal{L}_f$ of the following form.*

- *$\mathcal{L}_f$ reads variables x_V for some $V \subset S$ of size k.*
- *There are $k+1$ leaves denoted $L_1, \ldots L_{k+1}$, where L_j is labeled by a function $\ell_j \in \mathrm{XOR}(S \setminus V)$*[7]

We order V according to how variables are read by $\mathcal{L}_f$ and use V^j to denote the indices of the first j variables. The condition that x reaches L_j is given by a function in $g_j \in \mathrm{And}(V^j)$. We say that L_j accepts x if $g_j(x) = 1$ and $\ell_j(x) = 1$

We derive two consequences of Theorem 6.13.

Lemma 6.14. *Let f be as in Theorem 6.13. If $\mathbb{E}[f] \geq 5/6$, then there exists $g \in \mathrm{Or}(V)$ such that $g \leq f$ and $\mathbb{E}[g] \geq \mathbb{E}[f]^9$.*

Proof. Let $\mathbb{E}[f] = 1 - \varepsilon$ for $\varepsilon \leq \frac{1}{6}$. Note that

$$\varepsilon = \sum_{j=1}^{k+1} 2^{-j} \Pr[\ell_j(x) = 0]$$

Consider the smallest j such that ℓ_j is not the constant 1 function. Since $\ell_j \in \mathrm{XOR}(S \setminus V)$ and $\ell_j \neq 1$, ℓ_j rejects with probability at least $1/2$, hence $\varepsilon \geq 2^{-j-1}$.

[7] A decision list is a decision tree where the left child of every node is a leaf labeled by one of the functions ℓ_j. On an input x, the output is computed by traversing the tree until a leaf is reached and outputting the value computed by the function at the leaf.

The condition that x reaches one of $L_1, \dots, L_{j-1}$ is given by $g \in \mathrm{Or}(V^{j-1})$. Since $\ell_1 \equiv \ell_2 \equiv \cdots \equiv \ell_{j-1}$, we have that $g \leq f$ and $\mathbb{E}[g] = 1 - 2^{-j+1} \geq 1 - 4\varepsilon$. Since $\varepsilon \leq 1/6$, the inequality $(1-4\varepsilon) \geq (1-\varepsilon)^9$ holds. □

Lemma 6.15. *Let f be as in Theorem 6.13. There exist $h_1 \in \mathrm{And}(V)$ and $h_2 \in \mathrm{XOR}(S \setminus V)$ such that if we define $h = h_1 \wedge h_2$ then $h \leq f$ and $\mathbb{E}[h] \geq \mathbb{E}[f]/3$.*

Proof. Let L_j be the highest leaf in $\mathcal{L}_f$ which is not labeled 0. Set $h_1 = g_j$ and $h_2 = \ell_j$. It is easy to see that

$$\Pr_x[h(x) = 1] = \Pr_x[L_j \text{ accepts } x] \geq \frac{1}{3} \Pr_x[f(x) = 1].$$

□

We now prove Theorem 6.12.

Proof of Theorem 6.12. Let $f = f_1 \wedge f_2 \wedge \cdots \wedge f_m$, where $f_i \in \mathsf{BP(2,n)}$. Let $p = \mathbb{E}[f]$. Then, for $I = \{i : \mathbb{E}[f_i] < 5/6\}$, $|I| < \log_{6/5}(1/p)$. For $i \notin I$, let g_i be the function obtained from Lemma 6.14 and for $i \in I$, let h_i be the function obtained from Lemma 6.15. Let $g = (\wedge_{i \notin I} g_i) \wedge (\wedge_{i \in I} h_i)$. Then, clearly $g \in \mathsf{CNF}^{\oplus}$, $g \leq f$ and

$$\mathbb{E}[g] = \prod_{i \notin I} \mathbb{E}[g_i] \cdot \prod_{i \in I} \mathbb{E}[h_i] \geq \prod_{i \notin I} \mathbb{E}[f_i]^9 \cdot \prod_{i \in I} (\mathbb{E}[f_i]/3) \geq p^9 \cdot p \cdot \frac{1}{3^{|I|}} \geq p^{14}.$$

□

6.4 HSG for $\mathsf{BP(3,n)}$

We now combine the previous sections to prove Theorems 6.2, 6.3.

Proof of Theorem 6.2. Follows immediately from combining Theorem 6.5, 6.7, 6.12. □

Proof of Theorem 6.3. Let $f \in \mathsf{BP(3,n)}$ with $\mathbb{E}[f] \geq \varepsilon$. Let g, k be as given by Theorem 6.2 applied to f so that $\mathbb{E}[g] \geq \delta = (\varepsilon/n)^c$. Let $G' : \{0,1\}^s \to \{0,1\}^n$ be a PRG for $\mathsf{CNF}^{\oplus}$ with error at most $\delta/2$. By Theorem 8.2, there exists an explicit G' with seed-length $s = O(\log(n/\varepsilon) \cdot (\log\log(n/\varepsilon))^3)$.

Define, $G : \{0,1\}^{\log n + s} \to \{0,1\}^n$ as follows:

- Sample $r \sim [n]$ and $y \sim \{0,1\}^s$.
- Output r 0s followed by the first $n - r$ bits of $G'(y)$.

We claim that G is a $(\varepsilon, (\varepsilon/n)^{c+1})$-$\mathsf{HSG}$ for $\mathsf{BP(3,n)}$.

Assume that we guess $r = k$ correctly, which happens with probability $1/n$. G then simulates g on the string $G'(y)$. Since, $\mathbb{E}[g] \geq \delta$,

$$\mathop{\mathbb{P}}_{y \in \{0,1\}^s} \left[g(G'(y)) = 1\right] \geq \mathbb{E}[g] - \delta/2 \geq \delta/2.$$

Therefore,

$$\Pr_{k, y \in \{0,1\}^s} [f(G(y)) = 1] \geq \delta/2n.$$

The theorem now follows. □

7 PRGs for Read-Once CNFs

We construct a PRG for read-once CNFs (RCNFs) with a seed-length of $O((\log n) \cdot (\log \log n)^2)$ and error $1/\mathrm{poly}(n)$. As mentioned in the introduction, previously, only generators with seed-length $O(\log^2 n)$ were known for error $1/\mathrm{poly}(n)$. Besides being of interest on its own, this construction will play an important role in our HSG for width 3 branching programs. Our main construction and its analysis are similar in spirit to what we saw for combinatorial rectangles and will be based on Theorem 3.1.

Theorem 7.1. *For every $\varepsilon > 0$, there exists an explicit* PRG $G : \{0,1\}^r \to \{\pm 1\}^n$ *that fools all* RCNF*s on n-variables with error at most ε and seed-length $r = O((\log(n/\varepsilon)) \cdot (\log \log(n/\varepsilon))^3)$.*

The core of our construction will be a structural lemma that can be summarized as follows: The bias function of a random restriction of f where each variable has a small constant probability of being set has small L_1-norm sandwiching approximators.

Along with the structural lemma we shall also exploit the fact that for any RCNF, randomly restricting a constant fraction of the inputs simplifies the formula significantly: with high probability a size m RCNF upon a random restriction has size at most $\mathrm{poly}(\log n) \cdot m^\gamma$, where $\gamma < 1$ is a fixed constant. Theorem 7.1 is then proved using a recursive construction, where we use the above arguments for $O(\log \log n)$ steps.

7.1 Sandwiching Approximators for Bias Functions

For a function $f : \{\pm 1\}^n \to [0,1]$, a subset $I \subseteq [n]$ and $x \in \{\pm 1\}^I$, define $f_I(x) : \{\pm 1\}^I \to [0,1]$ by

$$f_I(x) = \mathop{\mathbb{E}}_{y \in_u \{\pm 1\}^{[n] \setminus I}} [f(x \circ y)],$$

where $x \circ y$ denotes the appropriate concatenation: $(x \circ y)_i = x_i$ if $i \in I$ and $(x \circ y)_i = y_i$ if $i \notin I$. We call f_I the "bias function" of the *restriction* (x, I).

We will show that for a RCNF f, and I chosen in an almost k-wise independent manner, the bias function f_I has small L_1-norm sandwiching approximators with very high probability (over the choice of I).

Lemma 7.2 (Main)**.** *There exists a constant α and $c > 0$ such that the following holds for every $\varepsilon > 0$ and $\delta < (\varepsilon/n)^c$. Let $f : \{\pm 1\}^n \to \{0,1\}$ be a* RCNF *and $I \sim \mathcal{D}(\alpha, \delta)$. Then, with probability at least $1 - \varepsilon$, f_I has ε-sandwiching approximators with* L_1*-norm at most*

$$L(n, \varepsilon) = (n/\varepsilon)^{c(\log \log(n/\varepsilon))^2}.$$

Proof. Let $f = C_1 \wedge C_2 \wedge \cdots \wedge C_m$. By abuse of notation, we will let C_i denote the set of variables appearing in C_i as well. In our analysis we shall group the clauses based on their widths. Let $\beta = 1 + 1/6$.

We first handle the case where f has bias at least ε, i.e., $\mathbb{P}[f(x) = 1] \geq \varepsilon$. Let $W_\ell = c_1 \log \log(n/\varepsilon)$ and $W_u = \log_2 m$ for c_1 a constant to be chosen later. Let f_ℓ be the RCNF containing all clauses of width less than W_ℓ and f_u the RCNF containing all clauses of width at least W_u. Let $T = \log_\beta(W_u/2W_\ell)$. For $w \in W_B \equiv \{\lfloor W_\ell \beta^r \rfloor : 0 \leq r \leq T\}$, let f_w be the RCNF containing all clauses with width in $[w, \beta w)$. Then,

$$f \equiv f_\ell \wedge (\wedge_{w \in W_B} f_w) \wedge f_u. \tag{7.1}$$

We will show that each of the functions f_ℓ, f_w, f_u have good sandwiching approximators. We then use Theorem 4.1 to conclude that f has good sandwiching approximators. The claim for f_ℓ follows immediately from Theorem 2.7. The main challenge will be in analyzing f_w (the analysis for f_u is similar). To show that f_w has good sandwiching approximators, we shall appeal to Theorem 3.2.

Observe that as f has bias at least ε, the number of clauses of width at most w in f is at most $2^w \log(1/\varepsilon)$. We will repeatedly use this fact. Let $\varepsilon_1 = \varepsilon/\mathrm{poly}(n, 1/\varepsilon)$ to be chosen later.

Sandwiching f_ℓ. As each clause in f_ℓ has width at most W_ℓ, the number of clauses in f_ℓ is at most $m_\ell \leq 2^{W_\ell} \log(1/\varepsilon) \leq (\log(n/\varepsilon))^{c_1+1}$. Thus, by Theorem 2.7, f_ℓ has ε_1-sandwiching approximators of L_1-norm at most $m_\ell^{O(\log(1/\varepsilon_1))} = (\log(n/\varepsilon))^{O(\log(1/\varepsilon_1))}$. As L_1-norm does not increase under averaging over a subset of the variables, it follows that $(f_\ell)_I$ has ε_1-sandwiching polynomials with the same L_1 norm bound:

$$(f_\ell)_I \text{ has } \varepsilon_1\text{-sandwiching approximators with } \mathsf{L}_1\text{-norm at most } (\log(n/\varepsilon))^{O(\log(1/\varepsilon_1))}. \tag{7.2}$$

Sandwiching f_w. Fix a $w \in W_B$. Note that f_w has $m_w < 2^{\beta w} \log(1/\varepsilon)$ clauses. Without loss of generality, suppose that $f_w = C_1 \wedge C_2 \wedge \cdots \wedge C_{m_w}$. Let $I \sim \mathcal{D}(\alpha, \delta)$.

Let $J \subseteq [m_w]$ be the set of all *good* clauses, $J = \{j : |C_j \cap I| \leq w/3\}$. Decompose $f_w = f'_w \wedge f''_w$, where $f'_w = \wedge_{j \in J} C_j$. We first show that $(f'_w)_I$ has good sandwiching approximators. We then show that f''_w has a small number, $\mathrm{poly}(\log(n/\varepsilon))$, of clauses with high probability over I. The intuition for the first step is that if each $|C_j \cap I|$ is small, then the randomness in the remaining variables damps the variance of the bias function f_I enough to guarantee existence of good sandwiching approximators via Theorem 3.2. For the second step, intuitively, as I picks each element with probability at most α, we expect $|C_j \cap I|$ to be about $\alpha|C_j| < \alpha(\beta w) \ll w/3$. Thus, the probability that $|C_j \cap I|$ is more than $w/3$ should be small so that the total number of bad clauses is small with high probability.

For brevity, suppose that $f'_w = C_1 \wedge \cdots \wedge C_{m'}$ and let $w_j = |C_j| \in [w, \beta w)$. For $x \in \{\pm 1\}^I$, and $j \in [m']$, define $g'_j : \{\pm 1\}^I \to [-1, 1]$ by

$$g'_j(x) = \begin{cases} -1/2^{w_j} & \text{if } x \text{ satisfies } C_j \cap I \\ 1/2^{|C_j \setminus I|} - 1/2^{w_j} & \text{otherwise} \end{cases}.$$

Then, for $p_j = 1 - 1/2^{w_j}$,

$$(f'_w)_I(x) = \prod_{j \in J} \left((1 - 1/2^{w_j}) - g'_j(x)\right) = \prod_{j \in J} \left(p_j - g'_j(x)\right) = \left(\prod_{j \in J} p_j\right) \cdot \prod_{j \in J} \left(1 - \frac{g'_j(x)}{p_j}\right).$$

Let $g_j(x) = g'_j(x)/p_j$. Then,

$$(f'_w)_I(x) = \prod_{j \in J} p_j \cdot \prod_{j \in J} (1 - g_j(x)).$$

By expanding the above expression we can write $(f'_w)_I(x) = \sum_{k=1}^{m'} c_k S_k(g_1, \ldots, g_{m'})$ where the coefficients c_k are at most 1 in absolute value. We will show that $g_1, \ldots, g_{m'}$ satisfy the conditions of Theorem 3.2.

Clearly, $g_1, \dots, g_{m'}$ are on disjoint subsets of x. Note that $g'_j(x) \in [-1/2^{2w/3}, 1/2^{2w/3}]$. Hence, as $p_j \geq 1/2$, $g_j(x) \in [-\sigma, \sigma]$ for $\sigma = 2/2^{2w/3}$. Now, as $w \geq c_1 \log\log(1/\varepsilon)$, $m' \leq 2^{\beta w} \log(1/\varepsilon) \leq 2^{(\beta+1/c_1)w}$. Thus, for $c_1 > 12$,

$$\sigma = \frac{2}{2^{2w/3}} \leq \frac{1}{(m')^{1/2+1/12}}.$$

Finally, note that each g_j has L_1-norm at most 2. This is because any clause, and hence g'_j, has L_1-norm at most 1. Therefore, the functions g_j satisfy the conditions of Theorem 3.2. Thus,

$$(f'_w)_I \text{ has } \varepsilon_1\text{-sandwiching approximators with } \mathsf{L}_1\text{-norm at most poly}(1/\varepsilon_1). \tag{7.3}$$

We are almost done, but for (f''_w). We will show that with high probability over I, (f''_w) has $O(\log(n/\varepsilon))$ clauses. To do so we will follow a standard argument for showing large deviation bounds using bounded independence.

For $i \in [w]$ and $j \in [m_w]$, let X_{ij} be the indicator variable that is 1 if the variable corresponding to the i'th literal in the j'th clause of f_w is included in I and 0 otherwise. Let

$$X = S_k\left(S_{w/3}(X_{11}, X_{21}, \dots, X_{w1}), \dots, S_{w/3}(X_{1m_w}, X_{2m_w}, \dots, X_{wm_w})\right).$$

Then, for any k,

$$\mathbb{P}[size(f''_w) \geq k] \leq \mathbb{E}[X].$$

To see this observe that whenever $size(f''_w) \geq k$, X is at least 1. Let us first calculate this expectation when the variables X_{ij} are truly independent. In this case, as $m_w \leq 2^w \log(1/\varepsilon)$, and each clause has at most βw variables,

$$\begin{aligned}\mathbb{E}[X] &= \binom{m_w}{k} \cdot \binom{\beta w}{w/3} \cdot \alpha^{wk/3} \\ &\leq 2^w \cdot \left(\frac{\beta e \log(1/\varepsilon)}{k}\right)^k \cdot (8\alpha)^{wk/3}.\end{aligned}$$

Therefore, for $\alpha = 1/32$ and $k = c_2 \max(\log(n/\varepsilon)/w, 1)$ for c_2 sufficiently large, $\mathbb{E}[X] \leq \varepsilon/2n$. Now, as the actual variables X_{ij} are δ-almost independent, and the polynomial defining X has at most $\binom{m_w}{k} \cdot \binom{\beta w}{w/3}$ terms, the expectation for $I \sim \mathcal{D}$ can be bounded by

$$\mathbb{E}[X] \leq \frac{\varepsilon}{2\log n} + \delta \cdot \binom{m_w}{k} \cdot \binom{\beta w}{w/3} = \frac{\varepsilon}{2n} + \delta \cdot 2^{O(wk)} \leq \frac{\varepsilon}{n},$$

for $\delta \leq (\varepsilon/n)^c$ for c a sufficiently large constant. Combining the above equations and applying Theorem 2.7, we get that with probability at least $1 - \varepsilon/n$, (f''_w) has ε_1-sandwiching polynomials with L_1-norm at most

$$k^{O(\log(1/\varepsilon_1))} = (\log(n/\varepsilon))^{O(\log(1/\varepsilon_1))}.$$

Therefore, from Equation (7.3) and Theorem 4.1, with probability at least $1 - \varepsilon/n$,

$$(f_w)_I \text{ has } O(\varepsilon_1)\text{-sandwiching approximators with } \mathsf{L}_1\text{-norm at most } (\log(n/\varepsilon))^{O(\log(1/\varepsilon_1))}. \tag{7.4}$$

Sandwiching f_u: A careful examination of the argument for f_w reveals that we used two main properties: there are at most $2^{\beta w}\log(1/\varepsilon)$ clauses in f_w and every clause has length at least w. Both of these are trivially true for f_u with $w = W_u$. Thus, the same argument applies. In particular, with probability at least $1-\varepsilon/n$,

$$(f_u)_I \text{ has } O(\varepsilon_1)\text{-sandwiching approximators with } \mathsf{L}_1\text{-norm at most } (\log(n/\varepsilon))^{O(\log(1/\varepsilon_1))}. \quad (7.5)$$

Now, observe that

$$f_I(x) = (f_\ell)_I(x)\cdot(f_u)_I(x)\cdot\prod_{w\in W_B}(f_w)_I(x).$$

Therefore, we can apply Theorem 4.1. In particular, by Equations 7.2, 7.4, 7.5, and a union bound, for $b = |W_B|+2 = O(\log\log n)$ we have: with probability at least $1-\varepsilon$, f_I has $(16^b\varepsilon_1)$-sandwiching polynomials with L_1-norm at most

$$4^b\cdot\left((\log(n/\varepsilon))^{O(b\,\log(1/\varepsilon_1))}\right) = (1/\varepsilon_1)^{O((\log\log(n/\varepsilon))^2)}.$$

The lemma now follows by setting $\varepsilon_1 = \varepsilon/n^{O(1)}$.

Handling Small Bias Case. We now remove the assumption that $\mathbb{E}[f]\geq\varepsilon$. Suppose $\mathbb{E}[f]\leq\varepsilon$. Consider the formula f' obtained from f by removing clauses in f until the first time $\mathbb{E}[f']$ exceeds ε. Then, $f\leq f'$ and $\varepsilon\leq\mathbb{E}[f']\leq 2\varepsilon$ (as each clause has probability at most $1/2$ of being false). We can use the upper approximator for f' as an upper approximator for f and constant zero as a lower approximator. This completes the proof of lemma. □

7.2 Restrictions Simplify RCNFs

We next argue that for restrictions (x,I) where (x,I) are chosen from almost-independent distributions as in the previous section, RCNFs simplify significantly and in particular have few surviving clauses with very high probability.

Let $I\sim\mathcal{D}(\alpha,\delta)$ be as in Lemma 7.2 and $x\sim\mathcal{D}$ be chosen from a δ_1-biased distribution with $\delta_1 = 1/\mathrm{poly}(n)$. We will show that fixing the variables in I according to x will make the number of clauses drop polynomially. Let α,β be the constants from Lemma 7.2.

Lemma 7.3. *There exists constants $c_2,\gamma>0$ such that the following holds for $\delta,\delta_1<(\varepsilon/n)^{c_2}$. Let $I\sim\mathcal{D}(\alpha,\delta)$ and $x\sim\mathcal{D}$ where $\mathcal{D}$ is a δ_1-biased distribution on $\{\pm1\}^n$. Let $f:\{\pm1\}^n\to\{0,1\}$ be a* RCNF *with $\mathbb{E}[f]\geq\varepsilon$. Let $g:\{\pm1\}^{[n]\setminus I}\to\{0,1\}$ be the* RCNF *obtained from f by fixing the variables in I to x. Then, with probability at least $1-\varepsilon$ over the choice of (x,I), g is a* RCNF *with at most $(\log(n/\varepsilon))^{c_2}\cdot m^{1-\gamma}$ clauses.*

Proof. As in the proof of Lemma 7.2, we shall do a case analysis based on the width of the clauses. Let f_ℓ, f_w, f_u and W_B be as in Equation (7.1). Note that the number of clauses in f_ℓ is at most $2^{W_\ell}\log(1/\varepsilon) = \mathrm{poly}(\log(n/\varepsilon))$. We will now reason about each of the f_w's for $w\in W_B$. The argument for f_u is similar and is omitted.

Let f_w have m_w clauses, where $m_w > 8\log(1/\varepsilon)$, otherwise there is nothing to prove. Without loss of generality, suppose that $f_w = C_1\wedge C_2\wedge\cdots\wedge C_{m_w}$ and $w_j = |C_j|$. Let Y_j be the indicator variable that is 1 if C_j survives in g (i.e., is not fixed to be *true*) and 0 otherwise. We first do the calculations assuming that the variables in x and I are truly independent and later transfer these bounds to the almost independent case.

Observe that $\mathbb{P}[Y_j = 1] = (1-\alpha/2)^{w_j} \leq (1-\alpha/2)^w$. Let $M, k < M$ be parameters to be chosen later. Then, as in the proof of Lemma 7.2,

$$\mathbb{P}\left[\sum_j Y_j > M\right] \cdot \binom{M}{k} \leq \mathbb{E}\left[S_k(Y_1, \ldots, Y_{m_w})\right] \leq \binom{m_w}{k} \cdot \left(1 - \frac{\alpha}{2}\right)^{wk}.$$

Here, the first inequality follows from observing that if $\sum_j Y_j > M$, then $S_k(Y_1, \ldots, Y_{m_2})$ is at least $\binom{M}{k}$. Therefore,

$$\mathbb{P}\left[\sum_j Y_j > M\right] \leq \left(\frac{m_w e}{M}\right)^k \cdot \left(1 - \frac{\alpha}{2}\right)^{wk}.$$

Now, setting $M = m_w^{1-\gamma}(e \log(1/\varepsilon))$ for a sufficiently small constant γ and using the fact that $m_w < 2^{\beta w} \log(1/\varepsilon)$, it follows that

$$\mathbb{P}[\sum_j Y_j > M\,] \leq \left(\frac{(2-\alpha)2^{\beta\gamma}}{2}\right)^{wk} < 2^{-\Omega(wk)},$$

for γ a sufficiently small constant. Thus, for $k = c_3 \max(\log(n/\varepsilon)/w, 1)$ and c_3 a sufficiently large constant, $\mathbb{P}[\sum_j Y_j > M\,] < \varepsilon/2n$. Now, as in the proof of Lemma 7.2, transferring the above calculations to the case of almost independent distributions only incurs an additional error of

$$err = (\delta + \delta_1) \cdot \binom{m_w}{k} \cdot 2^{\beta w} = (\delta + \delta_1) \cdot \mathrm{poly}(n, 1/\varepsilon).$$

Therefore, for $\delta, \delta_1 < (\varepsilon/n)^{c'}$ for a sufficiently large constant c', we get $\mathbb{P}[\sum_j Y_j > M\,] < \varepsilon/n$.

Hence, by a union bound over $w \geq W_\ell$, with probability at least $1-\varepsilon$, the number of surviving clauses in g is at most

$$size(f_\ell) + (e\log(1/\varepsilon)) \cdot \sum_{w \in W_B} m_w^{1-\gamma} \leq \mathrm{poly}(\log n) + (e \log(1/\varepsilon)) \cdot |B|^\gamma \cdot (\sum_w m_w)^{1-\gamma} <$$
$$\mathrm{poly}(\log n) + (e\log(1/\varepsilon)) \cdot |B|^\gamma \cdot m^{1-\gamma},$$

where the first inequality follows from the power-mean inequality. The claim now follows. □

In our recursive analysis we will also have to handle RCNFs that need not have high acceptance probabilities. The following corollary will help us do this.

Corollary 7.4. *Let constants c_2, γ and $\delta, \delta_1, I \sim \mathcal{D}(\alpha, \delta), x \sim \mathcal{D}$ be as in Lemma 7.3. Let $f : \{\pm 1\}^n \to \{0,1\}$ be a RCNF, and let $g : \{\pm 1\}^{[n]\setminus I} \to \{0,1\}$ be the RCNF obtained from f by fixing the variables in I to x. Then, with probability at least $1-\varepsilon$ over the choice of (x, I), there exist two RCNFs g_ℓ, g_u of size at most $(\log(n/\varepsilon))^{c_2} \cdot m^{1-\gamma}$ such that $g_\ell \leq g \leq g_u$ and $\mathbb{E}[g_u] - \mathbb{E}[g_\ell] \leq \varepsilon$.*

Proof. If $\mathbb{E}[f] \geq \varepsilon/2$, the claim follows from Lemma 7.3. Suppose $\mathbb{E}[f] \leq \varepsilon/2$. Let f' be the formula obtained from f by throwing away clauses until $\mathbb{E}[f']$ exceeds $\varepsilon/2$. Then, $\varepsilon/2 \leq \mathbb{E}[f'] \leq \varepsilon$. Let g' be the RCNF obtained from f' by restricting the variables in I to x. The claim now follow by applying Lemma 7.3 to f' and setting $g_\ell \equiv 0$ and $g_u \equiv g'$. □

7.3 A Recursive PRG Construction for RCNFs

We now use Lemmas 7.2 and 7.3 recursively to prove Theorem 7.1. The main intuition is as follows.

Let $\varepsilon = 1/\mathrm{poly}(n)$. Lemma 7.2 ensures that with high probability over the choice of I, f_I is fooled by small-bias spaces with bias $n^{-O((\log\log n)^2)}$ which can be sampled from using $O((\log n)(\log\log n)^2)$ random bits. Note that I can be sampled using $O(\log n)$ random bits.

Consider any fixed $I \subseteq [n]$ and $x \in \{\pm1\}^I$. We wish to apply the same argument to $f_{(x,I)} : \{\pm1\}^{[n]\setminus I} \to \{0,1\}$ to pick another set $I_1 \subseteq [n]$ and $x_1 \in \{\pm1\}^{I_1}$ and so on. The saving factor will be that most of the clauses in f will be determined by the assignment to x. In particular, by Lemma 7.3, with probability $1 - 1/\mathrm{poly}(n)$, $f_{(x,I)}$ has at most $\tilde{O}(n^{1-\gamma})$ clauses. By repeating this argument for $t = O_\gamma(\log\log n)$ steps we will get a RCNF with at most $\mathrm{poly}(\log n)$ clauses, which can be fooled directly. The total number of random bits used in this process will be $O((\log n)(\log\log n)^3)$.

Fix $\varepsilon > 0$ and let constants α, c be as in Lemma 7.2. Let $\mathcal{D}(\alpha,\delta)$ be a δ-almost independent distribution on $2^{[n]}$ with bias α. Finally, let $\mathcal{D}(\delta_1), \mathcal{D}(\delta_2)$ denote δ_1-biased and δ_2-biased distributions on $\{\pm1\}^n$ respectively for δ_1, δ_2 to be chosen later. Let $T = C\log\log n$ for C to be chosen later. Consider the following randomized algorithm for generating a string $z \in \{\pm1\}^n$.

- For $t = 1,\dots,T$, generate independent samples $z^1,\dots,z^T \sim \mathcal{D}(\delta_1)$ and $J_1,\dots,J_T \sim \mathcal{D}(\alpha,\delta)$.
- Let $I_1 = J_1$ and $I_t = J_t \setminus \left(\cup_{r=1}^{t-1} I_r\right)$ for $2 \le t \le T$. This is equivalent to sampling I_t from a δ-almost independent distribution with bias α from the set of subsets of as yet "uncovered" elements $[n] \setminus \cup_{r=1}^{t-1} I_r$.
- Let $x^t = (z^t)_{I_t}$. This is equivalent to sampling x^t using a δ_1-biased distribution over $\{\pm1\}^{I_t}$.
- Let $I = \cup_{t=1}^{T} I_t$ and $x = x^1 \circ x^2 \circ \cdots \circ x^T \in \{\pm1\}^I$ be the appropriate concatenation: for $i \in I$, $(x_i) = (x^t)_i$ if $i \in I_t$.
- Let $y \sim \mathcal{D}(\delta_2)$. The final generator output is defined by

$$G(z^1,\dots,z^T,J_1,\dots,J_T,y) = z, \text{ where } z_i = x_i \text{ if } i \in I \text{ and } z_i = y_i \text{ otherwise.} \tag{7.6}$$

To analyze our generator we first show that the restriction (x,I) preserves the bias of RCNFs. Let $L(n,\varepsilon)$ be the bound from Lemma 7.2.

Lemma 7.5. *For x, I defined as above, with probability at least $1 - \varepsilon T$ over the choice of I, for every* RCNF $f : \{\pm1\}^n \to \{0,1\}$,

$$\left| \mathop{\mathbb{E}}_{x}[f_I(x)] - \mathop{\mathbb{E}}_{y\in_u\{\pm1\}^I}[f_I(y)] \right| < \delta_1 \cdot L(n,\varepsilon) \cdot T + 2\,\varepsilon T.$$

Proof. We will prove the claim by a hybrid argument. For $j \le T$, let $y^j \sim \{\pm1\}^{I_j}$ and let $\mathcal{D}^j$ denote the distribution of $x^1 \circ x^2 \circ \cdots \circ x^j \circ y^{j+1} \circ \cdots \circ y^T$(the concatenation is done as in the definition of x). Note that $\mathcal{D}^{j-1}$ and $\mathcal{D}^j$ differ only in the j'th concatenation element, x^j, y^j. Further, $\mathcal{D}^0$ is uniformly distributed on $\{\pm1\}^I$ and $\mathcal{D}^T$ is the distribution of x. We will show that with probability at least $1-\varepsilon$, over the choice of I,

$$\left| \mathop{\mathbb{E}}_{a\sim\mathcal{D}^{j-1}}[f_I(a)] - \mathop{\mathbb{E}}_{a\sim\mathcal{D}^j}[f_I(a)] \right| < \delta_1 \cdot L(n,\varepsilon),$$

We couple the distributions $\mathcal{D}^{j-1}$ and $\mathcal{D}^j$ by drawing x^i for $i < j$ and let $I^j = \cup_{r \le j} I_r$. Now, as $y^{j+1}, \ldots, y^T$ are chosen uniformly at random,

$$\mathop{\mathbb{E}}_{a \sim \mathcal{D}^{j-1}}[f_I(a)] = \mathbb{E}\left[f_{I^j}(x^1 \circ \cdots \circ x^{j-1} \circ x^j)\right],$$
$$\mathop{\mathbb{E}}_{a \sim \mathcal{D}^{j}}[f_I(a)] = \mathbb{E}\left[f_{I^j}(x^1 \circ \cdots \circ x^{j-1} \circ y^j)\right].$$

Consider any fixing of the variables $x^1, \ldots, x^{j-1}$ and $I_1, \ldots, I_{j-1}$ and let $g : \{\pm 1\}^{[n] \setminus \cup_{r<j} I_r} \to \{0,1\}$ be the RCNF obtained from f under this fixing. Then, by Lemma 7.2, g_{I_j} is fooled by small-bias spaces: with probability $1 - \varepsilon$ over the choice of I_j,

$$\left|\mathop{\mathbb{E}}_{x^j}\left[g_{I_j}(x^j)\right] - \mathop{\mathbb{E}}_{y^j}\left[g_{I_j}(y^j)\right]\right| \le \delta_1 \cdot L(n, \varepsilon).$$

Combining the above three equations, we have with probability at least $1 - \varepsilon$ over I_j,

$$\begin{aligned}
\left|\mathop{\mathbb{E}}_{a \sim \mathcal{D}^{j-1}}[f_I(a)] - \mathop{\mathbb{E}}_{a \sim \mathcal{D}^{j}}[f_I(a)]\right| &= \left|\mathbb{E}\left[f_{I^j}(x^1 \circ \cdots \circ x^{j-1} \circ x^j)\right] - \mathbb{E}\left[f_{I^j}(x^1 \circ \cdots \circ x^{j-1} \circ y^j)\right]\right| \\
&= \left|\mathop{\mathbb{E}}_{x^1, \ldots, x^{j-1}} \mathop{\mathbb{E}}_{x^j}\left[g_{I_j}(x^j)\right] - \mathop{\mathbb{E}}_{x^1, \ldots, x^{j-1}} \mathop{\mathbb{E}}_{y^j}\left[g_{I_j}(y^j)\right]\right| \\
&\le \mathop{\mathbb{E}}_{x^1, \ldots, x^{j-1}}\left[\left|\mathop{\mathbb{E}}_{x^j}\left[g_{I_j}(x^j)\right] - \mathop{\mathbb{E}}_{y^j}\left[g_{I_j}(y^j)\right]\right|\right] \le \delta_1 \cdot L(n, \varepsilon).
\end{aligned}$$

The claim now follows by taking a union bound for $j = 1, \ldots, T$. □

We are now ready to prove our main PRG construction. The idea is to combine Lemmas 7.3, 7.5. For (x, I) chosen as in Lemma 7.5 we do not change the bias of the restricted function, on the other hand by iteratively applying 7.3 we can show that the resulting restricted RCNF has $(\log n)^{O(\log \log n)}$ clauses and hence is fooled by $n^{-O((\log \log n)^2)}$-biased distributions.

Proof of Theorem 7.1. Let I, x, y, z be as defined in Equation (7.6). Fix a RCNF $f : \{\pm 1\}^n \to \{0,1\}$. Let $g : \{\pm 1\}^{[n] \setminus I} \to \{0,1\}$ be the RCNF obtained by from f by fixing the variables in I to x. Let $I' = [n] \setminus I$. Note that

$$f_I(x) = \mathop{\mathbb{E}}_{y' \sim \{\pm 1\}^{I'}}\left[g(y')\right]. \tag{7.7}$$

We next argue that g is fooled by small-bias spaces with high probability over the choice of x, I. Observe that g can be viewed as obtained from f by iteratively restricting f according to $(x^1, I_1), (x^2, I_2), \ldots, (x^T, I_T)$ and all of these are independent of one another. Therefore, by Corollary 7.4 and a union bound, with probability at least $1 - \varepsilon \cdot T$, g has $O(\varepsilon T)$-sandwiching RCNFs g_ℓ, g_u of size at most

$$M = (\log(n/\varepsilon))^{c_2 T} \cdot m^{(1-\gamma)^T} = (\log(n/\varepsilon))^{O(\log \log n)},$$

for $T = C \log \log n$ and C a large constant. Hence, by Theorem 2.7, g_ℓ, g_u are ε-fooled by δ_2-biased distributions for $\delta_2 = M^{-O(\log(1/\varepsilon))}$. As g_ℓ, g_u sandwich g, it follows that g is $O(\varepsilon T)$-fooled by δ_2-biased distributions. As the above is true with probability at least $1 - \varepsilon T$ over the choice of (x, I), by taking expectation over (x, I) we get (y is δ_2-biased)

$$\mathop{\mathbb{E}}_{x, I}\left[\left|\mathop{\mathbb{E}}_{y}[g(y)] - \mathop{\mathbb{E}}_{y' \sim \{\pm 1\}^{I'}}\left[g(y')\right]\right|\right] = O(\varepsilon T). \tag{7.8}$$

Combining Equations 7.7, 7.8, we get

$$\begin{aligned}\mathbb{P}[f(z)=1] = \mathop{\mathbb{E}}_{x,I}\left[\mathop{\mathbb{E}}_{y}[g(y)]\right] &= \mathop{\mathbb{E}}_{x,I}\left[\mathop{\mathbb{E}}_{y'\sim\{\pm1\}^{I'}}[g(y')]\right] \pm O(\varepsilon T)\\ &= \mathop{\mathbb{E}}_{x,I}[f_I(x)] \pm O(\varepsilon T).\end{aligned}$$

Finally, note that for any $I \subseteq [n]$,

$$\mathop{\mathbb{P}}_{z'\sim\{\pm1\}^n}[f(z')=1] = \mathop{\mathbb{E}}_{x'\sim\{\pm1\}^I}[f_I(x')].$$

Combining the above two equations with Lemma 7.5, we get

$$\begin{aligned}\left|\mathbb{P}[f(z)=1] - \mathop{\mathbb{P}}_{z'\sim\{\pm1\}^n}[f(z')=1]\right| &\le \left|\mathop{\mathbb{E}}_{x,I}[f_I(x)] - \mathop{\mathbb{E}}_{x'\sim\{\pm1\}^I}[f_I(x')]\right| + O(\varepsilon T)\\ &\le \delta_1 \cdot L(n,\varepsilon) + O(\varepsilon T).\end{aligned}$$

Therefore, by setting $\delta_1 = \varepsilon/L$ the above error is at most $O(\varepsilon T)$. The number of bits used by the generator is

$$\begin{aligned}T \text{ (bits needed for } x^1, I_1) + \text{ (bits needed for y)} &= T\cdot O(\log n + \log(1/\delta_1)) + O(\log n + \log(1/\delta_2))\\ &= O((\log(n/\varepsilon))\cdot(\log\log(n/\varepsilon))^3).\end{aligned}$$

The theorem now follows by rescaling $\varepsilon = \varepsilon'/c'(\log\log n)$ for a large constant c'. □

8 A PRG for CNF$^{\oplus}$

We construct a PRG for the class of CNF$^{\oplus}$. The generator will be the same as in Theorem 7.1. The analysis will also be similar and in fact follow easily from Theorem 7.1. To do this, we shall use the following simple claim.

Lemma 8.1. *Let $f : \{\pm1\}^n \to \{0,1\}$ be a conjunction of parity constraints on n variables. Then, f has* L_1*-norm at most* 1.

Proof. Let $S_1, S_2, \ldots, S_m$ be the subsets defining the parity constraints in f. Then,

$$f(x) = \prod_{j=1}^{m}\left(\frac{1-\prod_{i\in S_j} x_i}{2}\right).$$

The lemma now follows. □

Theorem 8.2. *For every $\varepsilon > 0$, there exists an explicit* PRG *$G : \{0,1\}^r \to \{\pm1\}^n$ that fools all* CNF$^{\oplus}$ *formulas on n-variables with error at most ε and seed-length $r = O((\log(n/\varepsilon))\cdot(\log\log(n/\varepsilon))^3)$.*

Proof. Let G be the generator from Theorem 7.1. We will show that G fools CNF$^{\oplus}$ as well. This does not follow in a black-box manner from Theorem 7.1, but we will show analogues of Theorem 2.7, Lemma 7.2 and Lemma 7.3 hold so that the rest of the proof of Theorem 7.1 can be used as is.

Let $f : \{\pm1\}^n \to \{0,1\}$ be a CNF$^{\oplus}$ of size m. Let $f = g \wedge h$, where g has all the parity constraints of f and h the clauses.

First observe that by Lemma 8.1 and Theorem 2.7, a similar statement holds for f. Let P_ℓ, P_u be the ε-sandwiching approximators for h as guaranteed by Theorem 2.7. Then, $P'_\ell := g \cdot P_\ell$, $P'_u = g \cdot P_u$ are ε-sandwiching approximators for f and the L_1-norm of P'_ℓ (P'_u) is bounded by the L_1-norm of P_ℓ (P_u) by Lemma 8.1.

Note that for any subset $I \subseteq [n]$, $g_I : \{\pm 1\}^I \to \{0, 1\}$ is a constant function. Therefore, $f_I(x) = c_I \cdot h_I(x)$, where $c_I \leq 1$. Thus, by applying Lemma 7.2 to h, we get an analogous statement for f.

Finally, we show an analogue of Lemma 7.3. Suppose that f has acceptance probability at least ε. Then, g has at most $\log_2(1/\varepsilon)$ clauses. Therefore, by Lemma 7.3 applied to h, we also get a similar statement for f with a slightly worse constant of $c'_2 = c_2 + 1$. By arguing as in the proof of Corollary 7.4, we get a similar statement for f.

Examining the proof of Lemma 7.1 shows that given the above analogues of Theorem 2.7, Lemma 7.2 and Lemma 7.3, the rest of the proof goes through. The theorem follows. □

References

[AAI$^+$01] Manindra Agrawal, Eric Allender, Russell Impagliazzo, Toniann Pitassi, and Steven Rudich, *Reducing the complexity of reductions*, Computational Complexity **10** (2001), no. 2, 117–138.

[AGHP92] Noga Alon, Oded Goldreich, Johan Håstad, and René Peralta, *Simple construction of almost k-wise independent random variables*, Random Struct. Algorithms **3** (1992), no. 3, 289–304.

[Ajt83] Miklos Ajtai, *Σ_1^2-formula on finite structures*, Ann. Pure. Appl. Logic **24** (1983), 1–48.

[ASWZ96] Roy Armoni, Michael E. Saks, Avi Wigderson, and Shiyu Zhou, *Discrepancy sets and pseudorandom generators for combinatorial rectangles*, FOCS, 1996, pp. 412–421.

[AW85] Miklós Ajtai and Avi Wigderson, *Deterministic simulation of probabilistic constant depth circuits (preliminary version)*, FOCS, 1985, pp. 11–19.

[Baz09] Louay M. J. Bazzi, *Polylogarithmic independence can fool DNF formulas*, SIAM J. Comput. **38** (2009), no. 6, 2220–2272.

[BDVY09] Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff, *Pseudorandomness for width 2 branching programs*, Electronic Colloquium on Computational Complexity (ECCC) **16** (2009), 70.

[BL85] Michael Ben-Or and Nathan Linial, *Collective coin flipping, robust voting schemes and minima of banzhaf values*, FOCS, IEEE Computer Society, 1985, pp. 408–416.

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

[BTW98] Nader H. Bshouty, Christino Tamon, and David K. Wilson, *On learning width two branching programs*, Inf. Process. Lett. **65** (1998), no. 4, 217–222.

[BV10a] Andrej Bogdanov and Emanuele Viola, *Pseudorandom bits for polynomials*, SIAM J. Comput. **39** (2010), no. 6, 2464–2486.

[BV10b] Joshua Brody and Elad Verbin, *The coin problem and pseudorandomness for branching programs*, FOCS, 2010, pp. 30–39.

[CLO07] D.A. Cox, J.B. Little, and D. O'Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, Undergraduate texts in mathematics, no. v. 10, Springer, 2007.

[CRSW11] L. Elisa Celis, Omer Reingold, Gil Segev, and Udi Wieder, *Balls and bins: Smaller hash families and faster evaluation*, FOCS, 2011, pp. 599–608.

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

[DETT10] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani, *Improved pseudorandom generators for depth 2 circuits*, APPROX-RANDOM, 2010, pp. 504–517.

[EGL$^+$98] Guy Even, Oded Goldreich, Michael Luby, Noam Nisan, and Boban Velickovic, *Efficient approximation of product distributions*, Random Struct. Algorithms **13** (1998), no. 1, 1–16.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser, *Parity, circuits, and the polynomial-time hierarchy*, Mathematical Systems Theory **17** (1984), no. 1, 13–27.

[GMR12] Parikshit Gopalan, Raghu Meka, and Omer Reingold, *DNF sparsification and fast approximate counting*, 2012, To appear in CCC.

[Hås86] Johan Håstad, *Almost optimal lower bounds for small depth circuits*, STOC, 1986, pp. 6–20.

[HHR06] Iftach Haitner, Danny Harnik, and Omer Reingold, *On the power of the randomized iterate*, Advances in Cryptology—CRYPTO '06 (C. Dwork, ed.), Lecture Notes in Computer Science, Springer-Verlag, 2006.

[HVV04] Alexander Healy, Salil P. Vadhan, and Emanuele Viola, *Using nondeterminism to amplify hardness*, STOC, 2004, pp. 192–201.

[HVV06] ______, *Using nondeterminism to amplify hardness*, SIAM J. Comput. **35** (2006), no. 4, 903–931.

[Ind06] Piotr Indyk, *Stable distributions, pseudorandom generators, embeddings, and data stream computation*, J. ACM **53** (2006), no. 3, 307–323.

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson, *Pseudorandomness for network algorithms*, STOC, 1994, pp. 356–364.

[IW97] Russell Impagliazzo and Avi Wigderson, *$P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma*, STOC, 1997, pp. 220–229.

[JSZ85] W. B. Johnson, G. Schechtman, and J. Zinn, *Best constants in moment inequalities for linear combinations of independent and exchangeable random variables*, The Annals of Probability **13** (1985), no. 1, pp. 234–253.

[KLW10] Adam R. Klivans, Homin K. Lee, and Andrew Wan, *Mansour's conjecture is true for random DNF formulas*, COLT, 2010, pp. 368–380.

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

[KNR05] Eyal Kaplan, Moni Naor, and Omer Reingold, *Derandomized constructions of k-wise (almost) independent permutations*, 9th International Workshop on Randomization and Computation (RANDOM), 2005, pp. 354–365.

[LLSZ97] Nathan Linial, Michael Luby, Michael E. Saks, and David Zuckerman, *Efficient construction of a small hitting set for combinatorial rectangles in high dimension*, Combinatorica **17** (1997), no. 2, 215–234.

[Lov08] Shachar Lovett, *Unconditional pseudorandom generators for low degree polynomials*, STOC, 2008, pp. 557–562.

[LTW07] Chi-Jen Lu, Shi-Chun Tsai, and Hsin-Lung Wu, *Improved hardness amplification in np*, Theor. Comput. Sci. **370** (2007), no. 1-3, 293–298.

[Lu02] Chi-Jen Lu, *Improved pseudorandom generators for combinatorial rectangles*, Combinatorica **22** (2002), no. 3, 417–434.

[Nis91] Noam Nisan, *Pseudorandom bits for constant depth circuits*, Combinatorica **11** (1991), no. 1, 63–70. MR MR1112275 (92g:68055)

[Nis92] Noam Nisan, *Pseudorandom generators for space-bounded computation*, Combinatorica **12** (1992), no. 4, 449–461.

[NN93] Joseph Naor and Moni Naor, *Small-bias probability spaces: Efficient constructions and applications*, SIAM J. Comput. **22** (1993), no. 4, 838–856.

[NZ96] Noam Nisan and David Zuckerman, *Randomness is linear in space*, J. Comput. Syst. Sci. **52** (1996), no. 1, 43–52.

[O'D04] Ryan O'Donnell, *Hardness amplification within np*, J. Comput. Syst. Sci. **69** (2004), no. 1, 68–94.

[Pin94] Iosif Pinelis, *Optimum bounds for the distributions of martingales in banach spaces*, The Annals of Probability **22** (1994), no. 4, pp. 1679–1706 (English).

[Raz09] Alexander Razborov, *A simple proof of Bazzis theorem*, ACM Trans. Comput. Theory **1** (2009), no. 1, 3:1–3:5.

[Rei08] Omer Reingold, *Undirected connectivity in log-space*, Journal of the ACM **55** (2008), no. 4, Art. 17, 24. MR MR2445014

[Ros72] Haskell P. Rosenthal, *On the span in L^p of sequences of independent random variables. II*, Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability., Univ. California Press, 1972, pp. 149–167.

[RR99] Ran Raz and Omer Reingold, *On recycling the randomness of states in space bounded computation*, STOC, 1999, pp. 159–168.

[RTV06] Omer Reingold, Luca Trevisan, and Salil Vadhan, *Pseudorandom walks in regular digraphs and the RL vs. L problem*, Proceedings of the 38th Annual ACM Symposium on Theory of Computing (STOC '06), 21–23 May 2006, pp. 457–466.

[Siv02] D. Sivakumar, *Algorithmic derandomization via complexity theory*, IEEE Conference on Computational Complexity, 2002, p. 10.

[SZ95] Michael Saks and David Zuckerman, 1995, Unpublished manuscript.

[SZ99] Michael E. Saks and Shiyu Zhou, *$BP_H SPACE(S) \subseteq DSPACE(S^{3/2})$*, J. Comput. Syst. Sci. **58** (1999), no. 2, 376–403.

[SZ11] Jirí Síma and Stanislav Zák, *Almost k-wise independent sets establish hitting sets for width-3 1-branching programs*, CSR, 2011, pp. 120–133.

[Tre01] Luca Trevisan, *Extractors and pseudorandom generators*, J. ACM **48** (2001), no. 4, 860–879.

[Val77] Leslie G. Valiant, *Graph-theoretic arguments in low-level complexity*, Mathematical foundations of computer science (Proc. Sixth Sympos., Tatranská Lomnica, 1977), Springer, Berlin, 1977, pp. 162–176. Lecture Notes in Comput. Sci., Vol. 53. MR MR0660702 (58 #32067)

[Vio08] Emanuele Viola, *The sum of d small-bias generators fools polynomials of degree d*, IEEE Conference on Computational Complexity, 2008, pp. 124–127.

[Vio11] ———, *Randomness buys depth for approximate counting*, FOCS, 2011, pp. 230–239.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de