Pseudorandomness for read-k DNF formulas

Rocco A. Servedio* Columbia University

Li-Yang Tan† Stanford University

October 19, 2018

Abstract

The design of pseudorandom generators and deterministic approximate counting algorithms for DNF formulas are important challenges in unconditional derandomization. Numerous works on these problems have focused on the subclass of *small-read* DNF formulas, which are formulas in which each variable occurs a bounded number of times.

Our first main result is a pseudorandom generator which ε-fools M-term read-k DNFs using seed length $\mathrm{poly}(k, \log(1/\varepsilon)) \cdot \log M + O(\log n)$. This seed length is exponentially shorter, as a function of both k and $1/\varepsilon$, than the best previous PRG for read-k DNFs. We also give a deterministic algorithm that approximates the number of satisfying assignments of an M-term read-k DNF to any desired $(1+\varepsilon)$-multiplicative accuracy in time

$$\mathrm{poly}(n) \cdot \min\left\{(M/\varepsilon)^{\mathrm{poly}(k,\log(k/\varepsilon))}, (M/\varepsilon)^{\tilde{O}(\log((k\log M)/\varepsilon))}\right\}.$$

For any constant k this is a PTAS, and our runtime remains almost-polynomial ($M^{\tilde{O}(\log\log M)}$) for k as large as any $\mathrm{polylog}(M)$. Prior to our work, the fastest deterministic algorithm ran in time $M^{\tilde{\Omega}(\log M)}$ even for $k = 2$, and no PTAS was known for any non-trivial subclass of DNFs.

The common essential ingredients in these pseudorandomness results are new analytic inequalities for read-k DNFs. These inequalities may be of independent interest and utility; as an example application, we use them to obtain a significant improvement on the previous state of the art for agnostically learning read-k DNFs.

*Supported by NSF award CCF-1563155.

†Supported by NSF award CCF-1563122.

1 Introduction

DNF formulas—depth-2 circuits in which an AND of Boolean literals feeds into an output OR gate—play an important role in many branches of theoretical computer science. The most commonly used complexity measure for a DNF formula is its *size*, which is the number of terms (ANDs) feeding into the output OR gate, and which we shall denote by M throughout this paper. Another important parameter associated with a DNF formula is its *read number*; this is the maximum number of occurrences of any individual variable from $x_1, \ldots, x_n$ across the entire formula, and is a natural way of quantifying dependencies among the terms of the DNF. The class of read-k DNF formulas has been intensively studied, even in the $k = 1$ case of *read-once* DNF formulas, across a range of different fields including unconditional derandomization [EGL+98, CRS00, Baz03, DETT10, KLW10, GMR+12, BN17] and computational learning theory [HM91, AP92, BFJ+94, Han93, PR95, ABK+98, DMP99, KLW10].

The main contributions of this paper are new pseudorandom generators for read-k DNF formulas that exponentially improve on the prior state of the art, and fast deterministic approximate counting algorithms for read-k DNFs that achieve a strong *relative-error* guarantee (as opposed to the additive-error guarantee that has been the focus of prior work). The common essential ingredients in our results are new analytic inequalities for read-k DNFs, which we believe are of independent interest and utility.

We now discuss these problems, the background and context for each of them, and our new results in detail.

1.1 Pseudorandom generators for read-k DNFs

Background. We recall that for a class $\mathscr{C}$ of functions from $\{-1,1\}^n$ to $\{0,1\}$, a distribution $\mathcal{D}$ over $\{0,1\}^n$ *ε-fools $\mathscr{C}$ with seed length r* if (a) $\mathcal{D}$ can be sampled efficiently with r random bits (i.e. there is an efficient deterministic algorithm $\mathrm{Gen}_{\mathcal{D}} : \{-1,1\}^r \to \{-1,1\}^n$ which, on input a uniform string from $\{-1,1\}^r$, outputs a draw from $\mathcal{D}$), and (b) for every function $F \in \mathscr{C}$, we have

$$\left| \mathop{\mathbf{E}}_{\boldsymbol{s} \leftarrow \{-1,1\}^r} \left[F(\mathrm{Gen}_{\mathcal{D}}(\boldsymbol{s})) \right] - \mathbf{E} \left[F(\boldsymbol{x}) \right] \right| \leq \varepsilon.$$

Equivalently, we say that $\mathrm{Gen}_{\mathcal{D}}$ is an *ε-pseudorandom generator (ε-PRG) for $\mathscr{C}$ with seed length r*.

A number of researchers have studied the problem of obtaining explicit PRGs for read-k DNF formulas. In the $k = 1$ case of read-once DNFs, Chari, Rohatgi, and Srinivasan [CRS00], building on the work of Even et al. [EGL+98], obtained a PRG with seed length $O(\log M \cdot \log(1/\varepsilon) + \log n)$ by showing that suitable small-bias distributions fool read-once DNFs; this result was rediscovered by De et al. [DETT10]. Bazzi [Baz03] gave a PRG for read-once DNFs with seed length $O(\log M \cdot \log(1/\varepsilon) \cdot \log n)$ based on bounded independence rather than small-bias distributions. (Complementing these positive results, there has also been significant interest in lower bounds on the ability of such "generic" distributions to fool, or even hit, read-once DNFs [DETT10, LV17, BN17].) Additional motivation for designing optimal PRGs for read-once DNFs was given by Healy, Vadhan, and Viola, who established a connection between such PRGs and hardness amplification [HVV06]. Using a different approach based on iterative applications of "mild pseudorandom restrictions," Gopalan, Meka, Reingold, Trevisan, and Vadhan [GMR+12] gave an improved, near optimal, PRG with a $\tilde{O}(\log(n/\varepsilon))$ seed length for read-once DNFs. As mentioned in [Gop16], it is not known how to extend the techniques of [GMR+12] even to the case of read-k DNFs for $k = 2$.

Turning to read-k DNFs, an obvious but crucial qualitative difference between the $k = 1$ (read-once) and $k \geq 2$ cases is the lack of independence across terms in the latter, which introduces significant technical challenges. Klivans, Lee, and Wan [KLW10] gave a PRG for read-k DNFs formulas for general k; their seed length for M-term read-k DNFs is $\exp(O(k)) \cdot \mathrm{poly}(1/\varepsilon) \cdot \log M + O(\log n)$.[1]

Finally, we recall that for general M-term DNFs (without any restriction on the read parameter), the current best PRG has seed length $O(\log(M/\varepsilon) \log M \log\log M + \log n)$ [DETT10, Tal17]. Compared to the results on small-read DNF formulas discussed above, this seed length has a quadratic rather than linear dependence on $\log M$.

Our result. We give a PRG for M-term read-k DNF formulas with seed length $\mathrm{poly}(k, \log(1/\varepsilon)) \cdot \log M + O(\log n)$:

Theorem 1 (PRG for read-k DNFs)**.** *There is an ε-PRG for the class of M-term read-k DNFs with seed length*

$$O\big((k + \log(1/\varepsilon))^{3/2} \cdot \log^2(1/\varepsilon) \cdot \log(M/\varepsilon) + \log n\big).$$

Theorem 1 gives an exponential improvement of [KLW10]'s seed length in terms of both the read parameter k and the error parameter ε:

1.2 Agnostic learning of read-k DNFs

Background. Much work in computational learning theory has aimed at giving efficient uniform-distribution *agnostic learning* algorithms for various classes of functions [KKMS08, GKK08a, SSSS09, FGKP09, Fel10, DSFT+15]. We recall the basic definitions of this well-studied learning framework. Let $\mathscr{C}$ be a class of Boolean functions from $\{-1,1\}^n$ to $\{0,1\}$, and let $F : \{-1,1\}^n \to \{0,1\}$ be an arbitrary function. We define $\mathrm{opt}(F, \mathscr{C})$ to be the error of the best approximation for F in $\mathscr{C}$, i.e.

$$\mathrm{opt}(F, \mathscr{C}) = \min_{G \in \mathscr{C}} \Big\{ \mathbf{Pr}\left[G(\boldsymbol{x}) \neq F(\boldsymbol{x}) \right] \Big\}.$$

An algorithm A *agnostically learns $\mathscr{C}$ under the uniform distribution with membership queries* if for any function F and input parameter ε, given black-box access to F, with high probability A outputs a hypothesis $H : \{-1,1\}^n \to \{0,1\}$ such that $\mathbf{Pr}[H(\boldsymbol{x}) \neq F(\boldsymbol{x})] \leq \mathrm{opt}(F, \mathscr{C}) + \varepsilon$.

Gopalan, Kalai, and Klivans [GKK08a] gave a polynomial-time agnostic learning algorithm for the class of decision trees, but efficient agnostic learning of DNF formulas in this model remains an important open problem [GKK08b]. Klivans, Lee, and Wan [KLW10] studied this problem for the subclass of read-k DNFs, and gave an agnostic learning algorithm for M-term read-k DNFs which runs in time $\mathrm{poly}(n) \cdot (M/\varepsilon)^{\exp(O(k)) \cdot \log(1/\varepsilon)}$.

Our result. We obtain a significant improvement of [KLW10]'s result, replacing their doubly exponential running time dependence on k with a singly exponential one:

Theorem 2 (Agnostically learning read-k DNFs)**.** *The class of M-term read-k DNF formulas over $\{0,1\}^n$ can be agnostically learned to accuracy ε under the uniform distribution using membership queries in time* $\mathrm{poly}(n) \cdot (M/\varepsilon)^{O(k^{3/2}(\log(1/\varepsilon))^2)}$.

[1] [KLW10] claimed a seed length of $\exp(O(k)) \cdot \log(1/\varepsilon) \cdot \log M + O(\log n)$ but a close inspection of their construction shows that the actual seed length (resulting from a slight extension of their analysis) is $\exp(O(k)) \cdot \mathrm{poly}(1/\varepsilon) \cdot \log M + O(\log n)$ [Kli17]. See Remarks 8 and 11 in Section 5.1.

Prior to our work, there were no known $\mathrm{poly}(n) \cdot M^{\mathrm{poly}(k)}$-time learning algorithms for read-k DNFs, even in the significantly easier noiseless setting of uniform-distribution PAC learning (where the target function F is promised to lie in $\mathscr{C}$). Hancock and Mansour [HM91] gave a $\mathrm{poly}(n) \cdot M^{O(k)}$ time uniform-distribution learning algorithm for *monotone* read-k DNF in this easier setting; their algorithm did not require membership queries.

Remark 3. In the statement of Theorem 2, we have chosen to optimize the dependence on k. As we point out in Section 4.2, our techniques also give an algorithm that runs in time $\mathrm{poly}(n) \cdot (M/\varepsilon)^{O(k^2 \log(1/\varepsilon))}$ (indeed, via a somewhat simpler analysis than that of Theorem 2).

1.3 Relative error deterministic counting satisfying assignments of read-k DNFs

Background. Since exactly counting the number of satisfying assignments of a DNF formula is a well-known #P-complete problem, there has been signficant interest in developing efficient *approximate* counting algorithms. In particular, a number of researchers [AW85, LN90, LV96, LVW93, GMR13] have given deterministic algorithms for *additively* approximating the fraction of assignments in $\{-1,1\}^n$ that satisfy an M-term DNF formula. The strongest result of this sort to date is that of Gopalan, Meka, and Reingold [GMR13], who gave an algorithm running in time

$$\left(\frac{Mn}{\varepsilon}\right)^{\tilde{O}(\log\log(n)+\log\log(M)+\log(1/\varepsilon))}$$

for an additive ε-approximation.

For the more challenging task of achieveing a *multiplicative* $(1+\varepsilon)$-factor approximation, early work of Karp and Luby [KL83] gave an FPRAS, i.e. a randomized $\mathrm{poly}(M, n, 1/\varepsilon)$-time algorithm. (This seminal paper initiated the study of randomized algorithms for approximate counting problems, with DNF counting as its motivating example.) Achieving a full derandomizing the Karp–Luby FPRAS is viewed as an important open problem in unconditional derandomization, see e.g. Open Problem 2.36 of Vadhan's monograph [Vad12]. As we explain in Section 6, the technique of Karp and Luby can be viewed as a *deterministic* reduction from multiplicative $(1+\varepsilon)$-factor approximation for DNF counting to additive $\pm(\varepsilon/M)$-approximation for CNF counting (which they combine with a straightforward random sampling algorithm to achieve the required $\pm(\varepsilon/M)$ additive approximation). Therefore, one can derandomize Karp and Luby's randomized algorithm by using this reduction together with the best deterministic additive approximation algorithm for CNFs, the algorithm of [GMR13] described above[2]. This yields a deterministic multiplicative $(1+\varepsilon)$-factor approximation algorithm running in time

$$\left(\frac{Mn}{\varepsilon}\right)^{\tilde{O}(\log(M/\varepsilon)+\log\log n)},$$

which is the current fastest algorithm for multiplicative $(1+\varepsilon)$-approximation.

We note the wide gap between the best known deterministic runtimes for absolute and relative error: $M^{\tilde{O}(\log\log M)}$ versus $M^{\tilde{O}(\log M)}$ in the standard setting of $M = \mathrm{poly}(n)$, an exponential difference in the exponents.

[2] Recall that for absolute error, approximate counting of CNF satisfying assignments is equivalent to approximating counting of DNF satisfying assignments. (This is not the case for relative error.)

Our results. We show that a dramatically more efficient version of the Karp–Luby reduction holds for read-k DNFs: in order to achieve multiplicative $(1+\varepsilon)$-accuracy, it suffices to additively count to a much coarser error, namely $\pm(\varepsilon/k)$ rather than $\pm(\varepsilon/M)$ as in the general case. Combining this with the [GMR13] counting algorithm and with our new PRG for read-k DNFs (Section 1.1), we obtain the following result for relative-error counting of read-k DNFs:

Theorem 4 (Relative error deterministic counting of read-k DNFs)**.** *There is a deterministic algorithm which, given as input an M-term read-k DNF formula and an accuracy parameter $\varepsilon > 0$, runs in time*

$$\mathrm{poly}(n) \cdot \min\left\{(M/\varepsilon)^{\tilde{O}(\log((k\log M)/\varepsilon))}, (M/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}\right\}$$

and outputs a multiplicative $(1+\varepsilon)$-factor approximation to $\mathbf{Pr}[F(\boldsymbol{x})=1]$.

For any constant k this is a PTAS (running in time $M^{\mathrm{poly}(k)}$), and our runtime remains almost-polynomial ($M^{\tilde{O}(\log\log M)}$) for k as large as any polylog(M). Prior to our work, the fastest deterministic algorithm, even for $k=2$, was the $M^{\tilde{\Omega}(\log M)}$ time algorithm for general DNFs, and to our knowledge, no PTAS was known for any non-trivial subclass of DNFs. An intriguing direction for future work is to study whether our techniques can lead to improved relative error deterministic counting algorithms for general DNFs, with the end goal of an eventual full derandomization of the Karp–Luby FPRAS for DNF counting.

Monotone read-k DNFs. At the cost of only achieving a $(2+\varepsilon)$-factor approximation (rather than a $(1+\varepsilon)$-factor approximation), we give an even faster algorithm for *monotone* read-k width-w DNFs:

Theorem 5 ($(2+\varepsilon)$-factor approximation for monotone read-k DNFs)**.** *There is a deterministic algorithm which, given as input a monotone M-term width-w read-k DNF formula, runs in time*

$$\mathrm{poly}(n) \cdot M \cdot \min\left\{(kw/\varepsilon)^{\tilde{O}(\log(k\log(kw)/\varepsilon))}, (kw/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}\right\}$$

outputs a $(2+\varepsilon)$-factor approximation to $\mathbf{Pr}[F(\boldsymbol{x})=1]$.

Even if k and w are both as large as $\exp((\log M)^{0.49})$ the above running time is $\mathrm{poly}(n)\cdot M^{1+o(1)}$, almost linear in the number of terms.

1.4 Our techniques: A new analytic inequality for read-k DNFs

The unifying technical ingredient in this work is a new analytic inequality for read-k DNFs. This new result gives an essentially optimal bound on the expected number of terms in a read-k DNF that are satisfied by a uniform random assignment to the input variables, and is an exponential improvement on the best previous bound on this quantity.

Lemma 1.1 (Expected number of satisfied terms in a read-k DNF)**.** *Let $F = T_1 \vee \cdots \vee T_M$ be an M-term read-k DNF formula and let μ denote $\mathbf{Pr}[F(\boldsymbol{x})=1]$. Then*

$$\mathbf{E}\left[\sum_{i=1}^{M} T_i(\boldsymbol{x})\right] \le k\ln\left(\frac{1}{1-\mu}\right),$$

where $T_i(x)$ is 1 if input x satisfies term T_i and is 0 otherwise.

The upper bound of Lemma 1.1 is essentially the best possible, as can be seen by a straightforward analysis of a simple variant of the TRIBES DNF (see the discussion at the end of Section 3.1). Lemma 1.1 exponentially improves on a $16^k \ln(1/(1-\mu))$ upper bound that was proved in [KLW10] via a different argument that involved monotonicity and used the Four Functions Theorem. In contrast our proof, given in Section 3, does not involve monotonicity and the main tool used is Shearer's Lemma [CFGS86].

We give some high-level intuition as to why Lemma 1.1 gives us leverage for the various algorithmic problems we consider in this paper. Intuitively (thinking of μ as bounded away from 1), Lemma 1.1 says that "most" assignments to a read-k DNF do not satisfy "too many" distinct terms. To see why this might be useful, consider the extreme version of this property in which a DNF formula $F = T_1 \vee \cdots \vee T_M$ is such that *every* input assignment satisfies either no terms or exactly one term; that is, $\sum_{i=1}^{M} T_i(x) \leq 1$ for all $x \in \{-1,1\}^n$. (For example, the canonical conversion of a decision tree to a DNF results in a DNF which has this property.) Such a DNF formula is equivalent to a simple linear sum $T_1 + \cdots + T_M$ of the M terms which comprise it, rather than a logical OR of these terms; this linear structure greatly facilitates the algorithmic tasks that we consider. For example, a straightforward deterministic algorithm can exactly count satisfying assignments of such functions; a simple (ε/M)-biased distribution is a PRG with excellent seed length for such functions; and a polynomial-time agnostic learning algorithm (using membership queries) is known for such functions [GKK08a].

Returning to our actual read-k setting rather than the extreme "satisfy-once" formulas just considered, intuitively we are able to use bounds on the expected number of satisfied terms to ensure that every read-k DNF (approximately) has an analogous structure (though now a low-degree polynomial replaces a simple linear sum), which we then analogously exploit to give efficient algorithms and pseudorandomness constructions. At a technical level, our PRGs and agnostic learning algorithms for read-k DNFs (the first two of our three main results) are obtained by establishing the existence of polynomial approximators with suitable properties. Our general approach to obtaining such approximators follows the approach of [KLW10], though there are some differences in the specifics of our construction (see the discussion at the beginning of Section 4.2) and the overall bounds we obtain are significantly stronger than those of [KLW10] thanks to the quantitative improvement afforded by our Lemma 1.1.

2 Preliminaries

We view Boolean functions as taking *inputs* in $\{-1,1\}^n$, where we view -1 as TRUE and 1 as FALSE (this is more convenient for the Fourier representation). We use a different convention for the *outputs* of terms, Boolean functions, etc.; for these we take the outputs to be $\{0,1\}$ where 0 is FALSE and 1 is TRUE (this is more convenient for our constructions).

We represent an M-term DNF formula as $F = T_1 \vee \cdots \vee T_M$, where we view each term T_i as outputting either 0 or 1. We associate each term T_i with the set of variables that it contains, and thus in particular we write "$T_i \cap T_j \neq \emptyset$" to indicate that there is some variable that occurs in both T_i and T_j (such a variable could occur positively in one term and negatively in the other). Unless otherwise indicated, we write "$\boldsymbol{x}$" (bold font) to indicate a string drawn uniformly at random from $\{-1,1\}^n$. We write log to denote the logarithm base 2 and ln to denote the natural logarithm.

For $\boldsymbol{X}$ a random variable supported on a finite set S, we recall that the *entropy* of $\boldsymbol{X}$, denoted

$H(\boldsymbol{X})$, is

$$H(\boldsymbol{X}) = \sum_{s \in S} \mathbf{Pr}[\boldsymbol{X} = s] \cdot \log\left(\frac{1}{\mathbf{Pr}[\boldsymbol{X} = s]}\right)$$

and that $H(\boldsymbol{X}) \leq \log |S|$ (with equality if and only if $\boldsymbol{X}$ is uniform over $|S|$).

2.1 Polynomial representations, polynomial approximators, and agnostic learning

We recall that every function $f : \{-1,1\}^n \to \mathbb{R}$ has a unique *Fourier expansion*

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)\chi_S(x), \qquad \text{where} \qquad \chi_S(x) = \prod_{j \in S} x_j \quad \text{and} \quad \widehat{f}(S) = \mathbf{E}[f(\boldsymbol{x}) \cdot \chi_S(\boldsymbol{x})].$$

Thus the Fourier expansion is simply the (unique) multilinear polynomial agreeing with f on $\{-1,1\}^n$. The *Fourier degree* of f is $\max\{|S| : \widehat{f}(S) \neq 0\}$ (the degree of the polynomial given by the Fourier expansion). The *Fourier ℓ_1-norm*, or *spectral norm*, of f is $\|f\|_1 := \sum_{S \subseteq [n]} |\widehat{f}(S)|$.

A long line of work has underscored the usefulness of polynomial *approximators* which have small Fourier degree and/or small Fourier ℓ_1-norm. We recall the following basic results from [KLW10], which will be useful for us in constructing polynomial approximators for read-k DNF formulas:

Fact 2.1 (Facts 9 and 10 of [KLW10])**.**

1. *Let $p : \{-1,1\}^m \to \mathbb{R}$ be a polynomial with coefficients $\widehat{p}(S)$ for $S \subseteq [m]$, and let $q_1, \ldots, q_m : \{-1,1\}^n \to \{-1,1\}$ be arbitrary Boolean functions. Then $p(q_1, \ldots, q_m) = \sum_{S \subseteq [m]} \widehat{p}(S) \prod_{i \in S} q_i$ is a polynomial over $\{-1,1\}^n$ with spectral norm at most*

$$\sum_{S \subseteq [m]} |\widehat{p}(S)| \cdot \prod_{i \in S} \|q_i\|_1.$$

2. *For any term $T : \{-1,1\}^n \to \{0,1\}$ (i.e. any AND of literals), we have $\|T\|_1 = 1$.*

The following simple fact will also be useful:

Fact 2.2. *Let $a(t) = \sum_{i=0}^{s} a_i t^i$ be a univariate degree-s polynomial and let $p : \{-1,1\}^m \to \mathbb{R}$ be a multivariate polynomial. Then the spectral norm of $a(p(x_1, \ldots, x_m))$ is at most*

$$\left(\sum_{i=0}^{s} |a_i|\right) \cdot (\|p\|_1)^s .$$

Uniform-distribution agnostic learning. Gopalan, Kalai, and Klivans [GKK08a] showed that approximation by polynomials with small Fourier ℓ_1-norm implies efficient agnostic learnability:

Theorem 6 ([GKK08a])**.** *Let $\mathscr{C}$ be a class of functions from $\{-1,1\}^n$ to $\{0,1\}$ such that for every $g \in \mathscr{C}$, there is a polynomial p such that $\|p\|_1 \leq L$ and $\mathbf{E}[(p(\boldsymbol{x}) - g(\boldsymbol{x}))^2] \leq \varepsilon^2/2$. Then there is an algorithm that agnostically learns $\mathscr{C}$ under the uniform distribution with membership queries and runs in time* $\mathrm{poly}(n, L, 1/\varepsilon)$.

2.2 Sandwiching polynomial approximators and pseudorandom generators

As indicated by Theorem 6, polynomials that approximate Boolean functions are very useful for computational learning. Polynomials which satisfy a stronger requirement, known as *sandwiching approximation*, are known to be very useful for unconditional pseudorandomness; more precisely, they imply the existence of efficient *pseudorandom generators*, as described below.

Definition 1. *Let $f : \{-1,1\}^n \to \mathbb{R}$. Polynomials $p_\ell, p_u : \{-1,1\}^n \to \mathbb{R}$ are said to* δ-sandwich *f in ℓ_1 if (i) $p_\ell(x) \le f(x) \le p_u(x)$ for every $x \in \{-1,1\}^n$, and (ii) $\mathbf{E}\left[p_u(\boldsymbol{x}) - p_\ell(\boldsymbol{x})\right] \le \delta$.*

Sandwiching polynomials will be useful for us because of their close connection to *pseudorandom generators*.

The PRGs that we analyze are *ε-biased* distributions. We recall the following definition from [NN93]:

Definition 2 (ε-biased distributions)**.** *A distribution $\mathcal{D}$ over $\{-1,1\}^n$ is said to be* ε-biased *if for every nonempty $S \subseteq [n]$ it holds that*

$$\left| \mathop{\mathbf{E}}_{\boldsymbol{x} \leftarrow \mathcal{D}} \left[\prod_{i \in S} \boldsymbol{x}_i \right] \right| \le \varepsilon.$$

In [NN93] Naor and Naor gave constructions of ε-biased distributions over $\{-1,1\}^n$ that can be sampled efficiently with seed length $O(\log(1/\varepsilon) + \log n)$.

Our results will rely on a crucial connection between ε-biased distributions and sandwiching polynomials that have small ℓ_1-norm (see e.g. Appendix A of [Baz09]). This connection follows from linear programming duality.

Lemma 2.3 (Sandwiching polynomials yield PRGs via ε-biased distributions)**.** *Let $f : \{-1,1\}^n \to \mathbb{R}$ and suppose that p_ℓ, p_u are polynomials, each with Fourier ℓ_1-norm at most L, which δ-sandwich f in ℓ_1. Then any ε-biased distribution is a $(\delta + \varepsilon L)$-PRG for f.*

3 Bounding the expected number of satisfied terms in a read-k DNF

The setup. As indicated in the previous section, coming up with suitable polynomial approximators for a class of Boolean functions can lead to both pseudorandom generators and agnostic learning algorithms for the class. In contrast with previous approaches, which typically built polynomial approximators by analyzing the Fourier spectrum [Man95, LMN93, Baz09, Raz09], Klivans, Lee, and Wan [KLW10] developed an innovative technique that constructs polynomial approximators for DNF formulas based on *univariate polynomial interpolation*.

The idea underlying their basic approach, which we build on, is as follows: Let M be an M-term DNF formula $F = T_1 \vee \cdots \vee T_M$. Consider the integer-valued function

$$\mathbb{T}_F : \{-1,1\}^n \to \{0,1,\ldots,M\}, \quad \mathbb{T}_F(x) = \sum_{i=1}^{M} T_i(x) \tag{1}$$

which, on input x, outputs the number of terms satisfied by x (so in particular we have $\mathbb{T}_F(x) = 0$ iff $F(x) = 0$, and $\mathbb{T}_F(x) \in \{1,2,\ldots,M\}$ otherwise). Since each $T_i(x)$ is a single term, by linearity

it is easy to see that $\mathbb{T}_F(x)$ is a significantly simpler polynomial than the Fourier polynomial for F. The main insight of [KLW10] is to *compose* $\mathbb{T}_F(x)$ with a univariate polynomial $P_d(t)$ which is specially designed to output 0 on $t = 0$ and to output 1 on all $t \in [d]$. Intuitively, such a composed polynomial $P_d(\mathbb{T}_F(x))$ may be "much simpler" than the Fourier polynomial for F, and if almost every $x \in \{-1,1\}^n$ is such that at most d terms of F are satisfied by x, then $P_d(\mathbb{T}_F(x))$ should be a good approximator for F as desired. [KLW10] give a high-probability bound on the number of terms that are typically satisfied (equivalently, a tail bound on $\mathbb{T}_F(\boldsymbol{x})$ where $\boldsymbol{x}$ is uniform over $\{0,1\}^n$) in different kinds of DNF formulas such as random DNF and read-once DNF, and thus obtain pseudorandomness and agnostic learning results via this framework.

[KLW10] augments the above simple framework to analyze read-k DNF formulas. They define a different integer-valued function

$$\mathbb{A}_F : \{-1,1\}^n \to \{0,1,\dots,M\}, \quad \mathbb{A}_F(x) = \sum_{i=1}^{M} A_i(x), \tag{2}$$

where for each $i \in [M]$ the formula $A_i(x)$ is defined to be

$$A_i(x) := T_i \wedge \neg\, \phi_i(x), \quad \text{where } \phi_i(x) = \bigvee_{\substack{j<i \\ T_j \cap T_i \neq \emptyset}} T_j(x). \tag{3}$$

That is, $A_i(x) = 1$ iff x satisfies T_i but satisfies none of the earlier terms T_j that share any variable with T_i. It follows that $\sum_{i=1}^{M} A_i(x)$ is the number of disjoint terms of F satisfied by x, where this set of disjoint terms is formed by greedily including satisfied terms according to the ordering $T_1 \prec T_2 \prec \cdots \prec T_M$; note that similar to $\mathbb{T}_F(x)$, we have that $F(x) = 0$ iff $\mathbb{A}_F(x) = 0$, and $F(x) = 1$ iff $\mathbb{A}_F(x) \geq 1$. While the polynomial $\mathbb{A}_F(x)$ is somewhat more complex than $\mathbb{T}_F(x)$, it is still simple enough for $P_d(\mathbb{A}_F(x))$ to be a useful polynomial approximator for read-k DNF. To make this approach work, [KLW10] give a tail bound on $\mathbb{A}_F(\boldsymbol{x})$ by bounding the *expectation* $\mathbf{E}\left[\sum_{i=1}^{M} T_i(\boldsymbol{x})\right] = \mathbf{E}[\mathbb{T}_F(\boldsymbol{x})]$.

In Section 3.1 we give a new bound on $\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})]$, the expected number of terms satisfied by a random input $\boldsymbol{x}$, for F being a read-k DNF. As a function of k, our bound is an exponential (and optimal) improvement of the previous bound established in [KLW10]. Following [KLW10], we will use this bound on $\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})]$ to establish large-deviation bounds for the random variable $\mathbb{A}_F(\boldsymbol{x})$ in Section 4.1.

In Section 3.2 we give a bound on $\mathbf{E}[\mathbb{A}_F(\boldsymbol{x})]$ that holds for *all* monotone DNF formulas, without any restriction on their being read-k. We will use this bound in the analysis of Section 6.2.

3.1 Optimal bound on the expected number of terms satisfied in a read-k DNF

A key technical tool we use is *Shearer's Lemma*, which can be thought of as a quantitative sharpening of the sub-additivity of entropy.

Lemma 3.1 (Shearer's Lemma [CFGS86])**.** *Let $\boldsymbol{X}_1, \dots, \boldsymbol{X}_n$ be finitely supported random variables (not necessarily independent). Let $S_1, \dots, S_m \subseteq [n]$ be such that each $i \in [n]$ belongs to* at least *k of $S_1, \dots, S_m$. Then*

$$k \cdot H(\boldsymbol{X}_1, \dots, \boldsymbol{X}_n) \leq \sum_{j=1}^{m} H((\boldsymbol{X}_i)_{i \in S_j}).$$

For the sake of completeness we recall the simple proof of Lemma 3.1; we follow the expositions given in [Lov15, Rad03].

Proof of Lemma 3.1. The chain rule for entropy gives that

$$k \cdot H(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) = k \cdot (H(\boldsymbol{X}_1) + H(\boldsymbol{X}_2|\boldsymbol{X}_1) + \cdots + H(\boldsymbol{X}_n|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{n-1})). \tag{4}$$

On the other hand, if $S_j = \{i_1, \ldots, i_{s_j}\}$ with $i_1 < \cdots < i_{s_j}$, then

$$\begin{aligned} H((\boldsymbol{X}_i)_{i \in S_j}) &= H(\boldsymbol{X}_{i_1}) + H(\boldsymbol{X}_{i_2}|\boldsymbol{X}_{i_1}) + \cdots + H(\boldsymbol{X}_{i_s}|\boldsymbol{X}_{i_1}, \ldots, \boldsymbol{X}_{i_{s-1}}) \\ &\geq H(\boldsymbol{X}_{i_1}|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i_1-1}) + H(\boldsymbol{X}_{i_2}|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i_2-1}) + \cdots + H(\boldsymbol{X}_{i_s}|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i_s-1}), \end{aligned} \tag{5}$$

where the first line is the chain rule for entropy and the second is because additional conditioning can only decrease entropy. Summing (5) over all $j \in [m]$, the resulting RHS is at least (4) (by non-negativity of entropy and since by assumption each $H(\boldsymbol{X}_i|\boldsymbol{X}_1, \ldots, \boldsymbol{X}_{i-1})$ occurs at least k times in the sum), and the lemma is proved. □

Fix subsets $S_1, \ldots, S_m \subseteq [n]$ and let $J_1, \ldots, J_M$ be Boolean functions $J_j : \{-1, 1\}^n \to \{0, 1\}$ such that each J_j is a function of only the variables in S_j. (Note that J_j need not actually depend on every variable in S_j, but it may not depend on any variable outside of S_j.) We say that the family $J_1, \ldots, J_M$ is *read-k* if each input coordinate in $[n]$ feeds into at most k of the J_j's, i.e. $|\{j : i \in S_j\}| \leq k$ for all $i = 1, \ldots, n$. Note that specializing to the case in which each J_j is a disjunction, the function $G(x) = J_1(x) \wedge \cdots \wedge J_M(x)$ is a read-k CNF formula.

Let $\boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ be a random variable distributed uniformly over $\{-1, 1\}^n$ (so $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ are independent uniform $-1/1$ random variables). Define

$$p_j := \mathbf{Pr}\left[J_j(\boldsymbol{x}) = 1\right].$$

Lemma 3.2 (Bounding the acceptance probability of a read-k family.)**.** *Let $J_1, \ldots, J_M$ be a read-k family with $\mathbf{Pr}[J_j(\boldsymbol{x}) = 1] = p_j$. Then*

$$\mathbf{Pr}\left[J_1(\boldsymbol{x}) = \cdots = J_M(\boldsymbol{x}) = 1\right] \leq \left(\prod_{j=1}^{M} p_j\right)^{1/k}.$$

Proof. The proof closely follows the proof of Lemma 6.2 in [Lov15]. Let q denote the left-hand side, $q = \mathbf{Pr}[J_1(\boldsymbol{x}) = \cdots = J_M(\boldsymbol{x}) = 1]$. We may assume without loss of generality that for all $i \in [n]$ the value of $|\{j : i \in S_j\}|$ is exactly k (since if some $i \in [n]$ has $|\{j : i \in S_j\}| = k - \ell_i$ for some $\ell_i > 0$, we can simply add i to any ℓ_i of the sets S_j that don't already contain it.)

Let $A = \{x \in \{-1, 1\}^n : J_1(x) = \cdots = J_M(x) = 1\}$, and for each $j \in [M]$ let $A_j = \{x \in \{-1, 1\}^{S_j} : J_j(x) = 1\}$. We have that $|A| = q2^n$ and that each $|A_j| = p_j 2^{|S_j|}$.

Let $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ be a joint random variable such that $\boldsymbol{X} = (\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$ is uniform over A. Applying Shearer's Lemma to $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n)$, we get that

$$k \cdot H(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) \leq \sum_{j=1}^{M} H((\boldsymbol{X}_i)_{i \in S_j}).$$

The left-hand side is $k \cdot \log |A| = k(n + \log q)$. On the right-hand side, for each $j \in [M]$ we have that $(\boldsymbol{X}_i)_{i \in S_j}$ is supported on A_j, and hence $H((\boldsymbol{X}_i)_{i \in S_j}) \leq \log |A_j| = |S_j| + \log p_j$. It follows that

$$k(n + \log q) \leq \sum_{j=1}^{M} (|S_j| + \log p_j) = kn + \log \left(\prod_{j=1}^{M} p_j \right),$$

which gives the desired conclusion on rearrangement. □

With Lemma 3.2 in hand we are ready to prove Lemma 1.1. We recall its statement:

Lemma 3.3 (Lemma 1.1 restated: bound on expected number of satisfied terms in a read-k DNF)**.** *Let $F = T_1 \vee \cdots \vee T_M$ be any read-k DNF, let $\mu = \mathbf{Pr}[F(\boldsymbol{x})]$, and recall that $\mathbb{T}_F(x) = \sum_{j=1}^{M} T_j(x)$. Then*

$$\mathbf{E}\left[\mathbb{T}_F(\boldsymbol{x})\right] \leq k \ln \left(\frac{1}{1-\mu} \right).$$

Proof. Let G be the read-k CNF $G = C_1 \wedge \cdots \wedge C_M = \neg F$, so each clause C_j of G is the negation $\neg T_j$ of term T_j of F. Let $p_j = 1 - 2^{-|C_j|}$ be the probability that a uniform random input satisfies clause C_j of G. We have

$$\begin{aligned}
(1-\mu)^k = (1 - \mathbf{Pr}[F(\boldsymbol{x})])^k = \mathbf{Pr}[G(\boldsymbol{x})]^k &\leq \prod_{j=1}^{M} p_j && \text{(Lemma 3.2)} \\
&\leq \left(\frac{1}{M} \cdot \sum_{j=1}^{M} p_j \right)^M && \text{(AM-GM inequality)} \\
&= \left(1 - \frac{1}{M} \cdot \sum_{j=1}^{M} \mathbf{Pr}[T_j(\boldsymbol{x})] \right)^M && (p_j = 1 - \mathbf{Pr}[T_j(\boldsymbol{x})]) \\
&\leq \exp\left(- \sum_{j=1}^{M} \mathbf{Pr}[T_j(\boldsymbol{x})] \right) && (6) \\
&= \exp(-\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})]), && \text{(definition of } \mathbb{T}_F)
\end{aligned}$$

which upon rearrangement gives the claimed bound on $\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})]$. □

Discussion. The upper bound $\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})] \leq k \ln \frac{1}{1-\mu}$ given by Lemma 1.1 is exponentially stronger than the upper bound of $16^k \ln \frac{1}{1-\mu}$ which was proved, using the Four Functions Theorem, as Lemma 30 of [KLW10]. Lemma 1.1 is essentially optimal, as can be easily seen by considering the read-k DNF F obtained by OR-ing together k identical copies (over the same variables in each copy) of the width-w read-once TRIBES DNF (this is a monotone DNF with n/w terms each containing w distinct un-negated variables). A simple calculation shows that for this DNF $\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})] = \frac{kn}{w2^w}$, while the upper bound given by Lemma 1.1 is $\frac{kn}{w} \cdot \ln \frac{1}{1-2^{-w}}$, which is very close to $\frac{kn}{w2^w}$. (Indeed the only step in the proof of Lemma 1.1 which is not an equality for this F is (6), which is the simple inequality $1 - x \leq \exp(-x)$.)

3.2 Expected number of disjoint satisfied terms in a monotone DNF

Recall that given any M-term DNF formula F, the integer-valued function $\mathbb{A}_F(x)$ equals $\sum_{i=1}^{M} A_i(x)$, the number of disjoint terms satisfied by input x that are collected by a greedy procedure (see Equations (2) and (3)). In this section we give an upper bound on $\mathbf{E}[\mathbb{A}_F(x)]$ that holds for *any* monotone DNF (with no read-k requirement):

Lemma 3.4 (Expected number of *disjoint* satisfied terms in a monotone DNF)**.** *Let $F = T_1 \vee \cdots \vee T_M$ be a monotone M-term DNF formula and let μ denote $\mathbf{Pr}[F(\boldsymbol{x})]$. Then*

$$\mathbf{E}\left[\mathbb{A}_F(\boldsymbol{x})\right] \leq \ln\left(\frac{1}{1-\mu}\right).$$

Lemmas 1.1 and 3.4 are incomparable in two ways. The former bounds the (arguably more natural) quantity $\mathbb{T}_F(x)$, and is applicable to all (not just monotone) DNF formulas. However, the bound of Lemma 3.4 has no dependence on the read parameter k of the DNF F (as opposed to the linear dependence in Lemma 1.1). Like Lemma 1.1, Lemma 3.4 is not far from optimal: consider the OR of r variables $F(x) = x_1 \vee \cdots \vee x_r$. This monotone DNF has $\mathbf{E}[\mathbb{A}_F(\boldsymbol{x})] = r/2$, which is within a small constant factor of the upper bound $\ln(1/(1-\mu)) = (\ln 2) \cdot r \approx 0.693 \cdot r$.

The proof of Lemma 3.4 uses a straightforward corollary of the Fortuin–Kasteleyn–Ginibre (FKG) correlation inequality for monotone functions:[3]

Theorem 7 (FKG inequality)**.** *Let f and g be monotone functions over $\{-1,1\}^n$. Then*

$$\mathbf{Pr}\left[f(\boldsymbol{x})\right]\mathbf{Pr}\left[g(\boldsymbol{x})\right] \leq \mathbf{Pr}\left[f(\boldsymbol{x}) \wedge g(\boldsymbol{x})\right].$$

Corollary 3.5. *Let f, g and h be monotone functions over $\{-1,1\}^n$, where f and h depend on disjoint sets of variables. Then*

$$\mathbf{Pr}\left[f(\boldsymbol{x}) \wedge \neg\, g(\boldsymbol{x})\right] \leq \mathbf{Pr}\left[f(\boldsymbol{x}) \wedge \neg\, g(\boldsymbol{x}) \mid \neg\, h(\boldsymbol{x})\right]. \tag{7}$$

Proof. We may rewrite the LHS as $\mathbf{Pr}[f] - \mathbf{Pr}[f \wedge g]$ and may rewrite the RHS as

$$\frac{\mathbf{Pr}[f \wedge \neg\, h] - \mathbf{Pr}[f \wedge g \wedge \neg\, h]}{1 - \mathbf{Pr}[h]} = \frac{\mathbf{Pr}[f] - \mathbf{Pr}[f \wedge h] - \mathbf{Pr}[f \wedge g] + \mathbf{Pr}[f \wedge g \wedge h]}{1 - \mathbf{Pr}[h]}.$$

Thus (7) holds if and only if

$$(\mathbf{Pr}[f] - \mathbf{Pr}[f \wedge g]) \cdot (1 - \mathbf{Pr}[h]) \leq \mathbf{Pr}[f] - \mathbf{Pr}[f \wedge h] - \mathbf{Pr}[f \wedge g] + \mathbf{Pr}[f \wedge g \wedge h],$$

which on simplifying and rearranging is equivalent to

$$\mathbf{Pr}[f \wedge h] + \mathbf{Pr}[f \wedge g] \cdot \mathbf{Pr}[h] \leq \mathbf{Pr}[f] \cdot \mathbf{Pr}[h] + \mathbf{Pr}[f \wedge g \wedge h].$$

Since f and h are over disjoint sets of variables we have $\mathbf{Pr}[f \wedge h] = \mathbf{Pr}[f] \cdot \mathbf{Pr}[h]$, so the above further simplifies to $\mathbf{Pr}[f \wedge g] \cdot \mathbf{Pr}[h] \leq \mathbf{Pr}[f \wedge g \wedge h]$, which follows from the FKG inequality since both $f \wedge g$ and h are monotone. □

[3] We remark that Lemma 3.4 is essentially implicit in the proof of Lemma 30 of [KLW10]. As mentioned earlier, Lemma 30 of [KLW10] relies on the Four Functions Theorem, a generalization of the FKG inequality; here we have chosen to give an alternative proof that only uses the FKG inequality.

Proof of Lemma 3.4. We begin by writing

$$\begin{aligned}
\mathbf{Pr}\left[\neg F(\boldsymbol{x})\right] &= \mathbf{Pr}\left[\neg T_1(\boldsymbol{x}) \wedge \cdots \wedge \neg T_M(\boldsymbol{x})\right] \\
&= \prod_{i=1}^{M} \left(1 - \mathbf{Pr}\left[T_i(\boldsymbol{x}) \mid \neg T_1(\boldsymbol{x}) \wedge \cdots \wedge \neg T_{i-1}(\boldsymbol{x})\right]\right) \\
&\leq \left(1 - \frac{\sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \mid \neg T_1(\boldsymbol{x}) \wedge \cdots \wedge \neg T_{i-1}(\boldsymbol{x})\right]}{M}\right)^M. \qquad \text{(AM-GM inequality)}
\end{aligned}$$

Rearranging, recalling that $1 - \mu = \mathbf{Pr}[\neg F(\boldsymbol{x})]$, and using $\ln(1-x) \leq \exp(-x)$, we obtain

$$\sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \mid \neg T_1(\boldsymbol{x}) \wedge \cdots \wedge \neg T_{i-1}(\boldsymbol{x})\right] \leq \ln\left(\frac{1}{1-\mu}\right). \tag{8}$$

Since $\mathbb{A}_F(x) = \sum_{i=1}^{T} A_i(x)$, we have

$$\begin{aligned}
\mathbf{E}\left[\sum_{i=1}^{M} A_i(\boldsymbol{x})\right] &= \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \wedge \neg \phi_i(\boldsymbol{x})\right] && \text{(Definition of } A_i\text{)} \\
&\leq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \wedge \neg \phi_i(\boldsymbol{x}) \,\middle|\, \bigwedge_{\substack{j<i \\ T_j \cap T_i = \emptyset}} \neg T_j(\boldsymbol{x})\right] && \text{(Corollary 3.5)} \\
&\leq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \,\middle|\, \neg \phi_i(\boldsymbol{x}) \wedge \bigwedge_{\substack{j<i \\ T_j \cap T_i = \emptyset}} \neg T_j(\boldsymbol{x})\right] && \text{(using } \mathbf{Pr}[A \wedge B|C] \leq \mathbf{Pr}[A|B \wedge C]\text{)} \\
&= \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \mid \neg T_1(\boldsymbol{x}) \wedge \cdots \wedge \neg T_{i-1}(\boldsymbol{x})\right], && \text{(Definition of } \phi_i\text{)}
\end{aligned}$$

which, along with (8), completes the proof of Lemma 3.4. □

4 Large-deviation bounds, polynomial approximators, and agnostic learning for read-k DNFs

In this section we first (Section 4.1) establish large-deviation bounds for the random variable $\mathbb{A}_F(\boldsymbol{x}) = \sum_{i=1}^{M} A_i(\boldsymbol{x})$ when F is any read-k DNF. We then (Section 4.2) use these bounds to construct polynomial approximators which have small Fourier ℓ_1 norm; by Theorem 6, these yield agnostic learning results for read-k DNFs in the uniform-distribution membership-query model.

4.1 Large-deviation bounds for number of disjoint satisfied terms

Following [KLW10], we use our bound on $\mathbf{E}\left[\mathbb{T}_F(\boldsymbol{x})\right]$ (Lemma 1.1) to establish strong large-deviation bounds for the random variable $\mathbb{A}_F(\boldsymbol{x})$. The following lemma is reminiscent of Claim 27 and Lemma 31 of [KLW10], but is exponentially stronger thanks to Lemma 1.1, which lets us replace the '16^k' of [KLW10]'s Lemma 31 with a 'k.'

Lemma 4.1. *Let $F = T_1 \vee \cdots \vee T_M$ be an M-term read-k DNF and let μ denote $\mathbf{Pr}[F(\boldsymbol{x})]$. For all $j \in \mathbb{N}$,*

$$\mathbf{Pr}\left[\mathbb{A}_F(\boldsymbol{x}) = j\right] \leq \left(\frac{ek\ln(1/(1-\mu))}{j}\right)^j.$$

Proof. For a set $S \subseteq [M]$ and an input x, we first observe that $\wedge_{i\in S} A_i(x) = 0$ if there exist distinct indices $i < j \in S$ such that $T_i \cap T_j \neq \emptyset$. (To see this, note that if $T_i \cap T_j \neq \emptyset$ then T_i appears as a term in ϕ_j, so indeed $A_i(x) = T_i(x) \wedge \neg\, \phi_i(x)$ and $A_j(x) = T_j(x) \wedge \neg\, \phi_j(x)$ cannot be simultaneously satisfied.) With this observation we can bound:

$$\begin{aligned}
\mathbf{Pr}\left[\bigwedge_{i\in S} A_i(\boldsymbol{x})\right] &= \mathbf{Pr}\left[\bigwedge_{i\in S} T_i(\boldsymbol{x}) \wedge \neg\, \phi_i(\boldsymbol{x})\right] \cdot \mathbb{1}\left[\, T_i \cap T_j = \emptyset \;\forall\, i, j \in S \text{ with } i < j \,\right] \\
&\leq \mathbf{Pr}\left[\bigwedge_{i\in S} T_i(\boldsymbol{x})\right] \cdot \mathbb{1}\left[\, T_i \cap T_j = \emptyset \;\forall\, i, j \in S \text{ with } i < j \,\right] \\
&\leq \prod_{i\in S} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right].
\end{aligned}$$

Therefore

$$\begin{aligned}
\mathbf{Pr}\left[\mathbb{A}_F(\boldsymbol{x}) = j\right] \;\leq\; \sum_{\substack{S\subseteq[M]\\|S|=j}} \mathbf{Pr}\left[\bigwedge_{i\in S} A_i(\boldsymbol{x})\right] &\leq \sum_{\substack{S\subseteq[M]\\|S|=j}} \prod_{i\in S} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \\
&\leq \binom{M}{j}\left(\frac{\sum_{i=1}^M \mathbf{Pr}\left[T_i(\boldsymbol{x})\right]}{M}\right)^j \qquad \text{(AM-GM inequality)} \\
&\leq \left(\frac{ek\ln(1/(1-\mu))}{j}\right)^j. \quad \text{(Lemma 1.1 and } \tbinom{x}{y} \leq (\tfrac{ex}{y})^y\text{)}
\end{aligned}$$

This completes the proof. □

4.2 Approximating polynomials for read-k DNFs

Given their tail bound on $\mathbf{Pr}[\mathbb{A}_F(\boldsymbol{x}) = j]$ for F a read-k DNF, [KLW10] then shows that $P_d(\mathbb{A}_F(\boldsymbol{x}))$ is a polynomial approximator for F with ℓ_1 norm that is doubly exponential in k. Here P_d is the univariate polynomial mapping 0 to 0 and all points in $[d]$ to 1 that was described at the beginning of Section 3, where $d := \exp(O(k))\ln(1/\varepsilon))$. With our improved tail bound provided by Lemma 4.1, the same construction and analysis would show that we can instead take $d := O(k\ln(1/\varepsilon))$, and this would result in an ℓ_1-norm bound which is singly rather than doubly exponential in k as was the case in [KLW10].

In the interests of obtaining a further improvement in the dependence on k, we instead analyze a slightly different construction which uses a different univariate polynomial. (Briefly, the upshot is that our bound on the ℓ_1 norm will be $M^{O(k^{3/2})}$, whereas it would have been $M^{O(k^2)}$ had we used [KLW10]'s univariate polynomial P_d.) The analysis of this construction uses the following technical lemma:

Lemma 4.2. *For all integers $r \geq 1$ and $\varepsilon > 0$ there are univariate polynomials $\Psi_r^+, \Psi_r^- : \mathbb{R} \to \mathbb{R}$ of degree $O(\sqrt{r} \cdot \log(1/\varepsilon))$ satisfying*

1. $\Psi_r^+(0) = \Psi_r^-(0) = 0$;
2. *For* $j = 1, \ldots, r$, *it holds that* $\Psi_r^+(j) \in [1, 1+\varepsilon]$ *and* $\Psi_r^-(j) \in [1-\varepsilon, 1]$;
3. *For all* $j \geq r$,
   (a) $1 \leq \Psi_r^+(j) \leq (2j/r)^{O(\sqrt{r}\cdot\log(1/\varepsilon))}$.
   (b) $1 \geq \Psi_r^-(j) \geq -(2j/r)^{O(\sqrt{r}\cdot\log(1/\varepsilon))}$.

Proof. The desired polynomials are obtained by shifting, scaling, and powering the Chebyshev polynomial of the first kind in a fairly standard way, corresponding to well-known constructions of ε-approximating polynomials for the OR function (with a bit of care to ensure the upper and lower bounds stipulated in (2) and (3)). We recall that the k-th Chebychev polynomial T_k is a degree-k real univariate polynomial with the following properties:

(*i*) $|T_k(x)| \leq 1$ for all $|x| \leq 1$.

(*ii*) For all k, $T_k(1) = 1$. If k is odd then $T_k(-1) = -1$ and $T_k(x) < -1$ for $x < -1$.

(*iii*) The derivative satisfies $T_k'(x) \geq k^2$ for all $x \geq 1$.

Following [NS94] (with a slight twist), for Ψ_r^+ we choose $k = 2\lceil\sqrt{r}\rceil + 1$ (note that k is odd), we define $c = 1/T_k(\frac{r}{r-1})$, and we define

$$\Psi_r^+(j) = \frac{1 - \left(cT_k(\frac{r+(-2r+1)j/r}{r-1})\right)^\ell}{1-\varepsilon/4}, \tag{9}$$

where ℓ is the smallest odd integer that is at least $\log(4/\varepsilon)$. The claimed degree bound clearly holds. By property (*ii*) we have that $\Psi_r^+(0) = 0$, giving (1). Properties (*ii*) and (*iii*) ensure that $0 < c \leq 1/4$, and consequently for $j \in [1, r]$, by (*i*) we have that the numerator $1 - \left(cT_k(\frac{r+(-2r+1)j/r}{r-1})\right)^\ell$ is in $[1-\varepsilon/4, 1+\varepsilon/4]$ for $j \in [1, r]$, and hence $\Psi_r^+(j) \in [1, 1+\varepsilon]$ for such j, giving (2). To establish (3), we use the following standard fact from approximation theory (see e.g. [Riv74]):

Fact 4.3. *Let* $a(t)$ *be a polynomial of degree at most* d *for which* $|a(t)| \leq b$ *in the interval* $[-1, 1]$. *Then* $|a(t)| \leq b|2t|^d$ *for all* $|t| \geq 1$.

Taking $a(t) = \Psi_r^+(r \cdot (t+1)/2)$, we have that $|a(t)| \leq 1 + \varepsilon < 2$ for all $t \in [-1, 1]$, so we may apply Fact 4.3 to $a(t)$ taking $b = 2$, and we get that $|a(t)| \leq 2 \cdot |2t|^{O(\sqrt{r}\cdot\log(1/\varepsilon))}$ for all $t \geq 1$. This is equivalent to $\Psi_r^+(j) \leq 2 \cdot (2 \cdot (\frac{2j}{r} - 1))^{O(\sqrt{r}\cdot\log(1/\varepsilon))}$ for $j \geq r$, which gives the upper bound of part (3)(a), $\Psi_r^+(j) \leq (2j/r)^{O(\sqrt{r}\cdot\log(1/\varepsilon))}$. The lower bound, $1 \leq \Psi_r^+(j)$ for $j \geq r$, follows from (*ii*) since $\left(cT_k(\frac{r+(-2r+1)j/r}{r-1})\right)^\ell < 0$ (this uses that k is odd and ℓ is odd).

The construction for Ψ_r^- is very similar: we choose k, ℓ and c as before and now we define

$$\Psi_r^+(j) = \frac{1 - \left(cT_k(\frac{r+(-2r+1)j/r}{r-1})\right)^\ell}{1+\varepsilon/4},$$

where ℓ is the smallest *even* integer that is at least $\log(4/\varepsilon)$. Parts (1) and (2) are established essentially as before, as is the lower bound $\Psi_r^-(j) \geq -(2j/r)^{O(\sqrt{r\log(1/\varepsilon)})}$ of part (3)(b). Finally

the upper bound of (3)(b), $1 \geq \Psi_r^-(j)$ for $j \geq r$, in fact holds for all $j \in \mathbb{R}$ as an easy consequence of ℓ being even. □

Claim 4.4 (Approximating polynomials for small-width read-k DNFs)**.** *Let F be an M-term width-w read-k DNF, and let $\varepsilon > 0$. Then there is a polynomial P^+ of Fourier ℓ_1-norm at most $M^{O(\sqrt{k}\cdot(\log(1/\varepsilon))^2)} \cdot 2^{O(k^{3/2}w\cdot(\log(1/\varepsilon))^2)}$ that satisfies $\mathbf{E}[(P^+(\boldsymbol{x}) - F(\boldsymbol{x}))^2] \leq \varepsilon^2$ and upper sandwiches F, i.e. $P^+(x) \geq F(x)$ for all $x \in \{-1,1\}^n$.*

Proof. If $\mathbf{Pr}[F(\boldsymbol{x})] > 1 - \varepsilon^2/2$ then the constant 1 is the desired polynomial, so we assume that $\mathbf{Pr}[F(\boldsymbol{x})] \leq 1 - \varepsilon^2/2$ and show how to construct the desired upper sandwiching approximator P^+. Consider the polynomial $P^+ : \{-1,1\}^n \to \mathbb{R}$:

$$P^+(x) := \Psi_r^+(\mathbb{A}_F(x)),$$

where $r = Ck(\ln(1/\varepsilon))^2$ for some universal constant $C > 0$ that we fix later. Since $P^+(x) = 0$ for all x such that $\mathbb{A}_F(x) = 0$, $P^+(x) \geq 1$ for all x such that $\mathbb{A}_F(x) \geq 1$, and $\mathbb{A}_F(x) \geq 1$ if and only if $F(x) = 1$, we have that P^+ is an upper sandwich for F. To bound the error $\mathbf{E}[(P^+(\boldsymbol{x}) - F(\boldsymbol{x}))^2]$, we have that $\mathbf{E}\left[(P^+(\boldsymbol{x}) - F(\boldsymbol{x}))^2\right]$ is equal to

$$\begin{aligned}
&\sum_{j=0}^{M} \mathbf{Pr}\left[\mathbb{A}_F(\boldsymbol{x}) = j\right] \cdot \mathbf{E}\left[(P^+(\boldsymbol{x}) - F(\boldsymbol{x}))^2 \mid \mathbb{A}_F(\boldsymbol{x}) = j\right] \\
&\leq \frac{\varepsilon^2}{2} + \sum_{j=r+1}^{M} \mathbf{Pr}\left[\mathbb{A}_F(\boldsymbol{x}) = j\right] \cdot (\Psi_r^+(j) - 1)^2 && \text{(Lemma 4.2, items 1 and 2(a))} \\
&\leq \frac{\varepsilon^2}{2} + \sum_{j=r+1}^{\infty} \left(\frac{ek\ln(1/(1-\mu))}{j}\right)^j \cdot (2j/r)^{O(\sqrt{r}\cdot\log(1/\varepsilon))} \\
& && \text{(Lemma 4.1 and part 3(a) of Lemma 4.2)} \\
&\leq \frac{\varepsilon^2}{2} + \sum_{j=r+1}^{\infty} \left(\frac{2ek\ln(1/\varepsilon)}{j}\right)^j \cdot (2j/r)^{O(\sqrt{r}\cdot\log(1/\varepsilon))} && (\mu \leq \varepsilon^2/2) \\
&\leq \frac{\varepsilon^2}{2} + \sum_{j=r+1}^{\infty} \left(\frac{2ek\ln(1/\varepsilon)}{j}\right)^j \cdot (2j/r)^{j} && \text{(definition of } r) \\
&\leq \frac{\varepsilon^2}{2} + \sum_{j=r+1}^{\infty} \left(\frac{1}{2\ln(1/\varepsilon)}\right)^j && \text{(choice of } C \text{ in definition of } r) \\
&\leq \varepsilon^2. && \text{(definition of } r)
\end{aligned}$$

To bound the Fourier ℓ_1-norm, we view $P^+(x)$ as

$$P^+(x) = \Psi_r^+\left(A_1(x) + \cdots + A_M(x)\right), \quad \text{where } A_i(x) := T_i \wedge \neg\,\phi_i(x) \quad \text{and } \phi_i(x) = \bigvee_{\substack{j<i \\ T_j \cap T_i \neq \emptyset}} T_j(x).$$

Since F is a width-w read-k DNF formula each DNF $\phi_i(x)$ has at most kw terms; given this, a simple argument using Fact 2.1 (see Fact 25 of [KLW10]) gives that each A_i satisfies $\|A_i\|_1 \leq 2^{kw}$, and hence the argument $\mathbb{A}_F = A_1 + \cdots + A_M$ to Ψ_r^+ satisfies $\|\mathbb{A}_F\|_1 \leq M2^{kw}$. Turning to the

univariate function $\Psi_r^+(t)$, we recall that the Chebyshev polynomials satisfy the recurrence relation $T_0(x) = 1$, $T_1(x) = x$, $T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$, and from this it is clear that the k-th Chebyshev polynomial $T_n(x) = \sum_{i=0}^n c_i t^i$ satisfies $\sum_{i=0}^n |c_i| \leq 3^n$. Recalling (9), it follows easily that if we let $a_0, a_1, \ldots$ denote the coefficients of the the univariate degree-$O(\sqrt{r} \cdot \log(1/\varepsilon))$ polynomial $\Psi_r^+(j)$, we have that the sum of the magnitudes of the a_i's is at most $2^{O(\sqrt{r} \cdot \log(1/\varepsilon))}$. Now applying Fact 2.2, we get that the Fourier ℓ_1-norm $\|P^+\|_1$ is at most $2^{O(\sqrt{r} \cdot \log(1/\varepsilon))} \cdot (M2^{kw})^{O(\sqrt{r} \cdot \log(1/\varepsilon))} = M^{O(\sqrt{k} \cdot (\log(1/\varepsilon))^2)} \cdot 2^{O(k^{3/2} w \cdot (\log(1/\varepsilon))^2)}$ as claimed, and the proof is complete. □

4.3 Agnostic learning for read-k DNF formulas

Combining Theorem 6 and Claim 4.4, we immediately get a uniform-distribution membership-query agnostic learning algorithm for M-term read-k width-w DNFs under the uniform distribution that runs in time $\mathrm{poly}(n, M^{O(\sqrt{k} \cdot (\log(1/\varepsilon))^2)} \cdot 2^{O(k^{3/2} w \cdot (\log(1/\varepsilon))^2)})$. To extend this to general M-term read-k DNFs with no width restriction, we observe that any M-term DNF F can be modified to a DNF F' of width at most $w := \log(M/\varepsilon^2) < 2\log(M/\varepsilon)$ simply by deleting all terms of width greater than w, and the resulting DNF F' has $\mathbf{Pr}[F'(\boldsymbol{x}) \neq F(\boldsymbol{x})] \leq \varepsilon^2$, which is equivalent to $\mathbf{E}[(F'(\boldsymbol{x}) - F(\boldsymbol{x}))^2] \leq \varepsilon^2$. Using $(P^+ - F)^2 \leq 2(P^+ - F')^2 + 2(F' - F)^2$, we obtain the following:

Corollary 4.5 (Agnostically learning read-k DNF formulas; restatement of Theorem 2)**.** *There is an algorithm that agnostically learns the class of M-term read-k DNFs under the uniform distribution with membership queries and runs in time* $\mathrm{poly}(n, (M/\varepsilon)^{O(k^{3/2} \cdot (\log(1/\varepsilon))^2)})$.

This is a significant improvement of the previous best agnostic learning runtime for this class, due to [KLW10], which as described earlier was $\mathrm{poly}(n) \cdot (M/\varepsilon)^{\exp(O(k)) \log(1/\varepsilon)}$.

5 Sandwiching polynomials and PRGs for read-k DNFs

In this section we extend the construction of polynomial approximators from the previous section to obtain *sandwiching* polynomial approximators which, by Lemma 2.3, give unconditional PRGs for read-k DNFs.

5.1 Sandwiching polynomial approximators and fooling read-k DNFs: dealing with highly biased formulas

The proof of Claim 4.4 shows that if $\mathbf{Pr}[F(\boldsymbol{x})] \leq 1 - \varepsilon$ then there is a polynomial $P^+(x)$, constructed using Ψ_r^+, which is an upper sandwiching approximator for F. For such an F an entirely analogous construction using Ψ_r^- yields a lower approximator with the same degree and Fourier ℓ_1-norm, so the proof of Claim 4.4 in fact yields sandwiching polynomials for any read-k DNF F that satisfies $\mathbf{Pr}[F(\boldsymbol{x})] \leq 1 - \varepsilon$.

What about read-k DNFs F that have $\mathbf{Pr}[F(\boldsymbol{x})] > 1 - \varepsilon$? The constant 1 is an ε-approximating upper sandwiching polynomial, so it remains to construct a lower sandwiching polynomial for a read-k DNF F that has $\mathbf{Pr}[F(\boldsymbol{x})] > 1 - \varepsilon$.

Remark 8. As alluded to in the Introduction, there is a slight gap in the analysis of [KLW10] establishing their claimed PRG, arising from the case that we are now considering, where $\mathbf{Pr}[F(\boldsymbol{x}) = 1] = \mu > 1 - \varepsilon$ and we need to construct a *lower* sandwiching polynomial that ε-approximates F. [KLW10] claims (implicitly, at the very end of their Section 6) to give a lower sandwiching

polynomial $q_{F,d}$ with ℓ_1 bound $\|q_{F,d}\|_1 \leq M^{O(16^k \cdot \log(1/\varepsilon))}$, which would in turn yield their claimed $\exp(O(k)) \cdot \log(1/\varepsilon) \cdot \log M + O(\log n)$ PRG seed length. But in fact their analysis, which is based on Lemma 31 of their paper, only establishes a bound of $\|q_{F,d}\|_1 \leq M^{O(2^{4k} \cdot \log(1/(1-\mu)))}$ (observe that the parameter denoted "ε" in their Lemma 31 corresponds to the quantity $1 - \mathbf{Pr}[F] = 1 - \mu$). Since $1 - \mu$ can be arbitrarily small compared to ε, *a priori* this does not give any upper bound on the ℓ_1-norm of $q_{F,d}$ in terms of ε, and likewise does not translate into any seed length bound depending on ε (as opposed to depending on μ).

Towards the goal of constructing lower sandwiching polynomials, we prove the following lemma:

Lemma 5.1. *Let $0 < \varepsilon < 0.1$ and let F be an M-term read-k DNF over $\{0,1\}^n$ with $\mathbf{Pr}[F(\boldsymbol{x})] > 1 - \varepsilon^2$. Then for $k' := k + \log(1/\varepsilon) + O(1)$ there is a read-k' DNF F' over $\{0,1\}^{n+\log(1/\varepsilon)+O(1)}$ which lower sandwiches F (i.e. $F'(x) \leq F(x)$ for all x) and has $\mathbf{Pr}[F'(\boldsymbol{x})] \in [1-\varepsilon, 1-\varepsilon^2]$.*

Proof of Lemma 5.1. Given a DNF F we say that a DNF H is a *sub-DNF* of F if every term of H is also a term of F. It is clear that every sub-DNF of F lower sandwiches F, and also that if F is read-k then so is every sub-DNF of F. If any sub-DNF H of F has $\mathbf{Pr}[H(\boldsymbol{x})] \in [1-\varepsilon, 1-\varepsilon^2]$ then Lemma 5.1 holds for F, so in the rest of the proof of Lemma 5.1 we subsequently assume that F is such that every sub-DNF H of F has $\mathbf{Pr}[H(\boldsymbol{x})] \notin [1-\varepsilon, 1-\varepsilon^2]$.

The following terminology will be useful for us: given a DNF G and a term T in it, we write $\mathrm{unique}(T, G)$ to denote the probability

$$\mathrm{unique}(T, G) := \mathbf{Pr}\left[T(\boldsymbol{x}) = 1 \text{ and } T'(\boldsymbol{x}) = 0 \text{ for every other term } T' \text{ in } G \text{ besides } T\right].$$

Claim 5.2. *There is a sub-DNF H of F and a term T in H such that (i) $\mathbf{Pr}[H(\boldsymbol{x})] > 1 - \varepsilon^2$ and (ii) $\mathrm{unique}(T, H) > \varepsilon/2$.*

Proof of Claim 5.2. Consider the execution of the following simple iterative procedure run on F:

> **Algorithm** FindHeavyUniqueTerm(F)
>
> **Input:** A DNF F such that $\mathbf{Pr}[F(\boldsymbol{x})] > 1 - \varepsilon^2$ and no sub-DNF H of F has $\mathbf{Pr}[H(\boldsymbol{x})] \in [1-\varepsilon, 1-\varepsilon^2]$.
>
> **Output:** A sub-DNF H of F and a term T of H such that $\mathbf{Pr}[H(\boldsymbol{x})] > 1 - \varepsilon^2$ and $\mathrm{unique}(T, H) > \varepsilon/2$.
>
> 1. If some term T in F has $\mathrm{unique}(T, F) > \varepsilon/2$ then set $H = F$ and output (H, T).
> 2. Pick any term T in F and remove T from F. Go to Step 1.

To analyze FindHeavyUniqueTerm, first note that by assumption the initial argument F to FindHeavyUniqueTerm has $\mathbf{Pr}[F(\boldsymbol{x})] > 1 - \varepsilon^2$. If Step 2 is reached when F is some DNF F_{old} such that $\mathbf{Pr}[F_{\mathrm{old}}(\boldsymbol{x})] > 1 - \varepsilon^2$ and term T is removed to form F_{new}, then since T must satisfy $\mathrm{unique}(T, F_{\mathrm{old}}) \leq \varepsilon/2$ and $\mathbf{Pr}[F_{\mathrm{old}}(\boldsymbol{x})] = \mathbf{Pr}[F_{\mathrm{new}}(\boldsymbol{x})] + \mathrm{unique}(T, F_{\mathrm{old}})$, it must be the case that

$\mathbf{Pr}[F_{\text{new}}(\boldsymbol{x})] > 1-\varepsilon^2-\varepsilon/2 \geq 1-\varepsilon$; since F_{new} is a sub-DNF of F, it follows that $\mathbf{Pr}[F_{\text{new}}(\boldsymbol{x})]$ must be greater than $1-\varepsilon^2$. So every time Step 1 is reached, the current DNF F has $\mathbf{Pr}[F(\boldsymbol{x})] > 1-\varepsilon^2$.

If Step 1 were reached an $(M+1)$st time, the current DNF F would have no terms and hence would have $\mathbf{Pr}[F(\boldsymbol{x})] = 0$, but this would contradict $\mathbf{Pr}[F(\boldsymbol{x})] > 1-\varepsilon^2$. Hence in one of the first M executions of Step 1, it must be the case that some term T in F has $\text{unique}(T,F) > \varepsilon/2$. This proves the claim. □

We will combine Claim 5.2 with the following two facts, whose proofs are easy exercises:

Fact 5.3. *Let H be a DNF over variables $x_1,\dots,x_n$, T a term in H, and H' the DNF obtained by removing T from H, so $H = H' \vee T$ and*

$$\mathbf{Pr}[H(\boldsymbol{x})] = \mathbf{Pr}[H'(\boldsymbol{x})] + \text{unique}(T,H).$$

Let $G(y_1,\dots,y_r)$, $G:\{0,1\}^r \to \{0,1\}$ be any Boolean function over new variables $y_1,\dots,y_r$ and let p denote $\mathbf{Pr}_{\boldsymbol{y}\leftarrow\{0,1\}^r}[G(\boldsymbol{y})=1]$. Let $H^*(x_1,\dots,x_n,y_1,\dots,y_r)$ be the function

$$H^*(x_1,\dots,x_n,y_1,\dots,y_r) = H'(x_1,\dots,x_n) \vee (T(x_1,\dots,x_n) \wedge G(y_1,\dots,y_r)).$$

Then $\mathbf{Pr}_{(\boldsymbol{x},\boldsymbol{y})\leftarrow\{0,1\}^{n+r}}[H^*(\boldsymbol{x},\boldsymbol{y})=1] = \mathbf{Pr}_{\boldsymbol{x}\leftarrow\{0,1\}^n}[H'(\boldsymbol{x})] + p\cdot\text{unique}(T,H)$.

Fact 5.4. *Let $p\in[0,1]$ be a dyadic fraction $p = a/2^r$ for some integer $a\in[0,1,\dots,2^r]$. Then there is an r-term (and hence read-r) DNF G over $\{0,1\}^r$ such that $\mathbf{Pr}_{\boldsymbol{y}\leftarrow\{0,1\}^r}[G(\boldsymbol{y})=1] = p$.*

Let H and T be as given by Claim 5.2 and let H' be the DNF obtained from H by removing T. We have $\text{unique}(T,H) > \varepsilon/2$ and hence $\mathbf{Pr}[H'(\boldsymbol{x})] < 1-\varepsilon/2$ while $\mathbf{Pr}[H(\boldsymbol{x})] = \mathbf{Pr}[H'(\boldsymbol{x})] + \text{unique}(T,H) > 1-\varepsilon^2 > 1-\varepsilon/4$. It follows that there is a dyadic fraction $p = a/2^r$, where $r = \log(1/\varepsilon)+O(1)$, such that $\mathbf{Pr}[H'(\boldsymbol{x})] + p\cdot\text{unique}(T,H) \in [1-\varepsilon/2, 1-\varepsilon/4]$. Given this Lemma 5.1 follows from Facts 5.3 and 5.4, taking $F'(x_1,\dots,x_n)$ to be the function H^* from Fact 5.3, and observing that H' is a read-$(k' = k+\log(1/\varepsilon)+O(1))$-DNF (since H is a read-r DNF and G is an r-term read-r DNF over fresh variables). □

With Lemma 5.1 in hand, we can summarize what has been shown thus far as follows:

Theorem 9 (Sandwiching polynomials for read-k DNFs)**.** *Let $F:\{-1,1\}^n\to\{0,1\}$ be computed by an M-term read-k DNF, and let $\varepsilon>0$. There exists polynomials $P^+,P^-:\{-1,1\}^n\to\mathbb{R}$ such that*

$$P^-(x) \leq F(x) \leq P^+(x) \quad \text{for all } x\in\{-1,1\}^n,$$

and for $P\in\{P^+,P^-\}$ the following hold:

1. (Small Fourier ℓ_1)
$$\sum_{S\subseteq[n]} |\widehat{P}(S)| \leq (M/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}.$$
2. (L_1-approximators for F) $\mathbf{E}\left[|P(\boldsymbol{x})-F(\boldsymbol{x})|\right] \leq \varepsilon$.

Proof. Let F_ℓ be the DNF formula obtained from F by removing all terms of length greater than $\log(M/\varepsilon)$, and let F_u be the DNF formula obtained from F by trimming each term of length greater than $\log(M/\varepsilon)$ to contain (any set of) exactly $\log(M/\varepsilon)$ of its literals. It is easy to see that for all $x\in\{-1,1\}^n$ we have $F_\ell(x) \leq F(x) \leq F_u(x)$, and that $\mathbf{E}[F_u(\boldsymbol{x}) - F(\boldsymbol{x})] \leq \varepsilon$ and

$\mathbf{E}[F(\boldsymbol{x}) - F_\ell(\boldsymbol{x})] \le \varepsilon$. To prove the theorem it suffices to give an upper sandwiching polynomial P^+ for F_u and a lower sandwiching polynomial P^- for F_ℓ. Since the width of F_u is at most $w = \log(M/\varepsilon)$, the desired upper sandwiching polynomial P^+ for F_u is given by Claim 4.4. Turning to F_ℓ, if $\mathbf{Pr}[F_\ell(\boldsymbol{x})] \le 1 - \varepsilon^2$ then (as discussed at the start of Section 5.1) the desired polynomial P^- is given by the lower approximator analogue of Claim 4.4. Finally, if $\mathbf{Pr}[F_\ell(\boldsymbol{x})] > 1 - \varepsilon^2$, then the desired polynomial P^- is obtained by applying the lower approximator analogue of Claim 4.4 to the DNF F'_ℓ which lower sandwiches F_ℓ and is given by Lemma 5.1. □

Given Theorem 9 it is a simple matter to obtain Theorem 1:

Theorem 10 (Restatement of Theorem 1: a PRG for M-term read-k DNFs)**.** *There is an ε-PRG for the class of M-term read-k DNFs over $\{-1,1\}^n$ with seed length*

$$O\big(\log n + (k + \log(1/\varepsilon))^{3/2} \cdot (\log(1/\varepsilon))^2 \cdot \log(M/\varepsilon)\big).$$

Proof. The ℓ_1 norm of the sandwiching polynomials given by Theorem 9 is

$$(M/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2} \cdot (\log(1/\varepsilon))^2)}.$$

Hence by Lemma 2.3 any $(\varepsilon/(M/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2} \cdot (\log(1/\varepsilon))^2)})$-biased distribution 2ε-fools the class, and such distributions with the claimed seed length follow from the construction of Naor and Naor [NN93]. □

Remark 11. We observe that Lemma 5.1 can be used to patch the [KLW10] analysis but at the cost of a quantitative weakening of their claimed seed length. Since the [KLW10] claimed seed length for a read-k' DNF is $O(16^{k'} \cdot \log(1/\varepsilon) \cdot \log M + \log n)$, for $k' = k + \log(1/\varepsilon) + O(1)$ their analysis (augmented with Lemma 5.1) yields an actual seed length of $\exp(O(k)) \cdot \mathrm{poly}(1/\varepsilon) \cdot \log M + O(\log n)$. Note that the dependence on $1/\varepsilon$ is polynomial rather than logarithmic as was claimed in [KLW10]. (In contrast since our approach has a polynomial rather than exponential dependence on k, for us the use of Lemma 5.1 with $k' = k + \log(1/\varepsilon) + O(1)$ comes at a very small cost.)

6 Counting satisfying assignments of read-k DNFs

Early influential work of Karp and Luby [KL83] gave a randomized $\mathrm{poly}(n, M, 1/\varepsilon)$-time algorithm (i.e. an FPRAS) that approximates, to any desired $(1+\varepsilon)$-multiplicative accuracy, the fraction of satisfying assignments of an M-term n-variable DNF. At the heart of their algorithm is a simple and elegant deterministic reduction from *multiplicative* $(1+\varepsilon)$-approximation of DNFs to *additive* $\pm(\varepsilon/M)$-approximation of CNFs, which we now briefly describe.

The Karp–Luby reduction. The starting point of Karp and Luby's reduction is a basic identity concerning the quantity $\mathbf{Pr}[F(\boldsymbol{x})]$ we would like to multiplicatively approximate:

Fact 6.1. *Let $F = T_1 \vee \cdots \vee T_M$ be an M-term DNF formula. Then its fraction of satisfying assignments can be expressed as:*

$$\mathbf{Pr}\left[F(\boldsymbol{x})\right] = \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \cdot \mathbf{Pr}\left[\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}) \mid T_i(\boldsymbol{x})\right].$$

Proof. This holds because

$$\begin{aligned}\mathbf{Pr}\left[F(\boldsymbol{x})\right] &= \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \wedge (\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}))\right] \\ &= \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \cdot \mathbf{Pr}\left[\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}) \mid T_i(\boldsymbol{x})\right],\end{aligned}$$

where we have partitioned the set of satisfying assignments of F according to the first term T_i that each x satisfies. □

Writing $\tilde{\gamma}_i$ to denote an *additive* $\pm(\varepsilon/M)$-approximation of the quantity $\mathbf{Pr}[\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}) \mid T_i(\boldsymbol{x})]$, a straightforward argument (see e.g. Section 2 of [LV96]; we give a refined version of this argument in the proof of Theorem 4 below) shows that

$$\Gamma := \sum_{i=1}^{M} 2^{-|T_i|} \cdot \tilde{\gamma}_i = \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \cdot \tilde{\gamma}_i$$

is a *multiplicative* $(1 \pm \varepsilon)$-approximation to $\mathbf{Pr}[F(\boldsymbol{x})]$. To complete the reduction, we note that $\mathbf{Pr}[\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}) \mid T_i(\boldsymbol{x})]$ can be viewed as the fraction of satisfying assignments of a certain CNF (the CNF $\neg T_{i-1}(x) \wedge \cdots \wedge \neg T_1(x)$ restricted by the unique satisfying assignment $\rho \in \{-1, 1\}^{T_i}$ of the term T_i). Hence, the task of obtaining the M many estimates $\tilde{\gamma}_i$ is precisely that of additively approximating the acceptance probabilities of M many CNFs.

Applying the Karp–Luby reduction. Karp and Luby's randomized algorithm follows by combining the above reduction with a straightforward random sampling step to achieve the requisite $\pm(\varepsilon/M)$-approximation of CNFs.

To obtain a deterministic algorithm with a multiplicative $(1+\varepsilon)$ error guarantee, one can use a deterministic additive approximation algorithm for CNFs instead of random sampling. The current fastest such algorithm is due to Gopalan, Meka, and Reingold [GMR13]:

Theorem 12 (Absolute error approximate counting of CNFs)**.** *There is a deterministic algorithm which, given as input an n-variable M-clause CNF G and an accuracy parameter $\delta > 0$, runs in time $(M/\delta)^{\tilde{O}(\log((\log M)/\delta))}$ and outputs an estimate of the fraction of satisfying assignments of G to additive accuracy $\pm\delta$. That is, the algorithm outputs a value $\tilde{\gamma} \in [0, 1]$ satisfying*

$$\tilde{\gamma} = \mathbf{Pr}\left[G(\boldsymbol{x})\right] \pm \delta.$$

The Karp–Luby reduction along with this [GMR13] additive approximation algorithm (run M times with accuracy parameter $\delta := \varepsilon/M$ each time) yields a deterministic multiplicative $(1+\varepsilon)$-factor approximation algorithm for DNFs with overall runtime $(M/\varepsilon)^{\tilde{O}(\log(M/\varepsilon))}$.

We note the wide gap between the best known deterministic runtimes for absolute (additive) error and relative (multiplicative) error: $M^{\tilde{O}(\log \log M)}$ versus $M^{\tilde{O}(\log M)}$, an exponential difference in the exponents of the running times.

6.1 Faster relative error counting of read-k DNFs

As an easy consequence of our new structural results established in Section 3, we obtain a dramatically more efficient version of the Karp–Luby reduction for read-k DNFs. Combining this improved reduction with the [GMR13] algorithm and Theorem 1, our new PRG for read-k DNFs, we obtain correspondingly improved relative error counting algorithms.

Briefly, the connection to our structural results is as follows: in the Karp–Luby reduction, the accuracy to which one has to additively count is determined by the quantity $\mathbf{E}[\mathbb{T}_F(\boldsymbol{x})]$ defined in Section 3, the expected number of terms of F satisfied by a uniform random input $\boldsymbol{x}$. For a general M-term DNF this can be as large as $M\mu$ where $\mu = \mathbf{Pr}[F(\boldsymbol{x})]$, whereas for read-$k$ DNFs Lemma 1.1 gives the much smaller upper bound of $k\ln(1/(1-\mu)) \approx k\mu$. Consequently, in the Karp–Luby reduction, it suffices to additively count to a much coarser accuracy, namely (roughly) $\pm(\varepsilon/k)$ rather than $\pm(\varepsilon/M)$ as in the general case. We make this formal below but first we record the following useful technical proposition:

Proposition 6.2. *For $0 < \varepsilon \leq 1/e$, if $0 \leq \mu \leq 1-\varepsilon$ then*

$$\ln\left(\frac{1}{1-\mu}\right) \leq 2\mu\ln(1/\varepsilon).$$

Proof. If $\mu \geq 1/2$ then the claimed bound is immediate since $\ln\frac{1}{1-\mu} \leq \ln(1/\varepsilon) \leq 2\mu\ln(1/\varepsilon)$, and if $0 < \mu < 1/2$ then we have $\ln\frac{1}{1-\mu} = \ln(1+\frac{\mu}{1-\mu}) \leq \frac{\mu}{1-\mu} \leq 2\mu \leq 2\mu\ln(1/\varepsilon)$ (since $\varepsilon \leq 1/e$). □

Theorem 4. (Relative error approximate counting of read-k DNFs) *There is a deterministic algorithm which, given as input an M-term read-k DNF and an accuracy parameter $\varepsilon > 0$, runs in time*

$$\mathrm{poly}(n) \cdot \min\left\{(M/\varepsilon)^{\tilde{O}(\log((k\log M)/\varepsilon))}, (M/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}\right\}$$

and outputs a $(1+\varepsilon)$-multiplicative estimate of $\mathbf{Pr}[F(\boldsymbol{x})]$.

Proof. Let $F = T_1 \vee \cdots \vee T_M$. Our algorithm first computes, for each $i \in [M]$, an additive approximation $\tilde{\gamma}_i$ of the quantity $\gamma_i := \mathbf{Pr}[\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}) \mid T_i(\boldsymbol{x})]$ that is accurate to within $\pm\delta$ where $\delta := \varepsilon/(2k\ln(2/\varepsilon))$. Note that

$$\mathbf{Pr}\left[\neg T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg T_1(\boldsymbol{x}) \mid T_i(\boldsymbol{x})\right] = \mathbf{Pr}\left[G_i(\boldsymbol{x})\right],$$

where G_i is the i-term read-k CNF formula obtained by restricting $\neg T_{i-1}(x) \wedge \cdots \wedge \neg T_1(x)$ according to the unique satisfying assignment $\rho \in \{-1,1\}^{T_i}$ of the term T_i (i.e. $\rho_i = 1$ if x_i occurs positively in T_i and 0 if it occurs negatively). Therefore, we can run either the [GMR13] algorithm (Theorem 12) or the algorithm that corresponds to enumerating over all seeds of our PRG for read-k CNFs[4] that by Boolean duality, our (Theorem 1), with accuracy parameter $\delta = \varepsilon/(2k\ln(2/\varepsilon))$, to obtain all M of these estimates $\tilde{\gamma}_1, \ldots, \tilde{\gamma}_M$ in time

$$\mathrm{poly}(n) \cdot \min\left\{(M/\varepsilon)^{\tilde{O}(\log((k\log M)/\varepsilon))}, (M/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}\right\}.$$

[4]It is easy to confirm that, by Boolean duality, our structural result Lemma 1.1 has an exact analogue for the expected number of unsatisfied clauses in a read-k CNF formula, and that our PRG for read-k DNFs also extends to read-k CNFs.

Having obtained these M estimates our algorithm outputs

$$\Gamma := \min\left\{1, \sum_{i=1}^{M} 2^{-|T_i|}\,\tilde{\gamma}_i\right\} = \min\left\{1, \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right]\tilde{\gamma}_i\right\},$$

which we now claim satisfies $\Gamma = (1 \pm \varepsilon)\mu$ where μ denotes $\mathbf{Pr}[F(\boldsymbol{x})]$. The lower bound holds because

$$\begin{aligned}
\Gamma &\geq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \cdot \left(\gamma_i - \frac{\varepsilon}{2k\ln(2/\varepsilon)}\right) && \text{(Our choice of } \delta\text{)} \\
&= \mu - \frac{\varepsilon}{2k\ln(2/\varepsilon)} \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] && \text{(Fact 6.1)} \\
&\geq \mu - \frac{\varepsilon}{2k\ln(2/\varepsilon)} \cdot \mu \;>\; (1-\varepsilon)\mu,
\end{aligned}$$

where the penultimate inequality uses $\sum_{i=1}^{M} \mathbf{Pr}[T_i(\boldsymbol{x})] \geq \mathbf{Pr}[F(\boldsymbol{x})] = \mu$ (since every satisfying assignment of F has to satisfy at least one of its terms). For the upper bound, we have

$$\begin{aligned}
\Gamma &\leq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \left(\gamma_i + \frac{\varepsilon}{2k\ln(2/\varepsilon)}\right) && \text{(Our choice of } \delta\text{)} \\
&= \mu + \frac{\varepsilon}{2k\ln(2/\varepsilon)}\,\mathbf{E}\left[\,\mathbb{T}_F(\boldsymbol{x})\,\right] && \text{(Fact 6.1 and definition of } \mathbb{T}_F\text{)} \\
&\leq \mu + \underbrace{\frac{\varepsilon}{2\ln(2/\varepsilon)} \cdot \ln\left(\frac{1}{1-\mu}\right)}_{\Delta}. && \text{(Lemma 1.1)}
\end{aligned}$$

We consider two cases, depending on whether $\mu \leq 1 - (\varepsilon/2)$ or $\mu > 1 - (\varepsilon/2)$. In the first case, we apply Proposition 6.2 (with its "ε" parameter now instantiated as $\varepsilon/2$) to bound

$$\Delta \leq \frac{\varepsilon}{2\ln(2/\varepsilon)} \cdot 2\mu\ln(2/\varepsilon) = \varepsilon\mu,$$

and so indeed $\Gamma \leq (1+\varepsilon)\mu$. In the second case, since $\Gamma \leq 1$ we have that

$$\frac{\Gamma}{\mu} \leq \frac{1}{1-(\varepsilon/2)} \leq 1 + \varepsilon,$$

and hence again $\Gamma \leq (1+\varepsilon)\mu$. This completes the proof. □

6.2 Further improved runtime for monotone read-k DNFs that are not too wide

In this section we modify the Karp–Luby reduction to leverage our bounds on $\mathbf{E}[\mathbb{A}_F(\boldsymbol{x})]$ for monotone DNFs F (Section 3.2); recall that this is the expected number of *disjoint* terms of F satisfied by a uniform random input $\boldsymbol{x}$. At the cost of only achieving a $(2+\varepsilon)$-factor approximation (rather than a $(1+\varepsilon)$-factor approximation), we give a significantly faster algorithm for monotone read-k DNFs that are not too wide.

The starting point of our modified reduction is a variant of Fact 6.1 (recall our definition of ϕ_i in Equation (3)):

Fact 6.3. *Let $F = T_1 \vee \cdots \vee T_M$ be an M-term DNF formula. Then*

$$\mathbf{Pr}\left[F(\boldsymbol{x})\right] \leq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \cdot \mathbf{Pr}\left[\neg\,\phi_i(\boldsymbol{x}) \mid T_i(\boldsymbol{x})\right].$$

Proof. This holds because

$$\mathbf{Pr}\left[F(\boldsymbol{x})\right] \leq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \wedge \neg\,\phi_i(\boldsymbol{x})\right] = \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] \cdot \mathbf{Pr}\left[\neg\,\phi_i(\boldsymbol{x}) \mid T_i(\boldsymbol{x})\right],$$

where the inequality uses:

$$\begin{aligned}\mathbf{Pr}\left[F(\boldsymbol{x})\right] &= \sum_{i=1}^{M} \mathbf{Pr}\left[\,T_i(\boldsymbol{x}) \wedge (\neg\,T_{i-1}(\boldsymbol{x}) \wedge \cdots \wedge \neg\,T_1(\boldsymbol{x}))\,\right] \\ &\leq \sum_{i=1}^{M} \mathbf{Pr}\left[\,T_i(\boldsymbol{x}) \wedge \bigwedge_{\substack{j<i \\ T_j \cap T_i \neq \emptyset}} \neg\,T_j(\boldsymbol{x})\right] \;=\; \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \wedge \neg\,\phi_i(\boldsymbol{x})\right]. \qquad \square\end{aligned}$$

The key advantage of working with Fact 6.3 instead of Fact 6.1 comes from the fact that $\neg\,\phi_i$ is "much simpler" than $\neg\,T_{i-1}(x) \wedge \cdots \wedge \neg\,T_1(x)$: for a read-$k$ width-w DNF F, each $\neg\,\phi_i$ is a CNF with $O(kw)$ terms independent of i, whereas $\neg\,T_{i-1}(x) \wedge \cdots \wedge \neg\,T_1(x)$ has i terms (and i ranges from 1 to M). Consequently, in our modified reduction every one of the M CNFs that we additively count has size (number of terms) $O(kw)$, whereas in the Karp–Luby reduction these CNFs may have size as large as $\Omega(M)$.

Theorem 5. ($(2+\varepsilon)$-factor approximation for monotone read-k DNFs) *There is a deterministic algorithm which, given as input an M-term width-w read-k DNF and an accuracy parameter $\varepsilon > 0$, runs in time*

$$\mathrm{poly}(n) \cdot M \cdot \min\left\{(kw/\varepsilon)^{\tilde{O}(\log(k\log(kw)/\varepsilon))}, (kw/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}\right\}$$

and outputs a $(2+\varepsilon)$-factor estimate of $\mathbf{Pr}[F(\boldsymbol{x})]$.

Proof. Let $T_1 \vee \cdots \vee T_M$ be the terms of F. In close analogy with the algorithm in Theorem 4, our algorithm first computes, for each $i \in [M]$, an additive approximation $\tilde{\eta}_i$ of the quantity $\eta_i := \mathbf{Pr}[\neg\,\phi_i(x) \mid T_i(\boldsymbol{x})]$ that is accurate to within $\pm\delta$ where $\delta := \varepsilon/(2k)$. Note that

$$\mathbf{Pr}\left[\,\neg\,\phi_i(\boldsymbol{x}) \mid T_i(\boldsymbol{x})\,\right] = \mathbf{Pr}\left[H_i(\boldsymbol{x})\right],$$

where H_i is the CNF formula obtained by restricting $\neg\,\phi_i(x)$ according to the unique satisfying assignment $\rho \in \{-1,1\}^{T_i}$ of the term T_i. The key difference with Theorem 4—and the crux of our improvement here—is the fact that there are at most kw clauses in $\neg\,\phi_i(x)$. To see this, note that the number of clauses of $\neg\,\phi_i(x)$ is exactly the number of terms T_j in F such that $j < i$ and $T_j \cap T_i$; since $|T_i| \leq w$ and F is read-k, there can be at most $w(k-1)$ such terms.

Therefore, we can run either the [GMR13] algorithm (Theorem 12) or the algorithm that corresponds to enumerating over all seeds of our PRG for read-k CNFs (Theorem 1) of size $M = kw$, with accuracy parameter $\delta = \varepsilon/(2k)$, to obtain all M of these estimates $\tilde{\eta}_1, \ldots, \tilde{\eta}_M$ in time

$$\mathrm{poly}(n) \cdot M \cdot \min\left\{(kw/\varepsilon)^{\tilde{O}(\log(k\log(kw)/\varepsilon))}, (kw/\varepsilon)^{O((k+\log(1/\varepsilon))^{3/2}\cdot(\log(1/\varepsilon))^2)}\right\}.$$

Having obtained these M estimates our algorithm outputs

$$\Gamma := \min\left\{1, \sum_{i=1}^{M} 2^{-|T_i|}\,\tilde{\eta}_i\right\} = \min\left\{1, \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right]\tilde{\eta}_i\right\}$$

which we claim satisfies $(1-\varepsilon)\mu \leq \Gamma \leq (2+\varepsilon)\mu$, where μ denotes $\mathbf{Pr}[F(\boldsymbol{x})]$. The lower bound holds because

$$\begin{aligned}
\Gamma &\geq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right]\left(\eta_i - \frac{\varepsilon}{2k}\right) && \text{(Our choice of } \delta\text{)}\\
&\geq \mu - \frac{\varepsilon}{2k}\sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] && \text{(Fact 6.3 and } \mu = \mathbf{Pr}[F(\boldsymbol{x})]\text{)}\\
&\geq \mu - \frac{\varepsilon}{2k}\cdot\mu \;>\; (1-\varepsilon)\mu,
\end{aligned}$$

where the penultimate inequality uses $\sum_{i=1}^{M} \mathbf{Pr}[T_i(\boldsymbol{x})] \geq \mathbf{Pr}[F(\boldsymbol{x})] = \mu$. For the upper bound, we have

$$\begin{aligned}
\Gamma &\leq \sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right]\left(\eta_i + \frac{\varepsilon}{2k}\right) && \text{(Our choice of } \delta\text{)}\\
&= \left(\sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x}) \wedge \neg\,\phi_i(\boldsymbol{x})\right]\right) + \frac{\varepsilon}{2k}\sum_{i=1}^{M} \mathbf{Pr}\left[T_i(\boldsymbol{x})\right] && \text{(Definition of } \eta_i\text{)}\\
&= \mathbf{E}\left[\,\mathbb{A}_F(\boldsymbol{x})\,\right] + \frac{\varepsilon}{2k}\cdot\mathbf{E}\left[\,\mathbb{T}_F(\boldsymbol{x})\,\right] && \text{(Definitions of } \mathbb{T}_F \text{ and } \mathbb{A}_F\text{)}\\
&\leq \ln\left(\frac{1}{1-\mu}\right) + \frac{\varepsilon}{2}\cdot\ln\left(\frac{1}{1-\mu}\right) && \text{(Lemmas 3.4 and 1.1)}\\
&= \left(1+\frac{\varepsilon}{2}\right)\cdot\ln\left(\frac{1}{1-\mu}\right).
\end{aligned}$$

We consider two cases, depending on whether $\mu \geq 1/2$ or $\mu < 1/2$. In the first case, $\Gamma \leq 2\mu$ (since $\Gamma \leq 1$). In the second case, since $\ln(\frac{1}{1-\mu}) = \ln(1+\frac{\mu}{1-\mu}) \leq \frac{\mu}{1-\mu} \leq 2\mu$, we have

$$\left(1+\frac{\varepsilon}{2}\right)\cdot\ln\left(\frac{1}{1-\mu}\right) < \left(1+\frac{\varepsilon}{2}\right)\cdot 2\mu = (2+\varepsilon)\mu,$$

and this completes the proof. □

The most interesting parameter setting for Theorem 5 is to take ε to be a small absolute constant, yielding corollaries like the following:

Corollary 6.4. (2.01-factor approximation for monotone read-k DNFs) *There is a deterministic algorithm which, given as input an M-term width-w read-k DNF, runs in time*

$$\mathrm{poly}(n)\cdot M\cdot\min\left\{(kw)^{\tilde{O}(\log(k\log(kw)))}, (kw)^{O(k^{3/2})}\right\}$$

and outputs a 2.01-factor estimate of $\mathbf{Pr}[F(\boldsymbol{x})]$.

Acknowledgements

We thank Adam Klivans for helpful conversations. We also thank the anonymous SODA reviewers for their helpful comments and suggestions.

References

[ABK+98] Howard Aizenstein, Avrim Blum, Roni Khardon, Eyal Kushilevitz, Leonard Pitt, and Dan Roth. On learning read-k-satisfy-j DNF. *SIAM J. Comput.*, 27(6):1515–1530, 1998. 1

[AP92] Howard Aizenstein and Leonard Pitt. Exact learning of read-k disjoint DNF and not-so-disjoint DNF. In *Proceedings of the 5th Annual ACM Conference on Computational Learning Theory (COLT)*, pages 71–76, 1992. 1

[AW85] Miklós Ajtai and Avi Wigderson. Deterministic simulation of probabilistic constant depth circuits. In *Proceedings of the 26th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 11–19, 1985. 1.3

[Baz03] Louay Bazzi. *Minimum Distance of Error Correcting Codes versus Encoding Complexity, Symmetry, and Pseudorandomness.* PhD thesis, Massachusetts Institute of Technology, 2003. 1, 1.1

[Baz09] Louay Bazzi. Polylogarithmic independence can fool DNF formulas. *SIAM Journal on Computing*, 38(6):2220–2272, 2009. 2.2, 3

[BFJ+94] Avrim Blum, Merrick L. Furst, Jeffrey Jackson, Michael J. Kearns, Yishay Mansour, and Steven Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 253–262, 1994. 1

[BN17] Louay Bazzi and Nagi Nahas. Small-bias is not enough to hit read-once CNF. *Theory Comput. Syst.*, 60(2):324–345, 2017. 1, 1.1

[CFGS86] Fan Chung, Péter Frankl, Ronald Graham, and James Shearer. Some intersection theorems for ordered sets and graphs. *Journal of Combinatorial Theory (A)*, 43:23–37, 1986. 1.4, 3.1

[CRS00] Suresh Chari, Pankaj Rohatgi, and Aravind Srinivasan. Improved algorithms via approximations of probability distributions. *Journal of Computer and System Sciences*, 61(1):81–107, 2000. 1, 1.1

[DETT10] Anindya De, Omid Etesami, Luca Trevisan, and Madhur Tulsiani. Improved pseudorandom generators for depth 2 circuits. In *Proceedings of the 13th International Workshop on Randomization and Computation (RANDOM)*, pages 504–517, 2010. 1, 1.1

[DMP99] Carlos Domingo, Nina Mishra, and Leonard Pitt. Efficient Read-Restricted Monotone CNF/DNF Dualization by Learning with Membership Queries. *Machine Learning*, 37(1):89–110, 1999. 1

[DSFT+15] Dana Dachman-Soled, Vitaly Feldman, Li-Yang Tan, Andrew Wan, and Karl Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of the 26th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015. 1.2

[EGL+98] Guy Even, Oded Goldreich, Michael Luby, Noam Nisan, and Boban Velickovic. Efficient approximation of product distributions. *Random Struct. Algorithms*, 13(1):1–16, 1998. 1, 1.1

[Fel10] Vitaly Feldman. Distribution-specific agnostic boosting. In *Proceedings of the First Symposium on Innovations in Computer Science*, pages 241–250, 2010. 1.2

[FGKP09] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. On agnostic learning of parities, monomials, and halfspaces. *SIAM J. Comput.*, 39(2):606–645, 2009. 1.2

[GKK08a] Parikshit Gopalan, Adam Kalai, and Adam Klivans. Agnostically learning decision trees. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 527–536, 2008. 1.2, 1.4, 2.1, 6

[GKK08b] Parikshit Gopalan, Adam Kalai, and Adam Klivans. A query algorithm for agnostically learning DNF? In *Proceedings of the 21st Annual Conference on Learning Theory (COLT), Open Problem*, pages 515–516, 2008. 1.2

[GMR+12] Parikshit Gopalan, Raghu Meka, Omer Reingold, Luca Trevisan, and Salil P. Vadhan. Better pseudorandom generators from milder pseudorandom restrictions. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 120–129, 2012. 1, 1.1

[GMR13] Parikshit Gopalan, Raghu Meka, and Omer Reingold. DNF sparsification and a faster deterministic counting algorithm. *Comput. Complexity*, 22(2):275–310, 2013. 1.3, 6, 6, 6.1, 6.1, 6.2

[Gop16] Parikshit Gopalan. Pseudorandomness against bounded memory. Tutorial given at the Workshop on Algebraic Complexity Theory, Feburary 3–5, 2016. Video available at https://www.youtube.com/watch?v=a-dORbHQV-Q, 2016. 1.1

[Han93] Thomas Hancock. Learning $k\mu$ decision trees on the uniform distribution. In *Proceedings of the 6th Annual Conference on Computational Learning Theory (COLT)*, pages 352–360, 1993. 1

[HM91] Thomas Hancock and Yishay Mansour. Learning monotone k-μ DNF formulas on product distributions. In *Proceedings of the 4th Annual Conference on Computational Learning Theory (COLT)*, pages 179–193, 1991. 1, 1.2

[HVV06] Alexander Healy, Salil Vadhan, and Emanuele Viola. Using nondeterminism to amplify hardness. *SIAM Journal on Computing*, 35(4):903–931, 2006. 1.1

[KKMS08] Adam Kalai, Adam Klivans, Yishay Mansour, and Rocco Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008. 1.2

[KL83] Richard Karp and Michael Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Proceedings of the 24th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 56–64, 1983. 1.3, 6

[Kli17] Adam Klivans. Personal communication, 2017. 1

[KLW10] Adam Klivans, Homin Lee, and Andrew Wan. Mansour's conjecture is true for random DNF formulas. In *Proceedings of the 23rd Conference on Learning Theory (COLT)*, pages 368–380, 2010. 1, 1.1, 1.1, 1.2, 1, 1.4, 2.1, 2.1, 3, 3, 3, 3.1, 3, 4.1, 4.2, 4.2, 4.3, 8, 11

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform and learnability. *Journal of the ACM*, 40(3):607–620, 1993. 3

[LN90] Nathan Linial and Noam Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10(4):349–365, 1990. 1.3

[Lov15] Shachar Lovett. Information theory in combinatorics. Available at https://simons.berkeley.edu/talks/shachar-lovett-2015-01-13, 2015. 3.1, 3.1

[LV96] Michael Luby and Boban Veličković. On deterministic approximation of DNF. *Algorithmica*, 16(4-5):415–433, 1996. 1.3, 6

[LV17] Chin Ho Lee and Emanuele Viola. Some limitations of the sum of small-bias distributions. *Theory of Computing*, 13(16):1–23, 2017. 1.1

[LVW93] Michael Luby, Boban Veličković, and Avi Wigderson. Deterministic approximate counting of depth-2 circuits. In *Proceedings of the 2nd ISTCS*, pages 18–24, 1993. 1.3

[Man95] Yishay Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. *Journal of Computer and System Sciences*, 50:543–550, 1995. 3

[NN93] Joseph Naor and Moni Naor. Small-bias probability spaces: efficient constructions and applications. *SIAM J. Comput.*, 22(4):838–856, 1993. 2.2, 2.2, 5.1

[NS94] Noam Nisan and Mario Szegedy. On the degree of Boolean functions as real polynomials. *Comput. Complexity*, 4:301–313, 1994. 4.2

[PR95] Krishnan Pillaipakkamnatt and Vijay Raghavan. Read-twice DNF formulas are properly learnable. *Inf. Comput.*, 122(2):236–267, 1995. 1

[Rad03] Jaikumar Radhakrishnan. Entropy and counting. *Computational mathematics, modelling and algorithms*, 146, 2003. 3.1

[Raz09] Alexander Razborov. A simple proof of Bazzi's theorem. *ACM Transactions on Computation Theory*, 1(1):3, 2009. 3

[Riv74] Theodore Rivlin. *The Chebyshev Polynomials*. John Wiley and Sons, 1974. 4.2

[SSSS09] Shai Shalev-Shwartz, Ohad Shamir, and Karthik Sridharan. Agnostically learning halfspaces with margin errors. TTI Technical Report, 2009. 1.2

[Tal17] Avishay Tal. Tight bounds on the Fourier spectrum of AC^0. In *Proceedings of the 32nd Computational Complexity Conference (CCC)*. 2017. 1.1

[Vad12] Salil Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1-3):1–336, 2012. 1.3