Electronic Colloquium on Computational Complexity, Report No. 34 (2022)

Fourier Growth of Regular Branching Programs

Chin Ho Lee*
Harvard University
chlee@seas.harvard.edu

Edward Pyne
Harvard University
epyne@college.harvard.edu

Salil Vadhan†
Harvard University
salil_vadhan@harvard.edu

March 3, 2022

Abstract

We analyze the *Fourier growth*, i.e. the L_1 Fourier weight at level k (denoted $L_{1,k}$), of read-once regular branching programs. We prove that every read-once regular branching program B of width $w \in [1, \infty]$ with s accepting states on n-bit inputs must have its $L_{1,k}$ bounded by

$$\min\Big\{\mathbf{Pr}[B(U_n) = 1](w-1)^k, s \cdot O\big((n \log n)/k\big)^{\frac{k-1}{2}}\Big\}.$$

For any constant k, our result is tight up to constant factors for the AND function on $w-1$ bits, and is tight up to polylogarithmic factors for unbounded width programs. In particular, for $k = 1$ we have $L_{1,1}(B) \leq s$, with no dependence on the width w of the program.

Our result gives new bounds on the coin problem and new pseudorandom generators (PRGs). Furthermore, we obtain an explicit generator for unordered permutation branching programs of unbounded width with a constant factor stretch, where no PRG was previously known.

Applying a composition theorem of Błasiok, Ivanov, Jin, Lee, Servedio and Viola (RANDOM 2021), we extend our results to "generalized group products," a generalization of modular sums and product tests.

Keywords: pseudorandomness, space-bounded computation

*Supported by NSF grant CCF-1763299 and a Simons Investigator grant to S. Vadhan.
†Supported by NSF grant CCF-1763299 and a Simons Investigator grant.

ISSN 1433-8092

1 Introduction

Every Boolean function $f\colon \{-1,1\}^n \to \{0,1\}$ can be identified by its unique multilinear extension

$$f(x) := \sum_{S\subseteq[n]} \widehat{f}(S) \prod_{i\in S} x_i,$$

where the coefficients

$$\widehat{f}(S) := \mathop{\mathbf{E}}_{x\sim\{-1,1\}^n}\left[f(x)\prod_{i\in S} x_i\right]$$

are called the *Fourier coefficients* of f. Over the past few decades, the analysis of these coefficients of Boolean functions has become an indispensable tool in theoretical computer science and mathematics. We refer the readers to the excellent textbook by O'Donnell [O'D14] for a broad introduction.

Given the wide applicability of this tool, researchers have proposed and analyzed different quantitative measures of Fourier coefficients of Boolean functions. In this work we focus on the L_1 Fourier norm at level k:

Definition 1 (L_1 Fourier norm at level k)**.** The L_1 *Fourier norm of a function* $\{-1,1\}^n \to \{0,1\}$ at level k is

$$L_{1,k}(f) := \sum_{S\subseteq[n]:|S|=k} |\widehat{f}(S)|.$$

For a function class $\mathcal{F}$, we use $L_{1,k}(\mathcal{F})$ to denote $\max_{f\in\mathcal{F}} L_{1,k}(f)$.

The notion of *Fourier growth* is a convenient way of capturing the growth of $L_{1,k}$ with respect to levels k.

Definition 2 (Fourier growth)**.** A function class $\mathcal{F} \subseteq \{f\colon \{-1,1\}^n \to \{0,1\}\}$ has *Fourier growth* $L_1(a,b)$ if there exist constants a and b such that $L_{1,k}(\mathcal{F}) \le a \cdot b^k$ for every k.

By the Cauchy–Schwarz inequality, every Boolean function has its $L_{1,k}$ bounded by $\binom{n}{k}^{1/2}$, and thus has Fourier growth $L_1(1,\sqrt{n})$.

Fourier growth was first studied by Mansour to obtain sample-efficient algorithms for learning DNFs [Man95]. It was later formally introduced by Reingold, Steinke and Vadhan in [RSV13], where they constructed explicit unconditional pseudorandom generators for permutation branching programs. Subsequently, this notion has led to many exciting developments in learning theory [IRR+21, EI21] and pseudorandomness [CHRT18, CHHL19, FK18, CHLT19, CGL+21]. In recent years researchers have also discovered new applications to other areas such as separating quantum and classical computation [RT19, Tal20, BS21, SSW21, GRZ21], and proving correlation bounds with the Majority function (and its variants) [CHH+20, CGL+21, Vio21].

Thus given a function class, it has now become a natural question to analyze its Fourier growth. Indeed, in the past decade it has been shown that several well-studied classes of functions exhibit bounded Fourier growth. These include (parity) decision trees [OS07, BTW15, Tal20, SSW21, GTW21], constant-depth circuits [Man95, Tal17], subclasses of low-degree $\mathbb{F}_2$-polynomials [CHHL19, GTW21, CGL+21], low-degree real polynomials [IRR+21, EI21], functions with bounded sensitivity [GSTW16], product tests [Lee19], and read-once branching programs [RSV13, SVW17, CHRT18].

Motivated by derandomization of space-bounded algorithms, in this work we continue the line of research on the Fourier growth of *read-once branching programs*.

Definition 3 (Read-once branching programs)**.** An (unordered) *read-once branching program* B of length n and width w computes a function $B\colon \{-1,1\}^n \to \{0,1\}$. On input $x \in \{-1,1\}^n$, the program B fixes a permutation $\pi\colon [n] \to [n]$ and computes as follows. It starts at a fixed start state $v_1 \in [w]$. Then for $t = 1, \dots, n$, it reads the next input bit $x_{\pi(t)}$ and updates its state according to a transition function $B_t\colon [w] \times \{-1,1\} \to [w]$ by taking $v_{t+1} := B_t(v_t, x_{\pi(t)})$. Note that the transition function B_t can differ at each time step. The program has a fixed set of accept states $V_{\mathsf{acc}} \subseteq [w]$, and $B(x) = \mathbb{1}(v_{n+1} \in V_{\mathsf{acc}})$.

As we will not consider non-read-once branching programs in this work, henceforth we will often omit the word "read-once" and use "branching programs" to refer read-once branching programs for simplicity. Furthermore, as the Fourier growth of a function is unaffected by reordering the input bits, for the purpose of establishing $L_{1,k}$ bounds we can restrict our attention to the case where π is the identity permutation.

A well-studied subclass of branching programs is the class of *regular branching programs*. This model has received a lot of attention in the literature [RTV06, De11, BRRY14, RSV13, BHPP21], in part due to the fact that pseudorandomness against this restricted subclass sometimes suffices for pseudorandomness against general branching programs, and hence the derandomization of space-bounded computation [RTV06, BHPP21].

Definition 4 (Read-once regular branching programs)**.** A *read-once regular branching program* is a read-once branching program where for every time step t and state $v \in [w]$, there are exactly 2 pairs $(u, b) \in [w] \times \{-1,1\}$ such that $B_t(u, b) = v$.

A more restricted class that has also been well-studied is the class of *permutation branching programs*.

Definition 5 (Read-once permutation branching programs)**.** A *read-once permutation branching program* is a read-once regular branching program where for every time step t and state $v \in [w]$, if $B_t(u, b) = B_t(u', b')$ then either $u = u'$ or $b \neq b'$.

A recent line of works constructed explicit pseudorandom objects for regular and permutation branching programs of *unbounded* width with a bounded number of accept states[1] [HPV21, PV21, PV22, BHPP21], a model for which prior to these works even non-explicit constructions were not known to exist. Motivated by these results, we investigate the Fourier growth of these same models.

1.1 Our results

We obtain near-optimal $L_{1,k}$ bounds for regular branching programs of *any* width, improving the bounds in [RSV13] and obtaining the first non-trivial bounds for unbounded width programs.

Theorem 6. *Let $B\colon \{-1,1\}^n \to \{0,1\}$ be a regular branching program of width $w \in [1,\infty]$ with s accept states in its final layer. Then*

$$L_{1,k}(B) \le \min\Big\{ \underbrace{\mathbf{Pr}[B(U_n) = 1] \cdot (w-1)^k}_{1}, \underbrace{s \cdot O\left((n \log n)/k\right)^{\frac{k-1}{2}}}_{2} \Big\}.$$

Note that the two bounds are incomparable: the first bound is independent of the input length n, and the second bound is independent of the width w. The first bound is tight for the AND_{w-1}

[1] Note that unbounded width permutation programs with an unbounded number of accept states can compute arbitrary Boolean functions.

function on $w-1$ bits, which can be computed by a width-w permutation branching program, since

$$L_{1,k}(\mathsf{AND}_{w-1}) = 2^{-(w-1)} \cdot \binom{w-1}{k} = \mathbf{Pr}[\mathsf{AND}_{w-1}(U_{w-1}) = 1] \cdot \binom{w-1}{k}.$$

For $k = 1$, our second bound can be sharpened to $s \cdot \mathbf{Pr}[B(U_n) = 0]$ (see Theorem 21), which is also tight for the AND_2 function on 2 bits, which has $s = 1$, $\mathbf{Pr}[\mathsf{AND}_2(U_2) = 0] = 3/4$, and $L_{1,1}(\mathsf{AND}_2) = 3/4$.

We complement Theorem 6 by a lower bound showing that our second upper bound is in fact tight up to a factor of $\Theta_k(1) \cdot (\log n)^{\frac{k-1}{2}}$ for $k \geq 2$, even for the restricted subclass of permutation branching programs.

Proposition 7. *For all positive integers k, n, and s where $s \leq \sqrt{kn}$, there exists a permutation branching program $B\colon \{-1,1\}^n \to \{0,1\}$ of width $\Theta(\sqrt{kn})$ with s accept states such that $L_{1,k}(B) \geq \frac{s}{\sqrt{kn}} \cdot \Omega(n/k)^{k/2} = \Omega_k(1) \cdot s \cdot n^{\frac{k-1}{2}}$.*

We now make some remarks on Theorem 6. Previously, Reingold, Steinke and Vadhan proved an upper bound of $(2w^2)^k$ [RSV13]. Hence, our first upper bound improves their bound on two fronts. Our first improvement is a quadratic sharpening on the dependence on the width w. Our second improvement is the additional acceptance probability factor in our bounds, which, as we will discuss in the next section, has further implications. $L_{1,k}$ bounds with a dependence on the acceptance probability have proved to be useful, both in extending the bounds to higher levels $k' > k$ [CHRT18] and extending the bounds to other classes of tests [Lee19, BIJ+21]. Indeed, we obtain both our $L_{1,k}$ bounds for $k > 1$ by applying the reduction in [CHRT18] to bounds at a lower level, and this reduction requires obtaining an $L_{1,k}$ bound that scales linearly with respect to the acceptance probability of the function. We note that functions admitting $L_{1,k}$ bounds that scale linearly with acceptance probability include arbitrary Boolean functions [O'D14, Lee19], constant-width read-once branching programs [CHRT18], $\mathbb{F}_2$-polynomials [GTW21, BIJ+21], product tests with outputs $\{-1,1\}$ [Lee19, BIJ+21]. Therefore, Theorem 6 adds the class of regular branching programs to this list.

Our second upper bound gives the first non-trivial $L_{1,k}$ bounds for regular branching programs of unbounded width. Recall that every bounded function has its $L_{1,k}$ bounded by $\sqrt{\binom{n}{k}}$; so this upper bound is interesting only when $s = o(\sqrt{n/(k(\log n)^{k-1})})$.

Proposition 7 follows from the observation that symmetric $\mathbb{F}_2$-polynomials of degree w can be computed by a permutation branching program of width at most $2w$ [BGL06], where $L_{1,k}$ lower bounds on the former class were recently established in [BIJ+21]. For the same reason, Theorem 6 recovers the $L_{1,k}$ bounds for symmetric $\mathbb{F}_2$-polynomials in [BIJ+21, Theorem 8] with a different proof.

1.2 Applications

We describe several consequences of Theorem 6.

Coin problem. Let $X_\delta = (X_1, \ldots, X_n)$ be the distribution over $\{-1,1\}^n$, where the X_i's are independent and each X_i has expectation δ. The *δ-coin problem* studies the maximum advantage for a function class $\mathcal{F}$ to distinguish between the distributions X_δ and $X_0 = U_n$. This basic problem has been studied extensively for various restricted classes of tests, and has a wide range of applications in computational complexity, including circuit complexity [Ajt83, Val84, SV10, LSS+21, GII+19], pseudorandom generators [BV10], quantum computing [Aar10, AD11], streaming algorithms [BGW20], and multiparty computation [CDrI+13]. In particular, there has been a rich line of work on the coin problems for branching programs [BV10, Ste13, LV18, BGW20, BGZ21].

It is known that bounded Fourier growth of $\mathcal{F}$ implies an upper bound the coin problem for $\mathcal{F}$ (see [Lee19, Fact 9]). Thus we obtain the following corollary of Theorem 6.

Corollary 8. *There exists a constant $\alpha > 0$ such that the following holds. Let $B\colon \{-1,1\} \to \{0,1\}$ be a regular branching program of width $w \in [1,\infty]$ with s accept states. For every $\delta \le \alpha \max\{1/w, 1/\sqrt{n \log n}\}$, we have*

$$\left|\mathbf{E}[B(X_\delta)] - \mathbf{E}[B(X_0)]\right| \le \delta s + \delta^2 \cdot O\Big(\min\big\{w^2, s\sqrt{n\log n}\big\}\Big).$$

Moreover, Avishay Tal showed (see [Agr20, Lemma 9]) that if a class $\mathcal{F}$ is closed under restrictions, then $L_{1,1}$ bounds on $\mathcal{F}$ already implies bounds on the coin problem for $\mathcal{F}$. Since the class of permutation branching programs is closed under restrictions, we obtain the following stronger coin problem bounds for that class:

Corollary 9. *Let $B\colon \{-1,1\} \to \{0,1\}$ be a permutation branching program with s accept states. Then $|\mathbf{E}[B(X_\delta)] - \mathbf{E}[B(X_0)]| \le \frac{\delta}{1-\delta} \cdot s$.*

Claim 10. *For every $\delta > 0$ and positive integer $s \le 32/\delta$, there exists a permutation branching program B of length $32/\delta^2$ and width $128/\delta$ with s accept states such that $\mathbf{E}[B(X_\delta)] - \mathbf{E}[B(X_0)] \ge \frac{s\delta}{1000}$.*

Corollaries 8 and 9 can be interpreted as follows. Regular (and permutation) programs with a single accept state cannot distinguish (sufficiently small) biased coins from uniform much better than simply outputting their first input bit.

Previously Braverman, Rao, Raz, and Yehudayoff [BRRY14] obtained a coin problem bound of $\delta \cdot s \cdot (w-1)$ for width-w regular branching programs with s accept states. Corollaries 8 and 9 improve this to roughly $\delta \cdot s$ when δ is very small (Corollary 8) or when we restrict to permutation branching programs (Corollary 9). Claim 10 shows that the upper bound in Corollary 9 is tight up to constant factors.

Pseudorandom generators. Theorem 6 also implies new pseudorandom generators for permutation branching programs.

Definition 11 (Pseudorandom generators)**.** A function $G\colon \{0,1\}^s \to \{-1,1\}^n$ is a *pseudorandom generator (PRG)* for a function class $\mathcal{F}$ with seed length s and error ϵ, if for every $f \in \mathcal{F}$,

$$\left|\mathbf{E}[f(U_n)] - \mathbf{E}[f(G(U_s))]\right| \le \epsilon.$$

G is *explicit* if it can be computed in polynomial time.

Recall that we consider unordered branching programs, where a program can read its inputs in arbitrary order before its execution. Starting from the work of Bogdanov, Papakonstantinou, and Wan [BPW11], there has been extensive research on constructing pseudorandom generators for unordered branching programs [BPW11, IMZ19, RSV13, SVW17, HLV18, LV20, CHRT18, MRT19, FK18, DHH19, Lee19, DHH20, DMR+21], in search for new ideas for improving Nisan's PRG for ordered branching programs [Nis92], which remains the best PRG for derandomizing space-bounded computation to date. This line of research recently led to the first improvement over Nisan's PRG for the special case of width-3 (ordered) branching programs [MRT19].

Applying our $L_{1,k}$ bounds to the "polarizing random walk" framework of [CHHL19, CHLT19, CGL+21], we obtain the following pseudorandom generator.

Corollary 12. *There is an explicit pseudorandom generator for width-w permutation branching programs with seed length $w^2 \cdot O(\log(n/\epsilon))(\log(1/\epsilon) + \log\log n)$ and error ϵ.*

Corollary 12 gives a slight improvement on the PRG given by [CHHL19], reducing the dependence on width from w^4 to w^2, stemming directly from the $L_{1,k}(B) \le (w-1)^k$ bound in Theorem 6, which improves the $L_{1,k}(B) \le (2w^2)^k$ bound of [RSV13]. (Corollary 12 is for permutation branching programs rather than regular branching programs, because the polarizing random walk framework requires that the class is closed under restriction). By the reduction of [BHPP21], this also implies a *hitting set generator* (HSG) for permutation branching programs of *unbounded* width with seed length $O(1/\epsilon^2) \cdot \log(n/\epsilon)(\log(1/\epsilon) + \log\log n)$, quadratically improving the dependence on ϵ. (An ϵ-HSG for a class $\mathcal{F}$ is a function $G\colon \{0,1\}^s \to \{-1,1\}^n$ where for all $f \in \mathcal{F}$ with $\mathbf{Pr}[f(U_n) = 1] > \epsilon$ there is an $x \in \{0,1\}^s$ such that $f(G(x)) = 1$.)

From Corollary 9, we also obtain the first nontrivial pseudorandom generator that fools unordered permutation branching programs of unbounded width with constant factor stretch and constant error.[2] For simplicity we state our result for constant error, and do not optimize constants.

Let $H(x) := x\log(\frac{1}{x}) + (1-x)\log(\frac{1}{1-x})$ denote the binary entropy function.

Theorem 13. *Given any constant $\delta \in (0, 1/2)$ independent of n, there is an explicit PRG for unordered permutation branching programs with a single accept state with seed length $H(1/2 + 0.499\delta) \cdot n + o(n)$ and error $\frac{\delta}{1-\delta} + \frac{\delta}{100}$.*

This is proven by noting that with the specified seed length, we can approximately sample n independent δ-biased coins, which are pseudorandom by Corollary 9. We are not aware of any PRGs prior to our result.

As mentioned above, there exist explicit hitting-set generators (HSGs) with better seed length for this class [BHPP21]. For the easier case of *ordered* permutation programs, Hoza, Pyne, and Vadhan [HPV21] constructed an explicit PRG with significantly better seed length, namely $\widetilde{O}(\log n \cdot \log(1/\epsilon))$.

We note that our results do not give any PRGs for regular programs, because all of the methods for obtaining PRGs from Fourier growth bounds require the class to be closed under restrictions. In particular, even in the ordered setting, it remains unknown whether a nontrivial PRG for unbounded width regular programs exists.

Generalized group products. As mentioned in the previous section, $L_{1,k}$ bounds with the acceptance probability factor (as in Theorem 6) are useful for obtaining $L_{1,k}$ bounds for wider function classes. To make this precise, we recall the definition of *disjoint composition* of two function classes.

Definition 14 (Disjoint composition)**.** Let $\mathcal{F}$ be a class of functions from $\{-1,1\}^m$ to $\{-1,1\}$ and let $\mathcal{G}$ be a class of functions from $\{-1,1\}^\ell$ to $\{-1,1\}$. Define the class $\mathcal{F} \circ \mathcal{G}$ of *disjoint composition of $\mathcal{F}$ and $\mathcal{G}$* to be the class of all functions from $\{-1,1\}^{m\ell}$ to $\{-1,1\}$ of the form

$$h(x^1, \dots, x^m) = f(g_1(x^1), \dots, g_m(x^m)),$$

where $g_1, \dots, g_m \in \mathcal{G}$ are defined on m disjoint sets of variables and $f \in \mathcal{F}$.

Błasiok, Ivanov, Jin, Lee, Servedio and Viola [BIJ+21] showed that if both $\mathcal{F}$ and $\mathcal{G}$ are closed under negation of their outputs, and $\mathcal{F}$ is closed under restrictions, then $L_{1,k}$ bounds with the

[2] The co-HSG of [BHPP21] can be interpreted as an explicit PRG for permutation programs with error $1 - 1/(n+1)$.

acceptance probability factor for $\mathcal{F}$ and $\mathcal{G}$ imply $L_{1,k}$ bounds on the disjoint composition of $\mathcal{F}$ and $\mathcal{G}$. Specifically, if for every $1 \le k \le K$, we have $L_{1,k}(f) \le \mathbf{Pr}[f(U_m) = 1] \cdot b_{\mathsf{outer}}^k$ for every $f \in \mathcal{F}$ and $L_{1,k}(g) \le \mathbf{Pr}[g(U_\ell) = 1] \cdot b_{\mathsf{inner}}^k$ for every $g \in \mathcal{G}$, then for every function $h \in \mathcal{F} \circ \mathcal{G}$, we have that

$$L_{1,K}(h) \le \mathbf{Pr}[h(U_{m\ell}) = 1] \cdot (b_{\mathsf{inner}} b_{\mathsf{outer}})^K.$$

Therefore, we also obtain new $L_{1,k}$ bounds for the disjoint composition of permutation branching programs and other classes of functions that admit the acceptance probability factor in their $L_{1,k}$ bounds (see Section 1 for a list). As a concrete example of such composition, we introduce the class of *generalized group products.*

Definition 15 (Generalized group products)**.** A function $f\colon \{-1,1\}^n \to \{0,1\}$ is a (m, ℓ, G)*-group product* if there exist m disjoint subsets $I_1, \dots, I_m \subseteq [n]$ of size at most ℓ such that

$$f(x) = \mathbb{1}\left(\prod_{i=1}^m g_i^{f_i(x_{I_i})} \subseteq S\right),$$

for some subset $S \subseteq G$, group elements $g_i \in G$, and functions $f_i\colon \{-1,1\}^{I_i} \to \{0,1\}$. Here x_{I_i} are the $|I_i|$ bits of x indexed by I_i.

Note that generalized group products are unordered by definition. They are a generalization of several function classes that have received some attention in the past, including *modular sums* [LRTV09, MZ09, GKM18] (when G is the cyclic group and $\ell = 1$), product tests with outputs {-1,1} [HLV18, LV18, LV20, Lee19] (when $G = \{-1,1\}$), and unordered *combinatorial shapes* [GMRZ13, GKM18] (when $G = \mathbb{Z}_{m+1}$).

An (m, ℓ, G)-group product can be written as the disjoint composition of a width-$|G|$ permutation branching program and arbitrary Boolean functions on ℓ bits. Since both of these classes admit $L_{1,k}$ bounds with the acceptance probability factor, using the composition theorem of [BIJ+21] we obtain Fourier growth bounds for generalized group products.

Corollary 16. *Let $f\colon \{-1,1\}^n \to \{0,1\}$ be an (m,ℓ,G)-group product. Then $L_{1,k}(f) \le \mathbf{Pr}[f(U_n) = 1] \cdot O(\ell \cdot |G|)^k$.*

Corollary 16 extends the Fourier growth bounds for product tests studied in [Lee19] (where $G = \{-1,1\}$). Plugging our bounds into the polarizing random walk framework, we also obtain new pseudorandom generators for generalized group products.

Corollary 17. *There is an explicit pseudorandom generator for (m,ℓ,G)-group products with seed length $O(\ell \cdot |G|)^2 \cdot \log(n/\epsilon) \cdot (\log(1/\epsilon) + \log\log n)$ and error ϵ.*

Note that an $(m, 1, G)$-group product can be computed by a permutation branching program of width $|G|$, and a (m, ℓ, G)-group product can be computed by a general branching program of width $w = 2^\ell \cdot |G|$. When $\ell \ge 2$, we are not aware of any PRG that fools (m, ℓ, G)-group products better than unordered general branching programs. For the latter class, the current best PRGs are given by Forbes and Kelley [FK18] which, with the above choice of w, have seed lengths $O(\ell + \log(|G|) + \log(n/\epsilon)) \log^2 n$ and $\widetilde{O}(2^\ell + |G|) \log(n/\epsilon) \log n$. For comparison, for any error $\epsilon = O(1)$, our PRG for generalized group products has seed length $(\ell \cdot |G|)^2 \cdot \widetilde{O}(\log n)$, where is nearly optimal when $\ell \cdot |G| = O(1)$, whereas the Forbes–Kelley PRGs have seed lengths $\Omega(\log^2 n)$.

Finally, we note that when $G = \{-1,1\}$, there exists a PRG [Lee19, DHH20] with seed length $\widetilde{O}(\ell + \log(m/\epsilon)) + \mathrm{poly}(\log\log(n/\epsilon))$, which is nearly optimal.

1.3 Techniques

Our main contribution is a simple inductive proof for bounding the first level $L_{1,1}$ of a regular branching program in terms of the number of its accept states and *rejection* probability. Specifically, for a regular branching program B of s accept states, we prove that

$$L_{1,1}(B) \leq s \cdot \mathbf{Pr}[B(U_n) = 0]. \tag{1}$$

We prove Equation (1) by induction on n, the length of the program. We give some intuition for where Equation (1) came from. Let S be the set of accept states in the final layer. By regularity, the set of states T in the previous layer that lead to S must be at least the size of S. If they have the same size then the current layer is redundant. So we must have a nonempty set T_1 of vertices that have only one outgoing edge leading to S. Since these vertices also have one edge leading to the complement of S, they all contribute to the probability that the program rejects. This suggests bounding $L_{1,1}$ in terms of $|S|$ and the rejection probability. In the proof we use regularity of the program to relate $|S|$ to $|T|$ and $|T_1|$.

Our first $L_{1,k}$ bound $L_{1,k}(B) \leq \mathbf{Pr}[B(U_n) = 1] \cdot (w-1)^k$ then follows from the same inductive argument in [CHRT18], where the authors proved $L_{1,k}$ bounds for general constant-width branching programs. We note that this inductive argument relies on bounding the $L_{1,1}$ of the *local monotonization* of a branching program [BV10], which does not preserve the permutation property. Therefore, even for proving Fourier growth bounds of permutation programs, to apply this argument it is crucial to establish Equation (1) for the wider class of regular programs. Proving our second bound $L_{1,k}(B) \leq s \cdot O\left((n \log n)/k\right)^{\frac{k-1}{2}}$ is slightly more involved. Our proof combines the inductive idea in [CHRT18] with the "level-k inequalities" of Lee [Lee19] (Lemma 22), which give $L_{1,k}$ bounds for an arbitrary Boolean function in terms of its acceptance probability, and the approximator from Bogdanov, Hoza, Prakriya, and Pyne [BHPP21] (Lemma 23).

Given a regular branching program B of unbounded width, as in [BHPP21] we first construct a regular program B' that approximates B by rejecting all the states in B that can be reached with probability at most $q := \widetilde{O}(1/\sqrt{n})$. In [BHPP21], they observed that the probability that the program B accepts via *any* of these "sudden reject" states is at most q. So the error function $B - B'$ has small acceptance probability, and by the level-k inequalities it has small $L_{1,k}$. So it suffices to bound the $L_{1,k}$ of the approximator B'. We use the fact that B' has at most $1/q$ non-sudden-reject states in each layer, and so the total number of non-reject states in B' is bounded by $n/q = \mathrm{poly}(n)$. This allows us to apply an inductive argument to reduce bounding $L_{1,k}(B')$ to bounding (roughly) the product of $L_{1,k-1}(B')$ and $L_{1,1}(B')$. For $L_{1,k-1}(B')$ we again use the level-k inequalities, and for $L_{1,1}(B)$ we use the bound in (1). Note that while the states in B' all have reaching probability at least q in the original program B, some of them may have reaching probability much smaller than q in the approximator B'. To deal with this, we take a similar approach in [CHRT18] to handle states with small reaching probabilities separately.

Organization. We begin by introducing some notation in the next paragraph. In Section 2, we prove our $L_{1,k}$ bounds of permutation branching programs (Theorem 6 and Proposition 7) and generalized group products (Corollary 16). In Section 3, we prove our coin problem bounds (Corollaries 8 and 9 and Claim 10), and construct our pseudorandom generators for permutation programs (Corollary 12 and Theorem 13) and generalized group products (Corollary 17).

Notation. For a branching program B of length n and width w, we will view it as a directed layer graph with $n + 1$ layers of vertices denoted by $V_1, \ldots, V_{n+1}$, each consists of w vertices. For every two consecutive layers V_t and V_{t+1}, every vertex $u \in V_t$ has two outgoing edges labeled by $b \in \{-1, 1\}$, where the b-edge goes to the vertex $B_t(u, b)$ in V_{t+1}. We will overload notation and consider the transition function as a map $B_t\colon V_t \times \{-1, 1\} \to V_{t+1}$ in addition to thinking of it as a map $B_t\colon [w] \times \{-1, 1\} \to [w]$. Similarly, we will often think of the start state v_1 as being an element of V_1 instead of an element of $[w]$, and $V_{\mathsf{acc}} \subseteq V_{n+1}$ instead of $V_{\mathsf{acc}} \subseteq [w]$, etc.

For a vertex v in some layer V_t, we use $B_{\to v}$ to denote the sub-branching program of length $t-1$ but with v being the only accept vertex. We also use $B_{v\to}$ to denote the sub-branching program of length $n+1-t$ that starts at v and ends in V_n with accept vertices V_{acc}.

For ease of notation we use $\mu(f)$ to denote the expectation of f under uniform inputs.

2 $L_{1,k}$ bounds of regular branching programs

In this section we prove our $L_{1,k}$ bounds for regular branching programs (Theorem 6) and generalized group products (Corollary 16). We start with bounding the first level $L_{1,1}$ of regular branching programs.

Lemma 18. *Let $B\colon \{-1,1\}^n \to \{0,1\}$ be a regular branching program of width $w \in [1, \infty]$ with s accept states. Then*

$$L_{1,1}(B) \le \min\{s \cdot \mathbf{Pr}[B(U_n) = 0], \mathbf{Pr}[B(U_n) = 1] \cdot (w-1)\}.$$

Proof. We prove the first bound by induction on n. For $n = 0$ the bound is vacuous. Now assume it holds for $n-1$ and consider a regular program $B(x_1, \dots, x_n)$ with a set S of s accept states. Define the following 3 subsets of states in layer $n-1$, where T is the set of states with both of its edges leading to S, T_+ is the set of states with only 1-edges leading to S, and likewise for T_- and (-1)-edges. Observe that we can write B as

$$B(x_1, \dots, x_n) = g(x_1, \dots, x_{n-1}) + \frac{1+x_n}{2} g_+(x_1, \dots, x_{n-1}) + \frac{1-x_n}{2} g_-(x_1, \dots, x_{n-1}),$$

where g, g_+, g_- are functions computable by regular branching programs of length $n-1$ with T, T_+ and T_- as the sets of accept vertices, respectively. Note that $s = |T| + \frac{|T_-|+|T_+|}{2}$. Define $g_1 := g_- + g_+$ and $T_1 := T_+ \cup T_-$. Now observe that

$$\begin{aligned}
\left|\widehat{B}(\{i\})\right| &= \left|\widehat{g}(\{i\}) + \frac{1}{2}\left(\widehat{g_+}(\{i\}) + \widehat{g_-}(\{i\})\right)\right| \le \frac{1}{2}\left|\widehat{g}(\{i\})\right| + \frac{1}{2}\left|\widehat{g}(\{i\}) + \widehat{g_1}(\{i\})\right| \quad \text{for } i \in [n-1] \\
\left|\widehat{B}(\{n\})\right| &= \frac{1}{2}\left|\mu(g_+) - \mu(g_-)\right| \le \frac{1}{2}\left(\mu(g_+) + \mu(g_-)\right) \le \frac{1}{2}\mu(g_1) \\
\mu(B) &= \mu(g) + \frac{\mu(g_1)}{2}.
\end{aligned}$$

Finally, as T and T_1 are disjoint, the function $g + g_1$ is Boolean and is computable by a regular program of length $n-1$ with $|T| + |T_1|$ accept states, and $\mu(g + g_1) = \mu(g) + \mu(g_1)$. So applying our induction assumption on g and $g+g_1$, we have

$$\begin{aligned}
2L_{1,1}(B) = 2\sum_{i=1}^{n}\left|\widehat{B}(\{i\})\right| &\le \sum_{i=1}^{n-1}\left|\widehat{g}(\{i\})\right| + \sum_{i=1}^{n-1}\left|\widehat{(g+g_1)}(\{i\})\right| + \mu(g_1) \\
&= L_{1,1}(g) + L_{1,1}(g+g_1) + \mu(g_1) \\
&\le |T|\cdot\big(1-\mu(g)\big) + (|T|+|T_1|)\cdot\big(1-\mu(g+g_1)\big) + \mu(g_1) \\
&= \big(2|T|+|T_1|\big) - \Big(|T|\cdot\mu(g) + \big(|T|+|T_1|\big)\cdot\mu(g+g_1) - \mu(g_1)\Big) \\
&= 2s - \Big(\big(2|T|+|T_1|\big)\cdot\mu(g) + \big(|T|+|T_1|-1\big)\cdot\mu(g_1)\Big) \\
&\le 2s - \left(\big(2|T|+|T_1|\big)\cdot\mu(g) + \left(|T|+\tfrac{|T_1|}{2}\right)\cdot\mu(g_1)\right) \\
&= 2s - 2s\left(\mu(g) + \tfrac{\mu(g_1)}{2}\right) \\
&= 2s\cdot\big(1-\mu(B)\big),
\end{aligned}$$

where the last inequality uses that $\frac{|T_1|}{2}\cdot\mu(g_1)\ge\mu(g_1)$, since T_1 is either empty or has size at least 2.

To prove the second bound, suppose B is a regular program of width $w<\infty$ with a set S of accept states. For every state $v\in S$, the function $1-B_{\to v}$ is computable by a regular branching program with $w-1$ accept states. Since $L_{1,1}(B_{\to v}) = L_{1,1}(1-B_{\to v})$, it follows from the first bound we just proved that $L_{1,1}(B_{\to v}) \le \mathbf{Pr}[B_{\to v}(U_n)=1]\cdot(w-1)$. Summing over all the accept states $v\in S$ gives the second bound. $\square$

To obtain $L_{1,k}$ bounds at higher levels, we will apply the inductive argument in [RSV13, CHRT18]. We first recall the local monotonization of a branching program introduced in [BV10, CHRT18]. For a branching program B, we define the *local monotonization* B' of B by the following process. For every layer t, state $u\in V_t$, and input $b\in\{-1,1\}$, let $v_b := B_t(u,b)$ and define

$$B'_t(u,b) = \begin{cases} B_t(u,-b) & \text{if } \mu(B_{v_1\to}) < \mu(B_{v_{-1}\to}) \\ B_t(u,b) & \text{otherwise.}\end{cases}$$

In words, we swap the two outgoing edge-labels of u whenever $\mu(B_{v_1\to}) < \mu(B_{v_{-1}\to})$. As the underlying graph of B' remains the same as B, if B is regular then B' is also regular (with the same set of accept states). Also $\mu(B_{v\to}) = \mu(B'_{v\to})$ for every state v. By construction we have $|\widehat{B}(\{i\})| = \widehat{B'}(\{i\})$ for every $i\in[n]$.

The following claim reduces bounding $L_{1,k}$ of a branching program to bounding its $L_{1,k-1}$.

Claim 19 ([CHRT18])**.** *Let $B\colon\{-1,1\}^n\to\{0,1\}$ be a branching program, and B' be its local monotonization. Then $L_{1,k+1}(B)\le\sum_{i=1}^{n}\sum_{v\in V_i}\big(L_{1,k}(B_{\to v})\cdot\widehat{B'_{v\to}}(\{i\})\big)$.*

Proof. We have

$$\begin{aligned}
L_{1,k+1}(B) &= \sum_{i=1}^{n} \sum_{\substack{S\subseteq\{1,\ldots,i-1\}:\\ |S|=k}} \left|\widehat{B}(S\cup\{i\})\right| \\
&= \sum_{i=1}^{n} \sum_{\substack{S\subseteq\{1,\ldots,i-1\}:\\ |S|=k}} \left|\sum_{v\in V_i} \widehat{B_{\to v}}(S)\widehat{B_{v\to}}(\{i\})\right| \\
&\le \sum_{i=1}^{n} \sum_{\substack{S\subseteq\{1,\ldots,i-1\}:\\ |S|=k}} \sum_{v\in V_i} \left(\left|\widehat{B_{\to v}}(S)\right| \cdot \left|\widehat{B_{v\to}}(\{i\})\right|\right) \\
&= \sum_{i=1}^{n} \left(\sum_{v\in V_i} \sum_{\substack{S\subseteq\{i,\ldots,i-1\}:\\ |S|=k}} \left|\widehat{B_{\to v}}(S)\right|\right) \left|\widehat{B_{v\to}}(\{i\})\right| \\
&= \sum_{i=1}^{n} \sum_{v\in V_i} \left(L_{1,k}(B_{\to v}) \cdot \widehat{B'_{v\to}}(\{i\})\right).
\end{aligned}$$

□

Theorem 20. *Let $B\colon \{-1,1\}^n \to \{0,1\}$ be a regular branching program of width w. Then*

$$L_{1,k}(B) \le \mathbf{Pr}[B(U_n)=1]\cdot (w-1)^k.$$

Proof. Let B' be the local monotonization of B. By Claim 19,

$$\begin{aligned}
L_{1,k+1}(B) &\le \sum_{i=1}^{n} \sum_{v\in V_i} \left(L_{1,k-1}(B_{\to v}) \cdot \widehat{B'_{v\to}}(\{i\})\right) \\
&\le (w-1)^k \sum_{i=1}^{n} \sum_{v\in V_i} \mathbf{Pr}[B_{\to v}(U_i)=1]\cdot \widehat{B'_{v\to}}(\{i\}) \\
&= (w-1)^k \sum_{i=1}^{n} \sum_{v\in V_i} \mathbf{Pr}[B'_{\to v}(U_i)=1]\cdot \widehat{B'_{v\to}}(\{i\}) \\
&= (w-1)^k \sum_{i=1}^{n} \widehat{B'}(\{i\}) \\
&\le \mathbf{Pr}[B(U_n)=1]\cdot (w-1)^{k+1}.
\end{aligned}$$

□

Theorem 21. *Let $B\colon \{-1,1\}^n \to \{0,1\}$ be any regular branching program with s accepting states. Then*

$$L_{1,k}(B) \le s\,\mathbf{Pr}[B(U_n)=0]\cdot O\left(\frac{n}{k}\left(1+\frac{1}{k}\log\Big(\frac{n}{\mathbf{Pr}[B(U_n)=0]}\Big)\right)\right)^{\frac{k-1}{2}}.$$

We will use the following "L_1 level-k inequalities," which follows from applying Cauchy–Schwarz to Lemma 10 in [Lee19], and the observation that every non-constant Boolean function f has $\mu(f)\ge 2^{-n}$.

Lemma 22. *For every Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, we have*

$$L_{1,k}(f) \le \sqrt{\binom{n}{k}}\cdot \mu(f)\cdot O\left(\log\left(\frac{2}{\mu(f)^{1/k}}\right)\right)^{k/2} \le \mathbf{Pr}[f(U_n)=1]\cdot O(n)^{k/2}.$$

We also need the following lemma in [BHPP21], which follows from applying a union bound over all the s accept vertices to Claim 3.1 in [BHPP21].

Lemma 23 (Claim 3.1 in [BHPP21])**.** *Let $B\colon \{-1,1\}^n \to \{0,1\}$ be a regular branching program with s accept vertices. Let $V^\epsilon := \{v : \mu(B_{\to v}) \le \epsilon\}$ be the set of states in B that have at most ϵ probability of being reached over uniform inputs. Then for every state v,*

$$\Pr_{x\sim U_n}\left[B_{\to v}(x) = 1 \wedge B_{\to u}(x_1,\dots,x_t) = 1 \text{ for some } t \in [n] \text{ and } u \in V^\epsilon\right] \le s\cdot\epsilon.$$

Proof of Theorem 21. Let $\bar{\mu} := 1 - \mu(B)$, and define

$$q := \frac{\bar{\mu}}{s}\left(\frac{k}{n\log(ns/\bar{\mu})}\right)^{1/2}.$$

Let V^q be the set of states v in B with $\mu(B_{\to v}) \le q$. As in [BHPP21], we construct another regular program B' that approximates B as follows. For each state $u \in V^q$, we "sudden reject" u by rewiring its outgoing edges to an "unused" state. Specifically, we construct B' by modifying B as follows. We iterate each $u \in V^q$ and do the following: Suppose $u \in V_t \cap V^q$ for some layer t. Let $u' \in V_t$ be a state with $\mu(B_{\to u'}) = \mu(B_{u'\to}) = 0$. We swap the outgoing b-edges of u and u' for both $b \in \{-1,1\}$. Observe that for every state u in B', we have $\mu(B_{\to u}) \ge q$ and so in each layer of B' there are at most $1/q$ many non-sudden-reject states with $\mu(B'_{\to u}) > 0$.

We now bound above $L_{1,k+1}(B)$ by $L_{1,k+1}(B-B') + L_{1,k+1}(B')$. By Lemma 23,

$$\begin{aligned}&\Pr\big[(B-B')(U_n) = 1\big]\\ &\qquad = \Pr_{x\sim U_n}\left[B(x) = 1 \wedge B_{\to u}(x_1,\dots,x_t) = 1 \text{ for some } t \text{ and } u \in V^q\right] \le s\cdot q.\end{aligned}$$

As $B - B'$ has small acceptance probability, it follows from Lemma 22 that

$$\begin{aligned}L_{1,k+1}(B-B') &\le O(1)^k\sqrt{\binom{n}{k+1}}\cdot s\cdot q\cdot\left(\log\frac{2}{q^{1/(k+1)}}\right)^{\frac{k+1}{2}}\\ &\le O(1)^k\cdot\left(\frac{n}{k}\right)^{\frac{k+1}{2}}\cdot\bar{\mu}\cdot\left(\frac{k}{n\log(ns/\bar{\mu})}\right)^{1/2}\left(1+\frac{\log(ns/\bar{\mu})}{k}\right)^{\frac{k+1}{2}}\\ &\le O(1)^k\cdot\bar{\mu}\cdot\left(\frac{n}{k}\left(1+\frac{\log(ns/\bar{\mu})}{k}\right)\right)^{k/2}\left(\frac{n}{k}\cdot\frac{k}{n\log(ns/\bar{\mu})}\cdot\left(1+\frac{\log(ns/\bar{\mu})}{k}\right)\right)^{1/2}\\ &\le O(1)^k\cdot\bar{\mu}\cdot\left(\frac{n}{k}\left(1+\frac{\log(ns/\bar{\mu})}{k}\right)\right)^{k/2},\end{aligned}$$

It remains to bound $L_{1,k+1}(B')$. Let B'' be the local monotonization of B'. By Claim 19 and Lemma 22,

$$\begin{aligned}L_{1,k+1}(B') &\le \sum_{i=1}^{n}\sum_{v\in V_i'}\left(L_{1,k}(B'_{\to v})\cdot\widehat{B''_{v\to}}(\{i\})\right)\\ &\le O(1)^k\cdot\left(\frac{n}{k}\right)^{k/2}\cdot\sum_{i=1}^{n}\sum_{v\in V_i'}\left(\mu(B'_{\to v})\cdot\log\left(\frac{2}{\mu(B'_{\to v})^{1/k}}\right)^{k/2}\cdot\widehat{B''_{v\to}}(\{i\})\right).\end{aligned}$$

We separate the double sum above into two parts, depending on whether the states v can be reached with probability at least $q\bar{\mu}/n$.

We first consider those with reaching probability less than $q\bar{\mu}/n$, As the function $x \mapsto x\log(2/x^{1/k})^{k/2}$ is increasing for $x \in [0,1]$, we have

$$
\begin{aligned}
&\sum_{i=1}^{n} \sum_{\substack{v \in V_i': \\ \mu(B_{\to v}) < \frac{q\bar{\mu}}{n}}} \left(\mu(B_{\to v}) \cdot \log\left(\frac{2}{\mu(B_{\to v})^{1/k}}\right)^{k/2} \cdot \widehat{B''_{v\to}}(\{i\}) \right) \\
&\le O(1)^k \cdot \left(1 + \frac{\log(ns/\bar{\mu})}{k}\right)^{k/2} \cdot \frac{q\bar{\mu}}{n} \cdot \sum_{i=1}^{n} \sum_{\substack{v \in V_i' \\ 0 < \mu(B'_{\to v}) < \frac{q\bar{\mu}}{n}}} \widehat{B''_{v\to}}(\{i\}) \\
&\le O(1)^k \cdot \left(1 + \frac{\log(ns/\bar{\mu})}{k}\right)^{k/2} \bar{\mu},
\end{aligned}
$$

where the last inequality is because $|\widehat{B''}_{v\to}(\{i\})| \le 1$ and we are summing over at most $n \cdot 1/q$ many vertices.

For those states that are reached with probability at least $q\bar{\mu}/n$, we apply Lemma 22 and our $L_{1,1}$ bound in Lemma 18. We have

$$
\begin{aligned}
&\sum_{i=1}^{n} \sum_{\substack{v \in V_i': \\ \mu(B'_{\to v}) \ge \frac{q\bar{\mu}}{n}}} \left(\mu(B'_{\to v}) \cdot \log\left(\frac{2}{\mu(B'_{\to v})^{1/k}}\right)^{k/2} \cdot \widehat{B''_{v\to}}(\{i\}) \right) \\
&\le O\left(1 + \frac{\log(n/\bar{\mu})}{k}\right)^{k/2} \sum_{i=1}^{n} \sum_{\substack{v \in V_i': \\ \mu(B'_{\to v}) \ge \frac{q\bar{\mu}}{n}}} \left(\mu(B'_{\to v}) \cdot \widehat{B''_{v\to}}(\{i\}) \right),
\end{aligned}
$$

where by Lemma 18 we get

$$
\begin{aligned}
&\sum_{i=1}^{n} \sum_{\substack{v \in V_i': \\ \mu(B'_{\to v}) \ge \frac{q\bar{\mu}}{n}}} \left(\mu(B'_{\to v}) \cdot \widehat{B''_{v\to}}(\{i\}) \right) \\
&= \sum_{i=1}^{n} \sum_{\substack{v \in V_i': \\ \mu(B''_{\to v}) \ge \frac{q\bar{\mu}}{n}}} \left(\mu(B''_{\to v}) \cdot \widehat{B''_{v\to}}(\{i\}) \right) && (\mu(B''_{\to v}) = \mu(B'_{\to v})) \\
&\le \sum_{i=1}^{n} \sum_{v \in V_i'} \left(\mu(B''_{\to v}) \cdot \widehat{B''_{v\to}}(\{i\}) \right) && (\widehat{B''_{v\to}}(\{i\}) \ge 0) \\
&\le \sum_{i=1}^{n} \widehat{B''}(\{i\}) \le s \cdot \mathbf{Pr}[B''(U_n) = 0] \le 2s\bar{\mu},
\end{aligned}
$$

where we use $\mathbf{Pr}[B''(U_n) = 0] = \mathbf{Pr}[B'(U_n) = 0] \le \mathbf{Pr}[B(U_n) = 0] + sq \le 2\bar{\mu}$ in the last inequality. Hence,

$$
\begin{aligned}
L_{1,k+1}(B') &\le O(1)^k \cdot \left(\frac{n}{k}\right)^{k/2} \cdot \sum_{i=1}^{n} \sum_{v \in V_i'} \left(\mu(B'_{\to v}) \cdot \log\left(\frac{2}{\mu(B'_{\to v})^{1/k}}\right)^{k/2} \cdot \widehat{B''_{v\to}}(\{i\}) \right) \\
&\le O\left(\frac{n}{k}\left(1 + \frac{\log(ns/\bar{\mu})}{k}\right)\right)^{k/2} s\bar{\mu}.
\end{aligned}
$$

Therefore, we have

$$\begin{aligned}
L_{1,k+1}(B) &\leq L_{1,k+1}(B-B') + L_{1,k+1}(B') \\
&\leq O(1)^k \cdot s\bar{\mu} \cdot \left(\frac{n}{k}\left(1+\frac{\log(ns/\bar{\mu})}{k}\right)\right)^{k/2} \\
&\leq O(1)^k \cdot s\bar{\mu} \cdot \left(\frac{n}{k}\left(1+\frac{\log(n/\bar{\mu})}{k}\right)\right)^{k/2},
\end{aligned}$$

where the last inequality is because if $s \geq \sqrt{n}$, then the conclusion directly follows from Lemma 22; so we can assume $s \leq \sqrt{n}$. □

Theorem 6 now follows from Theorems 20 and 21.

Proof of Theorem 6. The first bound $L_{1,k}(B) \leq \mathbf{Pr}[B(U_n) = 1] \cdot (w-1)^k$ directly follows from Theorem 20. We now show that Theorem 21 implies the second bound $L_{1,k}(B) \leq s \cdot O((n \log n)/k)^{\frac{k-1}{2}}$. Let $\bar{\mu} := \mathbf{Pr}[B(U_n) = 0]$. As the function $x \mapsto x \log(2/x^{1/k})^{\frac{k-1}{2}}$ is increasing for $x \in [0,1]$, we have

$$\begin{aligned}
\bar{\mu}\left(1+\frac{\log(n/\bar{\mu})}{k}\right)^{\frac{k-1}{2}} &= \bar{\mu}\left(\frac{\log n}{k} + \log\left(\frac{2}{\bar{\mu}^{1/k}}\right)\right)^{\frac{k-1}{2}} \\
&\leq 2\max\left\{\bar{\mu}\left(\frac{\log n}{k}\right)^{\frac{k-1}{2}}, \bar{\mu}\log\left(\frac{2}{\bar{\mu}^{1/k}}\right)^{\frac{k-1}{2}}\right\} \qquad \leq 2(\log n)^{\frac{k-1}{2}}.
\end{aligned}$$

Hence, by Theorem 20,

$$L_{1,k}(B) \leq s \cdot \bar{\mu} \cdot O\left(\frac{n}{k}\left(1+\frac{\log(n/\bar{\mu})}{k}\right)\right)^{\frac{k-1}{2}} \leq s \cdot O\left(\frac{n \log n}{k}\right)^{\frac{k-1}{2}}.$$

□

We now prove Proposition 7. This is a direct consequence of a result of Błasiok, Ivanov, Jin, Lee, Servedio and Viola:

Theorem 24 (Theorem 24 of [BIJ+21])**.** *For all positive integers n and k where $k \leq n$, there is a symmetric $\mathbb{F}_2$-polynomial $p(x_1, \ldots, x_n)$ of degree a power of two in $[\sqrt{kn}, 8\sqrt{kn}]$ such that*

$$M_k(p) := \left|\sum_{|S|=k} \widehat{p}(S)\right| \geq (e^{-k}/2)\binom{n}{k}^{1/2}.$$

Their result is stated for $L_{1,k}(p)$, but the proof holds without modification for $M_k(p)$.

Proof of Proposition 7. Given n and k, let $p(x_1, \ldots, x_n)$ be the $\mathbb{F}_2$-symmetric polynomial in Theorem 24. As observed in [BIJ+21], as a consequence of a result of Bhatnagar, Gopalan, and Lipton [BGL06], p can be computed by a permutation branching program B of width $16\sqrt{kn}$. As $\sum_{|S|=k} \widehat{B}(S) = \sum_{v \in V_{\mathsf{acc}}} \sum_{|S|=k} \widehat{B_{\to v}}(S)$, the conclusion follows by an averaging argument. □

We end this section by proving the $L_{1,k}$ bounds for generalized group products. To do so, we recall the formal statement of the composition theorem of [BIJ+21].

Theorem 25 (Theorem 31 in [BIJ+21])**.** *Suppose $\mathcal{F}$ and $\mathcal{G}$ are closed under negation of their outputs. Let $g_1, \ldots, g_m \in \mathcal{G}$ and let $f \in \mathcal{F}$, where $\mathcal{F}$ is closed under restrictions. Suppose that for every $1 \leq k \leq K$, we have*

1. $L_{1,k}(f) \le \mathbf{Pr}[f(U_m) = 1] \cdot a_{\mathsf{outer}} \cdot b_{\mathsf{outer}}^k$ *for every* $f \in \mathcal{F}$*, and*

2. $L_{1,k}(g) \le \mathbf{Pr}[g(U_\ell) = 1] \cdot a_{\mathsf{inner}} \cdot b_{\mathsf{inner}}^k$ *for every* $g \in \mathcal{G}$*.*

Then for every function $h \in \mathcal{F} \circ \mathcal{G}$*, we have that*

$$L_{1,K}(h) \le \mathbf{Pr}[h(U_{m\ell}) = 1] \cdot a_{\mathsf{outer}} \cdot (a_{\mathsf{inner}} b_{\mathsf{inner}} b_{\mathsf{outer}})^K.$$

Proof of Corollary 16. An (m, ℓ, G)-product can be computed by the disjoint composition of a width-$|G|$ permutation branching program and arbitrary Boolean functions on ℓ bits, where both classes are closed under negation of their outputs and restrictions. Note that applying the map $f \mapsto 2f - 1$ to a $\{0, 1\}$-valued function f only affects its $L_{1,k}$ by at most a factor of 2. So we can apply Theorem 25 to Theorem 6 and Lemma 22. $\square$

3 Coin theorems and pseudorandom generators

In this section, we prove our coin problem bounds for regular and permutation branching programs (Corollaries 8 and 9 and Claim 10), and construct PRGs for permutation branching programs (Corollary 9 and Theorem 13) and generalized group products (Corollary 17).

We start with Corollary 8.

Proof of Corollary 8. Let B be a regular branching program. We identify B with its multilinear extension. By linearity of expectation and Theorem 6, we have

$$\begin{aligned}
\big|\mathbf{E}[B(X_\delta)] - \mathbf{E}[B(X_0)]\big| &= \big|B(\vec{\delta}) - B(\vec{0})\big| \\
&\le \sum_{k=1}^{n} \delta^k \sum_{|S|=k} \big|\widehat{B}(S)\big| \\
&\le \delta L_{1,1}(B) + \sum_{k=2}^{n} \delta^k L_{1,k}(B) \\
&\le \delta s + \sum_{k=2}^{n} \delta^k \min\{w^k, s \cdot O(\sqrt{n \log n}))^{k-1}\} \\
&\le \delta s + \delta^2 \cdot O\Big(\min\big\{w^2, s\sqrt{n \log n}\big\}\Big),
\end{aligned}$$

where the last inequality is because when $\delta \le \alpha \max\{1/w, 1/\sqrt{n \log n}\}$, then at least one of the summations $\sum_k (\delta w)^k$ and $\sum_k O(\delta\sqrt{n \log n})^k$ is a geometric sum with ratio at most $1/2$, and thus is bounded by twice of its first term. $\square$

Corollary 9 follows from applying a result of Avishay Tal establishing that $L_{1,1}$ bounds imply coin problem bounds for classes that are closed under restrictions to Theorem 6.

Lemma 26 (Lemma 3.2 in [Agr20])**.** *Let* $\mathcal{F}$ *be a function class that is closed under restrictions. Then for every* $f \in \mathcal{F}$*,*

$$\big|\mathbf{E}[f(X_\delta)] - \mathbf{E}[f(X_0)]\big| \le \ln\Big(\frac{1}{1-\delta}\Big) L_{1,1}(\mathcal{F}) \le \frac{\delta}{1-\delta} L_{1,1}(\mathcal{F}).$$

We now prove Claim 10. The idea is similar to proof idea behind Proposition 7. Here we give a self-contained argument. We approximate the Majority function on some $n = \Theta(1/\delta^2)$ bits by computing it correctly on inputs of Hamming weights between $n/2 + \Theta(\sqrt{n})$ and $n/2 - \Theta(\sqrt{n})$. This can be implemented by counting their Hamming weights modulo $\Theta(\sqrt{n})$ and hence can be done using a permutation program of width $\Theta(\sqrt{n})$.

Proof of Claim 10. Let $n := 32/\delta^2$ and $m := 64/\delta$. Consider the function $f\colon \{-(m-1), \dots, m\} \to \{0,1\}$ defined by $f(\ell) := 1$ if and only if $\ell \geq m/4$. We first construct the permutation program B, which on inputs x where $\sum_i x_i \in \ell + 2m\mathbb{Z}$ for some $\ell \in \{-(m-1), \dots, m\}$, outputs $B(x) := f(\ell)$. By counting modulo $2m$, this can be computed with width $2m$ and at least $m/2$ accept states. By the Chernoff bound,

$$\mathbf{Pr}[B(X_\delta) = 0] \leq \mathbf{Pr}\left[\left|\sum_{i=1}^{n} (X_\delta)_i\right| \geq m\right] + \mathbf{Pr}\left[\sum_{i=1}^{n} (X_\delta)_i < m/4\right] \leq 1/20.$$

Similarly, $\mathbf{Pr}[B(X_0) = 1] \leq 1/20$. Therefore, $\mathbf{E}[B(X_\delta)] - \mathbf{E}[B(X_0)] \geq 9/10$.

We now modify B by choosing s of its at most m many accept states uniformly at random, then letting B accept only at these s states and reject the rest of them. It follows by an averaging argument that there exists a choice of s accepting states such that the modified program B' satisfies

$$\mathbf{E}[B'(X_\delta)] - \mathbf{E}[B'(X_0)] \geq (s/m) \cdot (9/10) \geq s\delta/1000. \qquad \square$$

We now construct PRGs for bounded width permutation branching programs and generalized group products. We will use the following result that constructs PRGs from Fourier growth bounds using the "polarizing random walk framework."

Theorem 27 (Theorem 1.3 in [CHHL19])**.** *Let $\mathcal{F}$ be a function class on n bits that is closed under restrictions. Suppose $L_{1,k}(\mathcal{F}) \leq b^k$ for some $b \geq 1$. Then there exists an explicit pseudorandom generator for $\mathcal{F}$ with seed length $b^2 \cdot O(\log(n/\epsilon))(\log(1/\epsilon) + \log\log n)$ and error ϵ.*

Corollaries 12 and 17 then follow from applying Theorem 27 to Theorem 6 and Corollary 16, respectively.

We prove Theorem 13 by approximately sampling δ-biased coins. To do this efficiently, we follow the approach in [HLV18]. Recall that $H(x) = x\log(\frac{1}{x}) + (1-x)\log(\frac{1}{1-x})$ denotes the binary entropy function. For two distributions X and Y, we use $\|X - Y\|_1$ to denote their total variation distance.

Lemma 28. *Given $\delta > 0$, there is some $s = H(1/2 + 0.499\delta)n + o(n)$ and a polynomial-time computable function $f\colon \{0,1\}^s \to \{-1,1\}^n$ such that $\|X_\delta - f(U_s)\|_1 \leq \delta/100$.*

As its proof is a only a slight modification of the one in [HLV18], we defer it to the end of this section. To construct our PRG, it suffices to sample a distribution close to X_δ using Lemma 28.

Proof of Theorem 13. Let $f\colon \{0,1\}^s \to \{-1,1\}^n$ be the function obtained from Lemma 28 with the given δ, where

$$s \leq H(1/2 + 0.499\delta)n + o(n) + O(\log(1/\delta)) = \big(H(1/2 + 0.499\delta) + o(1)\big)n.$$

Let $B\colon \{-1,1\}^n \to \{0,1\}$ be a permutation branching program with a single accept state. Then

$$\begin{aligned}
\big|\mathbf{E}[B(U_n)] - \mathbf{E}[B(f(U_s))]\big| &\leq \big|\mathbf{E}[B(U_n)] - \mathbf{E}[B(X_\delta)]\big| + \big|\mathbf{E}[B(X_\delta)] - \mathbf{E}[B(f(U_s))]\big| \\
&\leq \big|\mathbf{E}[B(U_n)] - \mathbf{E}[B(X_\delta)]\big| + \delta/100 && \text{(Lemma 28)} \\
&\leq \frac{\delta}{1-\delta} + \frac{\delta}{100} && \text{(Corollary 9)},
\end{aligned}$$

proving the theorem. $\square$

It remains to prove Lemma 28. We will use a lemma in [HLV18] enabling us to approximately sample distributions.

Lemma 29 (Lemma 36 in [HLV18]). *Let D be a distribution on $[m]$. Suppose that given $i \in [m]$ we can compute in time polynomial in $O(\log m)$ the cumulative distribution $\mathbf{Pr}[D \le i]$. Then there is a* $\text{polylog}(mt)$*-time computable function f such that given any $t \ge 1$, f uses $s = \lceil \log(mt) \rceil$ bits to sample an element from the support of D such that $\|f(U_s) - D\|_1 \le 1/t$.*

We will also bound above the binomial coefficients in terms of the entropy function.

Fact 30. *For every $\rho > 0$ we have*

$$\log \binom{n}{\lceil n(1/2+\rho) \rceil} \le (1+o(1))n \cdot H(1/2+\rho).$$

We now prove the lemma, by giving an appropriate sampling procedure:

Proof of Lemma 28. Let X'_δ as the distribution over $\{0,1\}^n$, where the coordinates are independent and each coordinate is 1 with probability $1/2+\delta/2$ and 0 otherwise. Our sampling procedure below will sample a distribution D over $\{0,1\}^n$ that is close to X'_δ (over $\{0,1\}^n$), then apply $x_i \mapsto 2x_i - 1$ to each coordinate x_i of D to sample the target distribution over $\{-1,1\}^n$.

Consider the following procedure for sampling a string x from X'_δ. First sample the Hamming weight i of x according to $\text{Binomial}(n, 1/2+\delta/2)$, where each weight $i \in \{0,\dots,n\}$ is chosen with probability $\binom{n}{i}(1/2+\delta/2)^i(1/2-\delta)^{n-i}$. Then given $i \in \{0,\dots,n\}$, sample x uniformly from the set of strings with weight exactly i. By performing both steps in an approximate manner, we obtain f.

To do this, we apply Lemma 29 to sample the weight i from a distribution D (over $\{0,\dots,n\}$) that is within $\delta/300$ in total variation distance to $\text{Binomial}(n, 1/2+\delta/2)$, which costs $O(\log n + \log(1/\delta))$ bits. Given $i \sim D$, if $i < \lceil n(1/2+0.499\delta) \rceil$ then we return the all 0s string; otherwise, we apply Lemma 29 to sample from the set of strings of Hamming weight $i \ge \lceil n(1/2+0.499\delta) \rceil$.

As D is $(\delta/300)$-close to $|X'_\delta|$, for every sufficiently large n, we have

$$\mathbf{Pr}\big[D < \lceil n(1/2+0.499\delta) \rceil\big] \le \mathbf{Pr}\big[|X'_\delta| < \lceil n(1/2+0.499\delta) \rceil\big] + \delta/300 < \delta/150.$$

Here, we use Fact 30 to bound the log of the description size of the universe, i.e. the number of strings of some Hamming weight $i \ge \lceil n(1/2+0.499\delta) \rceil$, by

$$\log \binom{n}{\lceil n(1/2+0.499\delta) \rceil} \le (1+o(1))H(1/2+0.499\delta)n = H(1/2+0.499\delta)n + o(n).$$

Furthermore, Haramaty, Lee, and Viola show (in the proof of [HLV18, Lemma 35]) that we can sample from the distribution of strings of length n with Hamming weight i in time $\text{poly}(n)$. Thus, the total number of random bits required to sample a distribution within $\delta/100$ in total variation distance to X_δ is at most $s = H(1/2+0.499\delta) \cdot n + o(n) + o(n) + O(\log(1/\delta))$, and f can be computed in polynomial time as desired. $\square$

References

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In *STOC'10—Proceedings of the 2010 ACM International Symposium on Theory of Computing*, pages 141–150. ACM, New York, 2010. 1.2

[AD11] Scott Aaronson and Andrew Drucker. Advice coins for classical and quantum computation. In *Automata, languages and programming. Part I*, volume 6755 of *Lecture Notes in Comput. Sci.*, pages 61–72. Springer, Heidelberg, 2011. 1.2

[Agr20] Rohit Agrawal. Coin theorems and the Fourier expansion. *Chic. J. Theoret. Comput. Sci.*, pages Art. 4, 15, 2020. 1.2, 26

[Ajt83] M. Ajtai. Σ^1_1-formulae on finite structures. *Ann. Pure Appl. Logic*, 24(1):1–48, 1983. 1.2

[BGL06] Nayantara Bhatnagar, Parikshit Gopalan, and Richard J. Lipton. Symmetric polynomials over z_m and simultaneous communication protocols. *J. Comput. Syst. Sci.*, 72(2):252–285, 2006. 1.1, 2

[BGW20] Mark Braverman, Sumegha Garg, and David P. Woodruff. The coin problem with applications to data streams. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science*, pages 318–329. IEEE Computer Soc., Los Alamitos, CA, [2020] ©2020. 1.2, 1.2

[BGZ21] Mark Braverman, Sumegha Garg, and Or Zamir. Tight space complexity of the coin problem. *Electron. Colloquium Comput. Complex.*, page 83, 2021. 1.2

[BHPP21] Andrej Bogdanov, William Hoza, Gautam Prakriya, and Edward Pyne. Hitting sets for regular branching programs. *Electron. Colloquium Comput. Complex.*, page 143, 2021. 1, 1, 1, 1.2, 1.2, 2, 1.3, 1.3, 1.3, 2, 2, 23, 2

[BIJ+21] Jaroslaw Błasiok, Peter Ivanov, Yaonan Jin, Chin Ho Lee, Rocco A. Servedio, and Emanuele Viola. Fourier growth of structured F_2-polynomials and applications. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2021, August 16-18, 2021, University of Washington, Seattle, Washington, USA (Virtual Conference)*, volume 207 of *LIPIcs*, pages 53:1–53:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 1.1, 1.1, 1.1, 1.1, 1.1, 1.2, 1.2, 24, 2, 2, 25

[BPW11] Andrej Bogdanov, Periklis A. Papakonstantinou, and Andrew Wan. Pseudorandomness for read-once formulas. In *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science—FOCS 2011*, pages 240–246. IEEE Computer Soc., Los Alamitos, CA, 2011. 1.2, 1.2

[BRRY14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014. 1, 1.2

[BS21] Nikhil Bansal and Makrand Sinha. k-forrelation optimally separates quantum and classical query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2021, New York, NY, USA, 2021. 1

[BTW15] Eric Blais, Li-Yang Tan, and Andrew Wan. An inequality for the fourier spectrum of parity decision trees, 2015. 1

[BV10] Joshua Brody and Elad Verbin. The coin problem, and pseudorandomness for branching programs. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science—FOCS 2010*, pages 30–39. IEEE Computer Soc., Los Alamitos, CA, 2010. 1.2, 1.2, 1.3, 2

[CDrI+13] Gil Cohen, Ivan Bjerre Damgå rd, Yuval Ishai, Jonas Kölker, Peter Bro Miltersen, Ran Raz, and Ron D. Rothblum. Efficient multiparty protocols via log-depth threshold formulae (extended abstract). In *Advances in cryptology—CRYPTO 2013. Part II*, volume 8043 of *Lecture Notes in Comput. Sci.*, pages 185–202. Springer, Heidelberg, 2013. 1.2

[CGL+21] Eshan Chattopadhyay, Jason Gaitonde, Chin Ho Lee, Shachar Lovett, and Abhishek Shetty. Fractional Pseudorandom Generators from Any Fourier Level. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:24, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 1, 1, 1, 1.2

[CHH+20] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, Shachar Lovett, and David Zuckerman. XOR Lemmas for Resilient Functions against Polynomials. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 234–246, New York, NY, USA, 2020. Association for Computing Machinery. 1

[CHHL19] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. Pseudorandom generators from polarizing random walks. *Theory Comput.*, 15:Paper No. 10, 26, 2019. 1, 1, 1.2, 1.2, 27

[CHLT19] Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. Pseudorandom generators from the second Fourier level and applications to AC0 with parity gates. In *10th Innovations in Theoretical Computer Science*, volume 124. 2019. 1, 1.2

[CHRT18] Eshan Chattopadhyay, Pooya Hatami, Omer Reingold, and Avishay Tal. Improved pseudorandomness for unordered branching programs through local monotonicity. In *STOC'18—Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 363–375. ACM, New York, 2018. 1, 1, 1.1, 1.1, 1.1, 1.2, 1.3, 1.3, 1.3, 2, 2, 19

[De11] Anindya De. Pseudorandomness for permutation and regular branching programs. In *26th Annual IEEE Conference on Computational Complexity*, pages 221–231. IEEE Computer Soc., Los Alamitos, CA, 2011. 1

[DHH19] Dean Doron, Pooya Hatami, and William M. Hoza. Near-optimal pseudorandom generators for constant-depth read-once formulas. In *34th Computational Complexity Conference*, volume 137 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 16, 34. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2019. 1.2

[DHH20] Dean Doron, Pooya Hatami, and William M. Hoza. Log-seed pseudorandom generators via iterated restrictions. In *35th computational complexity conference*, volume 169 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 6, 36. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2020. 1.2, 1.2

[DMR+21] Dean Doron, Raghu Meka, Omer Reingold, Avishay Tal, and Salil Vadhan. Pseudorandom Generators for Read-Once Monotone Branching Programs. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, volume 207 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:21, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 1.2

[EI21] Alexandros Eskenazis and Paata Ivanisvili. Learning low-degree functions from a logarithmic number of random queries. *CoRR*, abs/2109.10162, 2021. 1, 1

[FK18] Michael A. Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *59th Annual IEEE Symposium on Foundations of Computer Science—FOCS 2018*, pages 946–955. IEEE Computer Soc., Los Alamitos, CA, 2018. 1, 1.2, 1.2

[GII+19] Alexander Golovnev, Rahul Ilango, Russell Impagliazzo, Valentine Kabanets, Antonina Kolokolova, and Avishay Tal. $AC^0[p]$ Lower Bounds Against MCSP via the Coin Problem. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 66:1–66:15, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. 1.2

[GKM18] Parikshit Gopalan, Daniel M. Kane, and Raghu Meka. Pseudorandomness via the discrete Fourier transform. *SIAM J. Comput.*, 47(6):2451–2487, 2018. 1.2, 1.2

[GMRZ13] Parikshit Gopalan, Raghu Meka, Omer Reingold, and David Zuckerman. Pseudorandom generators for combinatorial shapes. *SIAM J. Comput.*, 42(3):1051–1076, 2013. 1.2

[GRZ21] Uma Girish, Ran Raz, and Wei Zhan. Lower Bounds for XOR of Forrelations. In Mary Wootters and Laura Sanità, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, volume 207 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 52:1–52:14, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 1

[GSTW16] Parikshit Gopalan, Rocco A. Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions. *Electron. Colloquium Comput. Complex.*, 23:69, 2016. 1

[GTW21] Uma Girish, Avishay Tal, and Kewen Wu. Fourier growth of parity decision trees. In *36th Computational Complexity Conference*, volume 200 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 39, 36. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021. 1, 1, 1.1

[HLV18] Elad Haramaty, Chin Ho Lee, and Emanuele Viola. Bounded independence plus noise fools products. *SIAM J. Comput.*, 47(2):493–523, 2018. 1.2, 1.2, 3, 3, 3, 29, 3

[HPV21] William M. Hoza, Edward Pyne, and Salil Vadhan. Pseudorandom generators for unbounded-width permutation branching programs. In *12th Innovations in Theoretical Computer Science Conference*, volume 185 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 7, 20. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021. 1, 1.2

[IMZ19] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. *J. ACM*, 66(2):Art. 11, 16, 2019. 1.2

[IRR+21] Siddharth Iyer, Anup Rao, Victor Reis, Thomas Rothvoss, and Amir Yehudayoff. Tight bounds on the fourier growth of bounded functions on the hypercube. *CoRR*, abs/2107.06309, 2021. 1, 1

[Lee19] Chin Ho Lee. Fourier bounds and pseudorandom generators for product tests. In *34th Computational Complexity Conference*, volume 137. 2019. 1, 1.1, 1.1, 1.1, 1.2, 1.2, 1.2, 1.2, 1.2, 1.3, 2

[LRTV09] Shachar Lovett, Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom bit generators that fool modular sums. In *Approximation, randomization, and combinatorial optimization*, volume 5687 of *Lecture Notes in Comput. Sci.*, pages 615–630. Springer, Berlin, 2009. 1.2

[LSS+21] Nutan Limaye, Karteek Sreenivasaiah, Srikanth Srinivasan, Utkarsh Tripathi, and S. Venkitesh. A fixed-depth size-hierarchy theorem for $\mathrm{AC}^0[\oplus]$ via the coin problem. *SIAM J. Comput.*, 50(4):1461–1499, 2021. 1.2

[LV18] Chin Ho Lee and Emanuele Viola. The coin problem for product tests. *ACM Trans. Comput. Theory*, 10(3):Art. 14, 10, 2018. 1.2, 1.2

[LV20] Chin Ho Lee and Emanuele Viola. More on bounded independence plus noise: pseudorandom generators for read-once polynomials. *Theory Comput.*, 16:Paper No. 7, 50, 2020. 1.2, 1.2

[Man95] Yishay Mansour. An $O(n^{\log\log n})$ learning algorithm for DNF under the uniform distribution. volume 50, pages 543–550. 1995. Fifth Annual Workshop on Computational Learning Theory (COLT) (Pittsburgh, PA, 1992). 1, 1

[MRT19] Raghu Meka, Omer Reingold, and Avishay Tal. Pseudorandom generators for width-3 branching programs. In *STOC'19—Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 626–637. ACM, New York, 2019. 1.2, 1.2

[MZ09] Raghu Meka and David Zuckerman. Small-bias spaces for group products. In *Approximation, randomization, and combinatorial optimization*, volume 5687 of *Lecture Notes in Comput. Sci.*, pages 658–672. Springer, Berlin, 2009. 1.2

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. 1.2

[O'D14] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. 1, 1.1

[OS07] Ryan O'Donnell and Rocco A. Servedio. Learning monotone decision trees in polynomial time. *SIAM J. Comput.*, 37(3):827–844, 2007. 1

[PV21] Edward Pyne and Salil Vadhan. Pseudodistributions that beat all pseudorandom generators (extended abstract). In *36th Computational Complexity Conference*, volume 200 of *LIPIcs. Leibniz Int. Proc. Inform.*, pages Art. No. 33, 15. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2021. 1

[PV22] Edward Pyne and Salil Vadhan. Deterministic approximation of random walks via queries in graphs of unbounded size. In *Symposium on Simplicity in Algorithms (SOSA)*, pages 57–67. SIAM, 2022. 1

[RSV13] Omer Reingold, Thomas Steinke, and Salil Vadhan. Pseudorandomness for regular branching programs via Fourier analysis. In *Approximation, randomization, and combinatorial optimization*, volume 8096 of *Lecture Notes in Comput. Sci.*, pages 655–670. Springer, Heidelberg, 2013. 1, 1, 1, 1.1, 1.1, 1.2, 1.2, 2

[RT19] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In *STOC'19—Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 13–23. ACM, New York, 2019. 1

[RTV06] Omer Reingold, Luca Trevisan, and Salil Vadhan. Pseudorandom walks on regular digraphs and the **RL** vs. **L** problem. In *STOC'06: Proceedings of the 38th Annual ACM Symposium on Theory of Computing*, pages 457–466. ACM, New York, 2006. 1, 1

[SSW21] Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. *An Optimal Separation of Randomized and Quantum Query Complexity*, page 1289–1302. Association for Computing Machinery, New York, NY, USA, 2021. 1, 1

[Ste13] John Steinberger. The distinguishability of product distributions by read-once branching programs. In *2013 IEEE Conference on Computational Complexity—CCC 2013*, pages 248–254. IEEE Computer Soc., Los Alamitos, CA, 2013. 1.2

[SV10] Ronen Shaltiel and Emanuele Viola. Hardness amplification proofs require majority. *SIAM J. Comput.*, 39(7):3122–3154, 2010. 1.2

[SVW17] Thomas Steinke, Salil Vadhan, and Andrew Wan. Pseudorandomness and Fourier-growth bounds for width-3 branching programs. *Theory Comput.*, 13:Paper No. 12, 50, 2017. 1, 1.2

[Tal17] Avishay Tal. Tight bounds on the Fourier spectrum of $\mathbf{AC^0}$. In *32nd Computational Complexity Conference*, volume 79. 2017. 1

[Tal20] Avishay Tal. Towards optimal separations between quantum and randomized query complexities. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 228–239, 2020. 1, 1

[Val84] L. G. Valiant. Short monotone formulae for the majority function. *J. Algorithms*, 5(3):363–366, 1984. 1.2

[Vio21] Emanuele Viola. Fourier Conjectures, Correlation Bounds, and Majority. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming (ICALP 2021)*, volume 198 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 111:1–111:15, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 1

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il