Pseudorandom Generators from the Fourier Spectrum

Jason Gaitonde

Abstract

As we have seen from class, the construction of pseudorandom generators for certain classes of functions are often based on other "pseudorandom" objects, like ϵ-biased spaces, expanders, and extractors. Here, we will first discuss how the Fourier spectrum of Boolean functions is relevant in this regard, before turning to a new framework that develops pseudorandom generators for classes with sufficiently nice Fourier tails as introduced by Chattopadhyay, et al [1]. Their technique first constructs *fractional pseudorandom generators*, which are less rigid than standard pseudorandom generators because they need not lie on the vertices of the Boolean hypercube, and then adds them together to form a walk that quickly approaches $\{-1, +1\}^n$; the pseudorandom generator then works by rounding the final coordinates of the walk. This construction turns out to yield a single PRG with comparable seed-length to the best-known constructions for several natural classes of functions that satisfy the Fourier tail bounds.

We also discuss how the ideas from the paper have since been applied and advanced in other recent works. Somewhat surprisingly, this technique of using "small" random steps to analyze a larger random process was used by Raz and Tal in showing an oracle separation of **BQP** and **PH** [2]. Moreover, this same technique was used by Chattopadhyay, et al [3], to construct new pseudorandom generators that fool classes of functions whose second Fourier level is sufficiently bounded, as opposed to the entire tail. Proving such bounds for other natural classes of Boolean functions is an active area of interest.

1 Introduction

Randomness, and finding ways to reduce it, is of principal concern in complexity theory. A main goal is often to *derandomize* a randomized computation without degrading the time complexity of the algorithm. One natural approach to do so is to construct *pseudorandom generators* that take much less randomness and returns bits that "look" random to the perspective of the algorithm. An ambitious, but very plausible, goal is to show that every randomized computation that runs in polynomial time can be derandomized into another algorithm that computes the same language in polynomial time, i.e. $\mathsf{P} = \mathsf{BPP}$.

To construct pseudorandom generators, one often focuses on specific models of computation; for instance, consider Nisan's celebrated pseudorandom generator for space-bounded computation [4]. This can lead to specialized generators for specific classes. One main tool in proving that a certain construction works for some certain model is via Fourier analysis.

Here, we survey recent works that in some sense, go straight to the source to construct pseudorandom generators for large classes of functions that satisfy a Fourier tail bound. This approach proceeds by constructing a relaxation of pseudorandom generators, which are relatively easy to construct, and then combining them in a clever way that relates to the Fourier expansion of functions. We then show how these ideas were used in a very different problem to prove circuit lower bounds that imply the oracle separation of BQP, the set of languages computed by quantum algorithms in polynomial time, and PH, the polynomial hierarchy. We first introduce the topic by explaining

why the Fourier expansion is a natural tool to use in the construction of pseudorandom generators before surveying these new works.

2 Notation

2.1 Fourier Analysis

We quickly review some of the relevant basics from the Fourier analysis of Boolean functions [5]. For any function $f : \{\pm 1\}^n \to \{\pm 1\}$, we may express f in its *Fourier expansion*

$$f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(x), \tag{1}$$

where $\chi_S(x) = \prod_{i \in S} x_i$, and $\hat{f}(S) = \langle f, \chi_S \rangle = \mathbb{E}_{x \sim \{\pm 1\}^n}[f(x)\chi_S(x)] = 2^{-n} \sum_{x \in \{\pm 1\}^n} f(x)\chi_S(x)$.

Later on, we will extend Boolean functions f to a function on $[-1, 1]^n$ to $[-1, 1]$ using its Fourier expansion, that is, for all $x \in [-1, 1]$,

$$f(x) := \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(x); \tag{2}$$

we briefly observe that the Fourier expansion is the unique multilinear polynomial that agrees with f on $\{\pm 1\}^n$ as can be easily verified by interpolating the coefficients on all monomials of degree at most n, and therefore, the extension of any Boolean function on the Boolean hypercube to a multilinear function is unambiguous. The value of $f(x)$ in the cube $[-1, 1]^n$ has a particularly nice interpretation; if we define independent random variables $X_i \in \{\pm 1\}$ such that $\mathbb{E}[X_i] = x_i$, then we can easily compute

$$\mathbb{E}_X[f(X)] = \mathbb{E}_X\left[\sum_{S \subseteq [n]} \hat{f}(S)\chi_S(X)\right] = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{i \in S} \mathbb{E}[X_i] = \sum_{S \subseteq [n]} \hat{f}(S) \prod_{i \in S} x_i = f(x); \tag{3}$$

therefore, $f(x)$ is an expectation over the corners of the Boolean hypercube, which shows that f maps $[-1, 1]^n \to [-1, 1]$.

2.2 Pseudorandom Generators

Let $\mathcal{F}$ be a class of Boolean functions. We say that a pseudorandom generator (PRG) for the class $\mathcal{F}$ with error $\epsilon > 0$ is a random variable $X \in \{\pm 1\}^n$ such that

$$|\mathbb{E}_X[f(X)] - \mathbb{E}_U[f(U)]| \leq \epsilon, \quad \forall f \in \mathcal{F}, \tag{4}$$

where U denotes the uniform distribution over $\{\pm 1\}^n$. Of course, by the previous discussion, this is equivalent to

$$|\mathbb{E}_X[f(X)] - f(0)| \leq \epsilon, \quad \forall f \in \mathcal{F}, \tag{5}$$

where we consider the multilinear expansion of f. Note that $f(0) = \mathbb{E}_U[f(U)] = \hat{f}(\emptyset)$, by definition.

3 Warmup: PRGs from Small Spectral Norm

To see why the Fourier spectrum can be useful in the design of pseudorandom generators, we can use the Triangle Inequality to write

$$\begin{aligned}|\mathbb{E}_X[f(X)] - f(0)| &= |\mathbb{E}_X[\sum_{S\subseteq[n]} \hat{f}(S)\chi_S(X)] - \hat{f}(\emptyset)| \\ &= |\sum_{S\neq\emptyset} \hat{f}(S)\mathbb{E}_X[\chi_S(X)]| \\ &\leq \sum_{S\neq\emptyset} |\hat{f}(S)||\mathbb{E}_X[\chi_S(X)]|\end{aligned}$$

Write $\|\hat{f}\|_1 := \sum_{S\subseteq[n]} |f(S)|$ to be the L_1-spectral norm of f. If $\|\hat{f}\|_1 \leq \text{poly}(n)$ for all $f \in \mathcal{F}$, then these functions are particularly easy to fool; recall that an ϵ-biased space is a random variable $X \in \{\pm 1\}^n$ such that $|\mathbb{E}_X[\chi_S]| \leq \epsilon$ for all $\emptyset \neq S \subseteq [n]$. That is, an ϵ-biased space fools all parity tests with error at most ϵ. Such spaces can be constructed using $O(\log(n/\epsilon))$ random bits [6]; here, constructing a biased space X with bias $\epsilon/\text{poly}(n)$, we see that

$$|\mathbb{E}_X[f(X)] - f(0)| \leq \sum_{S\subseteq[n]} |\hat{f}(S)||\mathbb{E}_X[\chi_S(x)]| \leq \frac{\epsilon}{\text{poly}(n)}\|\hat{f}\|_1 \leq \epsilon; \tag{6}$$

therefore, X ϵ-fools $\mathcal{F}$. Such a construction takes $O(\log(\text{poly}(n)/\epsilon)) = O(\log(n/\epsilon))$ bits, so is particularly efficient.

3.1 Examples

Intuitively, functions with small L_1-spectral norm are necessarily concentrated on a small number terms in the Fourier basis. Indeed, by Parseval's theorem, the L_2-spectral norm (the sum of squares of the Fourier coefficients) is identically 1; in particular, by considering a function $f : \{\pm 1\}^n \to \{\pm 1\}$ as a vector of Fourier coefficients (with length 2^n), f necessarily must lie on the L_2-unit ball in $\mathbb{R}^{2^n}$. Of course, the L_1-unit ball is contained in the L_2-unit ball, and because all norms are equivalent in finite dimensional space, there is a smallest radius L_1-ball that contains the L_2-unit ball. A simple application of Cauchy-Schwarz implies that the smallest such L_1-ball has L_1 radius $2^{n/2}$; geometrically, the reason is that points on the L_2-ball that are not close to the unit basis vectors have large L_1 norm. In particular, this discussion implies that the L_1-spectral norm can be exponential in n, so clearly this method of using ϵ-biased spaces as pseudorandom generators cannot be sufficient for all, or even potentially many, functions. In fact, consider a *random* function $\mathbf{f}$, so that $\mathbf{f}(x) = 1$ with probability $1/2$ and -1 with probability $1/2$, independently across x. Then we can compute, for any $S \subseteq [n]$,

$$\begin{aligned}\mathbb{E}_{\mathbf{f}}[|\hat{\mathbf{f}}(S)|] &= \frac{1}{2^n}\mathbb{E}_{\mathbf{f}}\left[\left|\sum_{x\in\{\pm1\}^n} \mathbf{f}(x)\chi_S(x))\right|\right] \\ &= \frac{1}{2^n}\mathbb{E}_{\mathbf{f}}\left[\left|\sum_{x\in\{\pm1\}^n} \mathbf{f}(x)\right|\right] \\ &\approx \frac{\sqrt{\frac{2}{\pi}}}{2^{n/2}},\end{aligned}$$

where the second equality uses the fact that the terms $\mathbf{f}(x)\chi_S(x)$ are distributed the same as $\mathbf{f}(x)$ for all x, independently, and the last approximation uses the standard fact that the expected deviation of a random walk after 2^n steps is $\sqrt{\frac{2}{\pi}}2^{n/2}$. By linearity, this implies that

$$\mathbb{E}_{\mathbf{f}}[\|\hat{\mathbf{f}}\|_1] = \sum_{S\subseteq[n]} \mathbb{E}_{\mathbf{f}}[|\hat{\mathbf{f}}(S)|] \approx 2^{n/2}\sqrt{\frac{2}{\pi}}, \tag{7}$$

which is close to the tight bound. This implies that most functions have high L_1-spectral norm, so will not be fooled by this analysis.

Still, though this analysis has been fairly elementary, this analysis of the Fourier spectrum holds for some natural classes of functions. Here, we highlight some natural classes where these spectral bounds are enough:

Example 1 (Width 2 Oblivious Read-Once Branching Programs)**.** *Consider the class $\mathcal{B}^{ob}_{2,n}$ of oblivious, read-once branching programs (OROBP) of width 2 on n variables. An oblivious read-once branching program of width w on n variables is modeled by a layered directed graph with $n+1$ layers with at most w nodes each, where all nodes in a layer read the same, fixed, input bit, and accordingly move along the edge corresponding to the bit it read to the next layer. In the final layer, some nodes will be "accepting" and the rest will be "rejecting." The intuition for why these branching programs have small spectral norm is that they are constrained to be only slightly more complicated than parities.*

We will show that if $f \in \mathcal{B}^{ob}_{2,n}$, which we will represent as a function $f : \{\pm1\}^n \to \{0,1\}$ for convenience, then $\|\hat{f}\|_1 \leq n$. Note that by changing the range, the expectations in the definition of PRGs become just the probability of f accepting under U and X; it is easy to check that these definitions are the same up to a factor of 2. The interpretation of $f(x) = 1$ is that when the branching program reads in input x one bit at a time, it ends up at an "accept" state.

To see this, we proceed by induction. Suppose f is computed by an oblivious branching program of width 2 on 1 variable; then it is easy to see that the only possibilities are

$$f(x) = f(x_1) = \begin{cases} 1 & \text{if both edges go to "accept" state} \\ 0 & \text{if neither edge goes to "accept" state} \\ \frac{1}{2} + \frac{x_1}{2} & \text{if } f(1) = 1 \text{ and } f(-1) = 0 \\ \frac{1}{2} - \frac{x_1}{2} & \text{if } f(1) = 0 \text{ and } f(1) = 0. \end{cases} \tag{8}$$

In all of these cases, it is clear that $\|\hat{f}\|_1 \leq 1$. For the inductive step, suppose that the claim is true for all f computed by width 2, OROBPs on n variables. Let g be computed by a width 2, OROBP B on $n+1$ variables, which has $n+2$ layers. The key is to consider what B does on the first n variables.

Let v_1 be the first node in the $n+1$th layer of B, and v_2 be the second node. Then let $f_1 : \{\pm1\}^n \to \{0,1\}$ be the indicator of whether B reaches v_1 after reading the first n variables, and same with f_2 and v_2. Note that $f_2 = 1 - f_1$ and crucially, f_1 is computed by a width 2 OROBP, and so by induction has $\|\hat{f}_1\|_1 \leq n$. Let $g_1 : \{\pm1\} \to \{0,1\}$ be the action of B when starting on v_1 and reading the $n+1$th bit, and same with g_2 and v_2. It is then clear that

$$g(x_1, \ldots, x_{n+1}) = f_1(x_1, \ldots, x_n)g_1(x_{n+1}) + (1 - f_1(x_1, \ldots, x_n))g_2(x_{n+1}); \tag{9}$$

the interpretation of this formula is just that conditioned on where we are after reading n variables, only the action of B on the last variable matter. From this point, it is easy to explicitly upper bound

the L_1-spectral norm by considering cases. It is easy to check that the claim holds when $g_i = 0$ for either i, or when $g_1 = g_2$, so by exchanging f and $1 - f$ if necessary, the only cases to check are when $g_1 = 1$ and g_2 is one of the parities on x_{n+1} or when g_1 and g_2 compute distinct parities.

In the first case, when $g_2 = \frac{1}{2} + \frac{x_{n+1}}{2}$,

$$\begin{aligned} g(x_1, \ldots, x_{n+1}) &= f_1(x_1, \ldots, x_n) + (1 - f_1(x_1, \ldots, x_n))(\frac{1}{2} + \frac{x_{n+1}}{2}) \\ &= \frac{1}{2} + \frac{x_1}{2} + \frac{f_1(x_1, \ldots, x_n)}{2} + \frac{f_1(x_1, \ldots, x_n)x_{n+1}}{2} \end{aligned}$$

To conclude, we use the following lemma:

Lemma 2. *Let $f, g : \{\pm 1\}^n \to \mathbb{R}$. Then*

$$\|\widehat{f+g}\|_1 \leq \|\hat{f}\|_1 + \|\hat{g}\|_1 \tag{10}$$

$$\|\widehat{fg}\|_1 \leq \|\hat{f}\|_1 \|\hat{g}\|_1. \tag{11}$$

Proof. *The first equation is trivial, just using the fact that the Fourier Transform is linear and applying the normal Triangle Inequality to each Fourier coefficient. For the second condition, we can compute*

$$\begin{aligned} fg &= \Big(\sum_{S \subseteq [n]} \hat{f}(S)\chi_S\Big)\Big(\sum_{S \subseteq [n]} \hat{g}(S)\chi_S\Big) \\ &= \sum_{U \subseteq [n]} \Big(\sum_{S \Delta T = U} \hat{f}(S)\hat{g}(T)\Big)\chi_U, \end{aligned}$$

and therefore,

$$\begin{aligned} \|\widehat{fg}\|_1 &\leq \sum_{U \subseteq [n]} \Big|\sum_{S \Delta T = U} \hat{f}(S)\hat{g}(T)\Big| \\ &\leq \sum_{U \subseteq [n]} \sum_{S \Delta T = U} |\hat{f}(S)||\hat{g}(T)| \\ &= \sum_{S \subseteq [n]} \sum_{T \subseteq [n]} |\hat{f}(S)||\hat{g}(T)| \\ &= \Big(\sum_{S \subseteq [n]} |\hat{f}(S)|\Big)\Big(\sum_{S \subseteq [n]} |\hat{g}(S)|\Big) \\ &= \|\hat{f}\|_1 \|\hat{g}\|_1, \end{aligned}$$

as desired. ■

This lemma immediately implies $\|\hat{g}\|_1 \leq 1 + \|\hat{f}_1\|_1 \leq n + 1$, by the inductive hypothesis. The same occurs when $g_2(x_{n+1}) = \frac{1}{2} - \frac{x_1}{2}$. The other case is $g_1(x_{n+1}) = \frac{1}{2} + \frac{x_1}{2}$ and $g_2(x_{n+1}) = \frac{1}{2} - \frac{x_1}{2}$, in which case we can compute

$$\begin{aligned} g(x_1, \ldots, x_{n+1}) &= f(x_1, \ldots, x_n)(\frac{1}{2} + \frac{x_{n+1}}{2}) + (1 - f(x_1, \ldots, x_n))(\frac{1}{2} - \frac{x_{n+1}}{2}) \\ &= \frac{1}{2} - \frac{x_{n+1}}{2} + f(x_1, \ldots, x_n)x_{n+1}, \end{aligned}$$

in which case the lemma gives us the same bound, completing the induction. Therefore, all $f \in \mathcal{B}^{ob}_{2,n}$ *satisfy* $\|\hat{f}\|_1 \leq n$*, and so this class can be* ϵ*-fooled by an* ϵ/n*-biased space.*

Example 3 (Small Decision Trees)**.** *A special case of branching programs, a decision tree is a rooted binary tree representing a function* $\{\pm 1\}^n \to \mathbb{R}$ *where each internal node is labeled by a variable* x_i*, each outgoing edge is labeled* $\{-1, +1\}$ *and the leaves have function values. The decision tree* T *works by computation paths: on input* x*, if* T *is at an internal node with label* x_i*,* T *queries* x *at the* i*th location, then moves according to the edge with label* x_i*, until it reaches a leaf that gives the value of the function on this input. We stipulate that no path contains the same label more than once. We say the* size *of a decision tree is the number of leaves it has.*

It is an easy exercise to show that $\|\hat{f}\|_1 \leq \|f\|_\infty \cdot s$*, where* s *is the size of a decision tree computing* f *[5]. In particular, if* f *is a Boolean function, then* $\|\hat{f}\|_1 \leq s$*; as such for the set of all decision trees of polynomial size, for some fixed polynomial, can be fooled by a sufficiently good* ϵ*-biased space with seed length* $O(\log(n/\epsilon))$*.*

Example 4 (Concentrated Functions)**.** *We say a function* f *is* ϵ*-concentrated on* $\mathcal{S} \subseteq 2^{[n]}$ *if the sum of squares of the Fourier coefficients on* $2^{[n]} \setminus \mathcal{S}$ *is at most* ϵ*. It is easy to extend the above examples to show that the class of functions that is* ϵ*-concentrated on a class of subsets of polynomial size, for some fixed polynomial, can be fooled in an analogous way (for example, functions that have low degree in their Fourier expansion). We note here that this sort of spectral simplicity has implications in the polynomial-time learnability of functions [7].*

4 PRGs from Polarizing Random Walks

4.1 Intuition and Definitions

As these examples show, exploiting the Fourier spectrum can be quite fruitful and indeed, many analyses of candidate pseudorandom generators for a specific model of computation rely on delicate Fourier analysis. However, this discussion also suggests that one might be able to *directly* construct pseudorandom generators based on some Fourier property, not tethered to some form of computation.

In this section, we describe a recent approach from [1] that does exactly this. Let $\mathcal{F}$ be a class of Boolean functions on n variables that is closed under random restrictions (i.e for all $f \in \mathcal{F}$ and any fixing of any subset of variables in any input, the resulting function is in $\mathcal{F}$), and that satisfies the following L_1-Fourier tail bound: there exists constants $a, b \geq 1$ such that

$$\sum_{S \subseteq [n] : |S| = k} |\hat{f}(S)| \leq ab^k, \forall k \geq 1. \tag{12}$$

To fool such classes, one first defines a relaxation of pseudorandom generators:

Definition 4.1. Let $f : [-1, 1]^n \to [-1, 1]$ be a multilinear function. Then a random variable $X \in [-1, 1]^n$ is a *fractional pseudorandom generator* (FPRG) with error ϵ if

$$|\mathbb{E}_X[f(X)] - f(0)| \leq \epsilon. \tag{13}$$

X is a fractional pseudorandom generator for a class $\mathcal{F}$ with error ϵ if this holds for all $f \in \mathcal{F}$. If $X = G(U_r)$, where U_r is the uniform distribution on r bits, then we say X has seed length r.

In particular, when $\mathcal{F}$ is a class of n-variate Boolean functions, we consider the unique multilinear extension, just the Fourier expansion as explained above. Because fractional pseudorandom generators are not constrained to lie on corners of the Boolean hypercube, they are much easier to construct: in fact, just take $X \equiv 0$! In general, any X that is concentrated tightly around 0 will be a good fractional pseudorandom generator, just by continuity. If the end goal is to somehow use fractional generators to construct a bona fide pseudorandom generator, then we will need to somehow require that this does not happen using a variance condition:

Definition 4.2. A random variable $X = (X_1, \ldots, X_n)$ is p-noticeable if $\mathbb{E}_i[X_i^2] \geq p$ for all $i \in [n]$.

A good example of such a random variable is to take $G : \{\pm 1\}^r \to \{\pm 1\}^n$, and then define $X = pG(U_r)$. Then X has seed length r and is p^2-noticeable, as $X \in \{\pm p\}^n$.

4.2 Overview

To get from fractional pseudorandom generators to actual pseudorandom generators, [1] uses the following approach:

1. Start with a p-noticeable FPRG X for $\mathcal{F}$. By definition, $|\mathbb{E}_X[f(X)] - f(0)| \leq \epsilon$ for any $f \in \mathcal{F}$.
2. Add together independent copies of X to form steps in a random walk in $[-1, 1]^n$ to get a $1-q$-noticeable random variable, with the property that that these steps are small enough to not add much additional error, yet are large enough to require few steps to make the resulting random variable $1 - q$-noticeable.
3. Round to the nearest point $\{\pm 1\}^n$ and show that this does not add much error either when $1 - q$-noticeable.

4.3 The FPRG

To construct the FPRG, recall the intuition from above; by staying close to 0, one is able to fool $\mathcal{F}$, but we also want the FPRG to have variance so that it can serve as effective walk steps. This tension is where the Fourier tail bounds can be used: we will need the following construction.

Definition 4.3. A random variable $X \in \{\pm 1\}^n$ is ϵ-almost d-wise independent if any restriction of X to d coordinates has marginal distribution ϵ-close in statistical distance to the uniform distribution on d bits. In particular, $|\mathbb{E}_X[\prod_{i \in S} X_i]| \leq \epsilon$ for each $S \subseteq [n]$ with $|S| \leq d$.

We note that Naor and Naor give a construction that requires seed length $O(\log \log n + \log d + \log(1/\epsilon))$ [6]. We start with an ϵ-almost d-wise independent distribution, and then scale down by a factor which will be chosen according the b in the tail bounds. The original distribution will kill the small order terms in the Fourier expansion by independence, and the scaling will be multiplicative on higher order terms to kill those terms.

For convenience, we define the Fourier mass at level k:

Definition 4.4. The Fourier mass of f at level k is

$$W_k = \sum_{S \subseteq [n] : |S| = k} |\hat{f}(S)|. \tag{14}$$

Theorem 5 (Lemma 4.4 of [1]). *Fix $a, b \geq 1$. There exists a p-noticeable FPRG $X \in [-1,1]^n$ that fools any function with*

$$\sum_{S \subseteq [n]: |S| = k} |\hat{f}(S)| \leq ab^k, \forall k \geq 1, \tag{15}$$

with error ϵ, $p = \frac{1}{4b^2}$, and X has seed length $O(\log\log n + \log(a/\epsilon))$.

Proof. As mentioned above, we take an δ-almost d-wise independent distribution Z, where $d = \lceil \log(2a/\epsilon) \rceil, \delta = \epsilon/2a$, and $\beta = 1/2b$. We then construct $X = \beta Z$. The Naor and Naor construction gives the claimed seed length and by construction it is $1/(4p^2)$-noticeable. What remains to be checked is that it is a FPRG for these functions.

Using the calculation from Section 2, we have

$$\begin{aligned} |\mathbb{E}_X[f(X)] - f(0)| &\leq \sum_{\emptyset \neq S \subseteq [n]} |\hat{f}(S)||\mathbb{E}_X[\chi_S(X)]| \\ &= \sum_{\emptyset \neq S \subseteq [n]} \beta^{|S|} |\hat{f}(S)||\mathbb{E}_Z[\chi_S(Z)]| \end{aligned}$$

By definition, when $0 < |S| \leq d$, $|\mathbb{E}_Z[\chi_S(Z)]| \leq \delta$, and otherwise we can bound above by 1, so

$$\begin{aligned} |\mathbb{E}_X[f(X)] - f(0)| &\leq \sum_{\emptyset \neq S \subseteq [n]: |S| \leq d} \beta^{|S|} |\hat{f}(S)||\mathbb{E}_Z[\chi_S(Z)]| + \sum_{S \subseteq [n]: |S| > d} \beta^{|S|} |\hat{f}(S)||\mathbb{E}_Z[\chi_S(Z)]| \\ &\leq \delta \sum_{\emptyset \neq S \subseteq [n]: |S| \leq d} \beta^{|S|} |\hat{f}(S)| + \sum_{S \subseteq [n]: |S| > d} \beta^{|S|} |\hat{f}(S)||\mathbb{E}_Z[\chi_S(Z)]| \\ &\leq \delta \sum_{k=1}^{d} W_k \beta^k + \sum_{k=d+1}^{n} W_k \beta^k \\ &\leq a\delta \sum_{k=1}^{d} (b\beta)^k + a \sum_{k=d+1}^{n} (b\beta)^k \\ &\leq \frac{\epsilon}{2} + a2^{-d} \\ &\leq \epsilon, \end{aligned}$$

where we upper bound using formulas for infinite sums. ■

4.4 The Random Walk

To get a random walk that will quickly converge to the corners of the Boolean hypercube, the authors use the following construction:

Definition 4.5 (Random Walk Gadget). For $a_1, \ldots, a_t \in [-1,1]$, define $g_1(a_1) = a_1$, and then inductively define g_t for $t > 1$ by

$$g_t(a_1, \ldots, a_t) = g_{t-1}(a_1, \ldots, a_{t-1}) + (1 - |g_{t-1}(a_1, \ldots, a_{t-1})|)a_t. \tag{16}$$

Extend this to definition to vectors by doing the construction componentwise: that is, define $g_t^n : ([-1,1]^n)^t \to [-1,1]^n$ by

$$g_t^n(x_1, \ldots, x_t) = (g_t(x_{1,1}, \ldots, x_{t,1}), \ldots, g_t(x_{1,n}, \ldots, x_{t,n})). \tag{17}$$

In easier notation, if we generate independent FPRGs $X_1, \ldots, X_t \in [-1,1]^n$ and use them as steps in our walk, then we get the corresponding random variables $Y_1, \ldots, Y_t$ defined by

$$Y_1 = X_1 \tag{18}$$

$$Y_t = Y_{t-1} + \delta_{Y_{t-1}} \circ X_t, \tag{19}$$

where $\delta_y = (1 - |y_1|, \ldots, 1 - |y_n|)$ and $x \circ y$ is defined by $x \circ y = (x_1 y_1, \ldots, x_n, y_n)$. Geometrically, the quantity $1 - |(Y_{t-1})_i|$ gives the distance from $(Y_{t-1})_i$ to the closer of ± 1; in that sense, what this walk is doing physically is adding together the X_i, but scaling according to the distance from Y_{i-1} to the corners, which has the benefit of constricting the random walk to $[-1,1]^n$.

This construction is particularly natural in the following sense; we know that taking the first step $Y_1 = X_1$ does not lead to much error, because by definition of FPRGs, we have $|\mathbb{E}_{Y_1}[f(Y_1)] - f(0)| \leq \epsilon$. But what about the next step Y_2? By the Triangle Inequality, it will suffice to prove that $|\mathbb{E}_{X_1}[f(X_1)] - \mathbb{E}_{X_1,X_2}[g_2^n(X_1, X_2))]| \leq \epsilon$ as well. The way to do this will be to show that in the same way X_1 fools f near 0, the random step centered at any y will fool f near y.

Lemma 6 (Claim 3.3 of [1])**.** *Suppose $X \in [-1,1]^n$ is an FPRG for $\mathcal{F}$ with error ϵ. Then for any $f \in \mathcal{F}$ and $y \in [-1,1]^n$,*

$$|f(y) - \mathbb{E}[f(y + \delta_y \circ X)]| \leq \epsilon. \tag{20}$$

Proof. Note that the case when $y = 0$ is exactly the condition of being a FPRG. The intuition is the following: we have already seen that $f(y)$ has a natural interpretation as the expectation of f on certain random inputs. To get a similar bound when translating from 0 to y, we will want write $f(y)$ instead as $F(0)$ for an *random restriction* F of f; because $\mathcal{F}$ is closed under random restrictions, the fact that X is a PRG for $\mathcal{F}$ will imply the bound if this is done carefully.

The most natural way to enforce this is define a random function F whose value at x is the value of f on random inputs. Explicitly, define the random input $R(x)$ by sampling $R_1, \ldots R_n$ independently by $\Pr[R_i = \operatorname{sgn}(y_i)] = |y_i|$ and $\Pr[R_i = x_i] = 1 - |y_i|$. We then define $F(x) = f(R(x))$. From this definition, we get

$$\mathbb{E}_{R_i}[R_i] = \operatorname{sgn}(y_i)|y_i| + (1 - |y_i|)x_i = y_i + \delta_{y_i} \circ x_i; \tag{21}$$

by multilinearity, we thus have $\mathbb{E}_F[F(x)] = \mathbb{E}_R[f(R(x))] = f(\mathbb{E}_R[R(x)]) = f(y + \delta_y \circ x)$. In particular, we have $\mathbb{E}[F(0)] = f(y)$, exactly as we wanted. As a result, we have

$$|f(y) - \mathbb{E}[f(y + \delta_y \circ X)]| = |\mathbb{E}_F[F(0)] - \mathbb{E}_{F,X}[F(X)]| \leq \mathbb{E}_F[|F(0) - \mathbb{E}_X[F(X)]|]. \tag{22}$$

As we stated, though, as F is a random restriction, it is in $\mathcal{F}$ almost surely, and therefore, this term is bounded by ϵ by the fact that X is a FPRG for $\mathcal{F}$, as desired. ■

From the proof, we thus see that the steps in the random walk are naturally constructed when trying to construct random inputs whose expectation at 0 will be y. From this claim, we easily get the actual expected error analysis of the walk:

Lemma 7 (Claim 3.4 of [1])**.** *Suppose $X_1, \ldots, X_t \in [-1,1]^n$ are independent FPRGs for $\mathcal{F}$ with error ϵ. Then for any $f \in \mathcal{F}$,*

$$|\mathbb{E}_{X_1,\ldots,X_t}[f(g_t^n(X_1, \ldots, X_t))] - f(0)| \leq t\epsilon. \tag{23}$$

Proof. By writing this as a telescoping series and using the triangle inequality, we have

$$|\mathbb{E}_{X_1,\ldots,X_t}[f(g_t^n(X_1,\ldots,X_t))]-f(0)| \leq \sum_{i=1}^{t} |\mathbb{E}_{X_1,\ldots,X_i}[f(g_i^n(X_1,\ldots,X_i))]-\mathbb{E}_{X_1,\ldots,X_{i-1}}[f(g_{i-1}^n(X_1,\ldots,X_{i-1}))]|, \tag{24}$$

where we make the convention that $g_0 = 0$ as a function. We claim that each term is bounded above by ϵ, which implies the lemma.

This can be done by induction. The case that $i = 1$ is just the fact that X_1 is a FPRG. For $i > 1$, notice that

$$|\mathbb{E}_{X_1,\ldots,X_i}[f(g_i^n(X_1,\ldots,X_i))]-\mathbb{E}_{X_1,\ldots,X_{i-1}}[f(g_{i-1}^n(X_1,\ldots,X_{i-1}))]| \leq \\ \mathbb{E}_{X_1,\ldots,X_{i-1}}[|f(g_{i-1}^n(X_1,\ldots,X_{i-1}))] - \mathbb{E}_{X_i}[f(g_i^n(X_1,\ldots,X_i))]|].$$

By monotonicity, it thus suffices to show that for any $x_1,\ldots,x_{i-1} \in [-1,1]^n$, we have

$$|f(g_{i-1}^n(x_1,\ldots,x_{i-1}))] - \mathbb{E}_{X_i}[f(g_i^n(x_1,\ldots,x_{i-1},X_i))]|, \tag{25}$$

but this follows from Lemma 6.

■

In particular, this random walk only gives additive error on each step; if one can bound how many steps are needed before rounding, and then quantify how much error the rounding gets, one would get an error estimate for the resulting PRG. For the former, the analysis has nothing to do with FPRGs, but rather just martingale concentration:

Lemma 8 (Claim 3.5 of [1])**.** *Let $A_1,\ldots,A_t \in [-1,1]$ be symmetric, independent, p-noticeable random variables, and define $B_i = B_{i-1} + (1-|B_{i-1}|)A_i$. Then B_t is $1-3\exp(-tp/16)$-noticeable.*

Proof. The proof proceeds by considering the sequence of distances from $\{\pm 1\}$, that is $C_i := 1 - |B_i|$. One can show that $C_i \leq C_{i-1}A_i$, so by induction, $\sqrt{C_t} \leq \prod_{i=1}^{t}\sqrt{1-A_i}$. Taking expectations and using independence,

$$\mathbb{E}[\sqrt{C_t}] \leq \prod_{i=1}^{t}\mathbb{E}[\sqrt{1-A_i}]. \tag{26}$$

Because the A_i are identical, it suffices to bound a single term on the right. The Taylor expansion of $\sqrt{1-x}$ is $1-\frac{x}{2}-\frac{x^2}{8}-O(x^3)$, where each coefficient but the first is negative. Taking expectations and using symmetry of A_i to kill the odd moments, we deduce that

$$\mathbb{E}[\sqrt{1-A_i}] \leq 1 - \frac{\mathbb{E}[A_i^2]}{2} \leq 1 - \frac{p}{8} \leq \exp(-p/8), \tag{27}$$

using the bound $1 - x \leq e^{-x}$. Therefore,

$$\mathbb{E}[\sqrt{C_t}] \leq \exp(-pt/8). \tag{28}$$

Markov's inequality implies that

$$\Pr[C_t \geq \exp(-tp/8)] = \Pr[\sqrt{C_t} \geq \exp(-tp/16)] \leq \frac{\exp(-pt/8)}{\exp(-tp/16)} = \exp(-pt/16). \tag{29}$$

When $C_t \leq \exp(-tp/8) \leq \exp(-tp/16)$, using the inequality $1 - x^2 \leq 2 - 2|x|$ on the interval $[-1,1]$, we thus have $1 - B_t^2 \leq 2 - 2|B_t| = 2C_t \leq 2\exp(-pt/16)$, and otherwise, we can trivially bound $1 - B_t^2$ by 1. Therefore,

$$\mathbb{E}[1-B_t^2] \leq \mathbb{E}[1-B_t^2|C_t \leq \exp(-tp/8)] + (1)\Pr[C_t \geq \exp(-tp/8)] \leq 3\exp(-tp/16), \tag{30}$$

and rearranging gives the claim. ■

4.5 Error from Rounding

The final step is to show that rounding a $1-q$-noticeable random variable according to the sign of each coordinate does not add too much error. To do this, suppose that X is a $1-q$-noticeable random variable in $[-1,1]^n$. The proof proceeds by considering how much error comes from rounding any point $x \in [-1,1]$ in terms of x, and then averaging.

Let $W = (W_1, \ldots, W_n)$ be a random variable in $\{\pm 1\}$ such that $\mathbb{E}_W[W] = x$, as described before. Then as f is bounded in $[-1,1]$ on the cube, we have by multilinearity

$$|f(x) - f(\operatorname{sgn}(x))| = |\mathbb{E}_W[f(W)] - f(\operatorname{sgn}(x))| \leq 2\Pr[W \neq \operatorname{sgn}(x)], \tag{31}$$

as conditioned on $W = \operatorname{sgn}(x)$, the deviation is zero, and otherwise we can trivially upper bound by 2. By a union bound, this probability is bounded above by the sum over the probability for each coordinate, which is just $\frac{1-|x_i|}{2}$. As such, we have

$$\begin{aligned} |f(x) - f(\operatorname{sgn}(x))| &\leq 2\Pr[W \neq \operatorname{sgn}(x)] \\ &\leq 2\sum_{i=1}^{n} \Pr[W_i \neq \operatorname{sgn}(x_i)] \\ &\leq \sum_{i=1}^{n} 1 - |x_i|. \end{aligned}$$

By averaging over $x = X$, we get $|\mathbb{E}_X[f(X)] - \mathbb{E}_X[f(\operatorname{sgn}(X))]| \leq \sum_{i=1}^{n} \mathbb{E}_X[1 - |X_i|]$. Note that $1 - |x| \leq 1 - x^2$ on $[-1,1]$, and so we get

$$|\mathbb{E}_X[f(X)] - \mathbb{E}_X[f(\operatorname{sgn}(X))]| \leq \sum_{i=1}^{n} \mathbb{E}_X[1 - X_i^2] \leq qn, \tag{32}$$

by $1-q$-noticeability. Therefore, if X is an FPRG with error ϵ, and if $Z = \operatorname{sgn}(X)$, Z is a PRG with error

$$|f(0) - \mathbb{E}_Z[f(Z)]| \leq |f(0) - \mathbb{E}_X[f(X)]| + |\mathbb{E}_X[f(X)] - \mathbb{E}_X[f(\operatorname{sgn}(X))]| \leq \epsilon + qn. \tag{33}$$

4.6 Completing the Construction

We can finally put this together to construct a PRG for L_1-Fourier tail bounds:

Theorem 9 (Theorem 4.5 of [1])**.** *Let $\mathcal{F}$ be a family of n-variate Boolean functions closed under restriction satisfying*

$$\sum_{S \subseteq [n] : |S| = k} |\hat{f}(S)| \leq ab^k, \forall k \geq 1, \forall f \in \mathcal{F}. \tag{34}$$

Then for any error $\epsilon > 0$, there exists an explicit PRG $X \in \{\pm 1\}$ that ϵ-fools $\mathcal{F}$ with seed length $O(\log(n/\epsilon)(\log\log n + \log(ab/\epsilon)))$.

Proof. By Theorem 5, we will start with an explicit $1/(4b^2)$-noticeable FPRG for $\mathcal{F}$ (and determine seed length after checking parameters later). By Lemma 8, after taking $O(\log(n/\epsilon)b^2)$ copies of this FPRG (multiplying by a single uniform bit to guarantee symmetry), we have amplified this to a $1-q$-noticeable FPRG, where $q = 2\epsilon/n$; rounding this induces $\epsilon/2$ error as we saw above. Therefore, in the construction of the FPRG, we require it have error $\epsilon/(2O(\log(n/\epsilon)b^2))$ by Lemma 7, and therefore the Naor and Naor construction requires seed of size $O(\log\log n + \log(ab^2\log(n/\epsilon)/\epsilon)) = O(\log\log n + \log(ab/\epsilon))$, so in total the seed length $O(b^2\log(n/\epsilon)(\log\log n + \log(ab/\epsilon)))$ as claimed. ■

Pseudorandom Generators for Model Classes		
Function Class $\mathcal{F}$	b Tail Bound	PRG Seed Length
Functions with sensitivity $\leq s$	$O(s)$ [8]	$O(s^2 \log(n/\epsilon)(\log\log(n) + \log(s) + \log(1/\epsilon)))$
OROBPs of width w	$\log^w n$ [9]	$O(\log^{2w} n \log(n/\epsilon)(w \log\log(n) + \log(1/\epsilon)))$
AC^0 size $\leq s$ and depth $\leq d$	$2^{O(d)} \log^{d-1} s$ [10]	$O((\log^{2d-2} s)(\log(n/\epsilon))(\log\log(n) + d \log\log s + \log(1/\epsilon)))$

Figure 1: PRG constructions using known Fourier tail bounds.

Remark 10. *Note that when $b = \log^{O(1)} n$, then the $\log b$ term can be safely dropped.*

Many known classes of functions satisfy these Fourier tail bounds and are closed under restriction, so by plugging in the tail bounds, we can immediately plug in values for Theorem 9, which yields comparable generators to the state-of-the-art. See Figure 1. Crucially, this same construction holds across function classes; the construction is catered to the Fourier tail bounds, not to any particular model class.

5 Extensions

We now discuss some recent applications of these Fourier-analytic techniques, in particular in the recent oracle separation of BQP, the class of languages computed by bounded-error, polynomial-time quantum Turing machines, and PH, the polynomial hierarchy, by Raz and Tal [2].

5.1 Oracle Separation of BQP and PH

Through known reductions, the heart of the proof of the result is the construction of a distribution $\mathcal{D}$ such that a quantum algorithm can efficiently distinguish between $\mathcal{D}$ and the uniform distribution with noticeable advantage, whereas no Boolean circuit of small enough size and depth can gain an appreciable advantage. Formally, they prove that

Theorem 11 (Theorem 1.1 of [2])**.** *There exists an explicit distribution $\mathcal{D}$ on $\{\pm 1\}^{2N}$ such that*

1. *No Boolean circuit of quasipoly(N) size and constant depth can distinguish between $\mathcal{D}$ and U with advantage better than polylog(N)$/\sqrt{N}$.*
2. *There exists a quantum algorithm making one query to the input that runs in $O(\log N)$ time which distinguishes between $\mathcal{D}$ and U with advantage $\Omega(1/\log N)$.*

We sketch the proof of the first part of this theorem, as only the former relies on these Fourier techniques, and the latter follows from other known results beyond the scope of this paper. In this section, we define the distribution and sketch the key technical contribution of Raz and Tal, the circuit lower bound, using the techniques of the previous section.

5.1.1 The Distribution $\mathcal{D}$

Let $n \in \mathbb{N}$ and $N := 2^n$. The distribution $\mathcal{D}$ is constructed in the following manner by going through intermediate distributions and then rounding:

1. Sample $x_1, \ldots, x_N$ i.i.d. from a standard normal distribution $N(0,1)$. Let $x = (x_1, \ldots, x_N)$.

2. Define $y = H_N x$, where H_N is the Hadamard matrix defined by $(H_N)_{i,j} = (-1)^{\langle \lfloor i-1 \rfloor, \lfloor j-1 \rfloor \rangle}$, where $\lfloor m \rfloor$ denotes the binary representation of m and we take inner products. It is well-known that H_N is a symmetric, orthonormal matrix.

3. Define $z = (x, y)$ and note then that z is a zero-mean multivariate normal distribution with covariance matrix
$$\begin{pmatrix} I & H_N \\ H_N & I \end{pmatrix}. \tag{35}$$
Denote this intermediate distribution $\mathcal{G}$.

4. Set $\epsilon = 1/(24 \ln N)$. Sample $z \sim \mathcal{G}$ and output $\sqrt{\epsilon} z$. Denote this distribution $\mathcal{G}'$, and note this is just a scaled version of $\mathcal{G}$ with covariance matrix scaled by $(\sqrt{\epsilon})^2 = \epsilon$.

5. Define $\mathrm{trnc}(a) = \mathrm{proj}_{[-1,1]}(a) = \min(1, \max(-1, a))$ to be the projection of any number to the closest point in $[-1, 1]$. To get $\mathcal{D}$, sample $z \sim \mathcal{G}'$, take $\mathrm{trnc}(z)$ (considered componentwise), and then sample as follows: independently for each $i \in [2N]$, draw $z'_i = 1$ with probability $\frac{1+\mathrm{trnc}(z_i)}{2}$ and $z'_i = -1$ with probability $\frac{1-\mathrm{trnc}(z_i)}{2}$. Finally we output z' and denote this distribution by $\mathcal{D}$, noting that this takes values in $\{\pm 1\}$.

To gain some intuition behind these distributions, note that if f is any multilinear function, then by the same analysis as before, because $\mathcal{D}$ is an appropriately, independently rounded version of $\mathcal{G}'$,
$$\mathbb{E}_{z \sim \mathcal{G}'}[f(\mathrm{trnc}(z))] = \mathbb{E}_{z \sim \mathcal{D}}[f(z)]. \tag{36}$$
Moreover, because ϵ is quite small, the probability that $\mathrm{trnc}(z) \neq z$ is quite small, so we will also have
$$\mathbb{E}_{z \sim \mathcal{G}'}[f(\mathrm{trnc}(z))] \approx \mathbb{E}_{z \sim \mathcal{G}'}[f(z)]. \tag{37}$$
In particular, this means that it essentially suffices to show that $\mathcal{G}'$ fools bounded size circuits to show that $\mathcal{D}$ does so. The technique to do this will be entirely analogous to the analysis from before: one writes $\mathcal{G}'$ as a sum of many small parts of scaled down copies of $\mathcal{G}'$, by the fact the sum of Gaussians is still Gaussian. On each small "step," a very similar analysis to the above can be used to show that the higher Fourier coefficients and Gaussian moments will be killed by the scaling, while the Gaussians themselves dispense with the lower order terms. After adding these small parts together, the total error will remain quite small. Namely, Raz and Tal consider the distribution $\mathcal{G}'$ by setting $t = N$, $p = 1/\sqrt{t}$, and then define the intermediate steps $z^{\leq i} = p \sum_{j=1}^{i} z^j$. Then $z^{\leq t} \sim \mathcal{G}'$, and each individual $z^{\leq i}$ is just an intermediate step towards $z^{\leq t}$, in a completely analogous way as before.

To be more formal, using a careful analysis, they take the following steps. First, they show that adding small, scaled versions of $z \sim \mathcal{G}'$ do not change the expectation of a multilinear function near certain points z_0 near the origin whether or not we take the truncation, as mentioned before:

Lemma 12 (Claim 5.3 of [2])**.** *Suppose $0 \leq p, p_0$ have $p + p_0 \leq 1$ and suppose $f : \mathbb{R}^{2N} \to \mathbb{R}$ is multilinear mapping $\{\pm 1\} \to [-1, 1]$. Then if $z_0 \in [-p_0, p_0]^{2N}$,*
$$\mathbb{E}_{z \sim \mathcal{G}'}[|f(trnc(z_0 + p \cdot z)) - f(z_0 + p \cdot z)|] \leq 8N^{-1/2}. \tag{38}$$

From this point, their construction uses the following known Fourier tail bounds by Tal for bounded Boolean circuits:

Lemma 13 (Theorem 37 of [10]). *There exists $c > 0$ such that if $A : \{\pm 1\}^{2N} \to \{\pm 1\}$ is computable by a Boolean circuit with at most s gates and depth at most d, then for all $k \geq 0$,*

$$\sum_{S \subseteq [2N] : |S| = k} |\hat{A}(S)| \leq (c \log s)^{(d-1)k}. \tag{39}$$

Note that these are precisely the tail bounds discussed before, with $a = 1$ and $b = (c \log s)^{d-1}$.

Using these tail bounds and bounds on the moments of the Gaussian distribution (using Isserlis' Theorem and the covariance matrix), they are then able to prove a result that is entirely analogous to Lemma 7 from above: first, they show that taking a small step of $\mathcal{G}'$ near the origin will fool a circuit as above (exactly analogous to Theorem 5) by killing higher Fourier and Gaussian moment. Then they show that using the random restriction argument from Lemma 6 and the fact that the restriction of Boolean circuits are Boolean circuits of smaller size, we can iterate this argument using the intermediate distributions $z^{\leq i}$ and Lemma 12 from above to deduce the main technical circuit lower bound:

Theorem 14 (Theorem 1.1, Part 1 of [2]). *Let $A : \{\pm 1\}^{2N} \to \{\pm 1\}$ be a Boolean circuit of size s and depth d. Then*

$$|\mathbb{E}_{z' \sim \mathcal{D}}[A(z')] - A(0)| = |\mathbb{E}_{z' \sim \mathcal{D}}[A(z')] - \mathbb{E}_U[]A(U)]| \leq 32\epsilon (c \log s)^{2(d-1)} / \sqrt{N}, \tag{40}$$

where $\epsilon = 1/(24 \ln N)$ as before and c is from Lemma 14. In particular, if A is of size $\exp(\log^{O(1)}(N))$ and constant depth, then $|\mathbb{E}_{z \sim \mathcal{D}}[A(z)] - \mathbb{E}_U[A(U)]| \leq$ polylog$(N)/\sqrt{N}$.

In that way, the proof idea relies heavily on the Fourier analysis from before: first, use the Fourier expansion to think of Boolean functions as multilinear functions, then think of functions evaluated on points inside the cube $[-1, 1]^n$ as random restrictions over functions. As we showed, the benefit of doing this means that using Fourier tails, it is possible to show the error by perturbing any point with a small step using a good distribution is small, precisely because the small step corresponds to a scaling that kills off the higher Fourier terms. In [1], the goal was to add these small steps to get a good distribution that approximates uniform according to these functions with bounded Fourier tails, while in [2], the end goal is to find rewrite a distribution into smaller copies of itself to facilitate the analysis of hardness.

5.2 PRGs from Level Two Bounds

These ideas have also cycled back to their original motivation, the construction of pseudorandom generators. In particular, using a similar distribution as in the Raz and Tal result, the authors of [3] were able to construct pseudorandom generators that only require bounds on the second Fourier level, not the entire tail, at the risk of worse dependence on $1/\epsilon$. Their construction essentially derandomizes the Raz and Tal distribution by using code words to reduce the dimension, and so the seed length, of the Raz and Tal Gaussian construction, and then use a known result by [Kane15] that constructs approximate Gaussian distributions with small error in expectation. After this construction, the analysis from before that converts FPRGs to PRGs yields the following theorem that only relies on level 2 Fourier bounds:

Theorem 15 (Theorem 2.1 of [3]). *Let $\mathcal{F}$ be a family of Boolean functions on n-variables that is closed under restrictions. Then if there exists $t \geq 1$ such that*

$$\sum_{S \subseteq [n] : |S| = 2} |\hat{f}(S)| \leq t, \quad \forall f \in \mathcal{F}, \tag{41}$$

then for all $\epsilon > 0$, there exists an explicit pseudorandom generator for $\mathcal{F}$ with error ϵ with seed length $O((t/\epsilon)^{2+o(1)} polylog(n))$.

6 Conclusion

In this paper, we have discussed at length a new idea from [1] that proposes the analysis of Fourier tails to construct flexible and robust pseudorandom generators for a wide class of functions. Moreover, this analysis has proven useful in other regards, for as we sketched, these same exact ideas have recently lead to a circuit lower bound that implies the existence of an oracle $\mathcal{O}$ such that $\mathsf{BQP}^{\mathcal{O}} \not\subset \mathsf{PH}^{\mathcal{O}}$. As these ideas are quite new, it remains open to understanding how much more one can extract from them; for instance, are there better ways to put together FPRGs in a walk to get faster convergence? How much independence is needed? Understanding these questions, as well as studying better Fourier tail bounds for many natural classes of functions, will be crucial in determining how optimal this new framework will be.

References

[1] E. Chattopadhyay, P. Hatami, K. Hosseini, and S. Lovett, "Pseudorandom generators from polarizing random walks," in *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, pp. 1:1–1:21, 2018.

[2] R. Raz and A. Tal, "Oracle separation of BQP and PH," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 25, p. 107, 2018.

[3] E. Chattopadhyay, P. Hatami, S. Lovett, and A. Tal, "Pseudorandom generators from the second fourier level and applications to AC0 with parity gates," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 25, p. 155, 2018.

[4] N. Nisan, "Pseudorandom generators for space-bounded computation," *Combinatorica*, vol. 12, no. 4, pp. 449–461, 1992.

[5] R. O'Donnell, *Analysis of Boolean Functions*. New York, NY, USA: Cambridge University Press, 2014.

[6] J. Naor and M. Naor, "Small-bias probability spaces: Efficient constructions and applications," *SIAM journal on computing*, vol. 22, no. 4, pp. 838–856, 1993.

[7] E. Kushilevitz and Y. Mansour, "Learning decision trees using the fourier spectrum," *SIAM Journal on Computing*, vol. 22, no. 6, pp. 1331–1348, 1993.

[8] P. Gopalan, R. A. Servedio, and A. Wigderson, "Degree and sensitivity: Tails of two distributions," in *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pp. 13:1–13:23, 2016.

[9] E. Chattopadhyay, P. Hatami, O. Reingold, and A. Tal, "Improved pseudorandomness for unordered branching programs through local monotonicity," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 24, p. 171, 2017.

[10] A. Tal, "Tight bounds on the fourier spectrum of AC0," in *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pp. 15:1–15:31, 2017.