Analysis of Boolean Functions: Foundations and Applications to TCS

Avishay Tal (UC Berkeley)

Lectures 3 & 4

Lectures 3 & 4

Main character: **Fourier Growth** – a complexity measure for Boolean functions that captures the ability to ***aggregate weak k-wise correlations in the input.***

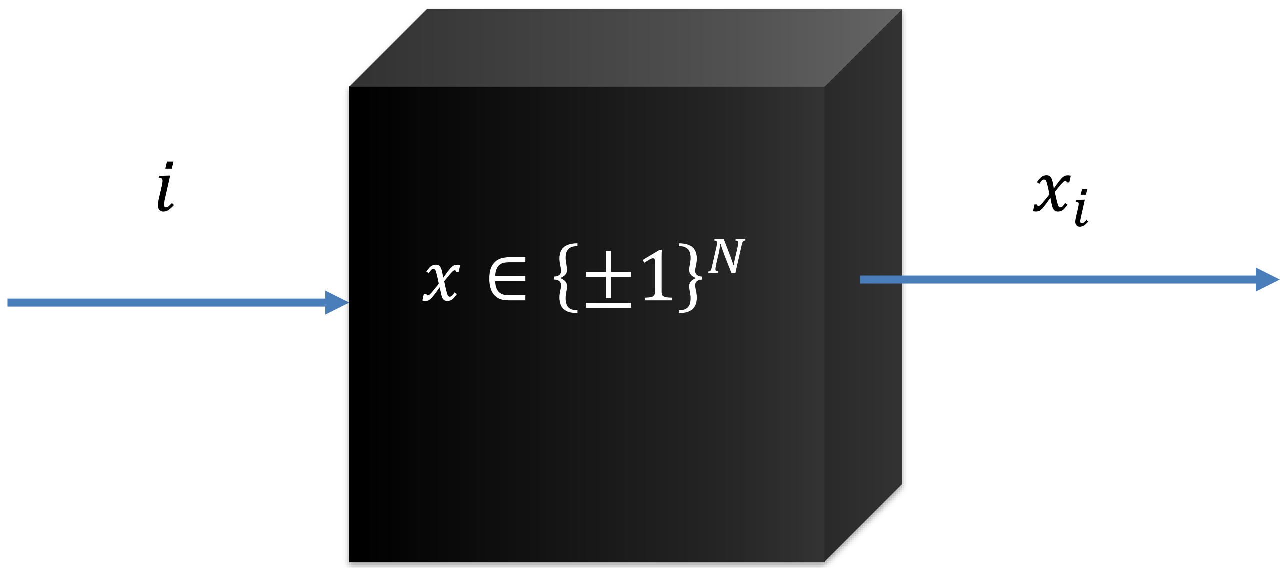
Applications:

1. **Quantum Advantage**
2. **Pseudo-randomness**
3. **Lower Bounds**

Quantum Advantage:

For which tasks do **quantum** algorithms provably outperform **classical** algorithms?

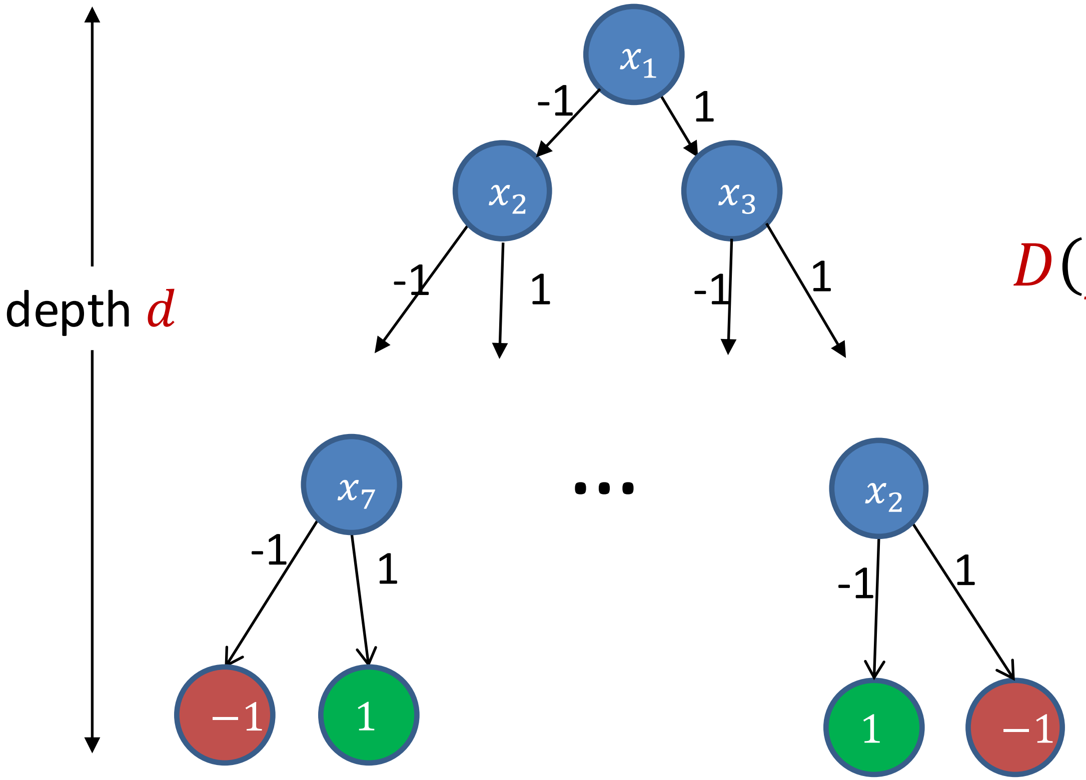
The Black-Box / Query Model



Typical Question: Does the black-box satisfy a property or not?

Query Complexity: How many queries (possibly adaptive) are needed to determine the property?

The Decision Tree Model



$f:\{-1,1\}^N \rightarrow \{-1,1\}$

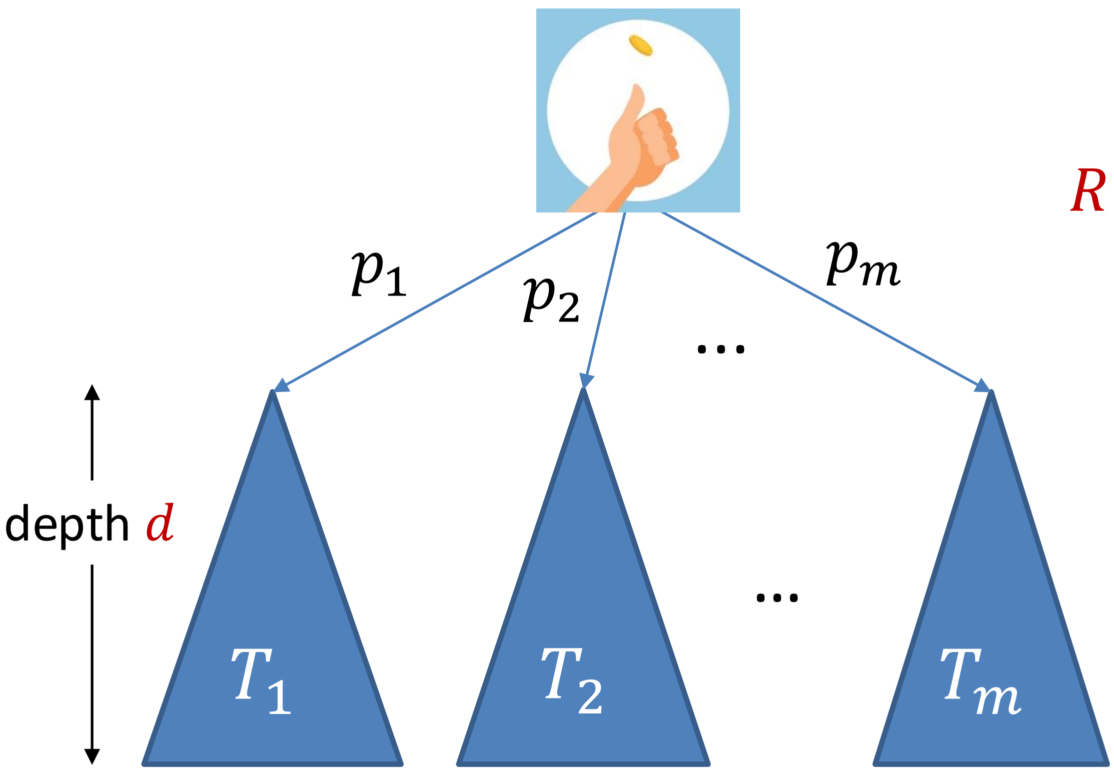
$D(f) =$ minimal depth of a decision tree computing f

$=$ deterministic query complexity of f

The Randomized Decision Tree Model

Randomized decision tree of depth d: a distribution over deterministic decision trees of depth at most d.

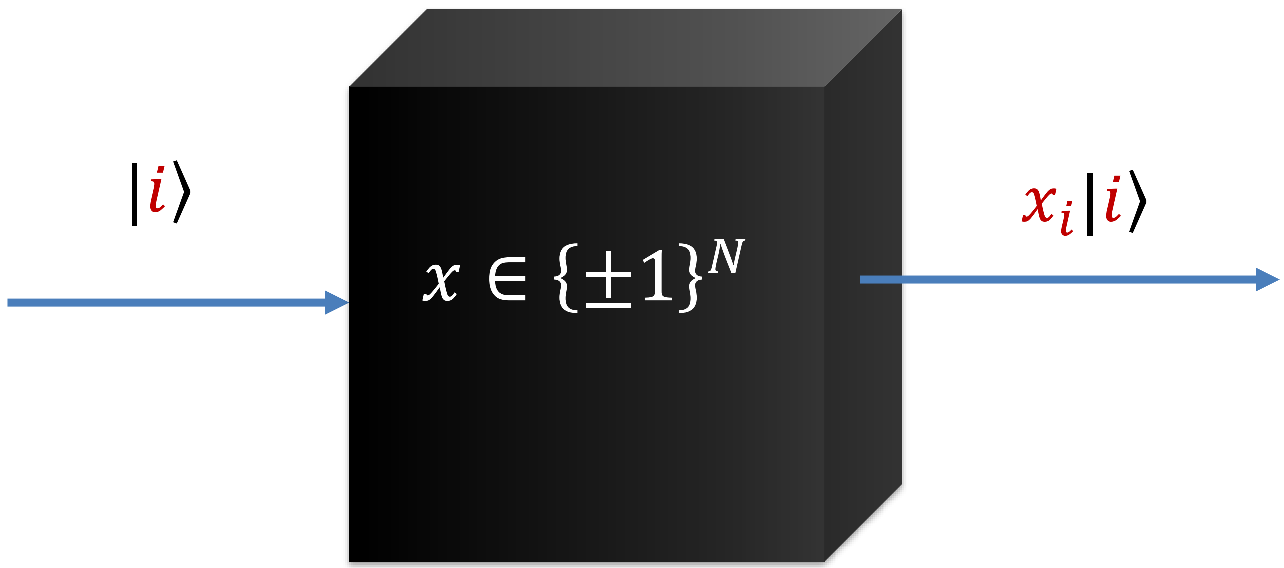
We say that a randomized decision tree computes f if its output equals $f(x)$ with probability at least $\frac{2}{3}$ for all $x \in \{-1,1\}^N$



$R(f) =$ minimal depth of a randomized decision tree computing f

$=$ randomized query complexity of f

Quantum Query Complexity



A query to the input applies the unitary transformation O_x that maps

$$|i\rangle \rightarrow x_i|i\rangle$$

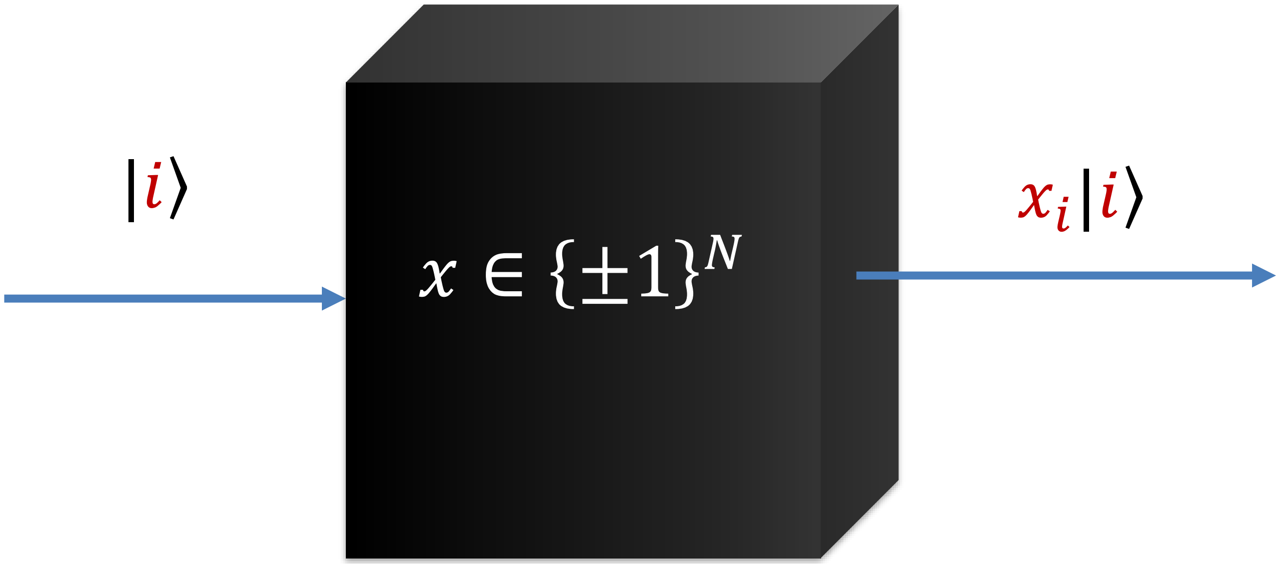
A t-query quantum algorithm applies

$$U_{t+1}\ O_x\ U_t \quad \dots \quad O_x\ U_3\ O_x\ U_2\ O_x\ U_1\ |0\rangle$$

where $U_1, \dots, U_{t+1}$ are unitary transformations that do not depend on x.

Finally: measure the state ➔ accept/reject based on outcome.

Quantum Query Complexity

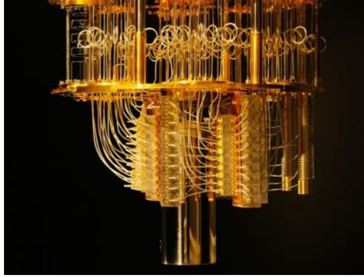


We say that a quantum query algorithm computes f if its output equals $f(x)$ with probability at least $\frac{2}{3}$ for all $x \in \{-1,1\}^N$

$Q(f) =$ minimal number of queries of a quantum query algorithm computing f

$=$ quantum query complexity of f

Quantum Advantage in Query Model



vs

Are quantum algorithms superior to randomized (or deterministic) algorithms in the query model?

[Grover'96]: **Quadratic speed-up**

[Aaronson, Ben-David, Kothari'16, T'20, Bansal, Sinha'21, Sherstov, Storozhenko, Wu'21]: **Super-quadratic speed-ups!**
Constructed a **total** function f_{cs} with $R(f_{cs}) \geq \widetilde{\Omega}(Q(f_{cs})^3)$

[Beals, Buhrman, Cleve, Mosca, de Wolf'98, Aaronson, Ben-David, Kothari, Rao, T' 21]:
For **total** functions $f\colon\{\pm 1\}^N \to \{\pm 1\}$ **at most polynomial speed-ups**:

$$R(f) \leq D(f) \leq O(Q(f)^4)$$

For **partial** functions $f\colon A \to \{-1,1\}$, $A \subseteq \{-1,1\}^N$ exponential separations exist [Simon'94, Shor'94, Childs, Cleve, Deotto, Farhi, Gutmann, Spielman'03, Aarondson, Ambainis'15], e.g. $Q(f) = O(1), R(f) = \sqrt{N}$

Motivation:

Identify a property that separates quantum from classical (query) algorithms

Recall: Discrete Fourier Analysis 101

The Fourier transform of a Boolean function f naturally defines a distribution D_f over sets $S \subseteq \{1, \dots, n\}$:

The probability to sample S from D_f equals $\hat{f}(S)^2$.

Denote by $\mathbf{W}^k[f] = \Pr_{S \sim D_f}[|S| = k] = \sum_{S:|S|=k} \hat{f}(S)^2$

Fourier Weight

Denote by $\mathbf{W}^{\geq k}[f] = \Pr_{S \sim D_f}[|S| \geq k] = \sum_{S:|S|\geq k} \hat{f}(S)^2$

Fourier Tail

Tails and Low-Degree Approximation Equivalence

Let $f: \{-1,1\}^n \to \mathbb{R}$. The truncated Fourier expansion of f at level k is a degree k polynomial defined by

$$f^{\leq k}(x) = \sum_{S:|S|\leq k} \hat{f}(S) \cdot \prod_{i \in S} x_i$$

By Parseval: $\mathbf{E}_x\left[\left(f(x) - f^{\leq k}(x)\right)^2\right] = \mathbf{W}^{>k}[f]$.

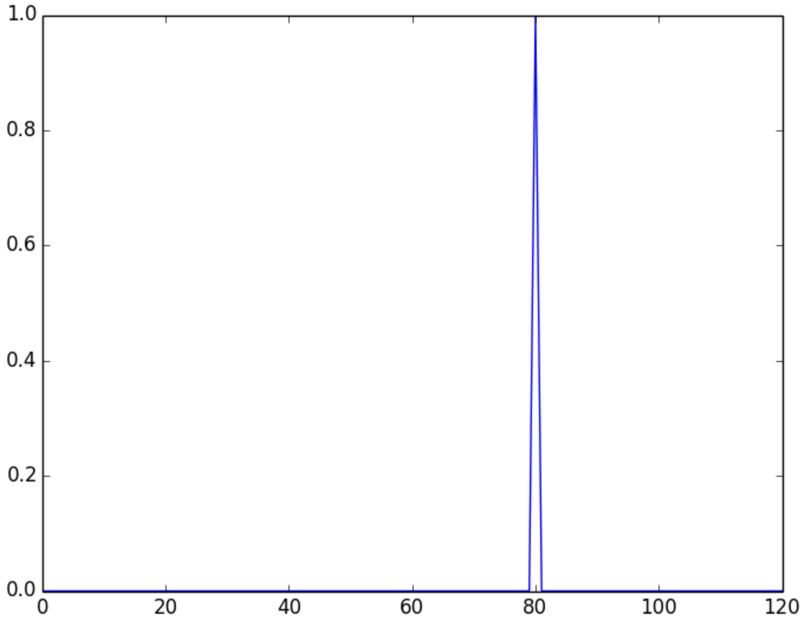
By Parseval: this is the best $\mathbf{L_2}$-approx. of f among degree k polys.

f has a degree-k $\mathbf{L_2}$-approximation with error ε **iff** $\mathbf{W}^{>k}[f] \leq \varepsilon$

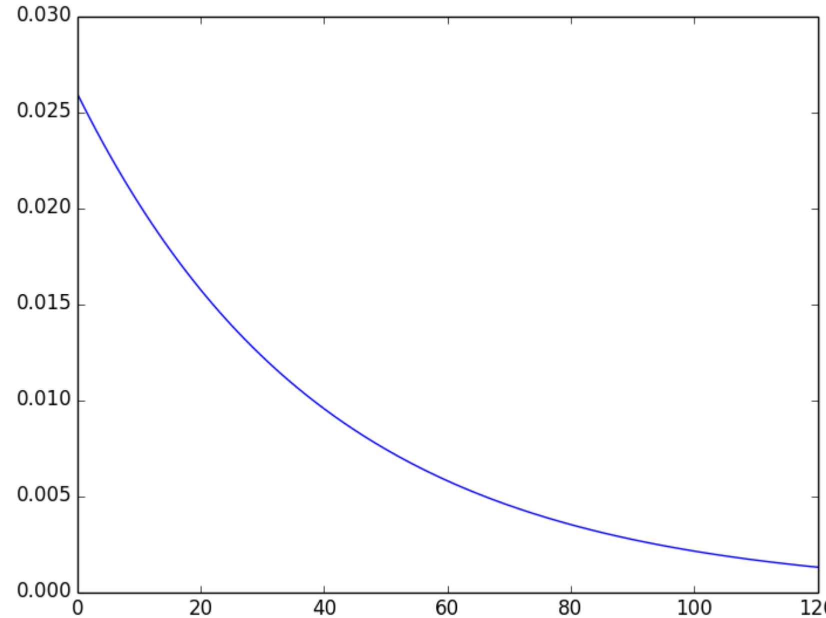
$$\text{Parity}(x_1, \ldots, x_n) = x_1 \cdot x_2 \cdot \ldots \cdot x_n$$

$$\text{Majority}(x_1, \ldots, x_n) = \text{sign}\left(\sum_i x_i\right)$$

$\mathbf{W}^k[\text{Parity}]$



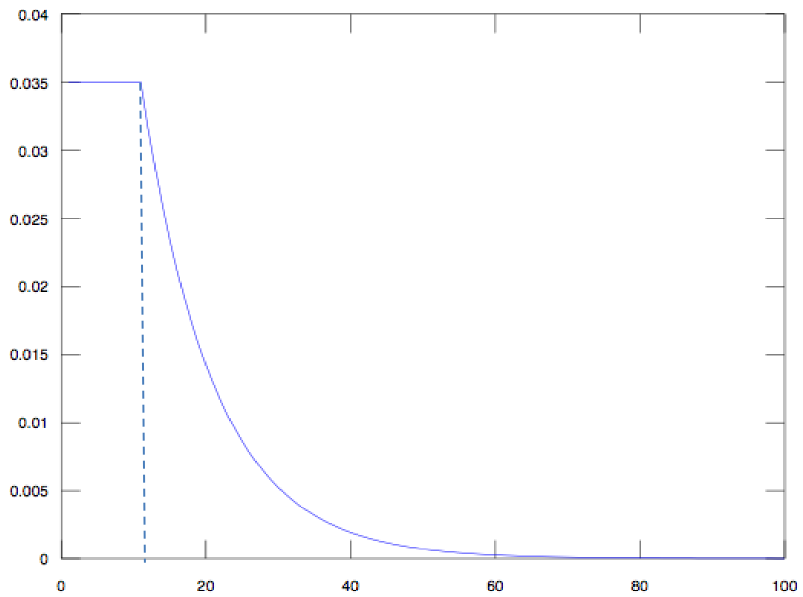
$\mathbf{W}^k[\text{Majority}]$



Exponentially Small Fourier Tails

Definition: f has **ESFT(t)** if for all k: $\mathbf{W}^{\geq k}[f] \leq e^{-k/t}$

$\mathbf{W}^{k}[f]$



t

Exponentially Small Fourier Tails

Definition: f has **ESFT(t)** if for all k: $\mathbf{W}^{\geq k}[f] \leq e^{-k/t}$

Several well-studied classes of Boolean functions have ESFT(t)

1. CNFs / **DNFs** formulae	**[H'86, LMN'89]**	$t = O(\log n)$
2. Formulae of size s	**[R'11]**	$t = O(\sqrt{s})$
3. Read-Once formulas	**[IK'14]**	$t = O(n^{0.31})$
4. Constant-depth circuits	**[T'14]**	$t = \mathrm{polylog}(n)$
5. Fncs with max-sensitivity s	**[GSTW'16]**	$t = O(s)$

"Excellent Low-Degree Approximations"

Equivalently: f in **ESFT(t)** if

$$\forall \epsilon > 0 \quad \exists p\colon \ \deg(p) \leq t \cdot \log(1/\epsilon), \qquad \|p - f\|_2 \leq \epsilon.$$

Correlation with Parity

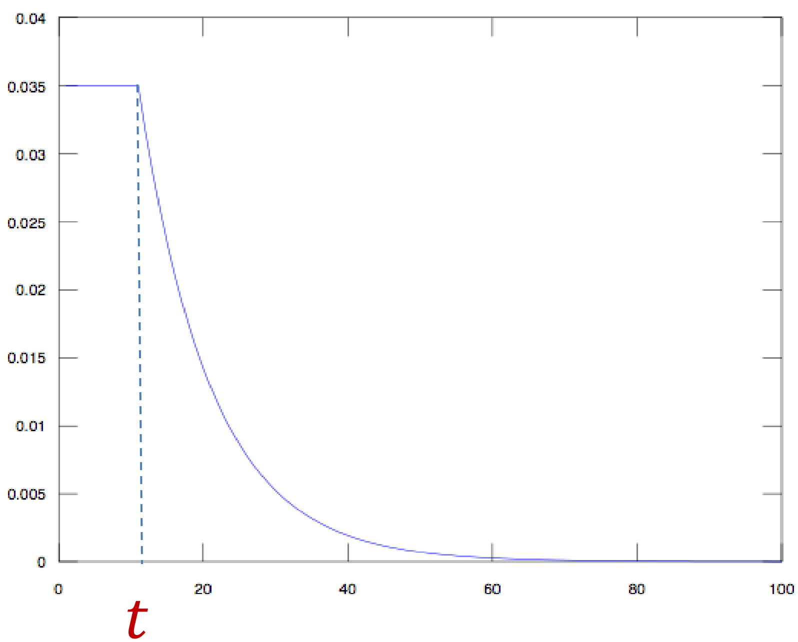
Observation: if f in **ESFT**(**t**), then

$$\mathbf{E}_x[f(x) \cdot \text{Parity}_n(x)] \leq e^{-n/2t}$$

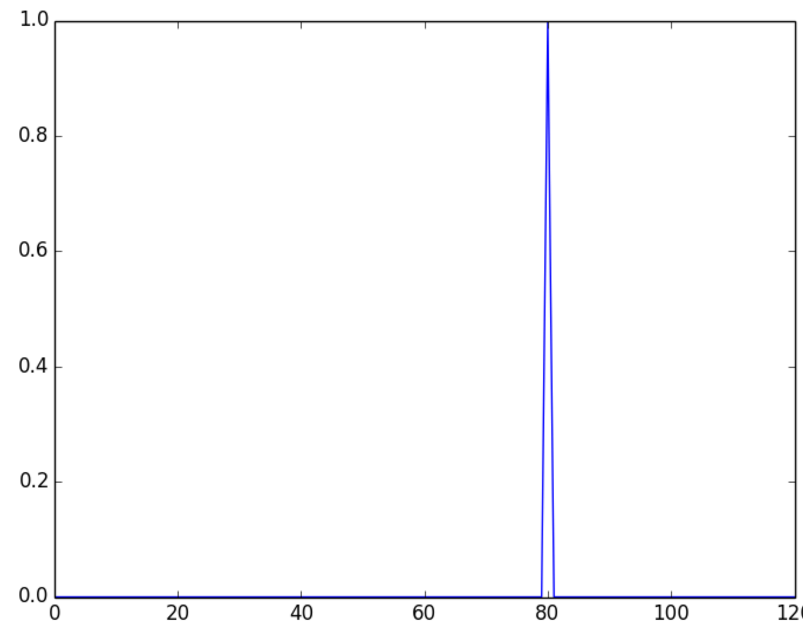
Proof:

$$|\mathbf{E}_x[f(x) \cdot \text{Parity}_n(x)]| = \left|\hat{f}(\{1, \dots, n\})\right| = \sqrt{\mathbf{W}^n[f]} \leq \sqrt{e^{-n/t}}$$

$\mathbf{W}^k[f]$



$\mathbf{W}^k[\text{Parity}]$



Different Notions of Fourier Concentration

TFAE:

- f has **Exponentially Small Fourier Tails:** $f \in \mathbf{ESFT}(t)$
- f has bounded **Fourier k-moments:**

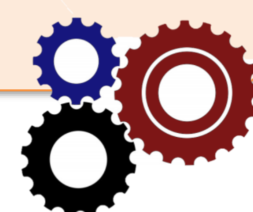
  $$\forall k:\ \mathbf{E}_{S\sim D_f}\left[\binom{|S|}{k}\right] \le O(t)^k$$

- f **simplifies under random restrictions:**

  $$\forall p, k:\ \underset{\substack{p \text{ random} \\ \text{restriction}}}{\mathbf{Pr}}\left[\deg(f_{\text{restricted}}) \ge k\right] \le O(pt)^k.$$

and they imply **using Booleanity**
L_1 degree-k sparsity:

$$\sum_{S:|S|=k} |\hat{f}(S)| = O(t)^k$$



Theorem: Let $f: \{-1,1\}^n \to \{-1,1\}$. Then,

$$\forall k: \sum_{S:|S|=k} \left|\hat{f}(S)\right| \leq 2^k \cdot \mathbf{E}_{S \sim D_f}\left[\binom{|S|}{k}\right]$$

Proof: Move to iPad

Separation between Quantum and Randomized Query Algorithms

First Try: Consider polynomial degree X

Problem: Both models are approximated by low-degree polynomials… [Nisan, Szegedy '92] [Beals, Buhrman, Cleve, Mosca, de Wolf '98]

But **these polynomials are very different**!

$L_{1,k}$ of a quantum query algorithm making $k/2$ queries can be $\sqrt{N^{k-1}}$

$L_{1,k}$ of a randomized query algorithm making d queries is at most $\sqrt{d^k}$

The Forrelation Problem [Aaronson'09]

The input to the (2-Fold) **Forrelation Problem** are two vectors

$$x, y \in \{-1,1\}^{N/2}.$$

The 2-Fold Forrelation Problem: distinguish between

`[Yes Instances]:` $(x,y) : \frac{\langle x, Hy \rangle}{N/2} \geq \tau$ Pairwise correlations $\pm\frac{\tau}{\sqrt{N}}$

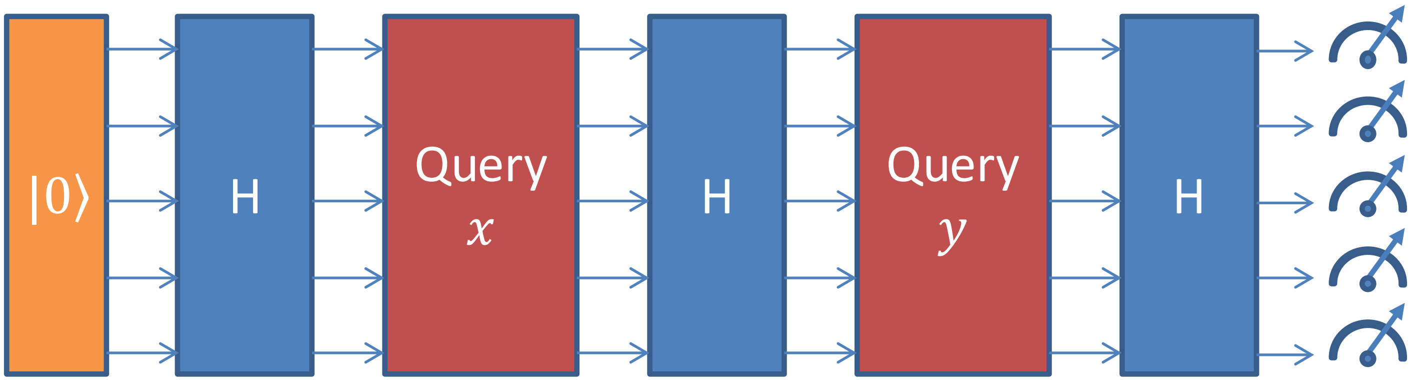
`[No  Instances]:` $(x,y) : \frac{\langle x, Hy \rangle}{N/2} \leq \tau/2$ No pairwise correlations

$$\frac{\langle x, Hy \rangle}{N/2} = \frac{1}{N/2} \sum_{i=1}^{N/2} \sum_{j=1}^{N/2} x_i \, H_{i,j} \, y_j$$

$$\boxed{H_{i,j} = \frac{(-1)^{<i,j>}}{\sqrt{N/2}}}$$

Quantum Algorithm for 2-Fold Forrelation

[Aaronson'09]:



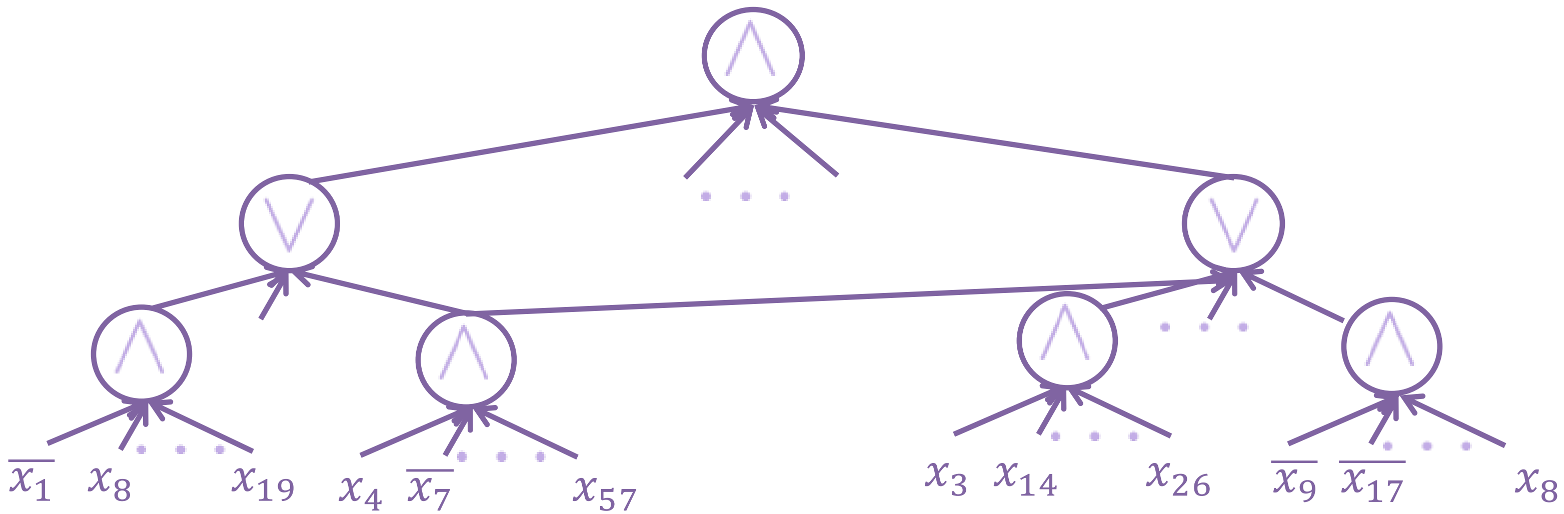
The probability of measuring the all 0's vector

$$\left(\frac{1}{N/2}\sum_{i=1}^{N/2}\sum_{j=1}^{N/2} x_i H_{i,j} y_j\right)^2 = \left(\frac{\langle x, Hy\rangle}{N/2}\right)^2$$

Main Technical Result [Raz-T'19]:

To solve Forrelation, f must have large $L_{1,2}(f)$.

Bounded Depth Circuits



AC⁰

- $poly(N)$ gates (**size** of the circuit)
- depth $d = O(1)$

Motivating Question: Separate **Quantum Log Time** from **AC⁰**

➔ Oracle Separation of **BQP** (Quantum Polynomial Time) from **PH** (The Polynomial Hierarchy)

What do we know about constant depth circuits ($\mathbf{AC^0}$)?

[Furst-Saxe-Sipser’81, Ajtai’83, Yao’85, Håstad’86]:

- The N-variate **Parity** function is not in $\mathbf{AC^0}$.

Proof technique:

- $\mathbf{AC^0}$ circuits can be well-approximated (in ℓ_2) by **low-degree polynomials** (over $\mathbb{R}$). **[Håstad’86, LMN’89]**
- **Parity** cannot.

Potential problem with the approach:

$O(\log N)$ time quantum algorithms are also well approximated by **low-degree polynomials**. **[BBCMW’98]**

The Difference between Quantum Log Time and AC^0

Both models are approximated by low-degree polynomials, but **these polynomials are very different**!

Quantum Log Time may require **dense** low-deg polynomials as in the case of **Aaronson's** algorithm:

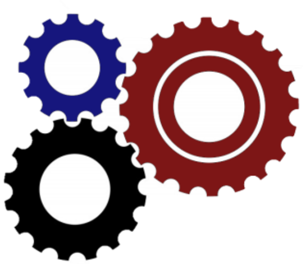
Degree: 2, #(monomials): $\Theta(N^2)$

Amplifies small pairwise correlations

[T'17]: AC^0 have **sparse** low-degree approximations:

$$\forall k\text{: } \#(\text{monomials of degree } k) \leq (\text{polylog } N)^k$$

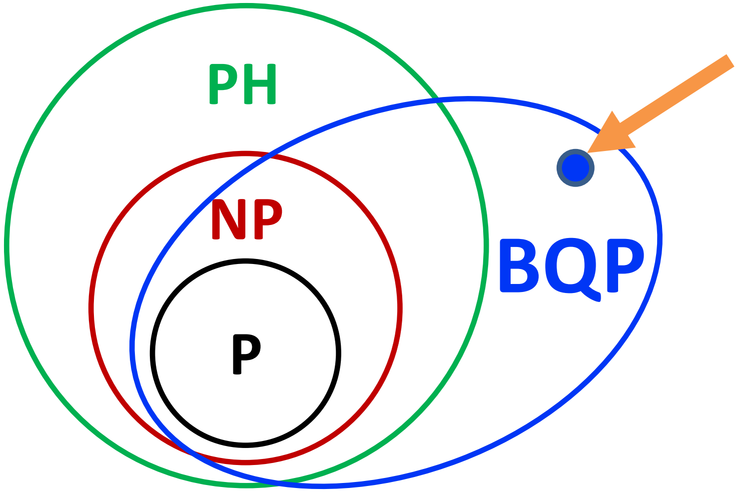
Does not amplify



Application

[Raz-T'19]:

$$\exists \text{ oracle } A: \quad \mathbf{BQP}^A \not\subseteq \mathbf{PH}^A$$



"Even if **P** were equal to **NP**, even making that strong assumption, that's not going to be enough to capture (the power of) quantum computing." **(Lance Fortnow)**

Distinguishing between Distributions

$z \in \{-1, +1\}^N$ sampled from the "Forrelation" distribution D

$z \in \{-1, +1\}^N$ sampled from the the uniform distribution U

One of these boxes is selected at random & given to you.

Can you tell which one is it?

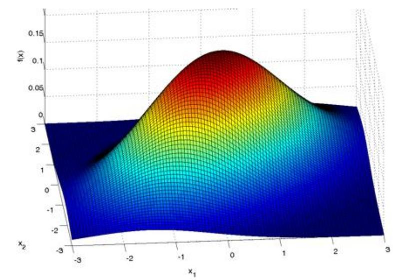
Sampling Forrelated Pairs

(Based on **Aaronson's** suggestion with some modifications)

Gaussian dist G over $\mathbb{R}^N$ ➔ Discrete dist D over $\{-1,1\}^N$

The Gaussian distribution G:

Sample $x_1, \dots, x_{N/2}$ i.i.d. $\mathcal{N}(0, \sigma^2)$



$$\vec{y} = H \cdot \vec{x}$$

Output $z = (x_1, \dots, x_{N/2}, y_1, \dots, y_{N/2})$.

$\boxed{\sigma^2 = 1/O(\log N)}$

The Discrete distribution D:

1. **Draw** $z \sim G$. If $z \notin [-1,1]^N$ ➔ abort
2. **Randomized Rounding:** For $i = 1, \dots, N$, draw independently $z_i' \in \{-1,1\}$ with $\mathbf{E}[z_i'] = z_i$.

The Fourier Expansion

The Fourier expansion of $f: \{-1,1\}^N \to \{-1,1\}$:

$$f(x) = \sum_{S \subseteq \{1,\dots,N\}} \hat{f}(S) \cdot \prod_{i \in S} x_i$$

$-1 \equiv$ True
$+1 \equiv$ False

For example: AND of 2 variables

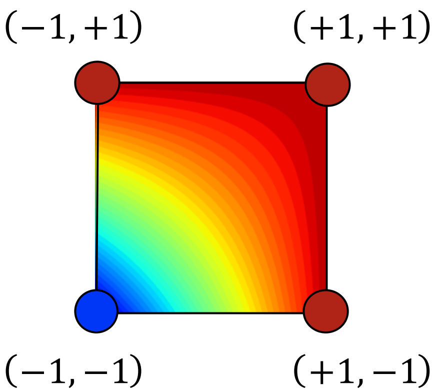
$$\text{AND}(x_1, x_2) = \frac{1}{2} + \frac{1}{2}x_1 + \frac{1}{2}x_2 - \frac{1}{2}x_1x_2$$

$$\text{AND}(+1,+1) = +\frac{1}{2} + \frac{1}{2} + \frac{1}{2} - \frac{1}{2} = +1.$$

$$\text{AND}(+1,-1) = +\frac{1}{2} + \frac{1}{2} - \frac{1}{2} + \frac{1}{2} = +1.$$

$$\text{AND}(-1,+1) = +\frac{1}{2} - \frac{1}{2} + \frac{1}{2} + \frac{1}{2} = +1.$$

$$\text{AND}(-1,-1) = +\frac{1}{2} - \frac{1}{2} - \frac{1}{2} - \frac{1}{2} = -1.$$



Fourier Expansion: a Bridge between Discrete and Continuous Settings

The Fourier expansion of $f: \{-1,1\}^N \to \{-1,1\}$:

$$f(x) = \sum_{S \subseteq \{1,\ldots,N\}} \hat{f}(S) \cdot \prod_{i \in S} x_i$$

Discrete **Gaussian**

Lemma: $\mathbf{E}_{z' \sim D}[f(z')] \approx \mathbf{E}_{z \sim G}[f(z)]$

Fact: $\mathbf{E}_{u \sim U}[f(u)] = f(\vec{0})$

Enough to show: For any f in **AC⁰**

$$\mathbf{E}_{z \sim G}[f(z)] \approx f(\vec{0})$$

Fourier Analytical Approach – First Attempt

$$\mathbf{E}_{z\sim G}[f(z)] - f(\vec{0}) =$$

$$= \sum_{|S|\geq 1} \hat{f}(S) \cdot \mathbf{E}_{z\sim G}\left[\prod_{i\in S} z_i\right]$$ **(By definition)**

$$= \sum_{\ell=1}^{N/2} \sum_{|S|=2\ell} \hat{f}(S) \cdot \mathbf{E}_{z\sim G}\left[\prod_{i\in S} z_i\right]$$ **(odd moments = 0)**

$$\leq \sum_{\ell=1}^{N/2} \sum_{|S|=2\ell} \left|\hat{f}(S)\right| \cdot \sigma^{2\ell} \cdot \frac{\ell!}{\sqrt{N/2}^{\ell}}$$ **(Isserlis' Theorem)**

$$\leq \sum_{\ell=1}^{N/2} \mathrm{polylog}(N)^{2\ell} \cdot \sigma^{2\ell} \cdot \frac{\ell!}{\sqrt{N/2}^{\ell}}$$

Contribution of
first $\tilde{O}(\sqrt{N})$ terms:
$\sigma^2 \cdot \mathrm{polylog}(N)/\sqrt{N}$

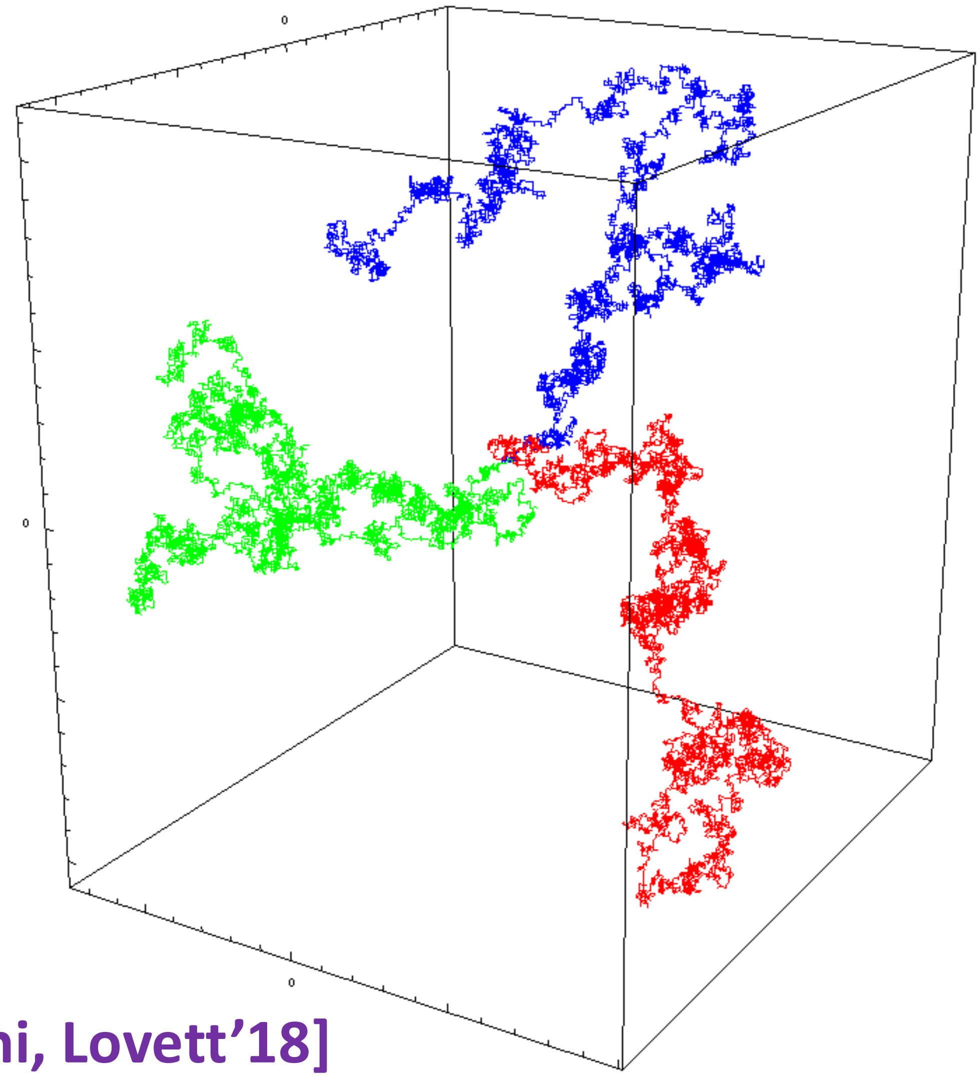
Contribution of larger terms?

Viewing $z \sim G$ as a result of a random walk

A Thought Experiment:

Instead of sampling $z \sim G$ at once, we sample t vectors $z^{(1)}, \dots, z^{(t)} \sim G$ independently, and take

$$z = \frac{1}{\sqrt{t}} \cdot \left(z^{(1)} + \cdots + z^{(t)} \right)$$



Based on the work of **[Chattopadhyay, Hatami, Hosseini, Lovett'18]**

Picture from http://en.wikipedia.org/wiki/Random_walk

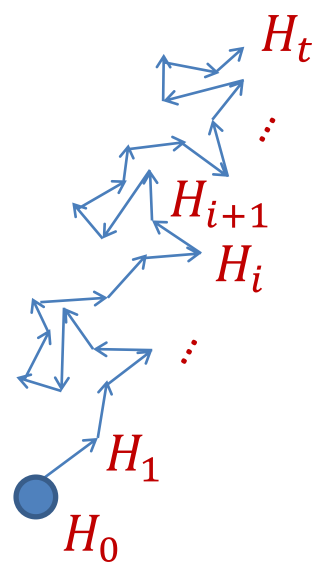
Viewing $z \sim G$ as a result of a random walk

Sample t vectors $z^{(1)}, \dots, z^{(t)} \sim G$

Define $t+1$ **hybrids:**

- $H_0 = \vec{0}$
- For $i = 1, \dots, t$

$$H_i = \frac{1}{\sqrt{t}} \cdot \left(z^{(1)} + \cdots + z^{(i)} \right)$$



Observe: $H_t \sim G$.

Taking $t \to \infty$ yields a Brownian motion.

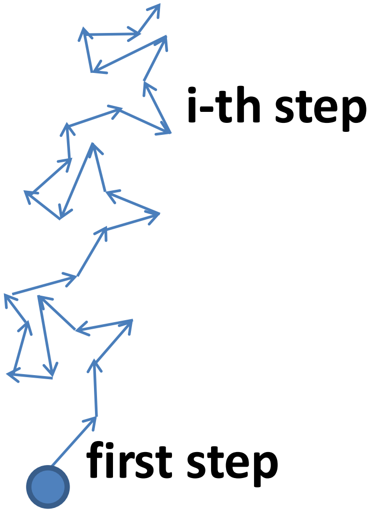
Suffices to take $t = \mathrm{poly}(N)$ for our analysis.

Claim: for $i = 0, \dots, t-1$,

$$|\mathbf{E}[f(H_{i+1})] - \mathbf{E}[f(H_i)]| \le \frac{\mathrm{polylog}(N)}{t\sqrt{N}}.$$

Proof by Picture

[CHHL'18]: i-th step ≈ first step, using closure under restrictions.



First Step: Simple Fourier Analysis
Only second level matters.

Base Case

$$\mathbf{E}[f(H_1)] - \mathbf{E}[f(H_0)]$$

$$= \mathbf{E}_{z\sim G}\left[f\left(\frac{1}{\sqrt{t}} z\right)\right] - f(\vec{0})$$

$$= \sum_{\ell=1}^{N/2} \sum_{|S|=2\ell} \hat{f}(S) \cdot \mathbf{E}_{z\sim G}\left[\prod_{i\in S} \frac{1}{\sqrt{t}} z_i\right]$$

$$\leq \sum_{\ell=1}^{N/2} \sum_{|S|=2\ell} \left|\hat{f}(S)\right| \cdot \frac{\sigma^{2\ell} \cdot \ell!}{t^{\ell} \cdot \sqrt{N/2}^{\ell}}$$

$$\leq \sum_{\ell=1}^{N/2} \mathrm{polylog}(N)^{2\ell} \cdot \frac{\sigma^{2\ell} \cdot \ell!}{t^{\ell} \cdot \sqrt{N/2}^{\ell}}$$

$$\leq \frac{\mathrm{polylog}(N)}{t\sqrt{N}} + o\left(\frac{1}{t\sqrt{N}}\right)$$

(for t large enough)

General Case: Reduction to Base Case

Lemma [CHHL'18]: for any fixed $v \in [-0.5, 0.5]^N$ the fnc

$$g(z) \stackrel{\text{def}}{=} f(v+z) - f(v)$$

can be written as $\mathbf{E}_\rho\left[f_\rho(2 \cdot z) - f_\rho(\vec{0})\right]$ where f_ρ is a random restriction of f (whose marginals depend on v).

Analysis of step i+1:

Conditioned on $H_i \in [-0.5, 0.5]^N$ (happens whp):

$$\begin{aligned}
&|\mathbf{E}[f(H_{i+1})] - \mathbf{E}[f(H_i)]| \\
&\leq \left|\mathbf{E}\left[f\left(H_i + \frac{1}{\sqrt{t}} \cdot z^{(i+1)}\right) - f(H_i)\right]\right| \\
&\leq \left|\mathbf{E}\left[f_\rho\left(\frac{2}{\sqrt{t}} \cdot z^{(i+1)}\right) - f_\rho(\vec{0})\right]\right| \leq \frac{\operatorname{polylog}(N)}{t\sqrt{N}}
\end{aligned}$$

$\blacksquare$

Recap

Main character: **Fourier Growth** – a complexity measure for Boolean functions that captures the ability to ***aggregate weak k-wise correlations in the input.***

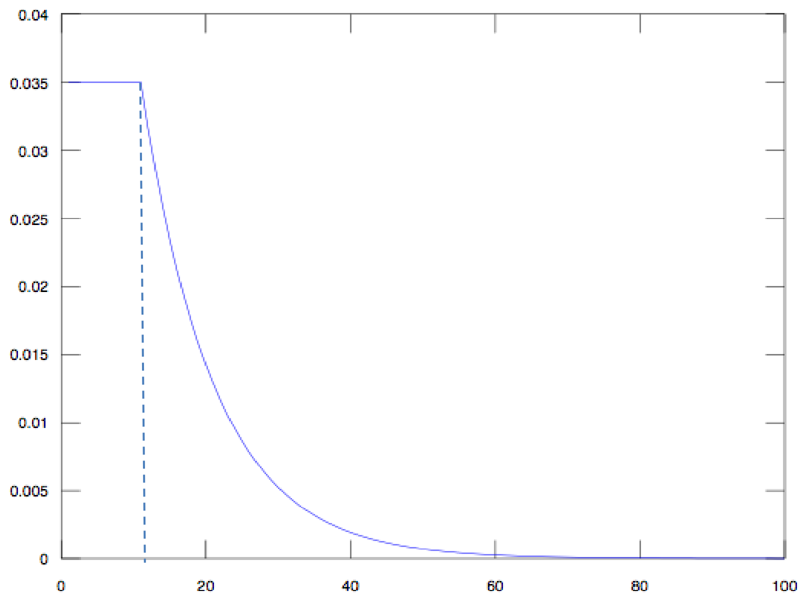
Applications:

1. **Quantum Advantage**
2. **Pseudo-randomness**
3. **Lower Bounds**

Exponentially Small Fourier Tails

Definition: f has **ESFT(t)** if for all k: $\mathbf{W}^{\geq k}[f] \leq e^{-k/t}$

$\mathbf{W}^{k}[f]$



t

Exponentially Small Fourier Tails

Definition: f has **ESFT(t)** if for all k: $\mathbf{W}^{\geq k}[f] \leq e^{-k/t}$

Several well-studied classes of Boolean functions have ESFT(t)

1. **CNFs** / **DNFs** formulae **[H'86, LMN'89]** $t = O(\log n)$
2. Formulae of size s **[R'11]** $t = O(\sqrt{s})$
3. Read-Once formulas **[IK'14]** $t = O(n^{0.31})$
4. Constant-depth circuits **[T'14]** $t = \mathrm{polylog}(n)$
5. Fncs with max-sensitivity s **[GSTW'16]** $t = O(s)$

"Excellent Low-Degree Approximations"

Equivalently: f in **ESFT(t)** if

$$\forall \epsilon > 0 \quad \exists p: \; \deg(p) \leq t \cdot \log(1/\epsilon), \quad \|p - f\|_2 \leq \epsilon.$$

Sparse Polynomial Approximations

Def'n: f in L_1(t) if $\forall k: \sum_{S:|S|=k} |\hat{f}(S)| \leq t^k$

Theorem [T'14]: If f is a **Boolean** function

f in **ESFT(t)** ➔ f in **L1(O(t))**

low degree approximations ➔ "sparse" approximations

But the latter is a much broader class!

- **Parity** in L_1(1). That is, **Parity** is sparse but of high degree.
- constant-width branching programs in **L_1(polylog(n))** **[CHRT'18]**
- **Most** Boolean functions are in **L_1(O(1))** !!

Which functions do not have "sparse" approximations?

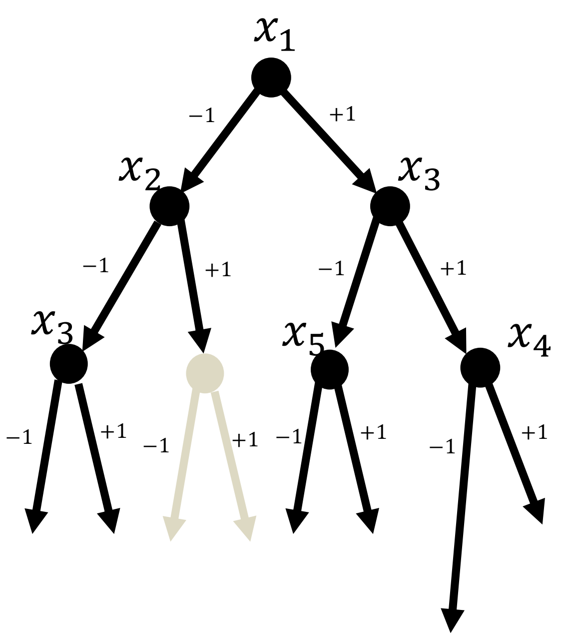
- **Majority** (Hardest function for this measure), **Forrelation**

Known Bounds on $L_{1,k}(f) = \sum_{S:|S|=k} |\hat{f}(S)|$

width-w DNFs/CNFs:	$L_{1,k} \lesssim w^k$	[Man95]
AC⁰ circuits of size s and depth d:	$L_{1,k} \lesssim (\log s)^{(d-1)k}$	[T17]
Boolean functions with sensitivity s:	$L_{1,k} \lesssim s^k$	[GSTW16]
regular width-w read-1 branching programs:	$L_{1,k} \lesssim (w)^k$	[RSV13,LPV22]
width-w read-1 branching programs:	$L_{1,k} \lesssim (\log n)^{wk}$	[CHRT18]
degree-d polynomials over F_2:	$L_{1,k} \lesssim \left(2^d \cdot k\right)^k$	[CHHL19]
depth-d (randomized) decision tree:	$L_{1,k} \lesssim \tilde{O}\left(\sqrt{d}\right)^k$	[T20, SSW21]
depth-d (randomized) parity decision tree:	$L_{1,k} \lesssim \tilde{O}\left(\sqrt{d}\right)^k$	[GTW21]
communication protocols of cost d:	$L_{1,k} \lesssim d^k$	[GRT21]
	$L_{1,1} \lesssim \sqrt{d}, \quad L_{1,2} \lesssim d^{3/2}$	[GSTW23]
Quantum query algorithms with r-rounds q-queries per round:		
	$L_{1,k} \lesssim (N^{\frac{1}{2}-\frac{1}{4r}} \cdot q)^k$	[GSTW24]

Most bounds are of the form $L_{1,k} \lesssim t^k$ for some parameter t

Proof Overview – $L_{1,1}$ for Decision Trees [OS'07]



$\ell_1 = \ell_3 = +1,$
$\ell_4 = -1$

- Let ℓ be a random root-to-leaf path
  - $\ell_i \in \{+1, -1\}$ iff x_i is queried in the path and fixed to ℓ_i
- Then

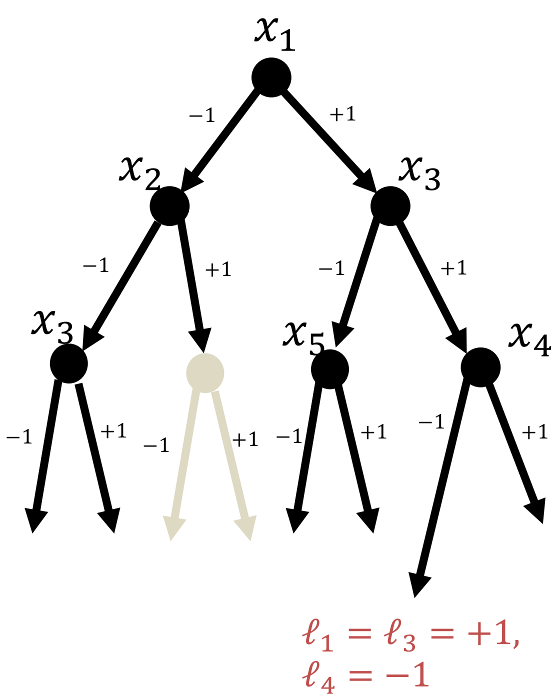
$$\hat{f}(\{i\}) = \mathbf{E}_x[f(x) \cdot x_i] = \mathbf{E}_\ell\big[f(\ell) \cdot \mathbf{E}_{x\sim\ell}[x_i]\big]$$
$$= \mathbf{E}_\ell[f(\ell) \cdot \ell_i]$$

- By negating x_i in f, we assume $\hat{f}(\{i\}) \geq 0$
- By querying dummy variables, WLOG the decision tree is full

- Then $L_{1,1}(f) = \sum_i \hat{f}(\{i\}) = \mathbf{E}_x[f(\ell) \cdot \sum_i \ell_i] \leq \mathbf{E}_x[|\sum_i \ell_i|]$

- $\sum_i \ell_i$ depends only on the number of $+1/-1$'s on the path
  $=$ the final state of a simple d-step drunkard walk
  $\mathbf{E}[|\sum_i \ell_i|] \approx \sqrt{d}$.

Proof Overview – $L_{1,2}$ for Decision Trees [T'20]



$\hat{f}(\{i,j\}) = \mathbf{E}_x[f(x) \cdot x_i x_j] = \mathbf{E}_\ell[f(\ell) \cdot \ell_i \cdot \ell_j].$

Can we assume $\hat{f}(\{i,j\}) \geq 0$?
Probably not. But we can write

$$L_{1,2}(f) = \sum_{i,j} |\hat{f}(\{i,j\})| = \sum_{i,j} a_{i,j} \cdot \hat{f}(\{i,j\})$$

for +-1 coefficients $a_{i,j} = \mathrm{sgn}(\hat{f}(\{i,j\}))$.

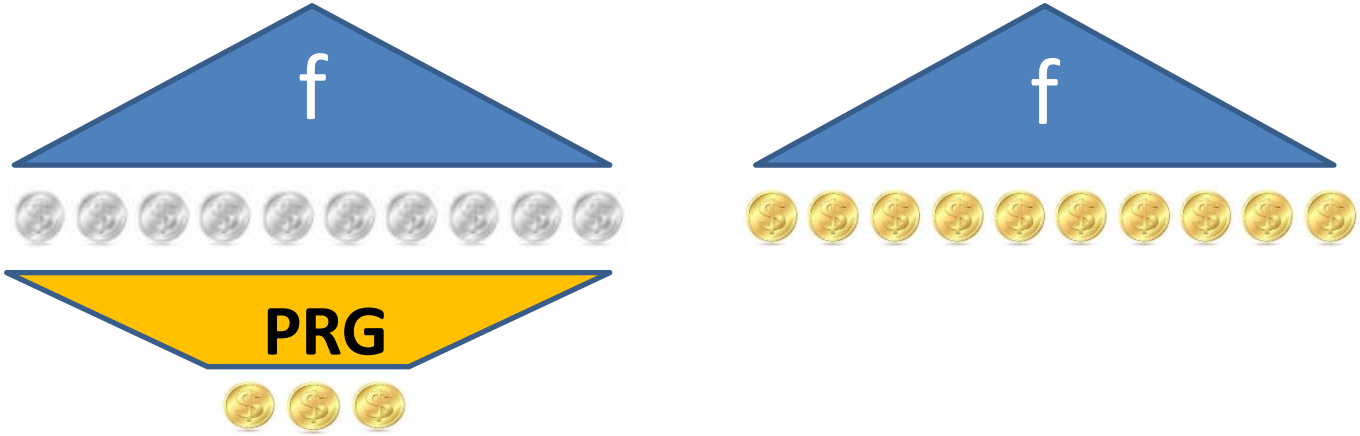
Then $L_{1,2}(f) = \mathbf{E}_\ell[f(\ell) \cdot \sum_{i,j} a_{i,j} \cdot \ell_i \ell_j] \leq \mathbf{E}_\ell[|\sum_{i,j} a_{i,j} \cdot \ell_i \ell_j|]$

$|\sum_{i,j} a_{i,j} \cdot \ell_i \ell_j|$ ~ a random 1-D walk with variabl

Reduces to $L_{1,1}$

Let $\ell^{(t)}$ be the evolution of ℓ after t queries.
If querying x_q in step t+1, then step size is
$|\sum_{i,j} a_{i,j} \cdot \ell_i^{(t+1)} \ell_j^{(t+1)} - \sum_{i,j} a_{i,j} \cdot \ell_i^{(t)} \ell_j^{(t)}| = \left|\sum_j a_{q,j} \cdot \ell_j^{(t)}\right|$

Applications to Pseudo-randomness



A distribution D over $\{\pm 1\}^n$ is **pseudorandom** for class C if

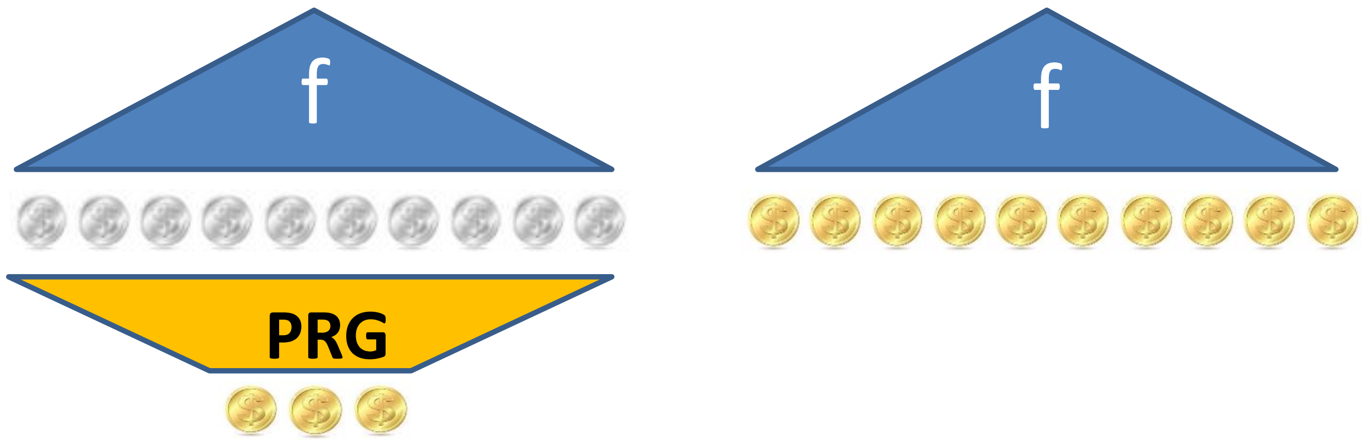
$$\forall f \in C: \qquad \mathbf{E}_{x\sim D}[f(x)] \approx_{\varepsilon} \mathbf{E}_{x\sim U}[f(x)]$$

A pseudo-random generator (**PRG**) for C is a function

$$\mathrm{PRG}: \{-1,1\}^s \to \{-1,1\}^n$$

such that $\mathrm{PRG}(U_s)$ is pseudorandom for C.

Applications to Pseudo-randomness



[CHLT'19] $\forall t$: a pseudo-random generator (**PRG**) for all functions f with $L_{1,2}(f) \leq t$ (assuming same holds for subfunctions of f) with seed length $s = O(t^2)$.

Build on [CHHL'18]: a PRG assuming $L_{1,k}(f) \leq t^k$ for all k.

PRG Construction

Observe that in the Forrelation analysis, we only relied on pairwise correlation of Gaussians being smaller than $1/L_{1,2}(f)$.

Lemma [CHLT'19] : We can sample n Gaussians with pairwise correlation δ with only $O\left(\frac{1}{\delta^2} \cdot \log^2(n)\right)$ seed.

But this gives us just a "Fractional PRG": a pseudorandom distribution D of points in $\mathbb{R}^n$ that is indistinguishable to f from uniform on $\{-1,+1\}^n$, and such that $\mathbf{E}_{x\sim D}\left[x_i^2\right] \geq 1/\log n$

Theorem [CHHL'18]: Fractional PRG ➔ PRG.

Open Problem

Conjecture [Chattopadhyay, Hatami, Lovett, T'19]:

Low-Degree $\boldsymbol{F_2}$-polynomials have sparse approximations.

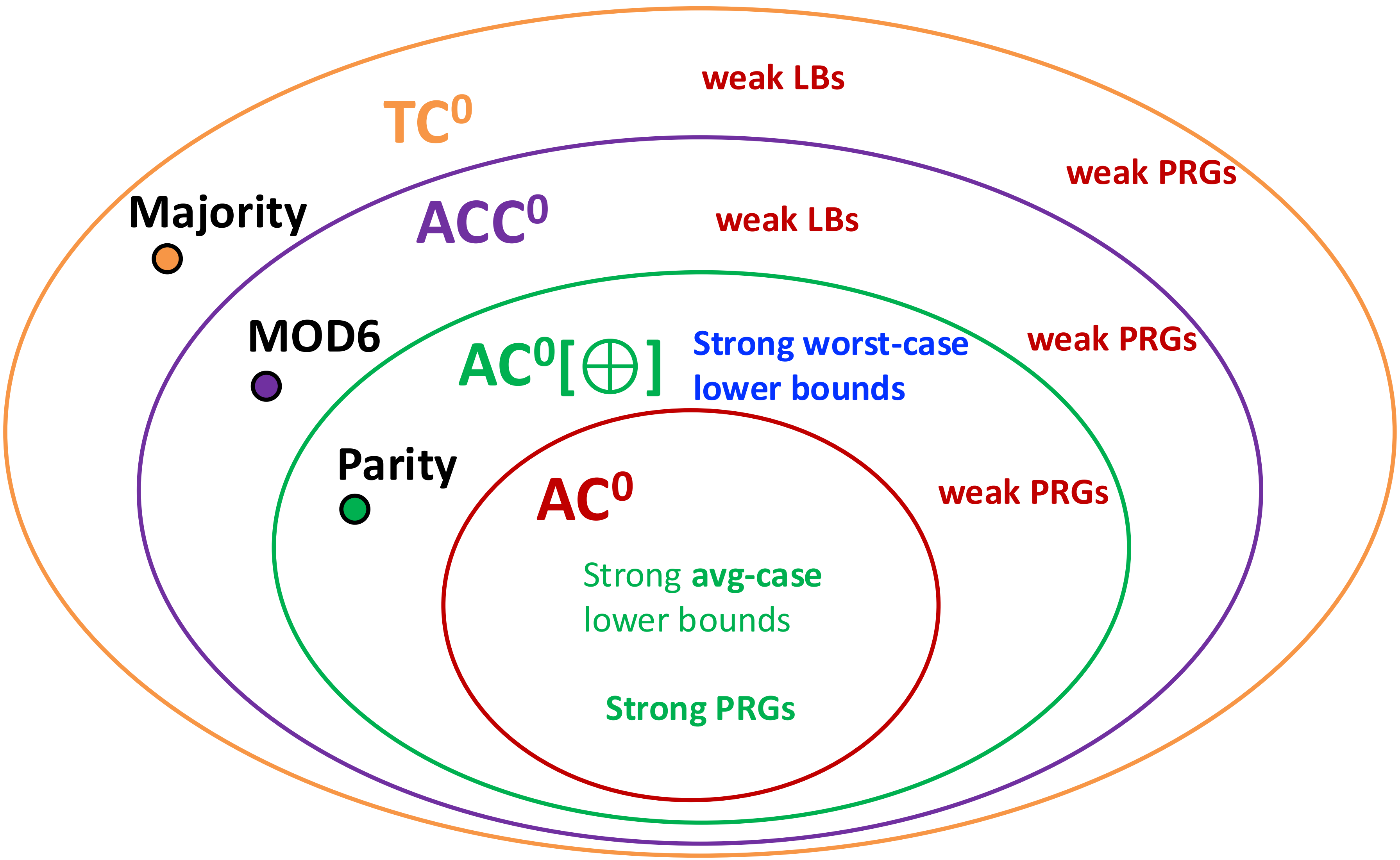
More Formally: If $p(x) \in \boldsymbol{F_2}[x_1, \dots, x_n]$ with $\deg(p) = d$,

then $f(x) = (-1)^{p(x)}$ has

$$\forall k: \sum_{S:|S|=k} \left|\hat{f}(S)\right| \leq O(d)^k.$$

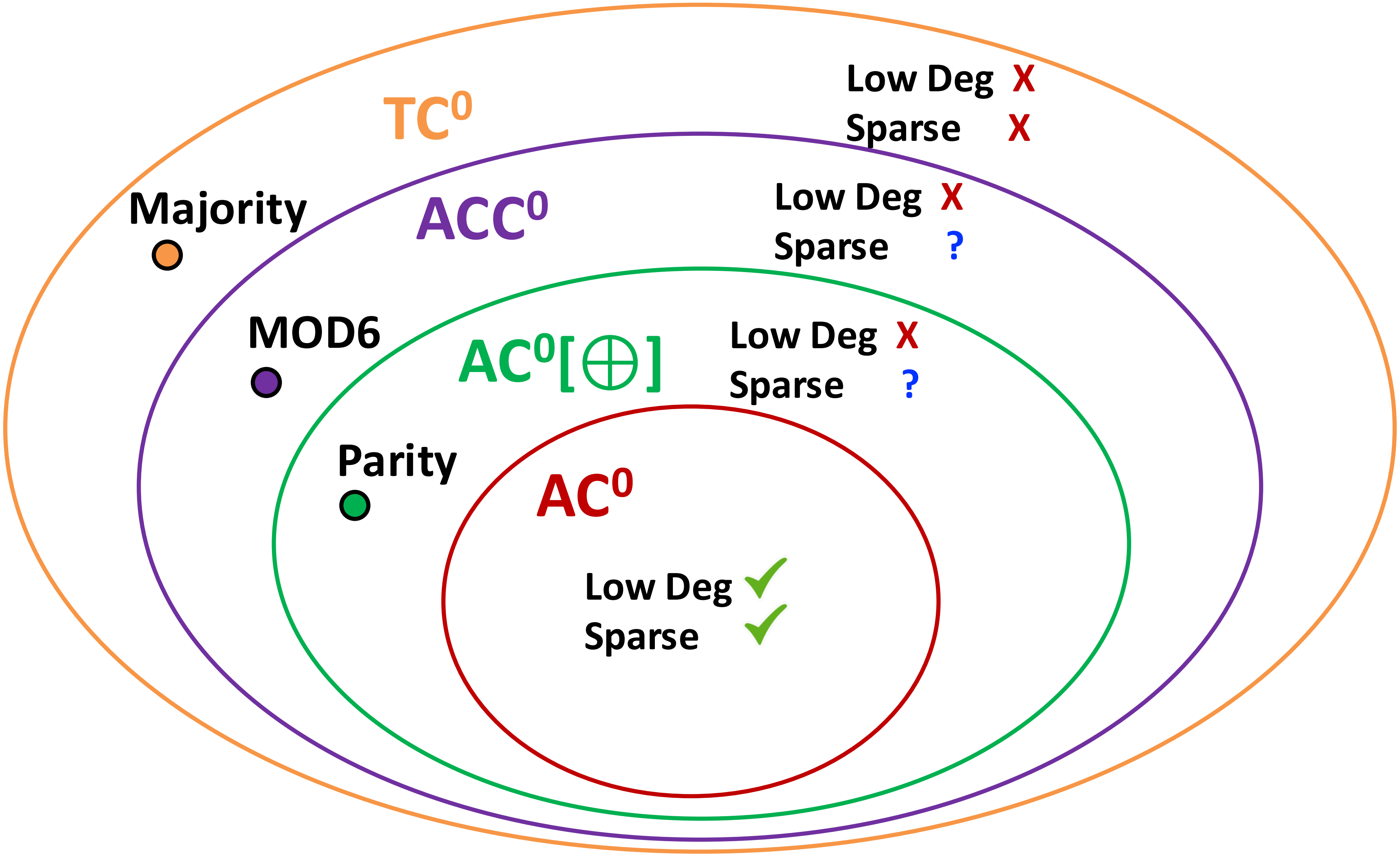
- We can prove the case k=1.
- Proving the case k=2 would yield pseudorandom generators that "look random" to low-degree $\boldsymbol{F_2}$-polynomials **(longstanding challenge)**

Circuit Complexity Frontier

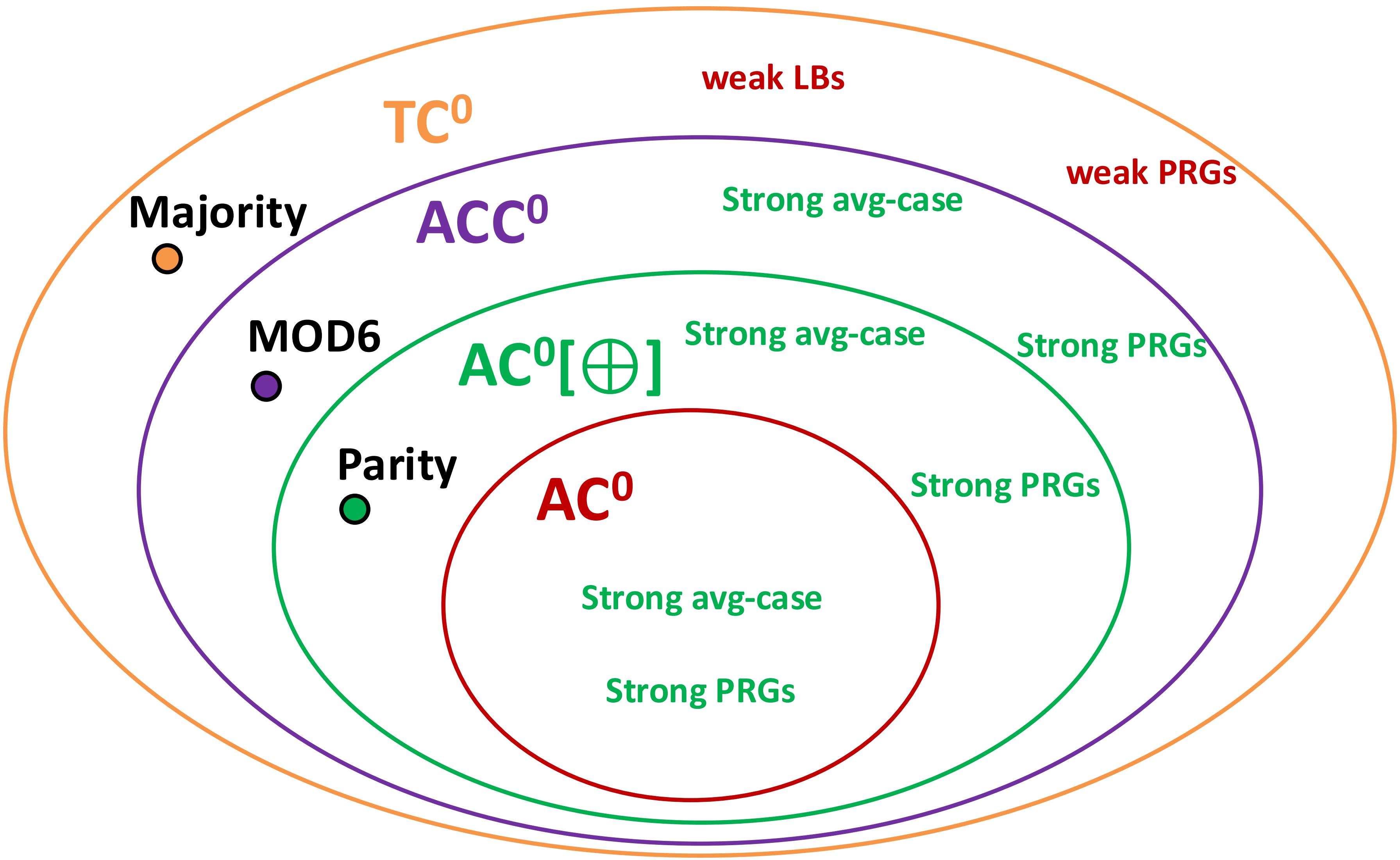


How broad are Sparse Polynomial Approximations?

Conjecture **[CHLT'19]:** $AC^0[\oplus]$, ACC^0 have sparse polynomial approximations



Circuit Complexity assuming Conjecture



Pseudo-randomness:

Can we **derandomize** any algorithm while increasing its **memory** by at most a constant?

Motivating Question: RL vs. L

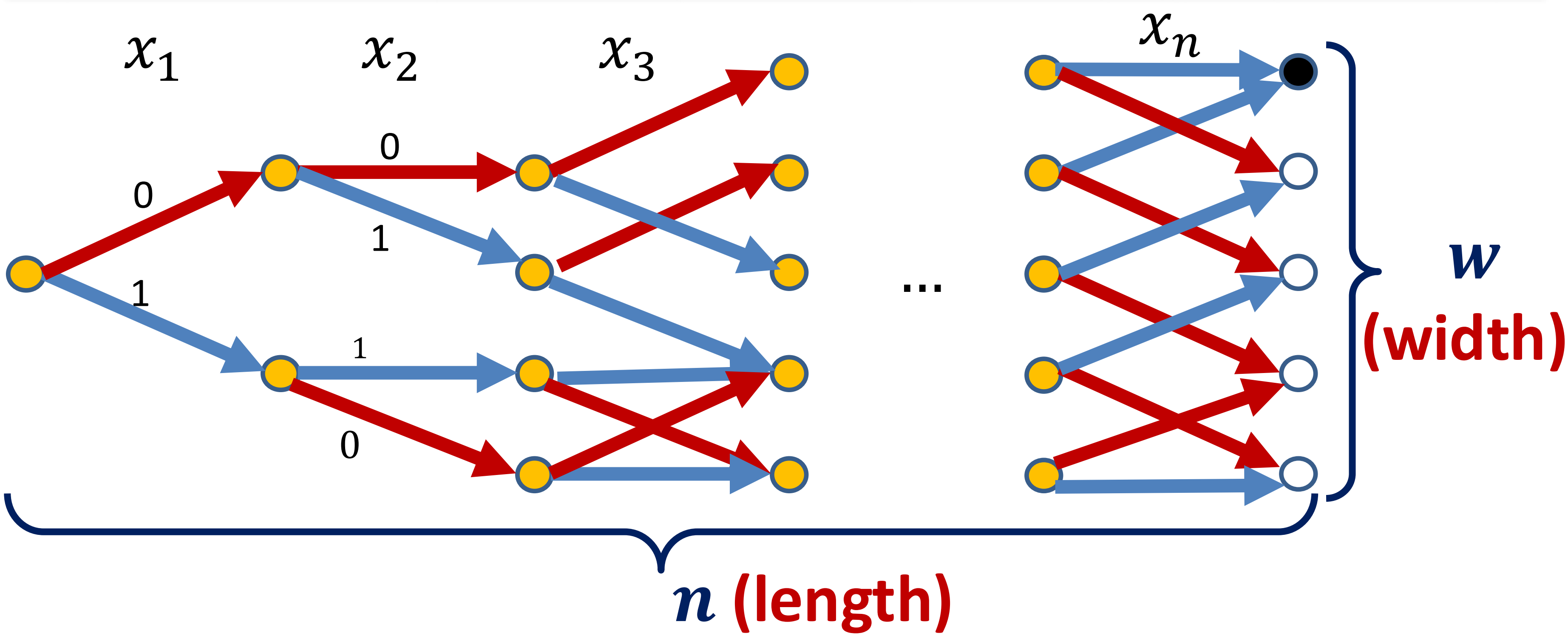
Open Question:

Does every problem solvable by a **randomized** algorithm with space s, is also solvable by a **deterministic** algorithm with space $O(s)$?

Suffices to focus on $s=O(\log n)$: does $RL = L$**?**

Randomized-Log-Space **Log-Space**

(Read-Once Oblivious) Branching Programs



- Each **layer** represents a **time step**
- Each **vertex** represents a **memory configuration**
- s memory bits ➔ width **at most** 2^s

PRGs for Branching Programs

[Nisan'90]: a **PRG** for length-n branching programs with seed-length:

- $O(\log^2 n)$ for width $\mathrm{poly}(n)$ (i.e., **Log-Space**).
- $O(\log^2 n)$ even for constant width

For width 2: seed length $O(\log n)$ suffices **[Saks-Zuckerman, Bogdanov-Dvir-Verbin-Yehudayoff]**

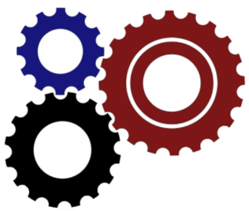
Nisan's PRG remains the state-of-the-art for width ≥ 4

Our Main Structural Result

[Chattopadyay-Hatami-Reingold-T'18]:

constant-width **branching programs** have **sparse polynomial approximations**:

$$\forall k: L_{1,k}(f) \leq (\text{polylog}\, n)^k$$



Applications:

1. **Exponentially** better **PRGs** for **unordered** branching programs **[CHRT'18, FK'18]**
2. **PRGs** for width-3 branching programs with seed-length $\tilde{O}(\log n)$ **[MRT'19]**
3. **PRGs** for read-once **AC^0** (and more) with seed-length $\tilde{O}(\log n)$ **[DHH'20, DMRTV'21]**

Open Problem

Show that the current construction by **[Forbes-Kelley'18]** works against any constant-width read-once branching programs with $\tilde{O}(\log n)$ seed length

Fourier Growth of Communication Protocols

Fourier Growth of Communication Protocols

Alice and Bob exchange d bits of communication and output a bit.

Their protocol defines a function $F: \{\pm 1\}^n \times \{\pm 1\}^n \to \{\pm 1\}$

What's the $L_{1,k}$ of F?
It could be arbitrarily large even with one bit of communication since Alice can compute an arbitrary function of x.

Fourier Growth of Communication Protocols

Alice and Bob exchange d bits of communication and output a bit.

Their protocol defines a function $F: \{\pm 1\}^n \times \{\pm 1\}^n \to \{\pm 1\}$

They attempt to compute an **XOR lifted function**:
Let $g: \{\pm 1\}^n \to \{\pm 1\}$ be a Boolean function (can be partial)
They want to compute $g(x \odot y)$ where $x \odot y$ is the bitwise product (XOR) of the strings

To succeed for any z in the domain of g, $g(z)$should be equal to $\mathbf{E}_{\mathbf{x},\mathbf{y}}[F(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \odot \mathbf{y} = z]$
➔ Fourier growth of the folded function $h(z) = \mathbf{E}_{\mathbf{x},\mathbf{y}}[F(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \odot \mathbf{y} = z]$

Fourier Growth of Communication Protocols

Alice and Bob exchange d bits of communication and output a bit.
Their protocol defines a function $F: \{\pm 1\}^n \times \{\pm 1\}^n \to \{\pm 1\}$.
Let $h(z) = \mathbf{E}_{\mathbf{x},\mathbf{y}}[F(\mathbf{x},\mathbf{y}) \mid \mathbf{x} \odot \mathbf{y} = z]$

Theorem [GRT21]: $L_{1,k}(h) \leq O(d)^k$

Theorem [GSTW23]: $L_{1,1}(h) \leq \sqrt{d}$, $L_{1,2}(h) \leq d^{3/2} \log(n)^{O(1)}$

Applications:

- New Proof for $\Omega(n)$ randomized communication complexity of Gap-Hamming-Problem **[Chakrabarti, Regev'10]**
- XOR-lift of **Forrelation$_2$:**
  - Requires $\widetilde{\Omega}(n^{1/3})$ randomized communication complexity
  - Can be computed in the simultaneous model using $\log(n)$ quantum communication, where each player implements an efficient quantum circuit of size $\mathrm{polylog}(n)$.

Open Problem

- Show $L_{1,2}(h) \leq d \log(n)^{O(1)}$
- Show $L_{1,k}(h) \leq O(\sqrt{d} \log n)^k$ for all k

- **The above conjecture** is implied by **lifting with any constant-size gadgets** (or even log-log size gadgets).

Summary

- Fourier L_1 degree-k sparsity (low $L_{1,k}$) as a **ubiquitous phenomenon**
- Separates quantum from classical query algorithms.
- Implies new oracle separations.
- Separates quantum from classical communication.
- Is useful for the design of pseudorandom generators for circuits

... and the design of pseudorandom generators against small space.

Connections to Open Problems:

- **RL** vs **L**
- Lifting with constant size gadgets
- PRGs and average-case lower bounds for **$\mathbf{AC^0[\oplus]}$, $\mathbf{ACC^0}$**

Thank You!