Boolean function $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$

$$\text{AND}_n(x) = \begin{cases} -1 \ (\text{TRUE}) & \text{if } x = (-1)^n \\ 1 \ (\text{FALSE}) & \text{otherwise} \end{cases}$$
A real polynomial $p$ $\epsilon$-approximates $f$ if

$$|p(x) - f(x)| < \epsilon \quad \forall x \in \{-1, 1\}^n$$

- $\tilde{\text{deg}}_\epsilon(f) =$ minimum degree needed to $\epsilon$-approximate $f$
- $\tilde{\text{deg}}(f) := \tilde{\text{deg}}_{1/3}(f)$ is the approximate degree of $f$
Threshold Degree

**Definition**

Let $f : \{-1, 1\}^n \rightarrow \{-1, 1\}$ be a Boolean function. A polynomial $p$ **sign-represents** $f$ if $\text{sgn}(p(x)) = f(x)$ for all $x \in \{-1, 1\}^n$.

**Definition**

The **threshold degree** of $f$ is $\min \deg(p)$, where the minimum is over all sign-representations of $f$.

- An equivalent definition of threshold degree is $\lim_{\epsilon \searrow 1} \tilde{\deg}_\epsilon(f)$. 

Upper bounds on $\tilde{\deg}_\epsilon(f)$ and $\deg_\pm(f)$ yield **efficient learning algorithms**.

- $\epsilon \approx 1/3$: Agnostic Learning [KKMS05]
- $\epsilon \approx 1 - 2^{-n^\delta}$: Attribute-Efficient Learning [KS04, STT12]
- $\epsilon \to 1$ (i.e., $\deg_\pm(f)$ upper bounds): PAC learning [KS01]

Upper bounds on $\tilde{\deg}_\epsilon(f)$ also imply fast algorithms for differentially private data release [TUV12, CTUW14].

Upper bounds on $\tilde{\deg}_\epsilon(f)$ and $\deg_\pm(f)$ for small formulas and threshold circuits $f$ yield state of the art formula size and threshold circuit lower bounds [Tal17, Forster02].
Upper bounds on $\widetilde{\deg}_\epsilon(f)$ and $\deg_\pm(f)$ yield efficient learning algorithms.

- $\epsilon \approx 1/3$: Agnostic Learning [KKMS05]
- $\epsilon \approx 1 - 2^{-n^\delta}$: Attribute-Efficient Learning [KS04, STT12]
- $\epsilon \to 1$ (i.e., $\deg_\pm(f)$ upper bounds): PAC learning [KS01]

- Upper bounds on $\widetilde{\deg}_{1/3}(f)$ also imply fast algorithms for differentially private data release [TUV12, CTUW14].
Upper bounds on $\tilde{\deg}_\varepsilon(f)$ and $\deg_\pm(f)$ yield efficient learning algorithms.

- $\varepsilon \approx 1/3$: Agnostic Learning [KKMS05]
- $\varepsilon \approx 1 - 2^{-n^\delta}$: Attribute-Efficient Learning [KS04, STT12]
- $\varepsilon \to 1$ (i.e., $\deg_\pm(f)$ upper bounds): PAC learning [KS01]

- Upper bounds on $\tilde{\deg}_{1/3}(f)$ also imply fast algorithms for differentially private data release [TUV12, CTUW14].
- Upper bounds on $\tilde{\deg}_\varepsilon(f)$ and $\deg_\pm(f)$ for small formulas and threshold circuits $f$ yield state of the art formula size and threshold circuit lower bounds [Tal17, Forster02].
Lower bounds on $\tilde{\deg}_\epsilon(f)$ and $\deg_\pm(f)$ yield lower bounds on:

- **Oracle Separations** [Bei94, BCHTV16]
- **Quantum query complexity** [BBCMW98]
- **Communication complexity** [She08, SZ08, CA08, LS08, She12]
  - Lower bounds hold for a communication problem related to $f$.
  - Via, e.g., a technique called the Pattern Matrix Method [She08].
Why Care About Approximate and Threshold Degree?

Lower bounds on $\tilde{\deg}_\epsilon(f)$ and $\deg_{\pm}(f)$ yield lower bounds on:

- **Oracle Separations** [Bei94, BCHTV16]
- **Quantum query complexity** [BBCMW98]
- **Communication complexity** [She08, SZ08, CA08, LS08, She12]

- Lower bounds hold for a communication problem related to $f$.
- Via, e.g., a technique called the Pattern Matrix Method [She08].

- $\epsilon \approx \frac{1}{3}$ $\implies$ $\mathsf{BQP}^{cc}$ lower bounds.
- $\epsilon \approx 1 - 2^{-n^\delta}$ $\implies$ $\mathsf{PP}^{cc}$ lower bounds
- $\epsilon \to 1$ (i.e., $\deg_{\pm}(f)$ lower bounds) $\implies$ $\mathsf{UPP}^{cc}$ lower bounds.
Why Care About Approximate and Threshold Degree?

Lower bounds on $\widetilde{\deg}_\epsilon(f)$ and $\deg_\pm(f)$ yield lower bounds on:

- **Oracle Separations** [Bei94, BCHTV16]
- **Quantum query complexity** [BBCMW98]
- **Communication complexity** [She08, SZ08, CA08, LS08, She12]
  - Lower bounds hold for a communication problem related to $f$.
  - Via, e.g., a technique called the Pattern Matrix Method [She08].

- $\epsilon \approx 1/3 \implies \text{BQP}^{cc}$ lower bounds.
- $\epsilon \approx 1 - 2^{-n^\delta} \implies \text{PP}^{cc}$ lower bounds
- $\epsilon \to 1$ (i.e., $\deg_\pm(f)$ lower bounds) $\implies \text{UPP}^{cc}$ lower bounds.

- Lower bounds on $\widetilde{\deg}_\epsilon(f)$ and $\deg_\pm(f)$ also yield efficient secret-sharing schemes [BIVW16]
Example 1: The Approximate Degree of $\text{AND}_n$
Example: What is the Approximate Degree of $\text{AND}_n$?

\[ \tilde{\deg}(\text{AND}_n) = \Theta(\sqrt{n}). \]

- Upper bound: Use **Chebyshev Polynomials**.
- The degree $d$ Chebyshev polynomial $T_d$ satisfies:
  - $|T_d(t)| \leq 1$ for all $t \in [-1, 1]$.
  - $T'_d(\pm 1) = d^2$. 

![Chebyshev Polynomial Graph]
Example: What is the Approximate Degree of $\text{AND}_n$?

$$\widetilde{\deg}(\text{AND}_n) = O(\sqrt{n}).$$

- After shifting a scaling, can turn degree $O(\sqrt{n})$ Chebyshev polynomial into a univariate polynomial $Q(t)$ that looks like:

![Graph showing $Q(-1+2/n) = 2/3$]

- Define $n$-variate polynomial $p$ via $p(x) = Q(\sum_{i=1}^{n} x_i/n)$.
- Then $|p(x) - \text{AND}_n(x)| \leq 1/3 \quad \forall x \in \{-1, 1\}^n$. 
Example: What is the Approximate Degree of $\text{AND}_n$?

\[ \text{NS92} \quad \tilde{\deg}(\text{AND}_n) = \Omega(\sqrt{n}). \]

- Lower bound: Use **symmetrization**.
- Suppose \( |p(x) - \text{AND}_n(x)| \leq 1/3 \quad \forall x \in \{-1, 1\}^n \).
- There is a way to turn \( p \) into a univariate polynomial \( p^{\text{sym}} \) that looks like this:

![Graph showing the approximation of AND function]

- **Claim 1:** \( \deg(p^{\text{sym}}) \leq \deg(p) \).
- **Claim 2:** Markov’s inequality \( \implies \deg(p^{\text{sym}}) = \Omega(n^{1/2}) \).
What if $\epsilon$ is “somewhat close” to 1?

- **Fact:** $\deg_{1-1/n}(\text{AND}_n) = 1$.
- **Proof:** Consider the approximation $1 - 1/n + \sum_{i=1}^{n} x_i/n$. 
Example 2: A Function With Large Approximate Degree For $\epsilon$ Exponentially Close to 1

Definition
Define the function ODD-MAX-BIT (OMB) via the following procedure: “For $i = 1, \ldots, n$, if $x_i = -1$, halt and output $(-1)^i$.”

- OMB is a decision list.
- Any decision list is also a linear-size DNF.

An example decision list on 4 variables.
Example 2: A Function With Large Approximate Degree For $\epsilon$ Exponentially Close to 1

**Definition**

Define the function ODD-MAX-BIT (OMB) via the following procedure: “For $i = 1, \ldots, n$, if $x_i = -1$, halt and output $(-1)^i$.”

- OMB is a decision list.
- Any decision list is also a linear-size DNF.

**Theorem (Beigel 1992, Klivans and Servedio 2004)**

*For any $d \geq 0$, $\widetilde{\deg}_\epsilon(OMB) = d$ for some $\epsilon = 1 - 2^{-\tilde{\Theta}(n/d^2)}$.*

**Special cases:**

- $\deg_{\pm}(OMB) = \widetilde{\deg}_{1-2^{-\Theta(n)}}(OMB) = 1$.
- $\widetilde{\deg}_\epsilon(OMB) = \tilde{\Theta}(n^{1/3})$ for $\epsilon = 1 - 2^{-n^{1/3}}$.
- $\deg_{1/3}(OMB) = \tilde{\Theta}(n^{1/2})$. 
In a $k$-decision list, each $C_i(x)$ is a conjunction of width $k$. 
In a $k$-decision list, each $C_i(x)$ is a conjunction of width $k$.

- Any $k$-decision list of length $\ell$ list is computed by a depth-3 circuit of bottom fan-in $O(k)$ and size $O(\ell)$. 
Our Main Result

Theorem (Main Theorem)

For any (large) constant $\Gamma > 0$ and (small) constant $\delta > 0$, there is an $O(\log n)$-decision list $f$ of length $\text{poly}(n)$ satisfying the following: $\deg_{\epsilon}(f) \geq n^{1/2-\delta}$ for $\epsilon = 1 - 2^{-n^\Gamma}$. 

Compare to prior work: Theorem (Beigel 1992, Klivans and Servedio 2004) For any $d \geq 0$, $\deg_{\epsilon}(\text{OMB}) = d$ for some $\epsilon = 1 - 2^{-\widetilde{\Theta}(n/d^2)}$.

In Main Theorem, $\deg_{\pm}(f) = O(\log n)$ and $\deg(f) = \widetilde{\Theta}(n^{1/2})$. So our $f$ can be sign-represented by a very low degree polynomial, but any polynomial of degree $\ll \deg(f)$ must incur extremely large error (superexponentially close to 1).

Proving this type of result requires fundamentally new techniques.
Our Main Result

Theorem (Main Theorem)

For any (large) constant \( \Gamma > 0 \) and (small) constant \( \delta > 0 \), there is an \( O(\log n) \)-decision list \( f \) of length \( \text{poly}(n) \) satisfying the following: \( \overline{\deg}_\epsilon(f) \geq n^{1/2-\delta} \) for \( \epsilon = 1 - 2^{-n^\Gamma} \).

Compare to prior work:

Theorem (Beigel 1992, Klivans and Servedio 2004)

For any \( d \geq 0 \), \( \overline{\deg}_\epsilon(\text{OMB}) = d \) for some \( \epsilon = 1 - 2^{-\tilde{\Theta}(n/d^2)} \).
Our Main Result

**Theorem (Main Theorem)**

For any (large) constant $\Gamma > 0$ and (small) constant $\delta > 0$, there is an $O(\log n)$-decision list $f$ of length $\text{poly}(n)$ satisfying the following: $\deg_\epsilon(f) \geq n^{1/2-\delta}$ for $\epsilon = 1 - 2^{-n^\Gamma}$.

- Compare to prior work:

**Theorem (Beigel 1992, Klivans and Servedio 2004)**

For any $d \geq 0$, $\deg_\epsilon(\text{OMB}) = d$ for some $\epsilon = 1 - 2^{-\tilde{\Theta}(n/d^2)}$.

- In Main Theorem, $\deg_\pm(f) = O(\log n)$ and $\deg(f) = \tilde{\Theta}(n^{1/2})$.

- So our $f$ can be sign-represented by a very low degree polynomial, but any polynomial of degree $\ll \deg(f)$ must incur extremely large error (superexponentially close to 1).

- Proving this type of result requires fundamentally new techniques.
Two Main Motivations for Our Main Result
First Motivation: PAC Learning DNFs

- The fastest known algorithm for PAC learning DNFs runs in time $2\tilde{O}(n^{1/3})$ [Klivans and Servedio 2001].
- Follows from the fact that for any DNF $f$, $\deg_{\pm}(f) = \tilde{O}(n^{1/3})$. 

Klivans and Servedio ask: for any DNF $f$, is it possible that $\tilde{\deg}_{\pm}(f) \leq \tilde{O}(n^{1/3})$ for $\epsilon = 1 - 2^{-n^{1/3}}$? An affirmative answer would yield a much simpler DNF learning algorithm. 

Our Main Theorem comes close to a negative resolution of their question.
The fastest known algorithm for PAC learning DNFs runs in time $2\tilde{O}(n^{1/3})$ [Klivans and Servedio 2001].

Follows from the fact that for any DNF $f$, $\deg_\pm(f) = \tilde{O}(n^{1/3})$.

How do Klivans and Servedio prove this?

- First, they turn any DNF into a (generalization of) a $k$-decision list, for some $k = \tilde{O}(n^{1/3})$.
- Second, they observe that any $k$-decision list $f$ satisfies $\deg_\pm(f) \leq k$.
- More specifically, $\deg_\epsilon(f) \leq k$ for $\epsilon = 1 - 2^{-n^k}$. 
The fastest known algorithm for PAC learning DNFs runs in time $2\tilde{O}(n^{1/3})$ [Klivans and Servedio 2001].

Follows from the fact that for any DNF $f$, $\deg_{\pm}(f) = \tilde{O}(n^{1/3})$.

How do Klivans and Servedio prove this?

- First, they turn any DNF into a (generalization of) a $k$-decision list, for some $k = \tilde{O}(n^{1/3})$.
- Second, they observe that any $k$-decision list $f$ satisfies $\deg_{\pm}(f) \leq k$.
- More specifically, $\tilde{\deg}_{\epsilon}(f) \leq k$ for $\epsilon = 1 - 2^{-n^k}$.

Klivans and Servedio ask: for any DNF $f$, is it possible that $\tilde{\deg}_{\epsilon}(f) \leq \tilde{O}(n^{1/3})$ for $\epsilon = 1 - 2^{-n^{1/3}}$?

An affirmative answer would yield a much simpler DNF learning algorithm.
First Motivation: PAC Learning DNFs

- The fastest known algorithm for PAC learning DNFs runs in time $2\tilde{O}(n^{1/3})$ [Klivans and Servedio 2001].
- Follows from the fact that for any DNF $f$, $\deg_\pm(f) = \tilde{O}(n^{1/3})$.
- How do Klivans and Servedio prove this?
  - First, they turn any DNF into a (generalization of) a $k$-decision list, for some $k = \tilde{O}(n^{1/3})$.
  - Second, they observe that any $k$-decision list $f$ satisfies $\deg_\pm(f) \leq k$.
  - More specifically, $\deg_\epsilon(f) \leq k$ for $\epsilon = 1 - 2^{-n^k}$.
- Klivans and Servedio ask: for any DNF $f$, is it possible that $\deg_\epsilon(f) \leq \tilde{O}(n^{1/3})$ for $\epsilon = 1 - 2^{-n^{1/3}}$?
- An affirmative answer would yield a much simpler DNF learning algorithm.
- Our Main Theorem comes close to a negative resolution of their question.
**Second Motivation: Complexity of AC$^0$**

- **PP** is the class of all languages solvable by polynomial time randomized algorithms that output the correct answer with probability strictly better than $1/2$.

- **PP** has a natural communication analog, **PP$^{cc}$**.

- Why is **PP$^{cc}$** important?
  - **PP$^{cc}(F)$** characterizes the margin complexity and discrepancy of $F$.
  - If $\text{PP}^{cc}(F) \geq d \Rightarrow F$ is not computed by Majority-of-Threshold Circuits of size $2^d$.

- Open question: How big can $\text{PP}^{cc}(F)$ be for an AC$^0$ function $F$? Can it be $\Omega(n)$?
**Second Motivation: Complexity of AC⁰**

- **PP** is the class of all languages solvable by polynomial time randomized algorithms that output the correct answer with probability strictly better than 1/2.
- **PP** has a natural **communication analog**, **PPcc**.
- Why is **PPcc** important?
  - **PPcc** (F) characterizes the **margin complexity** and **discrepancy** of F.
  - **PPcc** (F) ≥ d → F is not computed by Majority-of-Threshold Circuits of size 2ᵈ.
- Open question: How big can **PPcc** (F) be for an AC⁰ function F? Can it be Ω(n)?
- (Sherstov 2008): If \( \widetilde{\deg}_\epsilon(f) \geq d \) for \( \epsilon = 1 - 2^{-d} \), then f can be turned into a related function F satisfying **PPcc** (F) ≥ d.
Open question: How big can $\mathsf{PP}^{cc}(F)$ be for an $\mathsf{AC}^0$ function $F$?

History:

- Folklore: All depth-2 circuits $F$ have $\mathsf{PP}^{cc}(F) = O(\log n)$. 

Our work: For any constant $\delta > 0$, there is a depth-3 circuit $F$ with $\mathsf{PP}^{cc}(F) = \tilde{\Omega}(n^{1/2 - \delta})$. 

(Bun and Thaler 2015): A depth-3 circuit $F$ with $\mathsf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{2/5})$.

(Sherstov 2015): A depth-3 circuit $F$ with $\mathsf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{3/7})$ and a depth-4 circuit $F$ with $PP^{cc}(F) \geq \tilde{\Omega}(n^{1/2})$. 

Implication: Allender (1989) showed all of $\mathsf{AC}^0$ can be computed by quasipolynomial size depth-3 majority circuits. This cannot be improved to depth-2 majority circuits.
Second Motivation: Complexity of AC$^0$

- Open question: How big can $\mathbf{PP}^{cc}(F)$ be for an AC$^0$ function $F$?
- History:
  - Folklore: All depth-2 circuits $F$ have $\mathbf{PP}^{cc}(F) = O(\log n)$.
  - (Sherstov 2008, BVdW 2008): There is a depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq n^{1/3}$.
  - (Bun and Thaler 2015): A depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{2/5})$.
  - (Sherstov 2015): A depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{3/7})$ and a depth-4 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{1/2})$.

- Our work: For any constant $\delta > 0$, there is a depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{1/2-\delta})$. 
Open question: How big can $\mathsf{PP}^{cc}(F)$ be for an $\mathsf{AC}^0$ function $F$?

History:
- Folklore: All depth-2 circuits $F$ have $\mathsf{PP}^{cc}(F) = O(\log n)$.
- (Sherstov 2008, BVdW 2008): There is a depth-3 circuit $F$ with $\mathsf{PP}^{cc}(F) \geq n^{1/3}$.
- Implication: Allender (1989) showed all of $\mathsf{AC}^0$ can be computed by quasipolynomial size depth-3 majority circuits. This cannot be improved to depth-2 majority circuits.
Second Motivation: Complexity of $\text{AC}^0$

- Open question: How big can $\text{PP}^{cc}(F)$ be for an $\text{AC}^0$ function $F$?

- History:
  - Folklore: All depth-2 circuits $F$ have $\text{PP}^{cc}(F) = O(\log n)$.
  - (Sherstov 2008, BVdW 2008): There is a depth-3 circuit $F$ with $\text{PP}^{cc}(F) \geq n^{1/3}$.
    - Implication: Allender (1989) showed all of $\text{AC}^0$ can be computed by quasipolynomial size depth-3 majority circuits. This cannot be improved to depth-2 majority circuits.
  - (Bun and Thaler 2015): A depth-3 circuit $F$ with $\text{PP}^{cc}(F) \geq \tilde{\Omega}(n^{2/5})$.
  - (Sherstov 2015): A depth-3 circuit $F$ $\text{PP}^{cc}(F) \geq \tilde{\Omega}(n^{3/7})$ and a depth-4 circuit $F$ with $\text{PP}^{cc}(F) \geq \tilde{\Omega}(n^{1/2})$. 

Our work: For any constant $\delta > 0$, there is a depth-3 circuit $F$ with $\text{PP}^{cc}(F) \geq \tilde{\Omega}(n^{1/2} - \delta)$. 

Second Motivation: Complexity of AC$^0$

- Open question: How big can $\mathbf{PP}^{cc}(F)$ be for an AC$^0$ function $F$?

- History:
  - Folklore: All depth-2 circuits $F$ have $\mathbf{PP}^{cc}(F) = O(\log n)$.
  - (Sherstov 2008, BVdW 2008): There is a depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq n^{1/3}$.
    - Implication: Allender (1989) showed all of AC$^0$ can be computed by quasipolynomial size depth-3 majority circuits. This cannot be improved to depth-2 majority circuits.
  - (Bun and Thaler 2015): A depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{2/5})$.
  - (Sherstov 2015): A depth-3 circuit $F$ $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{3/7})$ and a depth-4 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{1/2})$.

- **Our work:** For any constant $\delta > 0$, there is a depth-3 circuit $F$ with $\mathbf{PP}^{cc}(F) \geq \tilde{\Omega}(n^{1/2 - \delta})$. 
These theorems show that $g \circ f$ is "harder to approximate" by low-degree polynomials than is $f$ alone. Here, $g \circ f = g(f, \ldots, f)$ is the block-composition of $g$ and $f$. 
Prior Techniques: Proving Hardness Amplification Theorems For Block-Composed Functions

These theorems show that $g \circ f$ is “harder to approximate” by low-degree polynomials than is $f$ alone.

Here, $g \circ f = g(f, \ldots, f)$ is the block-composition of $g$ and $f$. 
### Hardness-Amplification Theorems From Prior Work

<table>
<thead>
<tr>
<th>Theorem (Sherstov 2010)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $f$ be a Boolean function with $\widehat{\deg}<em>{1/2}(f) \geq d$. Let $F = \bigoplus_t \circ f$, where $\bigoplus_t$ is the parity function on $t$ bits. Then $\widehat{\deg}</em>{1-2^{-t}}(F) \geq d \cdot t$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (Bun and Thaler 2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $f$ be a Boolean function with $\widehat{\deg}_{-1/2}(f) \geq d$. Let $F = \text{OR}<em>t \circ f$. Then $\widehat{\deg}</em>{1-2^{-t}}(F) \geq d$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (Sherstov 2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $f$ be a Boolean function with $\widehat{\deg}_{-1/2}(f) \geq d$. Let $F = \text{OR}<em>t \circ f$. Then $\deg</em>{\pm}(F) = \Omega(\min{d, t})$.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theorem (Thaler 2014)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Let $f$ be a Boolean function with $\widehat{\deg}_{+1/2}(f) \geq d$. Let $F = \text{OMB}<em>t \circ f$. Then $\widehat{\deg}</em>{1-2^{-t}}(F) \geq d$.</td>
</tr>
</tbody>
</table>
Our Techniques: Beyond Block-Composed Functions
An $O(\log n)$-Decision List Harder to Approximate than OMB?

**Theorem (Beigel94, Thaler14)**

Let $F = \text{OMB}_t \circ \text{OR}_b$. Then $\overline{\deg}_{1-2^{-t}}(F) \geq \sqrt{b}$. E.g., if $t = n^{1/3}$ and $b = n^{2/3}$, then $\deg_{\epsilon}(F) \geq n^{1/3}$ for $\epsilon = 1 - 2^{-n^{1/3}}$.
An $O(\log n)$-Decision List Harder to Approximate than OMB?

**Theorem (Beigel94, Thaler14)**

Let $F = \text{OMB}_t \circ \text{OR}_b$. Then $\widetilde{\deg}_{1-2^{-t}}(F) \geq \sqrt{b}$. E.g., if $t = n^{1/3}$ and $b = n^{2/3}$, then $\deg_\epsilon(F) \geq n^{1/3}$ for $\epsilon = 1 - 2^{-n^{1/3}}$.

- Our goal is to modify $\text{OMB}_t \circ \text{OR}_b$ to obtain a function $f$ that is much harder to approximate by low-degree polynomials, while still ensuring that $f$ is computed by an $O(\log n)$-decision list.

**Theorem (Main Theorem)**

For any (large) constant $\Gamma > 0$ and (small) constant $\delta > 0$, there is an $O(\log n)$-decision list $f$ of length $\text{poly}(n)$ satisfying the following: $\widetilde{\deg}_\epsilon(f) \geq n^{1/2-\delta}$ for $\epsilon = 1 - 2^{-n^\Gamma}$. 

First attempt: Letting $\oplus_k$ denote the Parity function on $k$ bits, consider $F := \oplus_k \circ \text{OMB}_t \circ \text{OR}_b$. This is a $k$-decision list of length $n^k$. Unfortunately, this is too easy to approximate. Let $p$ approximate $\text{OMB}_b \circ \text{OR}_t$ to error $1 - \epsilon$. Then the polynomial $q(x_1,\ldots,x_k) = \prod_{i=1}^k p(x_i)$ approximates $F(x_1,\ldots,x_k)$ to error $1 - \epsilon^k$. Note: $q$ treats each of the $k$ "blocks" $x_i$ independently, and outputs the products of the $k$ results.
An $O(\log n)$-Decision List Harder to Approximate than OMB?

Theorem (Beigel94, Thaler14)

Let $F = \text{OMB}_t \circ \text{OR}_b$. Then $\deg_{1-2^{-t}}(F) \geq \sqrt{b}$. E.g., if $t = n^{1/3}$ and $b = n^{2/3}$, then $\deg_{\epsilon}(F) \geq n^{1/3}$ for $\epsilon = 1 - 2^{-n^{1/3}}$.

- Our goal is to modify $\text{OMB}_t \circ \text{OR}_b$ to obtain a function $f$ that is much harder to approximate by low-degree polynomials, while still ensuring that $f$ is computed by an $O(\log n)$-decision list.

- First attempt: Letting $\oplus_k$ denote the Parity function on $k$ bits, consider $F := \oplus_k \circ \text{OMB}_t \circ \text{OR}_b$.
  - This is a $k$-decision list of length $n^k$. 

Unfortunately, this is too easy to approximate. Let $p$ approximate $\text{OMB}_b \circ \text{OR}_t$ to error $1 - \epsilon$. Then the polynomial $q(x_1, \ldots, x_k) = \prod_{i=1}^{k} p(x_i)$ approximates $F(x_1, \ldots, x_k)$ to error $1 - \epsilon^k$. Note: $q$ treats each of the $k$ “blocks” $x_i$ independently, and outputs the products of the $k$ results.
An $O(\log n)$-Decision List Harder to Approximate than OMB?

**Theorem (Beigel94, Thaler14)**

Let $F = \text{OMB}_t \circ \text{OR}_b$. Then $\widetilde{\deg}_{1 - 2^{-t}}(F) \geq \sqrt{b}$. E.g., if $t = n^{1/3}$ and $b = n^{2/3}$, then $\deg_\epsilon (F) \geq n^{1/3}$ for $\epsilon = 1 - 2^{-n^{1/3}}$.

- Our goal is to modify $\text{OMB}_t \circ \text{OR}_b$ to obtain a function $f$ that is much harder to approximate by low-degree polynomials, while still ensuring that $f$ is computed by an $O(\log n)$-decision list.

- First attempt: Letting $\oplus_k$ denote the Parity function on $k$ bits, consider $F := \oplus_k \circ \text{OMB}_t \circ \text{OR}_b$.
  - This is a $k$-decision list of length $n^k$.

- Unfortunately, this is too easy to approximate.
  - Let $p$ approximate $\text{OMB}_b \circ \text{OR}_t$ to error $1 - \epsilon$.
  - Then the polynomial $q(x_1, \ldots, x_k) = \prod_{i=1}^k p(x_i)$ approximates $F(x_1, \ldots, x_k)$ to error $1 - \epsilon^k$. 
An $O(\log n)$-Decision List Harder to Approximate than OMB?

**Theorem (Beigel94, Thaler14)**

Let $F = \text{OMB}_t \circ \text{OR}_b$. Then $\overline{\deg}_{1 - 2^{-t}}(F) \geq \sqrt{b}$. E.g., if $t = n^{1/3}$ and $b = n^{2/3}$, then $\deg_\epsilon(F) \geq n^{1/3}$ for $\epsilon = 1 - 2^{-n^{1/3}}$.

- Our goal is to modify $\text{OMB}_t \circ \text{OR}_b$ to obtain a function $f$ that is much harder to approximate by low-degree polynomials, while still ensuring that $f$ is computed by an $O(\log n)$-decision list.

- First attempt: Letting $\oplus_k$ denote the Parity function on $k$ bits, consider $F := \oplus_k \circ \text{OMB}_t \circ \text{OR}_b$.
  - This is a $k$-decision list of length $n^k$.

- Unfortunately, this is too easy to approximate.
  - Let $p$ approximate $\text{OMB}_b \circ \text{OR}_t$ to error $1 - \epsilon$.
  - Then the polynomial $q(x_1, \ldots, x_k) = \prod_{i=1}^k p(x_i)$ approximates $F(x_1, \ldots, x_k)$ to error $1 - \epsilon^k$.
  - Note: $q$ treats each of the $k$ “blocks” $x_i$ independently, and outputs the products of the $k$ results.
Our $F$ first “pre-processes” its input $(x_1, \ldots, x_k)$ to obtain values $(u_1, \ldots, u_k) \in \{-1, 1\}^{(t \cdot b) \times k}$, which are then fed into $\oplus_k \circ \text{OMB}_t \circ \text{OR}_b$.

The pre-processing introduces dependencies between blocks. This ensures that an approximating polynomial for $F$ will be unable to treat them independently.

But the pre-processing is “mild” enough that $F$ is an $O(\log n)$-decision list of length $n^k$.

The larger $k$ is, the better our lower bound for $F$ (i.e., the lower bound holds for a larger $\Gamma$ and a smaller $\delta$).
Our $F$ first “pre-processes” its input $(x_1, \ldots, x_k)$ to obtain values $(u_1, \ldots, u_k) \in \{-1, 1\}^{(t \cdot b) \times k}$, which are then fed into $\oplus_k \circ \text{OMB}_t \circ \text{OR}_b$.

The pre-processing introduces dependencies between blocks.

- This ensures that an approximating polynomial for $F$ will be unable to treat them independently.
- But the pre-processing is “mild” enough that $F$ is an $O(\log n)$-decision list of length $n^k$.
- The larger $k$ is, the better our lower bound for $F$ (i.e., the lower bound holds for a larger $\Gamma$ and a smaller $\delta$).

Idea for $k = 2$.

- $F$ takes two input “blocks” $(x_1, x_2)$, with $x_1 \in \{-1, 1\}^{t \cdot b}$, and $x_2 \in \{-1, 1\}^{t \cdot b \cdot \log_2 (t \cdot b)}$.
- Turn $(x_1, x_2)$ into $(u_1, u_2) \in \{-1, 1\}^{t \cdot b} \times \{-1, 1\}^{t \cdot b}$ as follows:
Moving Beyond Block-Composition

- Our $F$ first “pre-processes” its input $(x_1, \ldots, x_k)$ to obtain values $(u_1, \ldots, u_k) \in \{-1, 1\}^{(t \cdot b) \times k}$, which are then fed into $\oplus_k \circ \text{OMB}_t \circ \text{OR}_b$.
- The pre-processing introduces dependencies between blocks.
  - This ensures that an approximating polynomial for $F$ will be unable to treat them independently.
  - But the pre-processing is “mild” enough that $F$ is an $O(\log n)$-decision list of length $n^k$.
  - The larger $k$ is, the better our lower bound for $F$ (i.e., the lower bound holds for a larger $\Gamma$ and a smaller $\delta$).
- Idea for $k = 2$.
  - $F$ takes two input “blocks” $(x_1, x_2)$, with $x_1 \in \{-1, 1\}^{t \cdot b}$, and $x_2 \in \{-1, 1\}^{t \cdot b \cdot \log_2(t \cdot b)}$.
  - Turn $(x_1, x_2)$ into $(u_1, u_2) \in \{-1, 1\}^{t \cdot b} \times \{-1, 1\}^{t \cdot b}$ as follows:
    - $u_1 = x_1$.
    - Let $i^* \in \{1, \ldots, t\}$ be the largest value such that $x_{1,i^*} = -1$.
    - $u_2$ is obtained from $x_2$ by testing each consecutive sequence of $\log_2(tb)$ bits for equality with (the binary representation of) $i^*$.
Schematic of Our Hard-To-Approximate $O(\log n)$-Decision List for $k = 2$
Subsequent Work and Open Questions

- (Bun and Thaler, 2017): A different hardness amplification technique that moves beyond block-composed functions.
  - For any constant $\delta > 0$, yielded a nearly-optimal $\Omega(n^{1-\delta})$ lower bound on the approximate degree of AC$^0$ (specifically, depth $\log(1/\delta)$).
  - Previous best lower bound for AC$^0$ was $\Omega(n^{2/3})$ (Aaronson and Shi, 2004).

- (Bun and Thaler 2018): Different refinements, showing that there is an AC$^0_c$ circuit of depth $O(1/\delta)$ and PP$^{cc}(F) \geq n^{1-\delta}$.

Conjecture: For any constant $\delta > 0$, there is a depth-3 AC$^0_c$ circuit $F$ with PP$^{cc}(F) \geq n^{1-\delta}$ (maybe even $\Omega(n)$).

Can we prove this by combining the techniques of this work with (Bun and Thaler, 2017/2018)? Can we extend our lower bound for $O(\log n)$-decision lists to DNFs, answering the question of Klivans and Servedio?
Subsequent Work and Open Questions

- (Bun and Thaler, 2017): A different hardness amplification technique that moves beyond block-composed functions.
  - For any constant $\delta > 0$, yielded a nearly-optimal $\Omega(n^{1-\delta})$ lower bound on the approximate degree of $\text{AC}^0$ (specifically, depth $\log(1/\delta)$).
  - Previous best lower bound for $\text{AC}^0$ was $\Omega(n^{2/3})$ (Aaronson and Shi, 2004).
- (Bun and Thaler 2018): Different refinements, showing that there is an $\text{AC}^0$ circuit of depth $O(1/\delta)$ and $\text{PP}^{cc}(F) \geq n^{1-\delta}$. 

Conjecture: For any constant $\delta > 0$, there is a depth-3 $\text{AC}^0$ circuit $F$ with $\text{PP}^{cc}(F) \geq n^{1-\delta}$ (maybe even $\Omega(n)$).

Can we prove this by combining the techniques of this work with (Bun and Thaler, 2017/2018)?

Can we extend our lower bound for $O(\log n)$-decision lists to DNFs, answering the question of Klivans and Servedio?
(Bun and Thaler, 2017): A different hardness amplification technique that moves beyond block-composed functions.

- For any constant $\delta > 0$, yielded a nearly-optimal $\Omega(n^{1-\delta})$ lower bound on the approximate degree of $AC^0$ (specifically, depth $\log(1/\delta)$).

- Previous best lower bound for $AC^0$ was $\Omega(n^{2/3})$ (Aaronson and Shi, 2004).

- (Bun and Thaler 2018): Different refinements, showing that there is an $AC^0$ circuit of depth $O(1/\delta)$ and $\text{PP}^{cc}(F) \geq n^{1-\delta}$.

**Conjecture**: For any constant $\delta > 0$, there is a depth-3 $AC^0$ circuit $F$ with $\text{PP}^{cc}(F) \geq n^{1-\delta}$ (maybe even $\Omega(n)$).

- Can we prove this by combining the techniques of this work with (Bun and Thaler, 2017/2018)?
Subsequent Work and Open Questions

- (Bun and Thaler, 2017): A different hardness amplification technique that moves beyond block-composed functions.
  - For any constant $\delta > 0$, yielded a nearly-optimal $\Omega(n^{1-\delta})$ lower bound on the approximate degree of $\text{AC}^0$ (specifically, depth $\log(1/\delta)$).
  - Previous best lower bound for $\text{AC}^0$ was $\Omega(n^{2/3})$ (Aaronson and Shi, 2004).
- (Bun and Thaler 2018): Different refinements, showing that there is an $\text{AC}^0$ circuit of depth $O(1/\delta)$ and $\text{PP}^{cc}(F) \geq n^{1-\delta}$.

**Conjecture:** For any constant $\delta > 0$, there is a depth-$3$ $\text{AC}^0$ circuit $F$ with $\text{PP}^{cc}(F) \geq n^{1-\delta}$ (maybe even $\Omega(n)$).

- Can we prove this by combining the techniques of this work with (Bun and Thaler, 2017/2018)?

- Can we extend our lower bound for $O(\log n)$-decision lists to DNFs, answering the question of Klivans and Servedio?
Thank you!