IDENTITY TESTING & LOWER BOUNDS
FOR
READ-\(k\) OBLIVIOUS ABPS

Ben Lee Volk

Joint with
Matthew Anderson
Michael A. Forbes
Ramprasad Saptharishi
Amir Shpilka
READ-ONCE OBLIVIOUS ABPS
Each $s \rightarrow t$ path computes multiplication of edge labels.
Program computes the sum of those over all $s \rightarrow t$ paths.
Read Once: Each var appears in one layer.
Each $s \rightarrow t$ path computes multiplication of edge labels
Program computes the sum of those over all $s \rightarrow t$ paths
Read Once: Each var appears in one layer
Equivalent: $f$ is the $(1, 1)$ entry of the iterated matrix product

$$\prod_{i=1}^{n} M_i(x_{\pi(i)})$$
We know a lot about ROABPs :)

• Exponential lower bounds [Nisan]
• Poly-time white-box PIT [Raz-Shpilka]
• Quasipoly-size hitting sets even when order is unknown [Forbes-Shpilka, Forbes-Shpilka-Saptharishi, Agrawal-Gurjar-Korwar-Saxena]

... and also PIT for sums of ROABPs and bounded-width ROABPs (all in this workshop).

This talk is about read-$k$ oblivious ABPs. (def: same as before except that now every variable appears in at most $k$ layers)
We know a lot about ROABPs :)  

- Exponential lower bounds [Nisan]
We know a lot about ROABPs :)  

- Exponential lower bounds [Nisan]  
- Poly-time white-box PIT [Raz-Shpilka]
We know a lot about ROABPs :)  

- Exponential lower bounds [Nisan]  
- Poly-time white-box PIT [Raz-Shpilka]  
- Quasipoly-size hitting sets even when order is unknown [Forbes-Shpilka, Forbes-Shpilka-Saptharishi, Agrawal-Gurjar-Korwar-Saxena]  

This talk is about read-$k$ oblivious ABPs. (def: same as before except that now every variable appears in at most $k$ layers)
We know a lot about ROABPs :)  
- Exponential lower bounds [Nisan]  
- Poly-time white-box PIT [Raz-Shpilka]  
- Quasipoly-size hitting sets even when order is unknown [Forbes-Shpilka, Forbes-Shpilka-Saptharishi, Agrawal-Gurjar-Korwar-Saxena]  
... and also PIT for sums of ROABPs and bounded-width ROABPs (all in this workshop).
We know a lot about ROABPs :)  
- Exponential lower bounds \[\text{[Nisan]}\]  
- Poly-time white-box PIT \[\text{[Raz-Shpilka]}\]  
- Quasipoly-size hitting sets even when order is unknown \[\text{[Forbes-Shpilka, Forbes-Shpilka-Saptharishi, Agrawal-Gurjar-Korwar-Saxena]}\]  

... and also PIT for sums of ROABPs and bounded-width ROABPs (all in this workshop).

This talk is about **read-\(k\) oblivious ABPs**.
Some Things You’ve All Heard About

We know a lot about ROABPs :)  
- Exponential lower bounds [Nisan]  
- Poly-time white-box PIT [Raz-Shpilka]  
- Quasipoly-size hitting sets even when order is unknown [Forbes-Shpilka, Forbes-Shpilka-Saptharishi, Agrawal-Gurjar-Korwar-Saxena]

... and also PIT for sums of ROABPs and bounded-width ROABPs (all in this workshop).

This talk is about read-$k$ oblivious ABPs.  
(def: same as before except that now every variable appears in at most $k$ layers)
Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])

Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])

Well-studied boolean analog for read-$k$ oblivious boolean branching programs:

$\exp(n = 2^k)$ lower bounds [Okolnishnikova, Borodin-Razborov-Smolensky] even for randomized and non-deterministic variants

PRG with seed length $p_s$ for size-$s$ programs [Impagliazzo-Meka-Zuckerman]
• Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])

• Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])

• Well-studied boolean analog for read-$k$ oblivious boolean branching programs:
  - $\exp(n = 2^k)$ lower bounds [Okolnishnikova, Borodin-Razborov-Smolensky] even for randomized and non-deterministic variants

• PRG with seed length $p^s$ for size-$s$ programs [Impagliazzo-Meka-Zuckerman]
• Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])
• Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])
**READING $k$ TIMES**

- Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])
- Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])
- Well-studied boolean analog
- Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])
- Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])
- Well-studied boolean analog

For read-$k$ oblivious boolean branching programs:
• Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])

• Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])

• Well-studied \texttt{boolean} analog

For read-$k$ oblivious \texttt{boolean} branching programs:

• $\exp(n/2^k)$ lower bounds [Okolnishnikova, Borodin-Razborov-Smolensky] even for randomized and non-deterministic variants
READING $k$ TIMES

- Generalizes sum of $k$ ROABPs (PIT by [Gurjar, Korwar, Saxena, Thierauf])
- Generalizes read-$k$ formulas (PIT by [Anderson, van Melkbeek, Volkovich])
- Well-studied boolean analog

For read-$k$ oblivious boolean branching programs:

- $\exp(n/2^k)$ lower bounds [Okolnishnikova, Borodin-Razborov-Smolensky] even for randomized and non-deterministic variants
- PRG with seed length $\sqrt{s}$ for size-$s$ programs [Impagliazzo-Meka-Zuckerman]
Lower Bound: There is a polynomial $f$ of $\mathsf{VP}$ that requires read-$k$ oblivious ABPs of width $\exp(n^k)$. PIT: There is a white-box* PIT algorithm for read-$k$ oblivious ABPs, of running time $\exp(n^{1/2^k})$. *only the order in which the variables appear is important
**READ-**$k$ **OBLIVIOUS ABPS**

**Lower Bound:** There is a polynomial $f \in VP$ that requires read-$k$ oblivious ABPs of width $\exp(n/k^k)$. 

*only the order in which the variables appear is important*
READ-\(k\) OBLIVIOUS ABPS

**Lower Bound:** There is a polynomial \(f \in \text{VP}\) that requires read-\(k\) oblivious ABPs of width \(\exp(n/k^k)\).

**PIT:** There is a white-box* PIT algorithm for read-\(k\) oblivious ABPs, of running time \(\exp(n^{1-1/2^{k-1}})\).

*only the order in which the variables appear is important
**Reminder:** \( f \in \mathbb{F}[x_1, \ldots, x_n], \, S \subseteq [n]. \)

\[
eval\text{-dim}_{S, \overline{S}}(f) = \dim \, \text{span} \{ f |_{x_S = \alpha} \mid \alpha \in \mathbb{F}^{|S|} \}.
\]
**EVALUATION DIMENSION**

**Reminder:** $f \in \mathbb{F}[x_1, \ldots, x_n]$, $S \subseteq [n]$.

$$\text{eval-dim}_{S,\overline{S}}(f) = \dim \text{ span } \{ f|_{x_S = \alpha} \mid \alpha \in \mathbb{F}^{|S|} \}.$$

Characterizes ROABP complexity:

**Theorem [Nisan]:** $f$ has ROABP of width $w$ in variable order $x_1, x_2, \ldots, x_n$ iff for every $i \in [n],$

$$\text{eval-dim}_{[i],[i]}(f) \leq w.$$
**EVALUATION DIMENSION**

**Reminder:** \( f \in \mathbb{F}[x_1, \ldots, x_n], S \subseteq [n]. \)

\[
eval-dim_{S, \overline{S}}(f) = \dim \text{ span } \{f |_{x_S = \alpha} \mid \alpha \in \mathbb{F}^{|S|}\}.
\]

Characterizes ROABP complexity:

**Theorem [Nisan]:** \( f \) has ROABP of width \( w \) in variable order \( x_1, x_2, \ldots, x_n \) iff for every \( i \in [n], \)

\[
eval-dim_{[i], \overline{[i]}}(f) \leq w.
\]

(same as rank of partial derivative matrix)
WARM-UP: 2-PASS ABP

Same as ROABP but with two “passes”:

$$f = (M_1^1(x_1)M_2^1(x_2) \cdots M_n^1(x_n) \cdot M_1^2(x_1)M_2^2(x_2) \cdots M_n^2(x_n))_{(1,1)}$$
WARM-UP: 2-PASS ABP

Same as ROABP but with two “passes”:

\[
\begin{array}{cccccccc}
& x_1 & x_2 & \cdots & x_{n-1} & x_n & x_1 & x_2 & \cdots & x_{n-1} & x_n \\
\end{array}
\]

\[
f = \left( M_1^1(x_1)M_2^1(x_2) \cdots M_n^1(x_n) \cdot M_1^2(x_1)M_2^2(x_2) \cdots M_n^2(x_n) \right)_{(1,1)}
\]

Fixing \( x_1 = \alpha_1 \):

\[
\left( N_1^1(\alpha_1)M_2^1(x_2) \cdots M_n^1(x_n) \cdot N_2^2(\alpha_1)M_2^2(x_2) \cdots M_n^2(x_n) \right)_{(1,1)}
\]
WARM-UP: 2-PASS ABP

Same as ROABP but with two “passes”:

\[
\begin{array}{cccccccccccc}
  x_1 & x_2 & \cdots & x_{n-1} & x_n & x_1 & x_2 & \cdots & x_{n-1} & x_n \\
\end{array}
\]

\[
f = \left( M_1^1(x_1)M_2^1(x_2)\cdots M_n^1(x_n) \cdot M_1^2(x_1)M_2^2(x_2)\cdots M_n^2(x_n) \right)_{(1,1)}
\]

Fixing \( x_1 = \alpha_1, x_2 = \alpha_2 \):

\[
\left( N_1^1(\alpha_1, \alpha_2)M_3^1(x_3)\cdots M_n^1(x_n) \cdot N_2^2(\alpha_1, \alpha_2)M_3^2(x_3)\cdots M_n^2(x_n) \right)_{(1,1)}
\]
WARM-UP: 2-PASS ABP

Same as ROABP but with two “passes”:

\[
f = (M_1^1(x_1)M_2^1(x_2)\cdots M_n^1(x_n) \cdot M_1^2(x_1)M_2^2(x_2)\cdots M_n^2(x_n))_{(1,1)}
\]

Fixing \(x_1, x_2, \ldots, x_i\):

\[
f|_{x_i} = (N_1^1(\alpha_1, \ldots, \alpha_i)M_{i+1}^1(x_{i+1})\cdots M_n^1(x_n)
\]

\[
N_2^2(\alpha_1, \ldots, \alpha_i)M_{i+1}^2(x_{i+1})\cdots M_n^2(x_n))_{(1,1)}
\]
WARM-UP: 2-PASS ABP

Same as ROABP but with two “passes”:

\[
\begin{array}{cccccccc}
  x_1 & x_2 & \cdots & x_{n-1} & x_n & x_1 & x_2 & \cdots & x_{n-1} & x_n \\
\end{array}
\]

\[
f = (M^1_1(x_1)M^1_2(x_2) \cdots M^1_n(x_n) \cdot M^2_1(x_1)M^2_2(x_2) \cdots M^2_n(x_n))_{(1,1)}
\]

Fixing \(x_1, x_2, \ldots, x_i\):

\[
f |_{x[i]=\alpha} = \left(N^1(\alpha_1, \ldots, \alpha_i)M^1_{i+1}(x_{i+1}) \cdots M^1_n(x_n) \right.
\]

\[
\left. N^2(\alpha_1, \ldots, \alpha_i)M^2_{i+1}(x_{i+1}) \cdots M^2_n(x_n) \right)_{(1,1)}
\]

Every restriction determined by \(N^1, N^2\) that have \(w^2\) entries.
### WARM-UP: 2-PASS ABP

Same as ROABP but with two “passes”:

\[
f = (M_1^1(x_1)M_2^1(x_2) \cdots M_n^1(x_n) \cdot M_1^2(x_1)M_2^2(x_2) \cdots M_n^2(x_n))(1,1)
\]

Fixing \(x_1, x_2, \ldots, x_i\):

\[
f|_{x[i]=a} = (N_1^1(\alpha_1, ..., \alpha_i)M_{i+1}^1(x_{i+1}) \cdots M_n^1(x_n) \\
N_2^1(\alpha_1, ..., \alpha_i)M_{i+1}^2(x_{i+1}) \cdots M_n^2(x_n))(1,1)
\]

Every restriction determined by \(N_1^1, N_2^1\) that have \(w^2\) entries. So eval-dim\(_{[i],[i]}(f) \leq w^4\).
**WARM-UP: 2-PASS ABP**

Same as ROABP but with two “passes”:

\[
\begin{array}{cccccccc}
  x_1 & x_2 & \cdots & x_{n-1} & x_n & x_1 & x_2 & \cdots & x_{n-1} & x_n \\
\end{array}
\]

\[
f = \left( M_1^1(x_1)M_2^1(x_2) \cdots M_n^1(x_n) \cdot M_1^2(x_1)M_2^2(x_2) \cdots M_n^2(x_n) \right)_{(1,1)}
\]

Fixing \( x_1, x_2, \ldots, x_i \):

\[
f|_{x[i]=\alpha} = \left( N^1(\alpha_1, \ldots, \alpha_i)M_{i+1}^1(x_{i+1}) \cdots M_n^1(x_n) \\
N^2(\alpha_1, \ldots, \alpha_i)M_{i+1}^2(x_{i+1}) \cdots M_n^2(x_n) \right)_{(1,1)}
\]

Every restriction determined by \( N^1, N^2 \) that have \( w^2 \) entries. So \( \text{eval-dim}_{[i],[i]}(f) \leq w^4 \). \( \implies f \) has width \( w^4 \) ROABP.
**Theorem:** If $f$ is computed by a width-$w$ $k$-pass ABP in variable order $x_1, x_2, \ldots, x_n$, then for every $i \in [n]$, eval-dim$_{[i],[i]}(f) \leq w^{2k}$. 

Up next: 2-pass, different order.
**GENERALIZE: \( k \)-PASS ABP**

**Theorem:** If \( f \) is computed by a width-\( w \) \( k \)-pass ABP in variable order \( x_1, x_2, \ldots, x_n \), then for every \( i \in [n] \), \( \text{eval-dim}_{[i], [i]}(f) \leq w^{2k} \).

In particular, \( f \) is computed by a ROABP of width \( w^{2k} \).
**GENERALIZE: \( k \)-PASS ABP**

**Theorem:** If \( f \) is computed by a width-\( w \) \( k \)-pass ABP in variable order \( x_1, x_2, \ldots, x_n \), then for every \( i \in [n] \), \( \text{eval-dim}_{[i],[i]}(f) \leq w^{2k} \).

In particular, \( f \) is computed by a ROABP of width \( w^{2k} \).

\( \implies \) Exp. lower bounds and quasi-poly PIT for \( k \)-pass ABPs.
**Theorem:** If $f$ is computed by a width-$w$ $k$-pass ABP in variable order $x_1, x_2, \ldots, x_n$, then for every $i \in [n]$, eval-$\text{dim}_{[i], [i]}(f) \leq w^{2k}$.

In particular, $f$ is computed by a ROABP of width $w^{2k}$.

$\implies$ Exp. lower bounds and quasi-poly PIT for $k$-pass ABPs.

Up next: 2-pass, different order.
Theorem: If $f$ is computed by a width-$w$ $k$-pass ABP in variable order $x_1, x_2, \ldots, x_n$, then for every $i \in [n]$, $\text{eval-dim}_{[i], [i]}(f) \leq w^{2k}$.

In particular, $f$ is computed by a ROABP of width $w^{2k}$.

$\implies$ Exp. lower bounds and quasi-poly PIT for $k$-pass ABPs.

Up next: 2-pass, different order.

this is already exponentially more powerful than ROABPs and even sums of ROABPs:
**Generalize: $k$-Pass ABP**

**Theorem:** If $f$ is computed by a width-$w$ $k$-pass ABP in variable order $x_1, x_2, \ldots, x_n$, then for every $i \in [n]$, $\text{eval-dim}_{[i],[i]}(f) \leq w^{2k}$.

In particular, $f$ is computed by a ROABP of width $w^{2k}$.

$\implies$ Exp. lower bounds and quasi-poly PIT for $k$-pass ABPs.

Up next: 2-pass, different order.

This is already exponentially more powerful than ROABPs and even sums of ROABPs: $\exists$ a polynomial computed by a 2-pass ABP with different orders that requires exponential width when computed as a sum of ROABPs.
2-PASS, DIFFERENT ORDER

| $x_1$ | $x_2$ | $\cdots$ | $x_{n-1}$ | $x_n$ | $x_8$ | $x_n$ | $\cdots$ | $x_2$ | $x_{n/2}$ |
**Theorem [Erdős-Szekeres]:** Every sequence of $n$ integers has a monotone subsequence of length $\sqrt{n}$. 

| $x_1$ | $x_2$ | $\cdots$ | $x_{n-1}$ | $x_n$ | $x_8$ | $x_n$ | $\cdots$ | $x_2$ | $x_{n/2}$ |
2-PASS, DIFFERENT ORDER

\[
\begin{array}{cccccccc}
  x_1 & x_2 & \cdots & x_{n-1} & x_n & x_8 & x_n & \cdots & x_2 & x_{n/2}
\end{array}
\]

**Theorem [Erdős-Szekeres]:** Every sequence of \( n \) integers has a monotone subsequence of length \( \sqrt{n} \).
Theorem [Erdős-Szekeres]: Every sequence of \( n \) integers has a monotone subsequence of length \( \sqrt{n} \).

Think of the ABP as computing a polynomial in the \( y \) vars over \( \mathbb{F}(\bar{y}) \) (i.e. all others vars are now “constants”)

<table>
<thead>
<tr>
<th>( x_1 )</th>
<th>( x_2 )</th>
<th>( \cdots )</th>
<th>( x_{n-1} )</th>
<th>( x_n )</th>
<th>( x_8 )</th>
<th>( x_n )</th>
<th>( \cdots )</th>
<th>( x_2 )</th>
<th>( x_{n/2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( y_1 ), ( \cdots ), ( y_{\sqrt{n}} )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2-PASS, DIFFERENT ORDER

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$\cdots$</th>
<th>$x_{n-1}$</th>
<th>$x_n$</th>
<th>$x_8$</th>
<th>$x_n$</th>
<th>$\cdots$</th>
<th>$x_2$</th>
<th>$x_{n/2}$</th>
</tr>
</thead>
</table>

Theorem [Erdős-Szekeres]: Every sequence of $n$ integers has a monotone subsequence of length $\sqrt{n}$.

Think of the ABP as computing a polynomial in the $y$ vars over $F(y)$ (i.e. all others vars are now “constants”)

What you get is a 2-pass ABP over $y$ vars.
2-PASS, DIFFERENT ORDER

<table>
<thead>
<tr>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$\cdots$</th>
<th>$x_{n-1}$</th>
<th>$x_n$</th>
<th>$x_8$</th>
<th>$x_n$</th>
<th>$\cdots$</th>
<th>$x_2$</th>
<th>$x_{n/2}$</th>
</tr>
</thead>
</table>

$y_1, \cdots, y_{\sqrt{n}}$

**Theorem [Erdős-Szekeres]:** Every sequence of $n$ integers has a monotone subsequence of length $\sqrt{n}$.

Think of the ABP as computing a polynomial in the $y$ vars over $\mathbb{F}(\bar{y})$ (i.e. all others vars are now “constants”)

What you get is a 2-pass ABP over $y$ vars. In other words, ignoring $\bar{y}$, for every $i \in [\sqrt{n}]$, eval-$\dim_{[i],\overline{[i]}}(f) \leq w^4$. 
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:

1. Find monotone subsequence $y$ of length $p$.
2. Plug-in hitting set for width $w$ to $y$.
3. Repeat with $y$ (plugging in a fresh copy of the hitting set each time).
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:

1. Find monotone subsequence \( y \) of length \( \sqrt{n} \)
PIT for 2-pass, different order

PIT algorithm:

1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:

1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\overline{y}$
   (plugging in a fresh copy of the hitting set each time)
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:

1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\bar{y}$
   (plugging in a fresh copy of the hitting set each time)
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:
1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\bar{y}$
   (plugging in a fresh copy of the hitting set each time)
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:

1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\bar{y}$
   (plugging in a fresh copy of the hitting set each time)
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:
1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\overline{y}$
   (plugging in a fresh copy of the hitting set each time)
PIT FOR 2-PASS, DIFFERENT ORDER

**PIT algorithm:**

1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\bar{y}$
   (plugging in a fresh copy of the hitting set each time)
PIT FOR 2-PASS, DIFFERENT ORDER

**PIT algorithm:**

1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\bar{y}$
   (plugging in a fresh copy of the hitting set each time)

**Running Time:** In total, $\approx \sqrt{n}$ copies of a $n^{\log n}$ size hitting set

$\rightarrow \approx n^{\sqrt{n}}$
PIT FOR 2-PASS, DIFFERENT ORDER

PIT algorithm:
1. Find monotone subsequence $y$ of length $\sqrt{n}$
2. Plug-in hitting set for width $w^4$ ROABPs to $y$
3. Repeat with $\overline{y}$
   (plugging in a fresh copy of the hitting set each time)

Running Time: In total, $\approx \sqrt{n}$ copies of a $n^{\log n}$ size hitting set
$\implies \approx n^{\sqrt{n}}$

Naturally generalizes to $k$ passes with different orders.
By repeatedly applying the Erdős-Szekeres theorem, we can find a subsequence of size $n^{1/2^{k-1}}$ which is monotone in each of the $k$ passes.
By repeatedly applying the Erdős-Szekeres theorem, we can find a subsequence of size $n^{1/2^{k-1}}$ which is monotone in each of the $k$ passes.
By repeatedly applying the Erdős-Szekeres theorem, we can find a subsequence of size $n^{1/2^{k-1}}$ which is monotone in each of the $k$ passes.

Same algorithm gives $n^{n^{1-1/2^{k-1}}}$ hitting set.
PIT FOR $k$-PASS, DIFFERENT ORDERS

By repeatedly applying the Erdős-Szekeres theorem, we can find a subsequence of size $n^{1/2^{k-1}}$ which is monotone in each of the $k$ passes.

Same algorithm gives $n^{n^{1-1/2^{k-1}}}$ hitting set.

This is still not a general read-$k$ oblivious ABP!
Begin by applying Erdős-Szekeres. Monotone sequences are not disjoint... BUT we can find a large set of the variables such that the resulting sequence is "regularly-interleaving":

first $X_1$ second $X_1$

first $X_2$ second $X_2$

first $X_t$ second $X_t$
Begin by applying Erdős-Szekeres.
Begin by applying Erdős-Szekeres.

| $x_1$ | $x_2$ | $x_3$ | $x_4$ | $x_1$ | $x_2$ | $x_5$ | $x_6$ | $x_3$ | $\ldots$ |

Monotone sequences are not disjoint...
READ-TWICE OBLIVIOUS ABPS

Begin by applying Erdős-Szekeres.

Monotone sequences are not disjoint...
BUT we can find a large set of the variables such that the resulting sequence is “regularly-interleaving”: 

$\begin{array}{cccccccc}
  x_1 & x_2 & x_3 & x_4 & x_1 & x_2 & x_5 & x_6 & x_3 & \cdots
\end{array}$
Begin by applying Erdős-Szekeres.

Monotone sequences are not disjoint...
BUT we can find a large set of the variables such that the resulting sequence is “regularly-interleaving”: 
This structure is enough to carry out the original argument: with respect to the variables $y$ in the regularly interleaving sequence ($|y| \approx \sqrt{n}$), the evaluation dimension is at most $w^4$. 

**REGULARLY INTERLEAVING SUBSEQUENCES**
This structure is enough to carry out the original argument: with respect to the variables $y$ in the regularly interleaving sequence ($|y| \approx \sqrt{n}$), the evaluation dimension is at most $w^4$.

Generalizes to read-$k$: apply Erdős-Szekeres to every sequence and make every pair regularly-interleaving.

Wrap-up: PIT algorithm with running time $\exp(n^{1-1/2^{k-1}})$ for read-$k$ oblivious ABPs.
• These arguments are sufficient to get a lower bound of roughly $\exp(n^{1/2^k})$
LOWER BOUNDS FOR READ-\(k\)

- These arguments are sufficient to get a lower bound of roughly \(\exp(n^{1/2^k})\)
- But actually, for a lower bound we don’t need to show that for every prefix \([i]\) the eval-dimension is small: it’s enough to show it is small for some prefix \([i]\)
These arguments are sufficient to get a lower bound of roughly $\exp(n^{1/2^k})$.

But actually, for a lower bound we don’t need to show that for every prefix $[i]$ the eval-dimension is small: it’s enough to show it is small for some prefix $[i]$.

That is, to show that if $f$ is computed by a read-$k$ oblivious ABP, then there is $i$ such that $\text{eval-dim}_{[i],[i]}(f) \leq w^{2^k}$.
Lower Bounds for Read-$k$

- These arguments are sufficient to get a lower bound of roughly $\exp(n^{1/2^k})$.
- But actually, for a lower bound we don’t need to show that for every prefix $[i]$ the eval-dimension is small: it’s enough to show it is small for some prefix $[i]$.
- That is, to show that if $f$ is computed by a read-$k$ oblivious ABP, then there is $i$ such that $\text{eval-dim}_{[i],[i]}(f) \leq w^{2k}$.
- This is very close to being true.
Claim: We can fix $n/10$ variables and partition the remaining to subsets $S$, $T$ with $|S|, |T| \geq n/k^k$ and $\text{eval-dim}_{S,T}(f) \leq w^{2k}$
Claim: We can fix \( n/10 \) variables and partition the remaining to subsets \( S, T \) with \( |S|, |T| \geq n/k^k \) and \( \text{eval-dim}_{S,T}(f) \leq w^{2k} \)

Proof: Partition program into \( r \) contiguous blocks.
Claim: We can fix $n/10$ variables and partition the remaining to subsets $S$, $T$ with $|S|, |T| \geq n/k^k$ and $\text{eval-dim}_{S,T}(f) \leq w^{2k}$

Proof: Partition program into $r$ contiguous blocks.
Claim: We can fix $n/10$ variables and partition the remaining to subsets $S, T$ with $|S|, |T| \geq n/k^k$ and $\text{eval-dim}_{S,T}(f) \leq w^{2k}$.

Proof: Partition program into $r$ contiguous blocks.
**Claim:** We can fix \( n/10 \) variables and partition the remaining to subsets \( S, T \) with \( |S|, |T| \geq n/k^k \) and eval-dim\(_{S,T}(f) \leq w^{2k} \)

**Proof:** Partition program into \( r \) contiguous blocks.

By averaging, \( \exists k \) blocks that contain all reads of \( n/\binom{r}{k} \) vars.
Claim: We can fix \( n/10 \) variables and partition the remaining to subsets \( S, T \) with \( |S|, |T| \geq n/k^k \) and eval-dim\(_{S,T}(f) \leq w^{2k} \)

Proof: Partition program into \( r \) contiguous blocks.

By averaging, \( \exists k \) blocks that contain all reads of \( n/\binom{r}{k} \) vars.
**EXPO\-NEN\-TIAL LOWER BOUND**

**Claim:** We can fix \( n/10 \) variables and partition the remaining to subsets \( S, T \) with \( |S|, |T| \geq n/k^k \) and \( \text{eval-dim}_{S,T}(f) \leq w^{2k} \)

**Proof:** Partition program into \( r \) contiguous blocks.

By averaging, \( \exists k \) blocks that contain all reads of \( n/r \binom{r}{k} \) vars. Call them \( S \) and fix all other vars in those blocks.
Claim: We can fix $n/10$ variables and partition the remaining to subsets $S, T$ with $|S|, |T| \geq n/k^k$ and $\text{eval-dim}_{S,T}(f) \leq w^{2k}$

Proof: Partition program into $r$ contiguous blocks.

By averaging, $\exists k$ blocks that contain all reads of $n/\binom{r}{k}$ vars. Call them $S$ and fix all other vars in those blocks.

$T =$ all remaining variables. Now compute $\text{eval-dim}_{S,T}$ using previous arguments.
Claim: We can fix $n/10$ variables and partition the remaining to subsets $S, T$ with $|S|, |T| \geq n/k^k$ and $\text{eval-dim}_{S,T}(f) \leq w^{2k}$

Proof: Partition program into $r$ contiguous blocks.

By averaging, $\exists k$ blocks that contain all reads of $n/\binom{r}{k}$ vars. Call them $S$ and fix all other vars in those blocks.

$T = \text{all remaining variables}$. Now compute $\text{eval-dim}_{S,T}$ using previous arguments.

if $r = 10k^2$ we fix at most $n/10$ vars and $|S| \geq n/k^k$. 
Claim: We can fix $n/10$ variables and partition the remaining to subsets $S$, $T$ with $|S|, |T| \geq n/k^k$ and $\text{eval-dim}_{S,T}(f) \leq w^{2k}$

Proof: Partition program into $r$ contiguous blocks. By averaging, $\exists k$ blocks that contain all reads of $n/{r \choose k}$ vars. Call them $S$ and fix all other vars in those blocks.

$T = \text{all remaining variables}$. Now compute $\text{eval-dim}_{S,T}$ using previous arguments.

if $r = 10k^2$ we fix at most $n/10$ vars and $|S| \geq n/k^k$.

what's left is to find a polynomial such that $\text{eval-dim}_{S,T} \geq 2^\min\{|S|, |T|\}$
Lower Bound: An \( \exp(n/k) \) lower bound on any read-
\( k \)-oblivious ABP computing some polynomial \( f \in \text{VP} \).

PIT: A white-box PIT algorithm for read-
\( k \)-oblivious ABPs, with running time \( \exp(n/1^{1/2}k) \).
**Summary**

**Lower Bound:** An $\exp(n/k^k)$ lower bound on any read-$k$ oblivious ABP computing some polynomial $f \in VP$. 
**Summary**

**Lower Bound:** An $\exp(n/k^k)$ lower bound on any read-$k$ oblivious ABP computing some polynomial $f \in VP$.

**PIT:** A white-box PIT algorithm for read-$k$ oblivious ABPs, with running time $\exp(n^{1-1/2^{k-1}})$. 
OPEN PROBLEMS

• Faster PIT algorithm
• A complete black-box test (no dependence on order)
• "Tighter" lower bounds (e.g. a hierarchy theorem for read-
  $k$ ABPs)
• Non-oblivious? (open even for $k = 1$)
• Connections with pseudorandomness for boolean branching

Thank You
OPEN PROBLEMS

• Faster PIT algorithm
OPEN PROBLEMS

- Faster PIT algorithm
- A complete black-box test (no dependence on order)

Thank You
OPEN PROBLEMS

- Faster PIT algorithm
- A complete black-box test (no dependence on order)
- “Tighter” lower bounds (e.g. a hierarchy theorem for read-$k$ ABPs)
OPEN PROBLEMS

- Faster PIT algorithm
- A complete black-box test (no dependence on order)
- “Tighter” lower bounds (e.g. a hierarchy theorem for read-\(k\) ABPs)
- Non-oblivious? (open even for \(k = 1\))
OPEN PROBLEMS

• Faster PIT algorithm
• A complete black-box test (no dependence on order)
• “Tighter” lower bounds (e.g. a hierarchy theorem for read-$k$ ABPs)
• Non-oblivious? (open even for $k = 1$)
• Connections with pseudorandomness for boolean branching programs?

THANK YOU