A Tight Lower Bound for Entropy Flattening

Yi-Hsiu Chen\textsuperscript{1} Mika Göös\textsuperscript{1} Salil Vadhan\textsuperscript{1} Jiapeng Zhang\textsuperscript{2}

\textsuperscript{1}Harvard University, USA
\textsuperscript{2}UC San Diego, USA

June 23, 2018
Agenda

1. Problem Definition / Model
2. Cryptographic Motivations
3. Proof Techniques
Flatness

**Definition (Entropies)**

Let $X$ be a distribution over $\{0, 1\}^n$. Define the surprise of $x$ to be $H_X(x) = \log(1/\Pr[X = x])$.

$$H_{sh}(X) \overset{\text{def}}{=} \mathbb{E}_{x \sim X} [H_X(x)],$$

$$H_{\min}(X) \overset{\text{def}}{=} \min_x H_X(x),$$

$$H_{\max}(X) \overset{\text{def}}{=} \log |\text{Supp } X| \leq \max_x H_X(x).$$

- $H_{\min}(X) \leq H_{sh}(x) \leq H_{\max}(X)$ \quad (The gap can be $\Theta(n)$.)
- A source $X$ is **flat** iff $H_{sh}(X) = H_{\min}(X) = H_{\max}(X)$. 
Entropy Flattening

\[
\text{Input source } X \xrightarrow{\text{Flattening Algorithm } A} \text{Output source } Y
\]

\[
(H_{sh}(Y) \approx H_{\min}(Y) \approx H_{\max}(Y))
\]

nearly flat
Entropy Flattening

- Entropies of the output and input sources are monotonically related.

\[ H_{sh}(Y) \approx H_{\text{min}}(Y) \approx H_{\text{max}}(Y) \]
Entropy Flattening

Input source $X$ \quad Flattening Algorithm $A$ \quad Output source $Y$

- Entropies of the output and input sources are monotonically related.

$nearly flat$

$(H_{sh}(Y) \approx H_{\min}(Y) \approx H_{\max}(Y))$

- Shannon gap
- Flattening
- $\min/\max$ gap

\[ X_L \quad X_H \quad Y_L \quad Y_H \]
Entropy Flattening

Entropy Flattening Problem

Find an flattening algorithm $A$:

- If $H_{sh}(X) \geq \tau + 1$, then $H_{\varepsilon \text{min}}(Y) \geq k + \Delta$.
- If $H_{sh}(X) \leq \tau - 1$, then $H_{\varepsilon \text{max}}(Y) \leq k - \Delta$. 

Smooth Entropies

$H_{\varepsilon \text{min}}(Y) \geq k$ if $\exists Y'$ s.t. $H_{\text{min}}(Y) \geq k$ and $d_{TV}(Y,Y') \leq \varepsilon$.

$H_{\varepsilon \text{max}}(Y) \leq k$ if $\exists Y'$ s.t. $H_{\text{max}}(Y) \leq k$ and $d_{TV}(Y,Y') \leq \varepsilon$. 

**Entropy Flattening**

### Entropy Flattening Problem

Find an flattening algorithm $A$:

- If $H_{sh}(X) \geq \tau + 1$, then $H_{\min}^\varepsilon(Y) \geq k + \Delta$.
- If $H_{sh}(X) \leq \tau - 1$, then $H_{\max}^\varepsilon(Y) \leq k - \Delta$.

### Smooth Entropies

- $H_{\min}^\varepsilon(Y) \geq k$ if $\exists Y'$ s.t. $H_{\min}(Y) \geq k$ and $d_{TV}(Y, Y') \leq \varepsilon$.
- $H_{\max}^\varepsilon(Y) \leq k$ if $\exists Y'$ s.t. $H_{\max}(Y) \leq k$ and $d_{TV}(Y, Y') \leq \varepsilon$. 
Solution: Repetition

Theorem ([HILL99, HR11])

- \( X \): a distribution over \( \{0, 1\}^n \).
- Let \( Y = (X_1, \ldots, X_q) \) where \( X_i \)'s are i.i.d. copies of \( X \).

\[
H^\varepsilon_{\min}(Y), H^\varepsilon_{\max}(Y) \in H_{sh}(Y) \pm O \left( n \sqrt{q \log(1/\varepsilon)} \right) \\
q \cdot \left( H_{sh}(X) \pm O \left( n \sqrt{\frac{\log(1/\varepsilon)}{q}} \right) \right)
\]

(Asymptotic Equipartition Property (AEP) in information theory)
Solution: Repetition

Theorem ([HILL99, HR11])

- $X$: a distribution over $\{0, 1\}^n$.
- Let $Y = (X_1, \ldots, X_q)$ where $X_i$'s are i.i.d. copies of $X$.

\[
H^\varepsilon_{\text{min}}(Y), H^\varepsilon_{\text{max}}(Y) \in H_{\text{sh}}(Y) \pm O \left( n\sqrt{q \log(1/\varepsilon)} \right)
\]

\[
q \cdot \left( H_{\text{sh}}(X) \pm O \left( n\sqrt{\frac{\log(1/\varepsilon)}{q}} \right) \right)
\]

(Asymptotic Equipartition Property (AEP) in information theory)

- $q = O(n^2)$ is sufficient for the constant entropy gap.
- $q = \Omega(n^2)$ is needed due to anti-concentration results. [HR11]
Query Model

The Model:

- **Input source**: encoded by a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ and defined as $f(U_n)$.
- **Flattening algorithm**: oracle algorithm $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$ has query access to $f$.
- **Output source**: $A^f(U_{n'})$.
- **Example**: $A^f(r_1, \ldots, r_q) = (f(r_1), \ldots, f(r_q))$
Query Model

The Model:

- **Input source**: encoded by a function $f : \{0, 1\}^n \to \{0, 1\}^m$ and defined as $f(U_n)$.
- **Flattening algorithm**: oracle algorithm $A^f : \{0, 1\}^{n'} \to \{0, 1\}^{m'}$ has query access to $f$.
- **Output source**: $A^f(U_{n'})$.
- **Example**: $A^f(r_1, \ldots, r_q) = (f(r_1), \ldots, f(r_q))$

**Def: Flattening Algorithm**

- $H_{sh}(f(U_n)) \geq \tau + 1 \Rightarrow H_{\text{min}}^\varepsilon(A^f(U_{n'})) \geq k + \Delta$
- $H_{sh}(f(U_n)) \leq \tau - 1 \Rightarrow H_{\text{max}}^\varepsilon(A^f(U_{n'})) \leq k - \Delta$
Query Model

The Model:

- **Input source**: encoded by a function $f : \{0, 1\}^n \rightarrow \{0, 1\}^m$ and defined as $f(U_n)$.
- **Flattening algorithm**: oracle algorithm $A^f : \{0, 1\}^{n'} \rightarrow \{0, 1\}^{m'}$ has query access to $f$.
- **Output source**: $A^f(U_{n'})$.
- **Example**: $A^f(r_1, \ldots, r_q) = (f(r_1), \ldots, f(r_q))$

### Def: Flattening Algorithm

<table>
<thead>
<tr>
<th>Condition</th>
<th>Implication</th>
</tr>
</thead>
<tbody>
<tr>
<td>$H_{sh}(f(U_n)) \geq \tau + 1$</td>
<td>$H_{\min}^e(A^f(U_{n'})) \geq k + \Delta$</td>
</tr>
<tr>
<td>$H_{sh}(f(U_n)) \leq \tau - 1$</td>
<td>$H_{\max}^e(A^f(U_{n'})) \leq k - \Delta$</td>
</tr>
</tbody>
</table>

More powerful:

- Querying correlated positions or even in an adaptive way.
- Computation on the query inputs. e.g., hashing.
Theorem

Flattening algorithms for $n$-bit oracles $f$ require $\Omega(n^2)$ oracle queries.
Main Theorems

**Theorem**

Flattening algorithms for $n$-bit oracles $f$ require $\Omega(n^2)$ oracle queries.

**Def: SDU Algorithm**

- $H_{sh}(f(U_n)) \geq \tau + 1 \implies d_{TV}(A^f(U_{n'}), U_{m'}) < \varepsilon$.
- $H_{sh}(f(U_n)) \leq \tau - 1 \implies \text{Supp} \left( A^f(U_{n'}) \right)/2^{m'} \leq \varepsilon$.

Flattening Algorithm $\iff$ SDU Algorithm

(Reduction between two NISZK-complete problems [GSV99])

**Theorem**

SDU algorithms for $n$-bit oracles $f$ require $\Omega(n^2)$ oracle queries.
Example: OWF $f \rightarrow$ PRG $g^f$ ([HILL90, Hol06, HHR06, HRV10, VZ13]):

1. Create a gap between “pseudoentropy” and (true) entropy.
2. Guess the entropy threshold $\tau$ (or other tricks).
3. Flatten entropies.
4. Extract the pseudorandomness (via universal hashing).
Connection to Cryptographic Constructions

Example: OWF $f \rightarrow$ PRG $g^f$ ([HILL90, Hol06, HHR06, HRV10, VZ13]):

1. Create a gap between “pseudoentropy” and (true) entropy.
2. Guess the entropy threshold $\tau$ (or other tricks). $\tilde{O}(n)$ queries
3. Flatten entropies. $\tilde{O}(n^2)$ queries
4. Extract the pseudorandomness (via universal hashing).

- Overall, the best PRG makes $\tilde{O}(n^3)$ queries to the one-way function [HRV10, VZ13].
- From regular one-way function, Step 3 is unnecessary, and so $\tilde{O}(n)$ query is sufficient. [HHR06]
Connection to Cryptographic Constructions

**Example:** OWF $f \rightarrow$ PRG $g^f$ ([HILL90, Hol06, HHR06, HRV10, VZ13]):

1. Create a gap between “pseudoentropy” and (true) entropy.
2. Guess the entropy threshold $\tau$ (or other tricks). $\tilde{O}(n)$ queries
3. Flatten entropies. $\tilde{O}(n^2)$ queries
4. Extract the pseudorandomness (via universal hashing).

- Overall, the best PRG makes $\tilde{O}(n^3)$ queries to the one-way function [HRV10, VZ13].
- From regular one-way function, Step 3 is unnecessary, and so $\tilde{O}(n)$ query is sufficient. [HHR06]

- Holenstein and Sinha ([HS12]) prove that any black-box construction requires $\tilde{\Omega}(n)$ queries. (From Step 2. Applicable to regular OWF)
Connection to Cryptographic Constructions

**Example:** OWF $f \rightarrow$ PRG $g^f$ ([HILL90, Hol06, HHR06, HRV10, VZ13]):

1. Create a gap between “pseudoentropy” and (true) entropy.
2. Guess the entropy threshold $\tau$ (or other tricks). $\tilde{O}(n)$ queries
3. Flatten entropies. $\tilde{O}(n^2)$ queries
4. Extract the pseudorandomness (via universal hashing).

   - Overall, the best PRG makes $\tilde{O}(n^3)$ queries to the one-way function [HRV10, VZ13].
   - From regular one-way function, Step 3 is unnecessary, and so $\tilde{O}(n)$ query is sufficient. [HHR06]

---

- Holenstein and Sinha ([HS12]) prove that any black-box construction requires $\tilde{\Omega}(n)$ queries. (From Step 2. Applicable to regular OWF)

Can we do better in the entropy flattening step?
Overview of the Proof

Def: SDU Algorithm

- \( H_{sh}(f(U_n)) \geq \tau + 1 \) \( \Rightarrow \) \( d_{TV}(A^f(U_{n'}), U_{m'}) < \varepsilon \).
- \( H_{sh}(f(U_n)) \leq \tau - 1 \) \( \Rightarrow \) \( \text{Supp}(A^f(U_{n'}))/2^{m'} \leq \varepsilon \).

1. Construct distributions \( D_H \) and \( D_L \):
   - Sample \( f \) from \( D_H \), then \( H_{sh}(f(U_n)) \geq \tau + 1 \) w.h.p.
   - Sample \( f \) from \( D_L \), then \( H_{sh}(f(U_n)) \leq \tau - 1 \) w.h.p.

2. \( A \) cannot “behave very different” on both distributions by making only \( q = o(n^2) \) queries.
Construction of $f$

- Partition the domain into $s$ blocks, each with $t$ elements ($s \cdot t = 2^n$)
  - Concentrated: map to the same element.
  - Scattered: map to all distinct elements.
Construction of $f$

- Partition the domain into $s$ blocks, each with $t$ elements ($s \cdot t = 2^n$)
  - Concentrated: map to the same element.
  - Scattered: map to all distinct elements.

\[ f \{0, 1\}^n \downarrow \{0, 1\}_m \]

\[ \begin{array}{cccccccc}
\text{Sca} & \text{Sca} & \text{Con} & \text{Sca} & \cdots & \text{Con} & \text{Sca} \\
\end{array} \]

\[ 2^{3n/4} \text{ blocks} \]

\[ 2^{n/4} \text{ elements} \]

- $\geq s \cdot (1/2 + 4/n)$ blocks are scattered $\Rightarrow H_{\text{sh}}(f) \geq 7n/8 + 1$
- $\leq s \cdot (1/2 - 4/n)$ blocks are scattered $\Rightarrow H_{\text{sh}}(f) \leq 7n/8 - 1$
\( \mathcal{D}_H \) and \( \mathcal{D}_L \)

Randomly partition \( \{0, 1\}^n \) into \( 2^{3n/4} \) blocks.

1. Randomly partition \( \{0, 1\}^n \) into \( 2^{3n/4} \) blocks.
**$\mathcal{D}_H$ and $\mathcal{D}_L$**

1. Randomly partition $\{0, 1\}^n$ into $2^{3n/4}$ blocks.
2. Decide each block to be scattered or concentrated.
   - $\mathcal{D}_H$: scattered with probability $(1/2 + 5/n)$, then w.h.p., $\geq s \cdot (1/2 + 4/n)$ blocks are scattered
   - $\mathcal{D}_L$: scattered with probability $(1/2 - 5/n)$, then w.h.p., $\leq s \cdot (1/2 - 4/n)$ blocks are scattered
$\mathcal{D}_H$ and $\mathcal{D}_L$

1. Randomly partition $\{0, 1\}^n$ into $2^{3n/4}$ blocks.
2. Decide each block to be scattered or concentrated.
   - $\mathcal{D}_H$: scattered with probability $(1/2 + 5/n)$,
     then w.h.p., $\geq s \cdot (1/2 + 4/n)$ blocks are scattered
   - $\mathcal{D}_L$: scattered with probability $(1/2 - 5/n)$,
     then w.h.p., $\leq s \cdot (1/2 - 4/n)$ blocks are scattered
3. Random mapping:
   - *Randomly* map each element in a scattered block.
   - Map all $t$ elements in a concentrated block to a *random* target.
Fix an SDU algorithm $A^{(\cdot)}$. An input $w$ is **block-compatible** (B.C.) for $f$ if each block is queried (when evaluating $A^f(w)$) at most once.
Intuitions for the Hard Distributions

Fix an SDU algorithm $A^\cdot$. An input $w$ is block-compatible (B.C.) for $f$ if each block is queried (when evaluating $A^f(w)$) at most once.

Why random partition?

- Hard to make correlated queries.
- When partitioning in many $(2^{3n/4})$ blocks, it is block-compatible w.h.p over $f$. 
Intuitions for the Hard Distributions

Fix an SDU algorithm $A^{(\cdot)}$. An input $w$ is \textbf{block-compatible} (B.C.) for $f$ if each block is queried (when evaluating $A^f(w)$) at most once.

Why random partition?
- Hard to make correlated queries.
- When partitioning in many ($2^{3n/4}$) blocks, it is block-compatible w.h.p over $f$.

Why random mapping?
- Conditioning on B.C., an algorithm cannot distinguish scattered or concentrated blocks.
- $O(n)$ queries is sufficient if the algorithm knows the block is scattered or concentrated!
Proof Overview

We will focus on the event \( \exists \text{B.C. } w, A^f(w) = z \).

By the definition of SDU algorithm, there exists \( z \in \{0, 1\} \),

\[
\Pr_{f \sim D} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \geq 1 - \varepsilon \geq \Theta(1)
\]

\[
\Pr_{f \sim D} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \leq \varepsilon
\]

which concludes that \( q = \Omega(n^2) \) (or \( \Omega(n^2 \log(1/\varepsilon)) \)).
Proof Overview

We will focus on the event \( \exists \text{B.C. } w, A^f(w) = z \).

By the definition of SDU algorithm, there exists \( z \in \{0, 1\}^{m'} \) (most \( z \)),

\[
\Pr_{f \sim \mathcal{D}_H} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \geq 1 - \varepsilon \geq \Theta(1)
\]

\[
\Pr_{f \sim \mathcal{D}_L} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \leq \varepsilon
\]
We will focus on the event \( \exists \text{B.C. } w, A^f(w) = z \).

By the definition of SDU algorithm, there exists \( z \in \{0, 1\}^{m'} \) (most \( z \)),

\[
\Pr_{f \sim \mathcal{D}_H} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \geq 1 - \varepsilon \geq \Theta(1)
\]

\[
\Pr_{f \sim \mathcal{D}_L} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \leq \varepsilon
\]

**Main Technical Lemma**

Suppose \( A^f \) algorithm makes \( q \) oracle queries, for most \( z \in \{0, 1\}^{m'} \),

\[
\Pr_{f \sim \mathcal{D}_H} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \leq 2^{O(\frac{q}{n^2})} \cdot \Pr_{f \sim \mathcal{D}_L} \left[ \exists \text{B.C. } w, A^f(w) = z \right] + o(\varepsilon)
\]
Proof Overview

We will focus on the event $\exists \text{B.C. } w, A^f(w) = z$.

By the definition of SDU algorithm, there exists $z \in \{0, 1\}^{m'}$ (most $z$),

$$\Pr_{f \sim \mathcal{D}_H} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \geq 1 - \varepsilon \geq \Theta(1)$$

$$\Pr_{f \sim \mathcal{D}_L} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \leq \varepsilon$$

Main Technical Lemma

Suppose $A^f$ algorithm makes $q$ oracle queries, for most $z \in \{0, 1\}^{m'}$,

$$\Pr_{f \sim \mathcal{D}_H} \left[ \exists \text{B.C. } w, A^f(w) = z \right] \leq 2^{O\left(\frac{q}{n^2}\right)} \cdot \Pr_{f \sim \mathcal{D}_L} \left[ \exists \text{B.C. } w, A^f(w) = z \right] + o(\varepsilon)$$

which concludes that $q = \Omega(n^2)$ (or $\Omega(n^2 \log(1/\varepsilon))$).
Primitive Intuition for Distinguishing $\mathcal{D}_L$ and $\mathcal{D}_H$

We flip coins to decide each block is scattered or concentrated.

$$\mathcal{D}_H : \text{Bern}(1/2 + 5/n) \quad \mathcal{D}_L : \text{Bern}(1/2 - 5/n)$$

- How many queries to distinguish two cases with constant probability?
Primitive Intuition for Distinguishing $\mathcal{D}_L$ and $\mathcal{D}_H$

We flip coins to decide each block is scattered or concentrated.

$\mathcal{D}_H : \text{Bern}(1/2 + 5/n)$ \quad $\mathcal{D}_L : \text{Bern}(1/2 - 5/n)$

- How many queries to distinguish two cases with constant probability?
- 1 query: $\frac{1/2 + 5/n}{1/2 - 5/n} \approx 1 + 20/n$
- $q$ queries: $(1 + 20/n)^q = 2^{O(q/n)}$
We flip coins to decide each block is scattered or concentrated.

\[ D_H : \text{Bern}(1/2 + 5/n) \quad D_L : \text{Bern}(1/2 - 5/n) \]

- How many queries to distinguish two cases with constant probability?
  - 1 query: \( \frac{1/2 + 5/n}{1/2 - 5/n} \approx 1 + 20/n \)
  - \( q \) queries: \( (1 + 20/n)^q = 2^{O(q/n)} \)
- We can afford more: w.h.p. fraction of scattered blocks \( \in \left( \frac{1}{2} \pm \frac{6}{n} \right) \).
Primitive Intuition for Distinguishing $D_L$ and $D_H$

We flip coins to decide each block is scattered or concentrated.

$$D_H : \text{Bern}(1/2 + 5/n) \quad D_L : \text{Bern}(1/2 - 5/n)$$

- How many queries to distinguish two cases with constant probability?
- 1 query: $\frac{1/2+5/n}{1/2-5/n} \approx 1 + 20/n$
- $q$ queries: $(1 + 20/n)^q = 2^{O(q/n)}$
- We can afford more: w.h.p. fraction of scattered blocks $\in \left(\frac{1}{2} \pm \frac{6}{n}\right)$.
- Conditioning on the “balance” event, the ratio is at most

$$\left(1 + \frac{20}{n}\right)^{q\cdot(1/2+6/n)} \times \left(1 - \frac{20}{n}\right)^{q\cdot(1/2-6/n)}$$

$$\leq \left(1 + \frac{20}{n}\right)^{12q/n} \times \left(1 - \frac{20}{n}\right)^{-12q/n} = 2^{O(q/n^2)}.$$
We flip coins to decide each block is scattered or concentrated.

\[ \mathcal{D}_H : \text{Bern}(1/2 + 5/n) \quad \mathcal{D}_L : \text{Bern}(1/2 - 5/n) \]

- How many queries to distinguish two cases with constant probability?
- 1 query: \( \frac{1/2+5/n}{1/2-5/n} \approx 1 + 20/n \)
- \( q \) queries: \((1 + 20/n)^q = 2^{O(q/n)}\)
- We can afford more: w.h.p. fraction of scattered blocks \( \in \left( \frac{1}{2} \pm \frac{6}{n} \right) \).
- Conditioning on the “balance” event, the ratio is at most

\[
(1 + 20/n)^q \cdot (1/2+6/n) \times (1 - 20/n)^q \cdot (1/2-6/n) \\
\leq (1 + 20/n)^{12q/n} \times (1 - 20/n)^{-12q/n} = 2^{O(q/n^2)}. 
\]

- **Warning!** To distinguish two cases in “NISZK”-sense (instead of BPP) \( O(n) \) queries are sufficient.
Comparison to [Lovett Zhang 17]

Entropy reversal: $A$ has to make **exponentially** many queries such that
- $f(U_n)$ has high entropy $\Rightarrow A^f(U_{n'})$ has small support.
- $f(U_n)$ has low entropy $\Rightarrow A^f(U_{n'})$ is close to uniform

(They ruled out efficient black-box reduction between SZK and NISZK)
Comparison to [Lovett Zhang 17]

Entropy reversal: \( A \) has to make \textbf{exponentially} many queries such that

- \( f(U_n) \) has high entropy \( \Rightarrow A^f(U_n') \) has small support.
- \( f(U_n) \) has low entropy \( \Rightarrow A^f(U_n') \) is close to uniform

(They ruled out efficient black-box reduction between SZK and NISZK)

\textbf{Lemma (Lemma in [LZ17])}

\[
\Pr_{f \sim \mathcal{D}_H} \left[ \exists B.C. \; w, A^f(w) = z \right] \geq \Pr_{f \sim \mathcal{D}_L} \left[ \exists B.C. \; w, A^f(w) = z \right] + \text{negl}
\]

\textbf{Lemma (This work)}

\[
\Pr_{f \sim \mathcal{D}_H} \left[ \exists B.C. \; w, A^f(w) = z \right] \leq 2^{O\left(\frac{q}{n^2}\right)} \cdot \Pr_{f \sim \mathcal{D}_L} \left[ \exists B.C. \; w, A^f(w) = z \right] + \text{negl}
\]
Technical Sketch

Let \( \{w_1, \ldots, w_{2n'}\} = \{0, 1\}^{n'} \). \( W_\ell = \{w_1, \ldots, w_\ell\} \).

\[
\Pr \left[ \exists w, A^f(w) = z \right] = \sum_\ell \Pr \left[ w_\ell \text{ is the “first” } w \text{ s.t. } A^f(w) = z \right]
\]
Let \( \{w_1, \ldots, w_{2n'}\} = \{0, 1\}^{n'} \). \( W_\ell = \{w_1, \ldots, w_\ell\} \).

\[
\Pr \left[ \exists w, A^f(w) = z \right] = \sum_\ell \Pr \left[ w_\ell \text{ is the "first" w s.t. } A^f(w) = z \right]
\]

\[
= \sum_\ell \Pr \left[ \nexists w \in W_{\ell-1} \text{ s.t. } A^f(w) \neq z \mid A^f(w_\ell) = z \right] \times \Pr \left[ A^f(w_\ell) = z \right]
\]

\[
= \sum_\ell \left( 1 - \Pr \left[ \exists w, \tilde{A}^f(w) = z \mid A^f(w_\ell) = z \right] \right) \times \Pr \left[ A^f(w_\ell) = z \right]
\]

where \( \tilde{A}^f(w) = \begin{cases} A^f(w) & \text{if } w \in W_\ell \\ \bot & \text{Otherwise} \end{cases} \)
We proved the $\Omega(n^2)$ lower bound for flattening entropy.
Conclusion

- We proved the $\Omega(n^2)$ lower bound for flattening entropy.
- Flattening entropy is an important step in constructing PRG, UOWHF and bit commitment from OWF.

Is the step necessary?
- If Yes $\Rightarrow \tilde{\Omega}(n^2)$ query lower bound for OWF $\rightarrow$ PRG
- Can the lower bound combined with the $\Omega(n^3)$ one in [HS12]?
  - If Yes $\Rightarrow \tilde{\Omega}(n^3)$ query lower bound for OWF $\rightarrow$ PRG (tight!)

Thanks!
Conclusion

- We proved the $\Omega(n^2)$ lower bound for flattening entropy.
- Flattening entropy is an important step in constructing PRG, UOWHF and bit commitment from OWF.
- Is the step necessary?
  - If Yes $\Rightarrow \tilde{\Omega}(n^2)$ query lower bound for OWF $\rightarrow$ PRG
We proved the $\Omega(n^2)$ lower bound for flattening entropy.

Flattening entropy is an important step in constructing PRG, UOWHF and bit commitment from OWF.

Is the step necessary?
If Yes $\Rightarrow \tilde{\Omega}(n^2)$ query lower bound for OWF $\rightarrow$ PRG

Can the lower bound combined with the $\Omega(n)$ one in [HS12]?
If Yes $\Rightarrow \tilde{\Omega}(n^3)$ query lower bound for OWF $\rightarrow$ PRG (tight!)
Conclusion

- We proved the $\Omega(n^2)$ lower bound for flattening entropy.
- Flattening entropy is an important step in constructing PRG, UOWHF and bit commitment from OWF.
- Is the step necessary?
  - If Yes $\Rightarrow \tilde{\Omega}(n^2)$ query lower bound for OWF $\rightarrow$ PRG
- Can the lower bound combined with the $\Omega(n)$ one in [HS12]? If Yes $\Rightarrow \tilde{\Omega}(n^3)$ query lower bound for OWF $\rightarrow$ PRG (tight!)

Thanks!