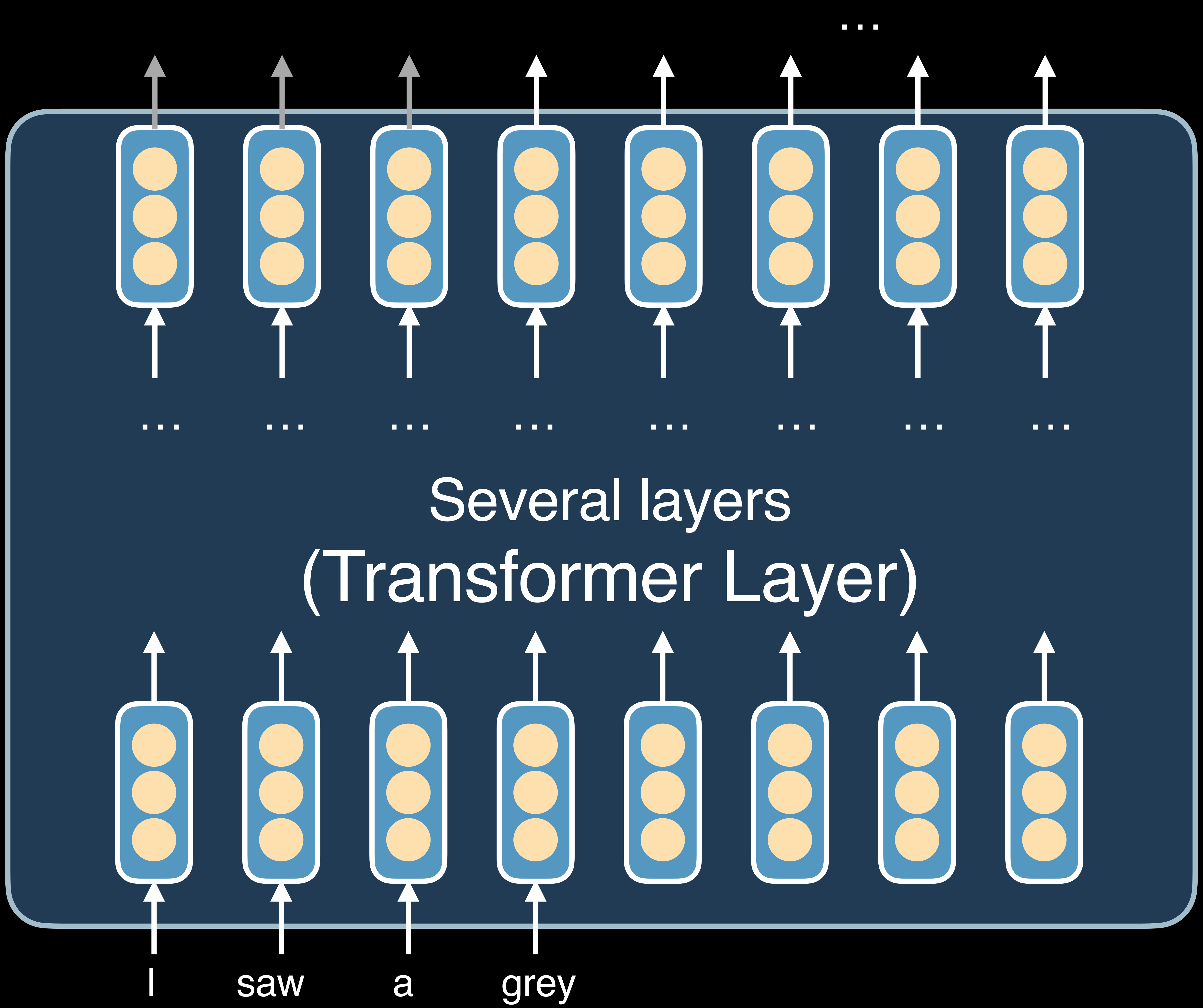
Analyzing and Mitigating Repetitions in Neural Text Generation

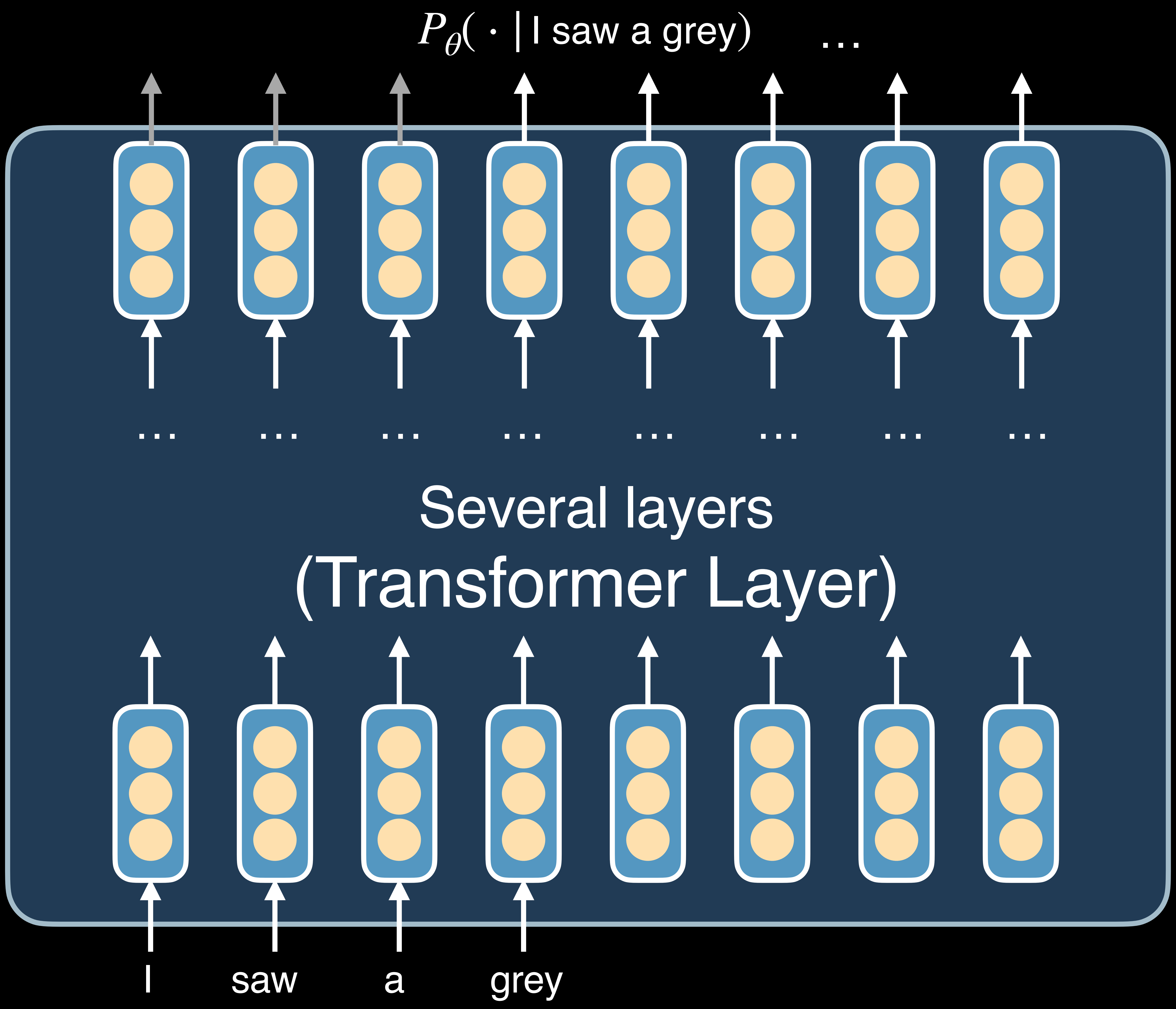
Jin Xu[1], Xiaojiang Liu[4], Jianhao Yan[2], Deng Cai[3], Huayang Li[4], Jian Li[1]
[1]Tsinghua University,
[2]Westlake University, [3]The Chinese University of Hong Kong,
[4]Apple Inc

- Introduction
- Related Work
- Analyzing Repetition Problems
- DITTO - a Method to Mitigate Repetitions
- Experiments
- Future Work

Pre-trained Language Models (PLMs) based on Transformer



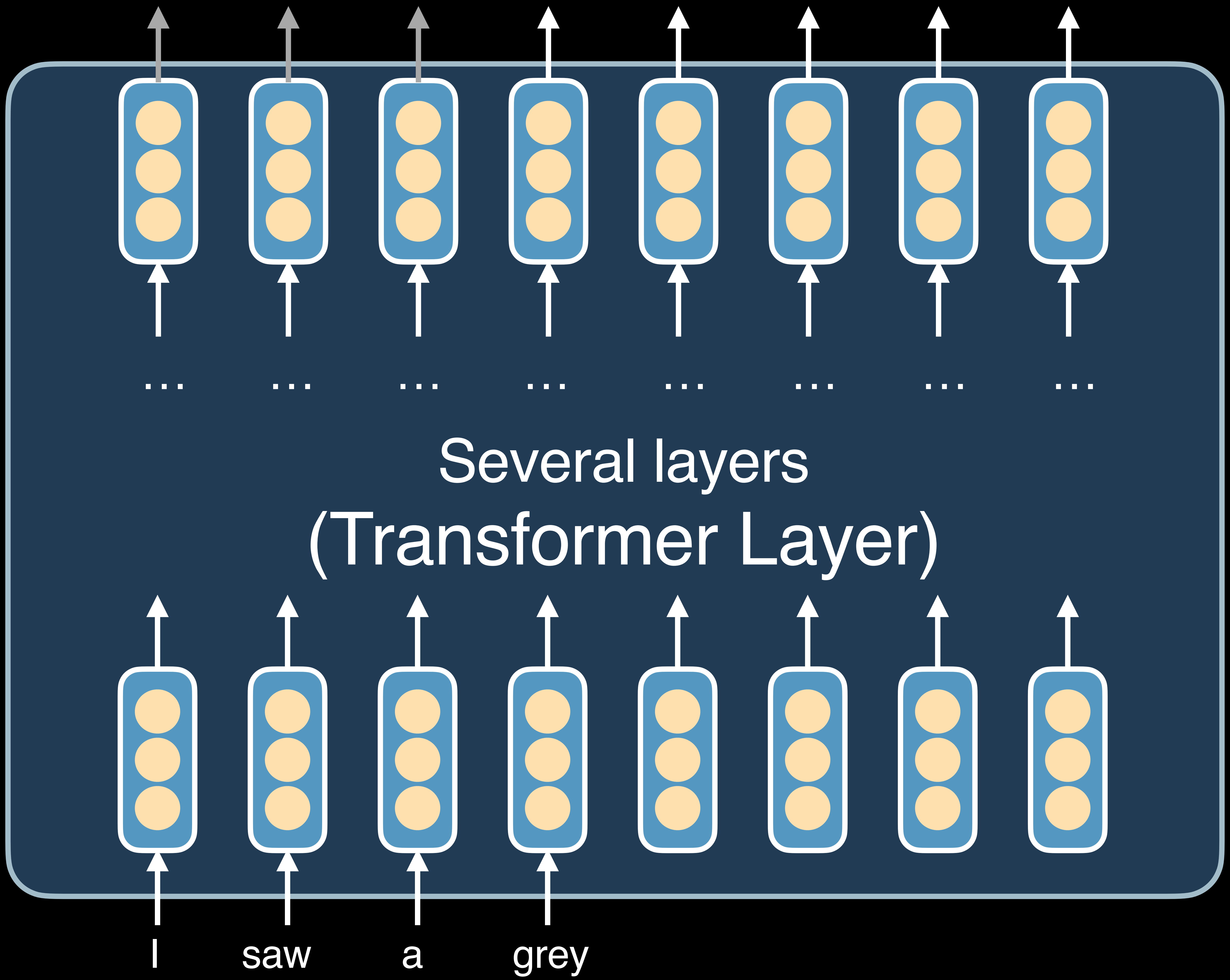
Pre-trained Language Models (PLMs) based on Transformer



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

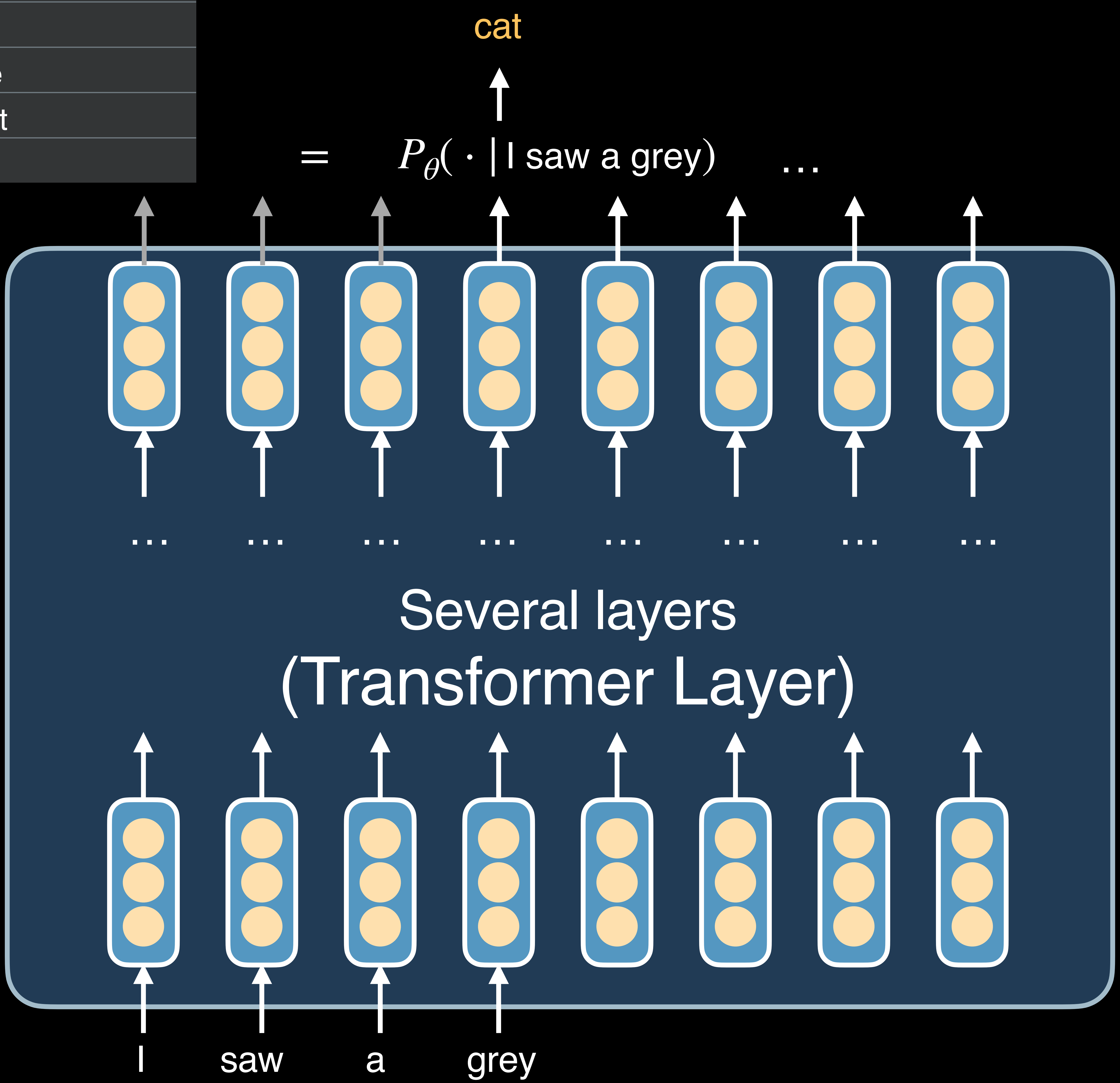
$= \quad P_\theta(\cdot \mid \text{I saw a grey}) \quad \ldots$



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

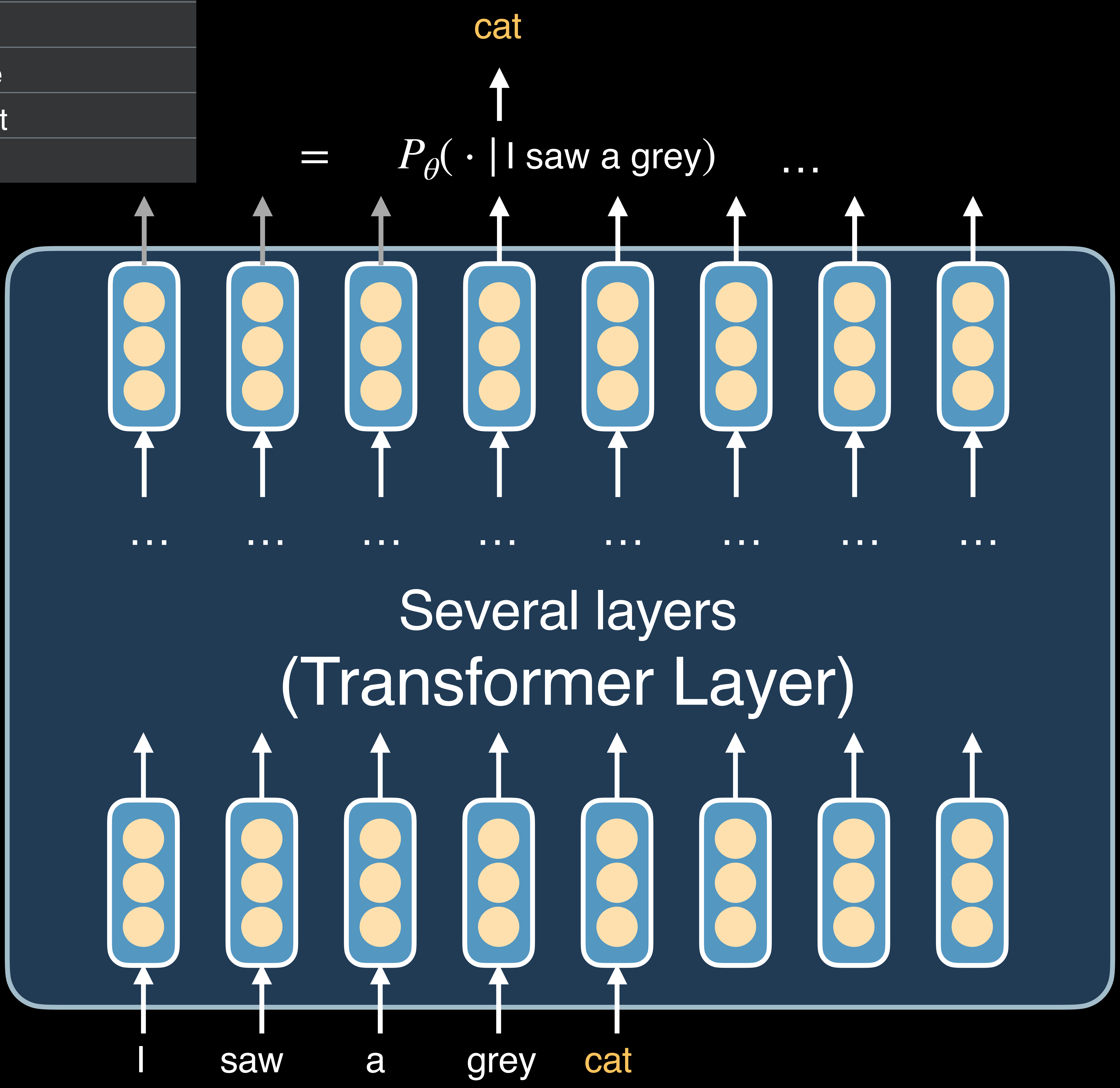
$= \quad P_\theta(\cdot \mid \text{I saw a grey})$



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

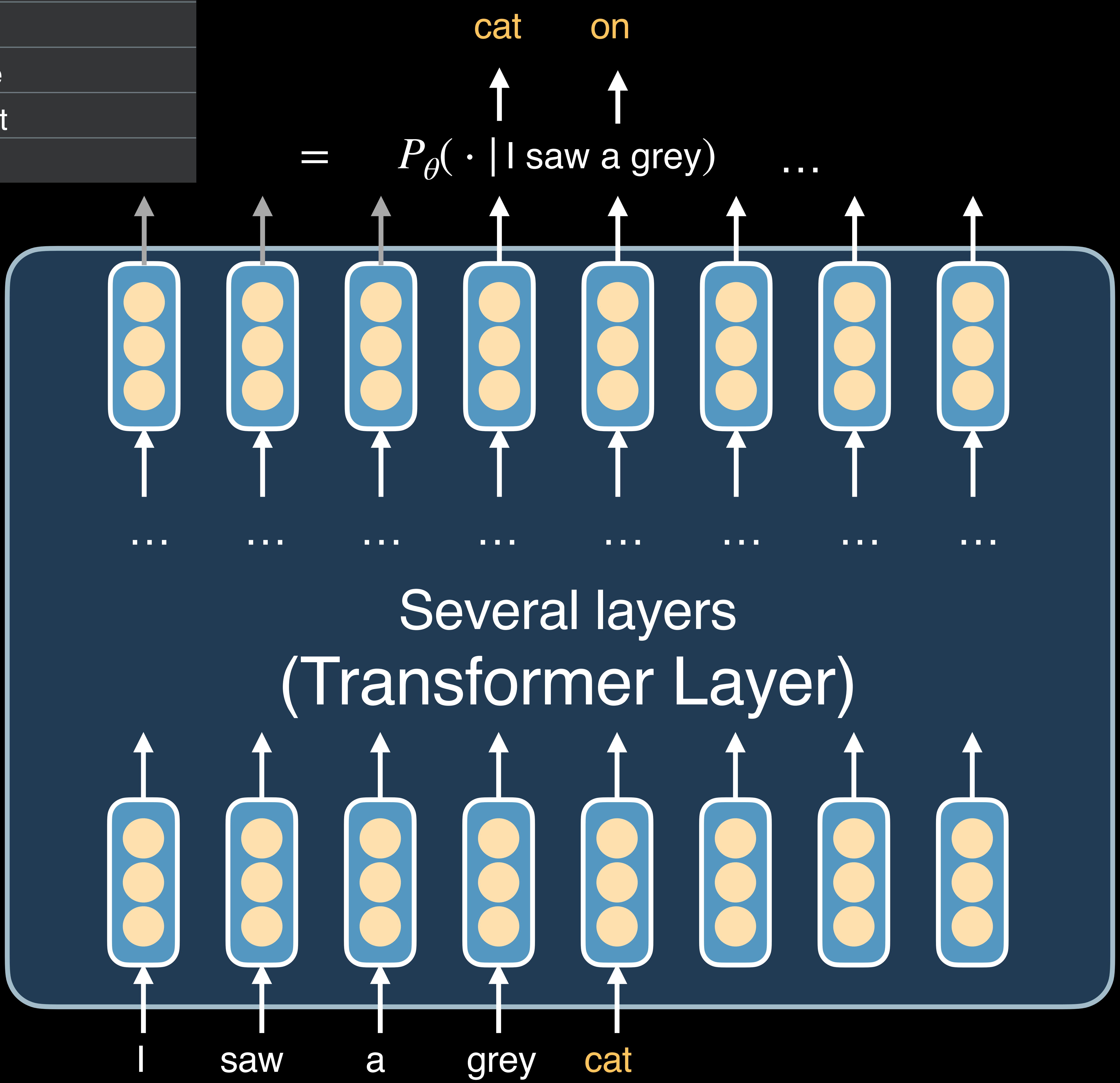
$= P_\theta(\cdot \mid \text{I saw a grey}) \quad \ldots$



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

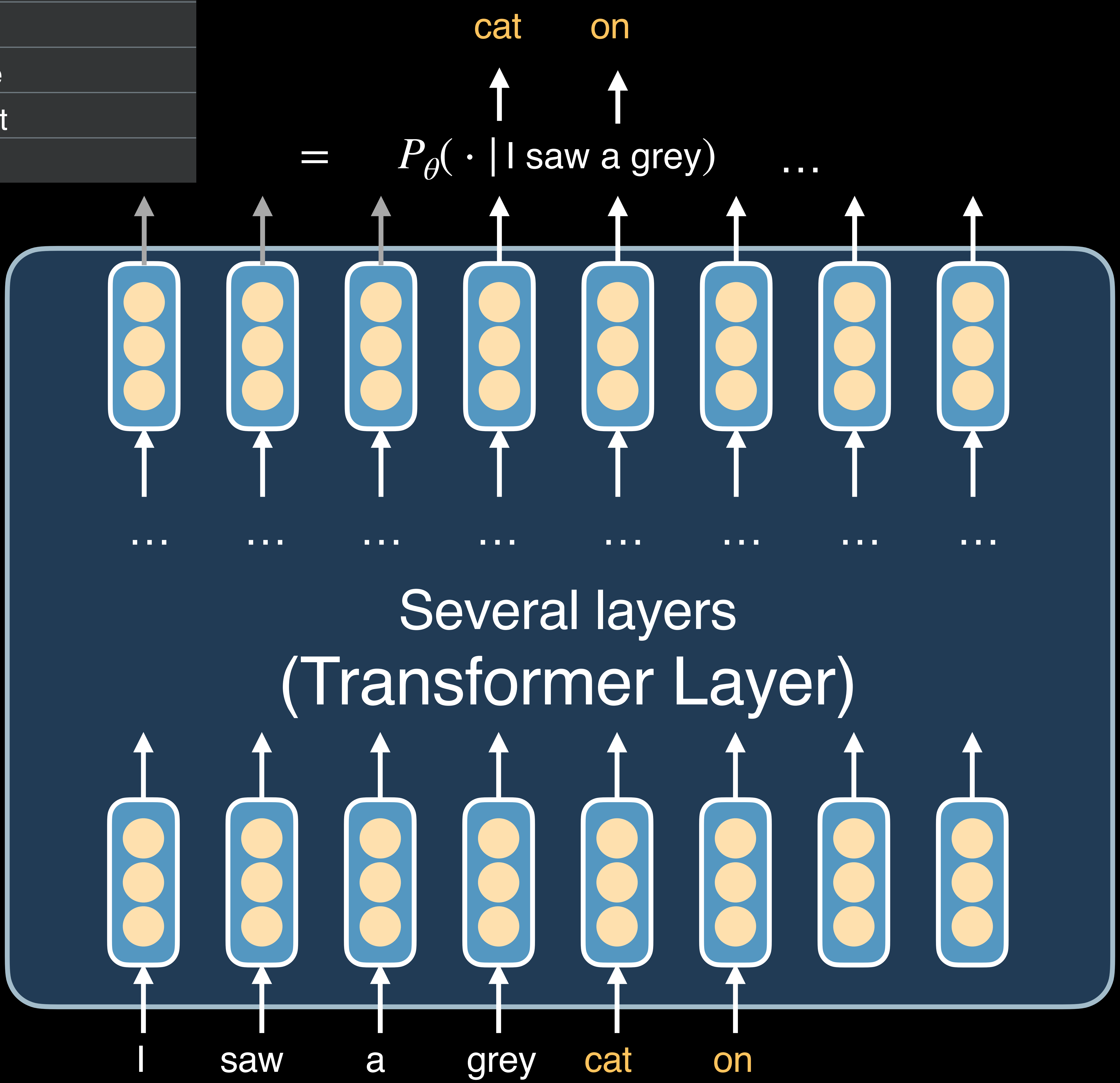
$= P_\theta(\,\cdot \mid \text{I saw a grey})$ …



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

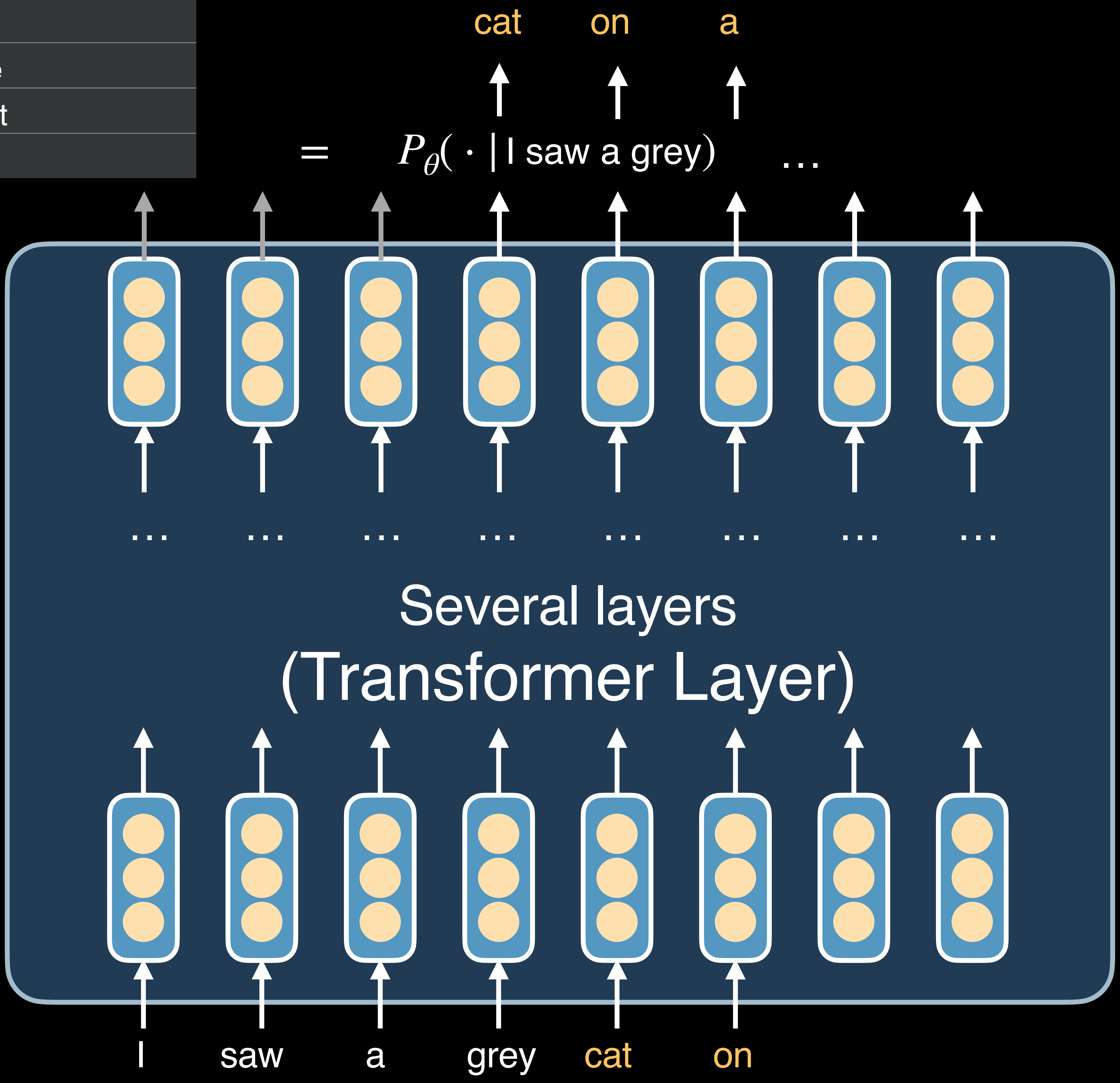
$= P_\theta(\cdot \mid \text{I saw a grey})$ …



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

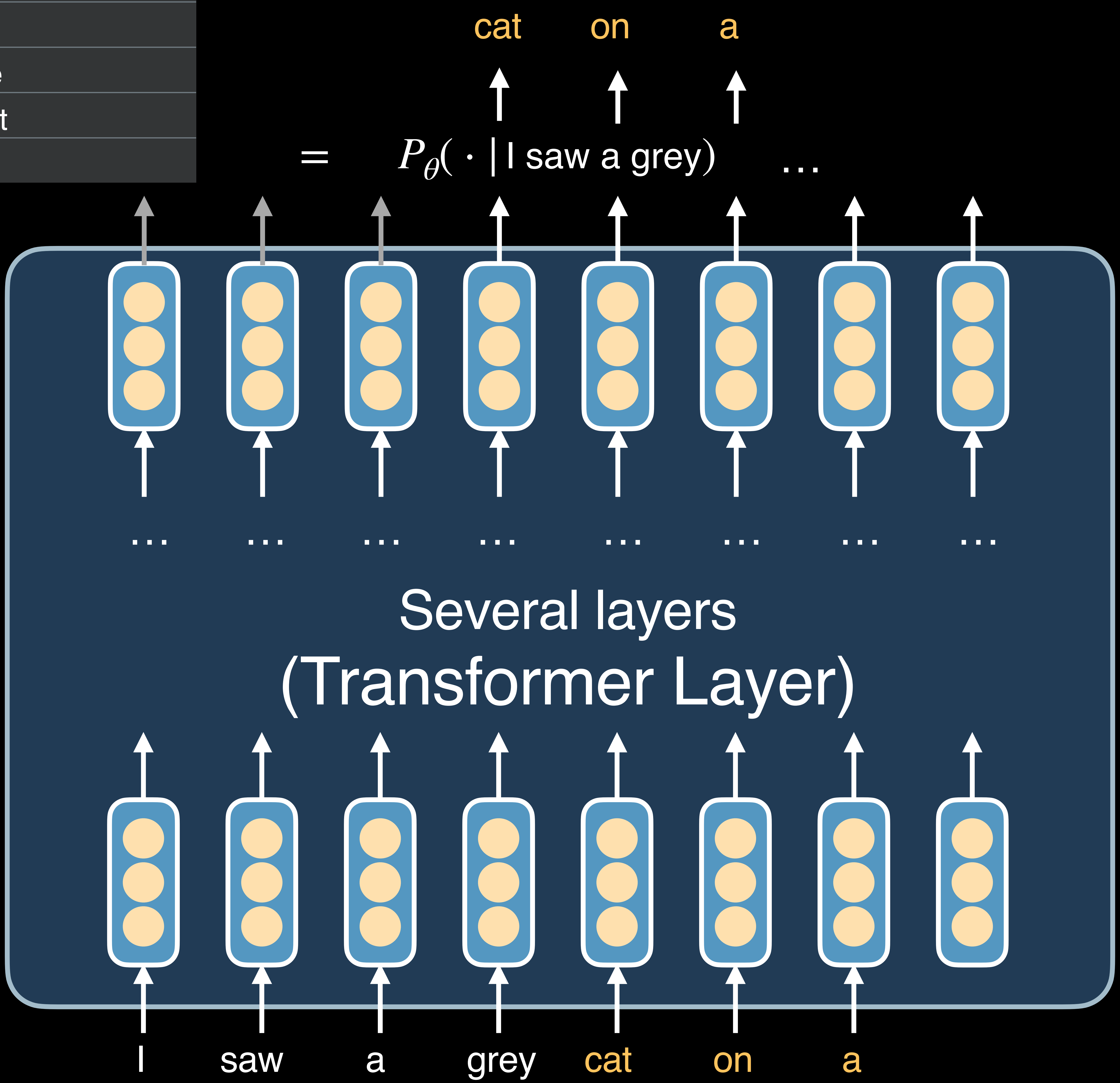
$= P_\theta(\cdot \mid \text{I saw a grey})$ …



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

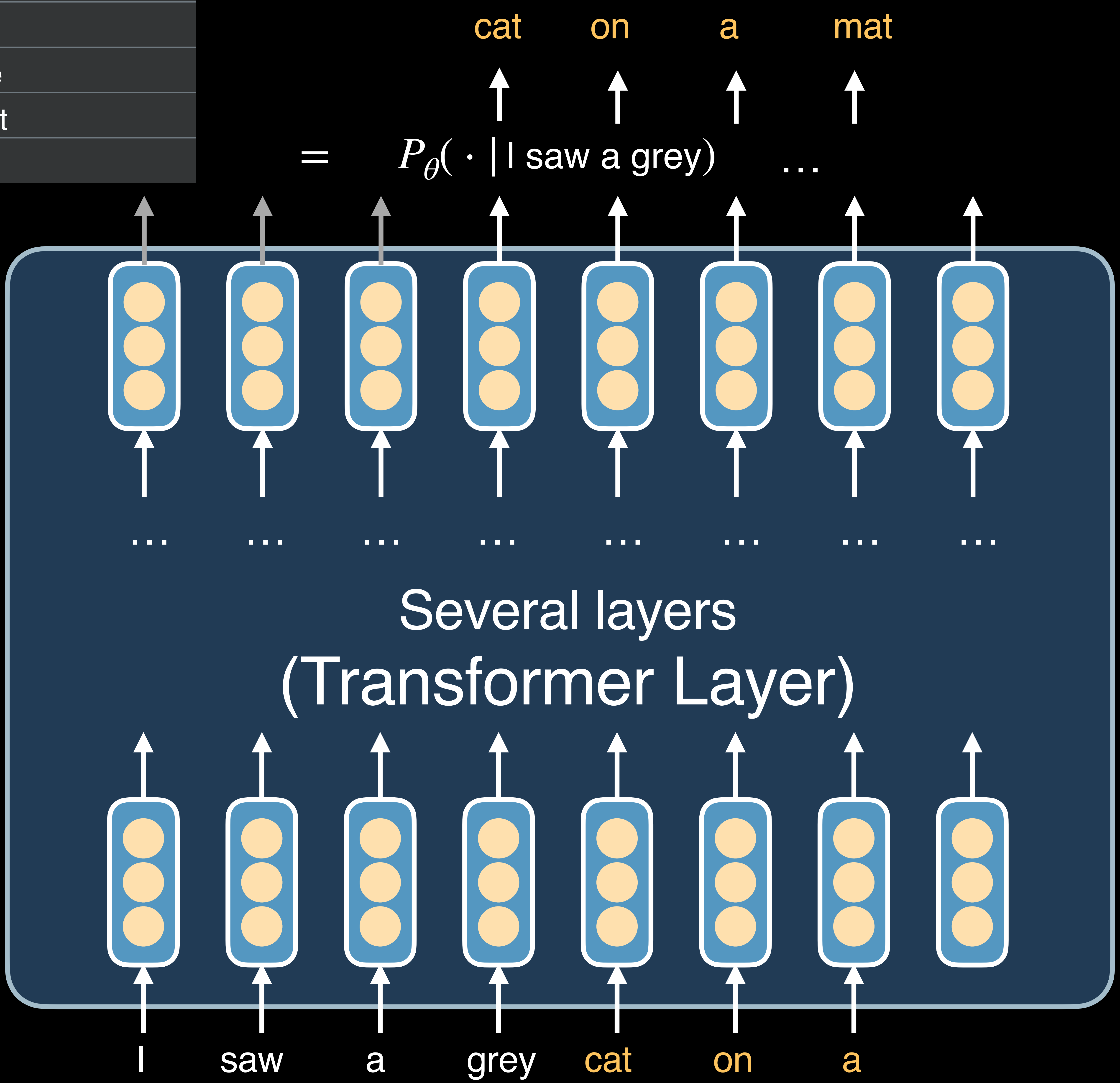
$= \quad P_\theta(\cdot \mid \text{I saw a grey}) \quad \ldots$



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

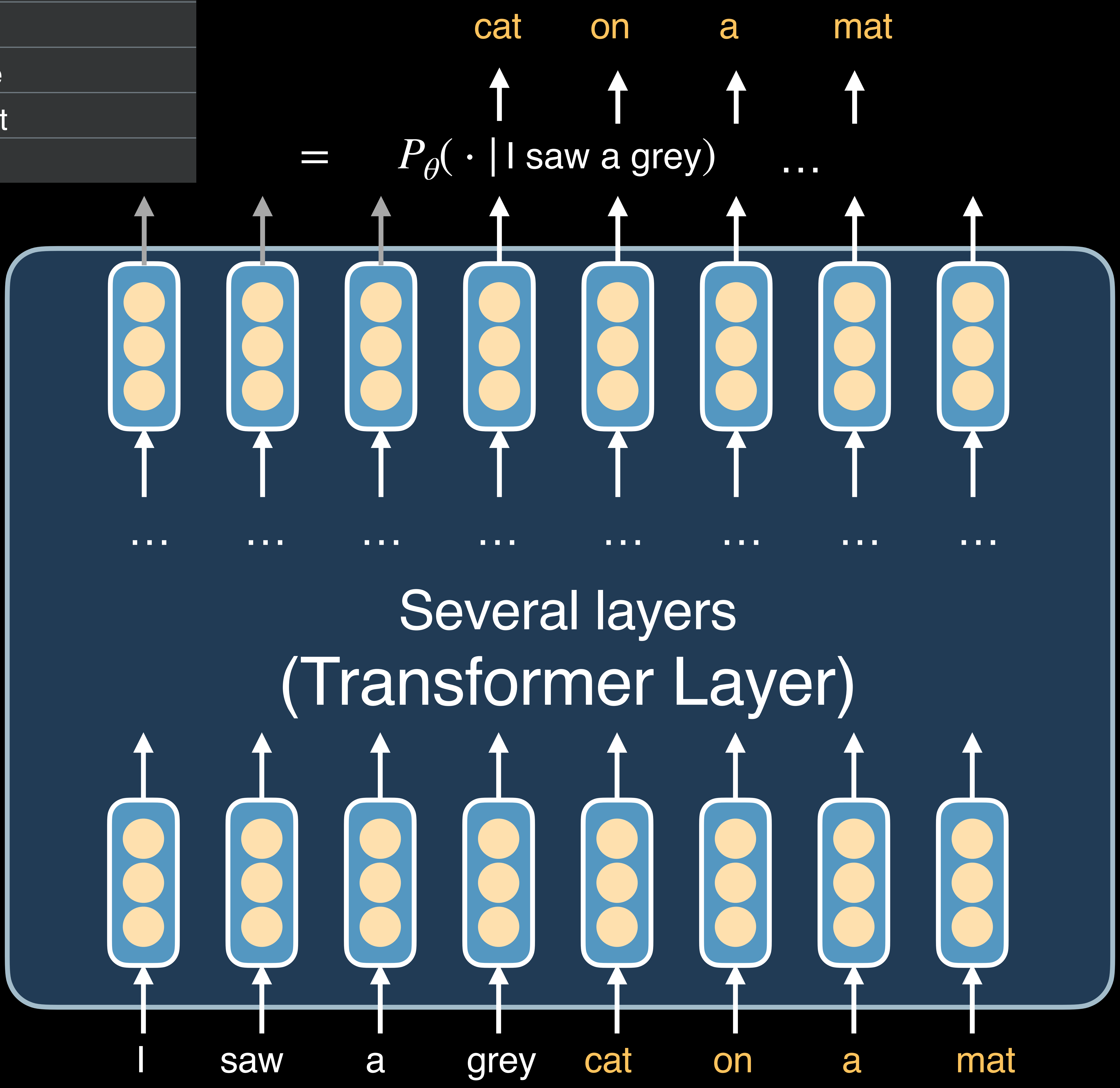
$= P_\theta(\cdot \mid \text{I saw a grey})$ …



Pre-trained Language Models (PLMs) based on Transformer

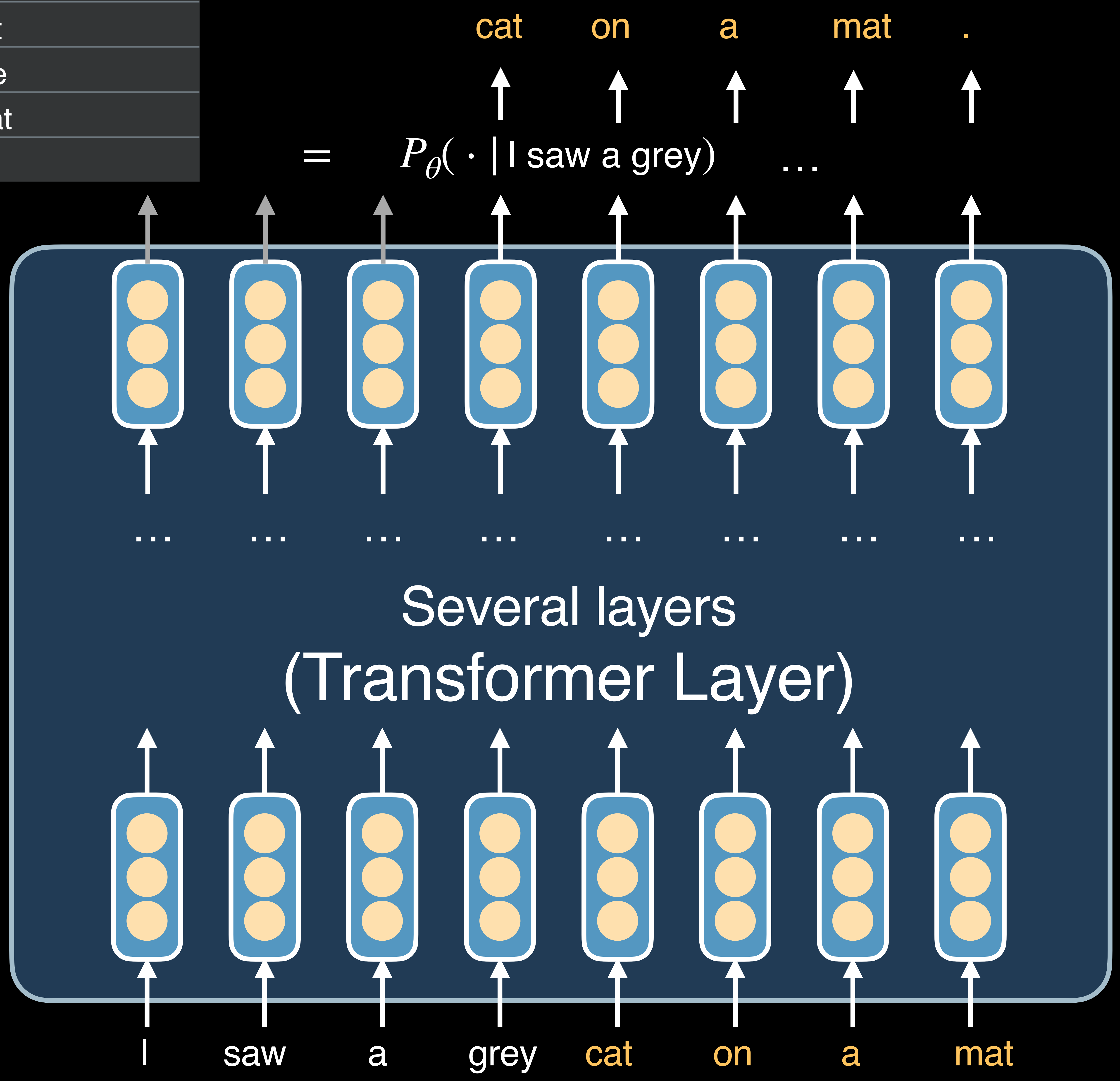
Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…

$= P_\theta(\cdot \mid \text{I saw a grey})$ …



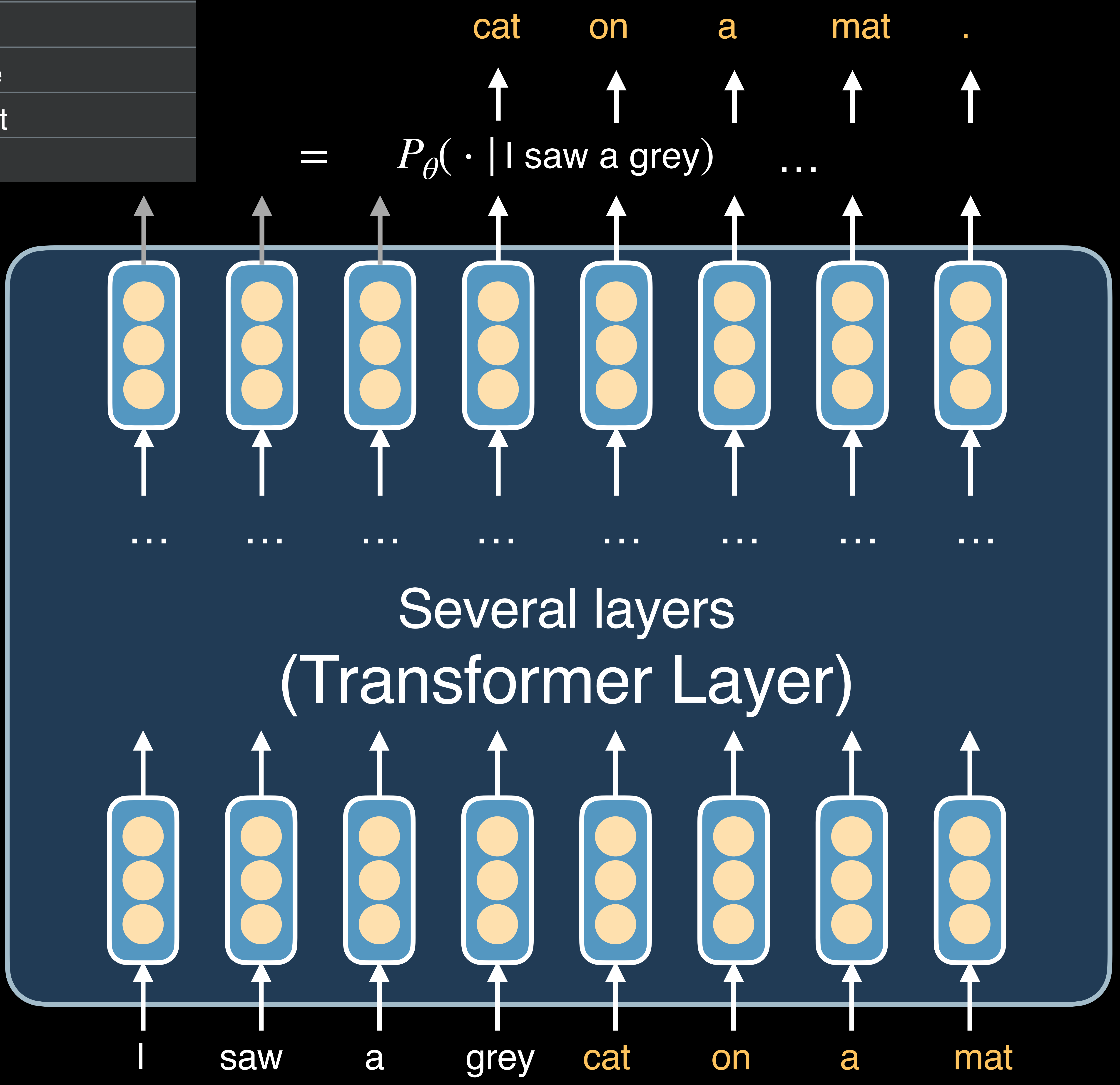
Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…



Pre-trained Language Models (PLMs) based on Transformer

Prob	Vocabulary
0.01	the
0.09	cat
0.03	like
0.08	mat
…	…



- Learn next token distribution

$$P_\theta(\cdot \mid x_1, \cdots, x_{t-1})$$

- Decode auto-regressively [*Greedy Decoding*]

$$x_t = \arg\max P_\theta(\cdot \mid x_1, \cdots, x_{t-1})$$

Language Models Prefer Repetitions

Language Models Prefer Repetitions

Context:

No significant craters intersect the rim , and it is sloped about 1 @.@ 5 °toward the direction 50 90 °from the Earth .

Language Models Prefer Repetitions

Context:

No significant craters intersect the rim , and it is sloped about 1 @.@ 5 °toward the direction 50 90 °from the Earth .

Greedy Decoding [select $x_t = \arg\max P_\theta(\cdot \mid x_1, \cdots, x_{t-1})$] :

Language Models Prefer Repetitions

Context:

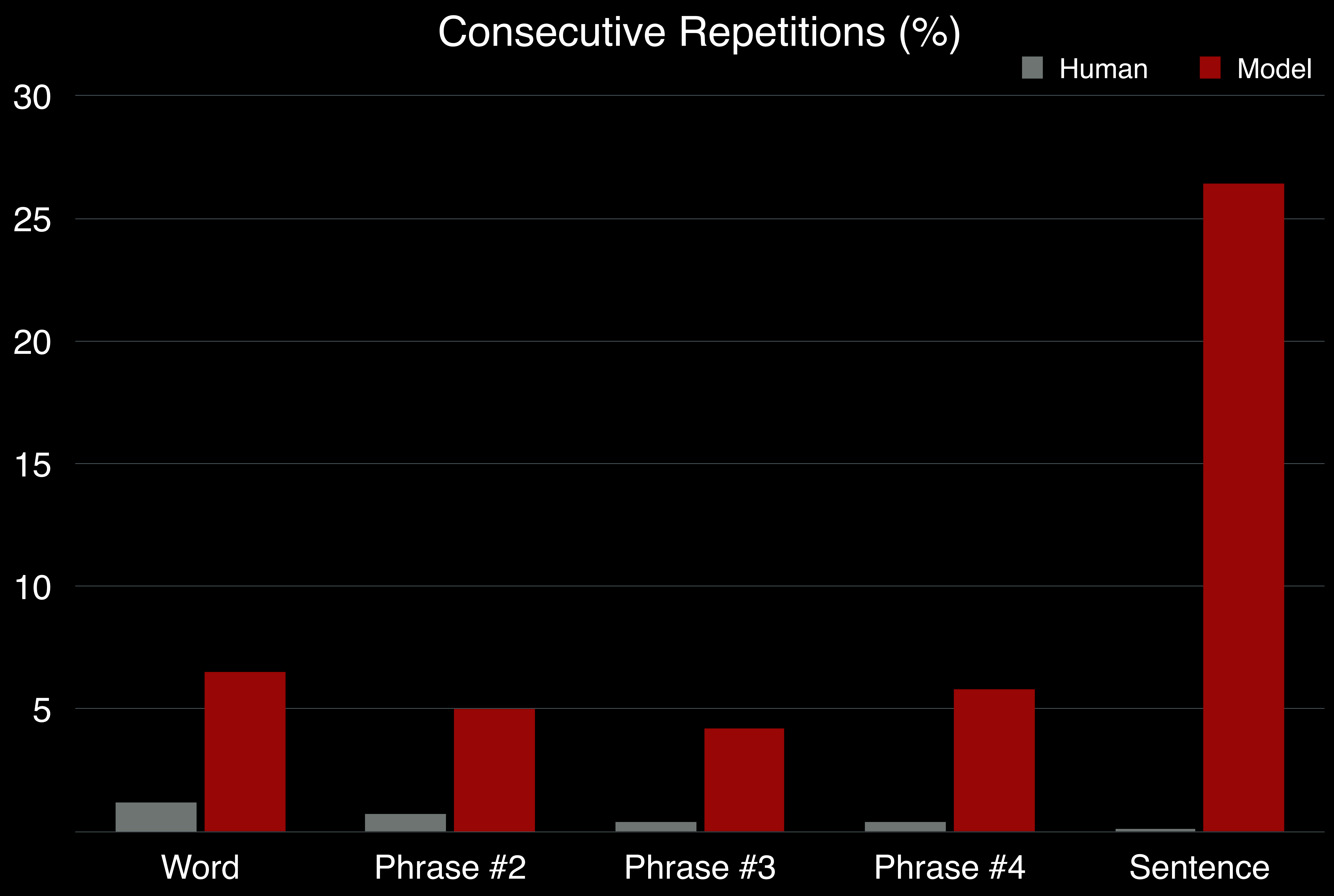
No significant craters intersect the rim , and it is sloped about 1 @.@ 5 °toward the direction 50 90 °from the Earth .

Greedy Decoding [select $x_t = \arg\max P_\theta(\cdot \mid x_1, \cdots, x_{t-1})$] :

The age of the crater is about 3 @.@ 6 billion years and it has been in the proximity of the south lunar pole for at least 10 @,@ 000 years . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . …

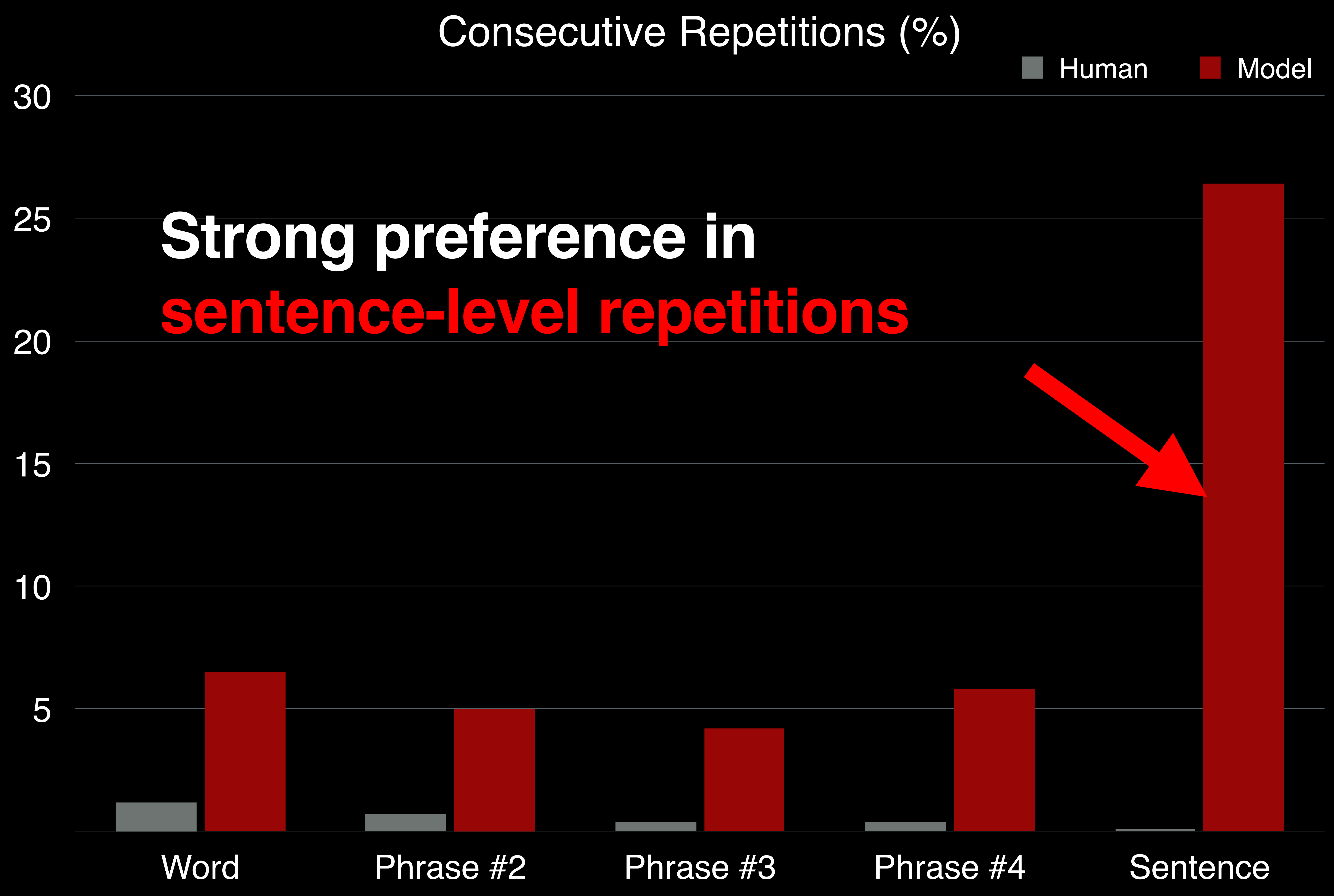
Discrepancy Between Model Generations and Human Languages

- Consecutive Repetitions
  - **Word-level**: hello hello hello ….
  - **Phrase-level**: hello world hello world …
  - **Sentence-level**: I love oranges . I love oranges . …



Discrepancy Between Model Generations and Human Languages

- Consecutive Repetitions
  - **Word-level**: hello hello hello ….
  - **Phrase-level**: hello world hello world …
  - **Sentence-level**: I love oranges . I love oranges . …



- Introduction
- **Related Work**
- Analyzing Repetition Problems
- DITTO - a Method to Mitigate Repetitions
- Experiments
- Future Work

Existing Methods in Mitigating Repetitions

- Rectify model distribution error by forbidding repetition when **decoding**
  - N-gram Blocking.
    - E.g., $P_\theta(\cdot \mid \text{A grey cat on the table. I have a grey})$ =

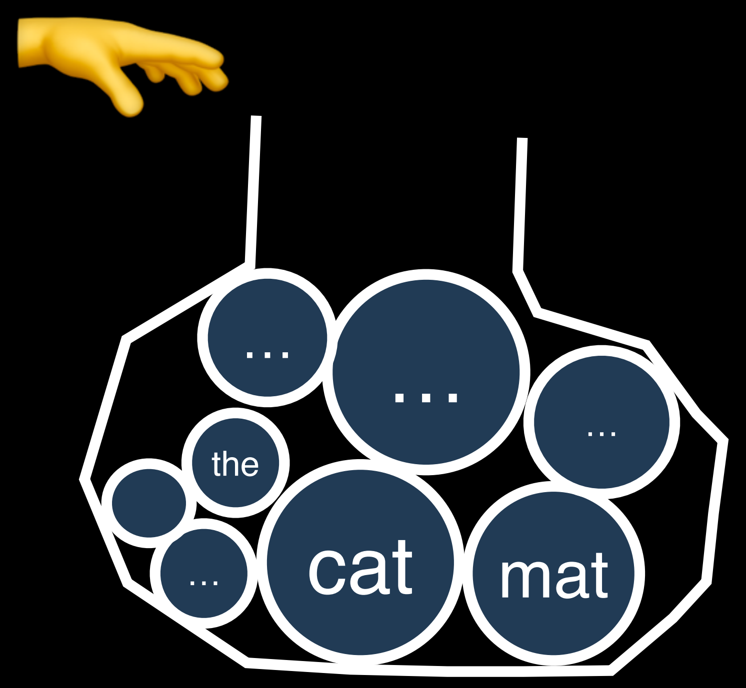
Vocabulary	the	~~cat~~	mat	…
Prob	0.01	~~0.09~~	0.07	

Existing Methods in Mitigating Repetitions

- Rectify model distribution error by forbidding repetition when **decoding**
  - N-gram Blocking.
    - E.g., $P_\theta(\,\cdot \mid \text{A grey cat on the table. I have a grey})$ =

Vocabulary	the	~~cat~~	mat	…
Prob	0.01	~~0.09~~	0.07	

  - Stochastic Sampling such as Top-p and Top-k.
    - E.g., randomly sample token according to token probs

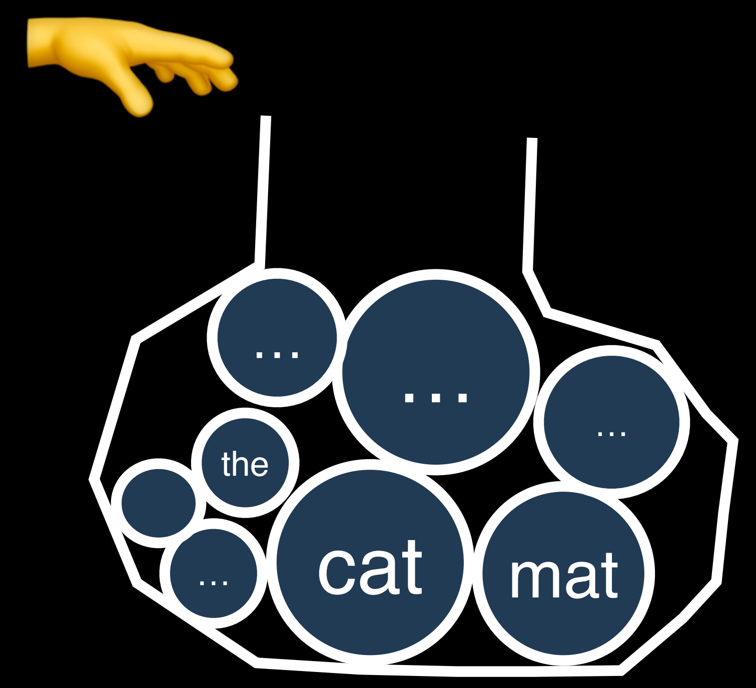


Existing Methods in Mitigating Repetitions

- Rectify model distribution error by forbidding repetition when **decoding**
  - N-gram Blocking.
    - E.g., $P_\theta(\cdot \mid \text{A grey cat on the table. I have a grey})$ =

Vocabulary	the	~~cat~~	mat	…
Prob	0.01	~~0.09~~	0.07	

  - Stochastic Sampling such as Top-p and Top-k.
    - E.g., randomly sample token according to token probs



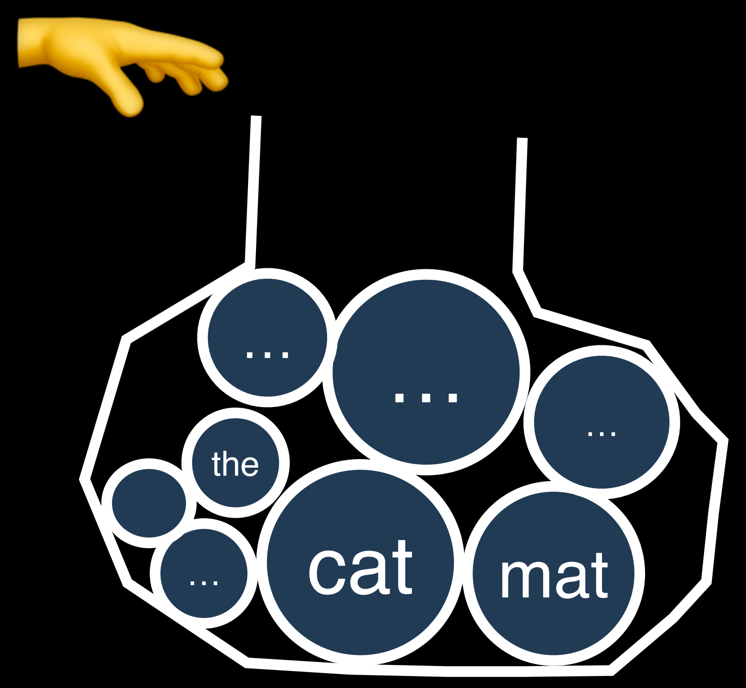
- However,
  - **The underlying model distribution is clearly problematic.**

Existing Methods in Mitigating Repetitions

- Rectify model distribution error by forbidding repetition when **decoding**
  - N-gram Blocking.
    - E.g., $P_\theta(\cdot \mid \text{A grey cat on the table. I have a grey}) =$

Vocabulary	the	~~cat~~	mat	…
Prob	0.01	~~0.09~~	0.07	

  - Stochastic Sampling such as Top-p and Top-k.
    - E.g., randomly sample token according to token probs



- However,
  - **The underlying model distribution is clearly problematic.**
- Our work
  - Analyze how sentence repetition occurs
  - Propose a novel **training-based** model to improve model distribution
  - Compatible with various decoding algorithms

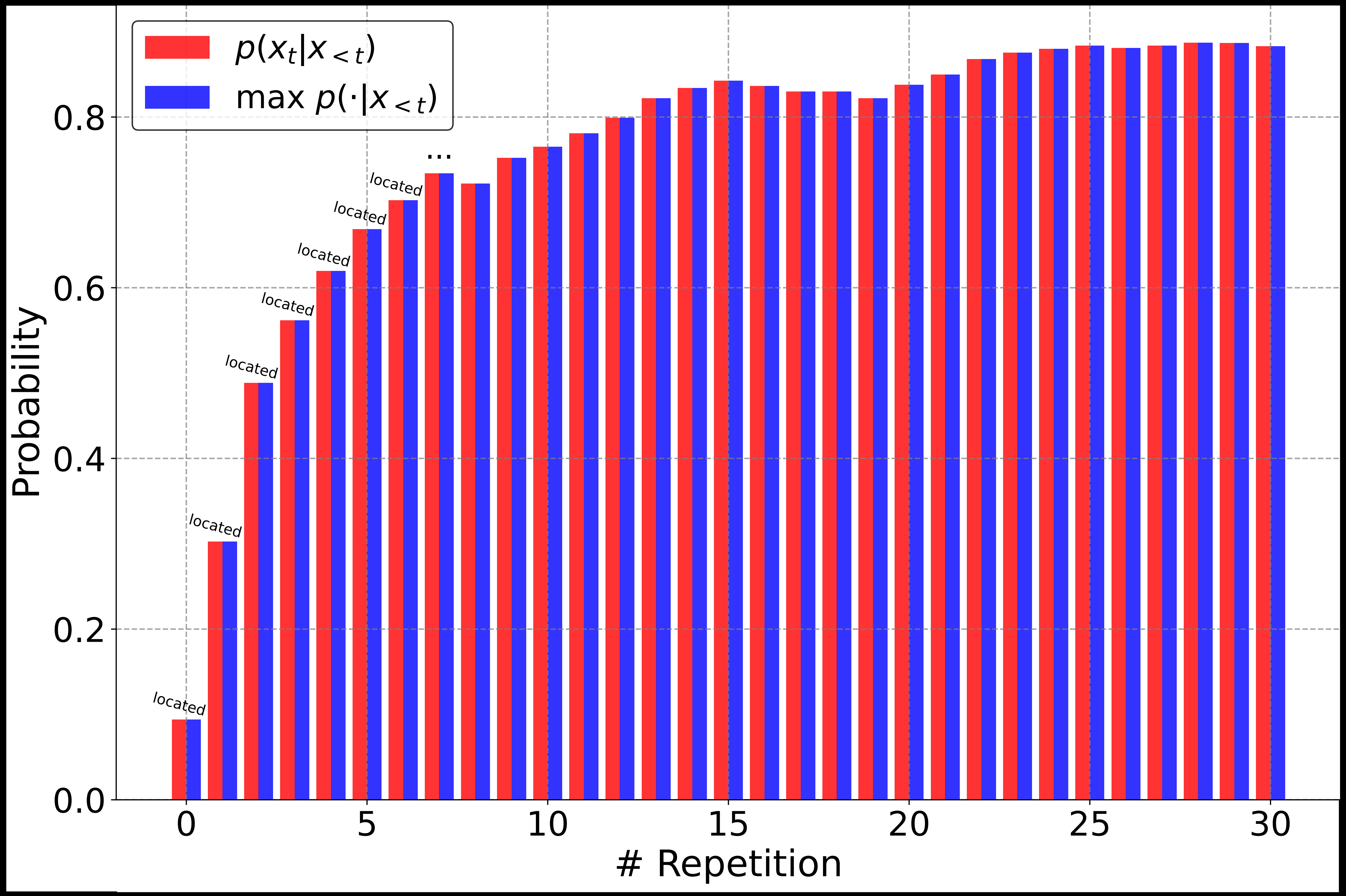
- Introduction
- Related Work
- **Analyzing Repetition Problems**
- DITTO - a Method to Mitigate Repetitions
- Experiments
- Future Work

Case Study

Context: No significant craters intersect the rim , and it is sloped about 1 @.@ 5 °toward the direction 50 90 °from the Earth .

Greedy Decoding:

The age of the crater is about 3 @.@ 6 billion years and it has been in the proximity of the south lunar pole for at least 10 @,@ 000 years . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . …



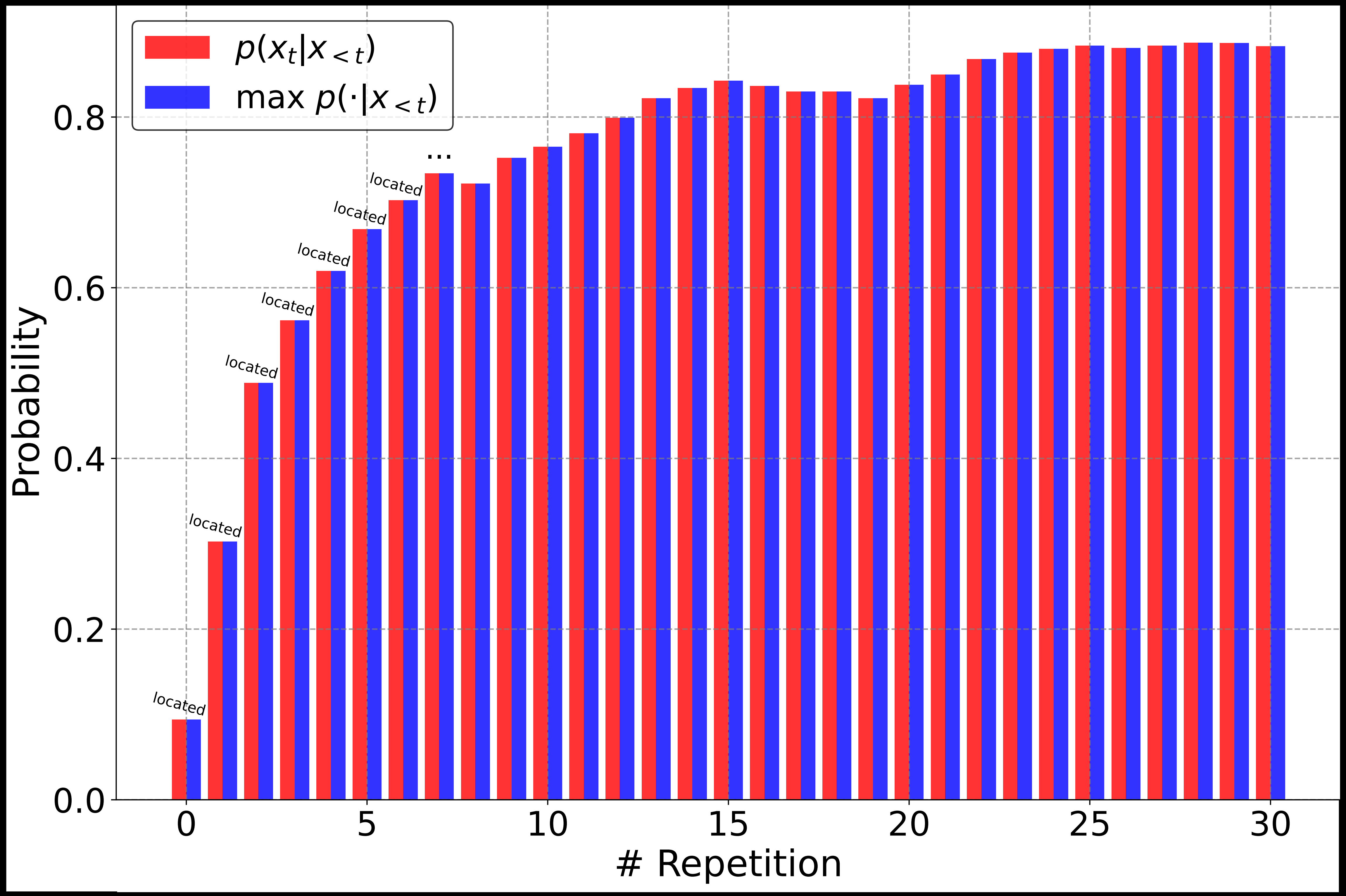
Case Study

The probability of repetition (in red) increases almost monotonically

Context: No significant craters intersect the rim , and it is sloped about 1 @.@ 5 °toward the direction 50 90 °from the Earth .

Greedy Decoding:

The age of the crater is about 3 @.@ 6 billion years and it has been in the proximity of the south lunar pole for at least 10 @,@ 000 years . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . The South Crater is located on the southern edge of the northern highlands . …



Statistically Investigate the Repetition Issue

Why the first sentence repetition occurs?

Why model gets stuck in sentence-level loop?

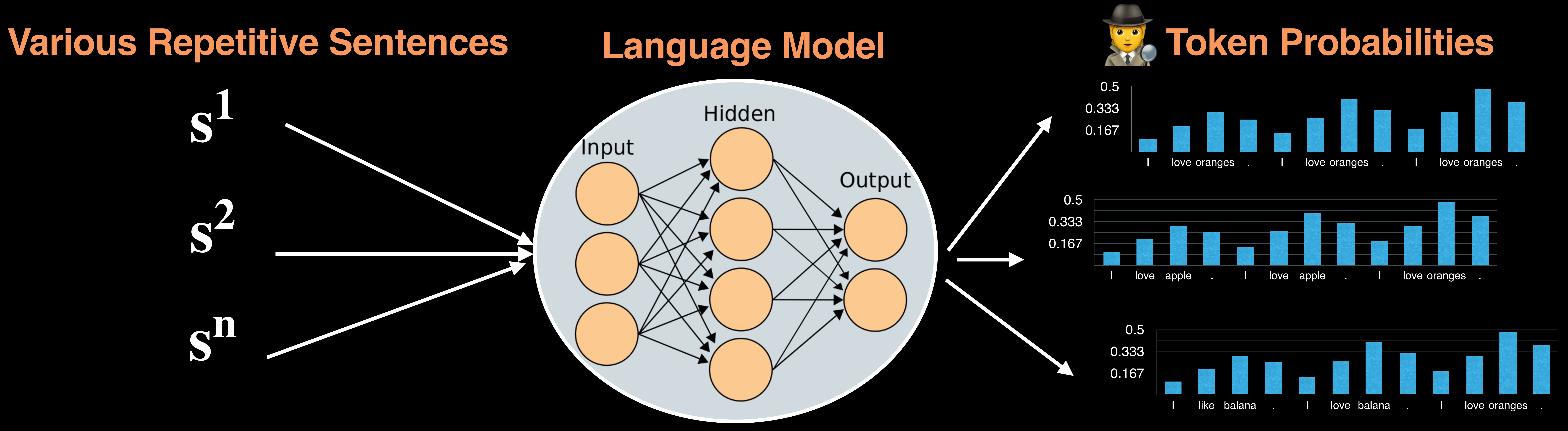
What kind of sentences are more likely to be repeated?

Statistically Investigate the Repetition Issue

Why the first sentence repetition occurs?

Why model gets stuck in sentence-level loop?

What kind of sentences are more likely to be repeated?

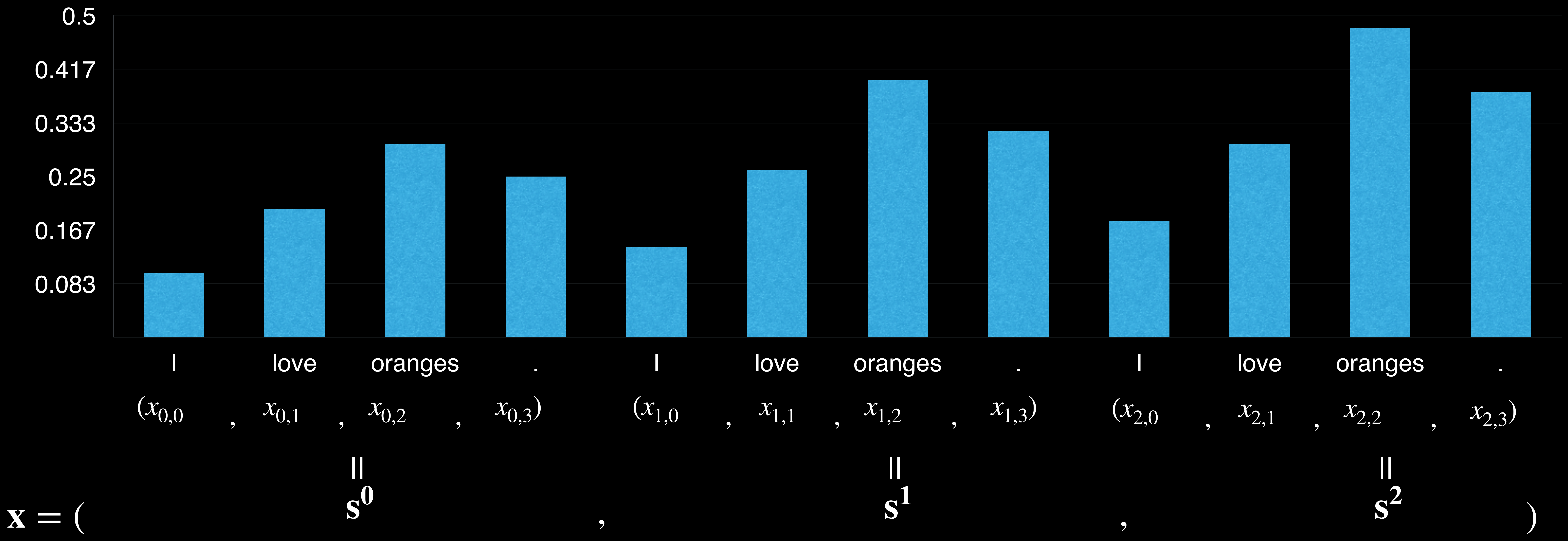


Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathrm{s}}} \sum_{l=1}^{L_{\mathrm{s}}} \mathbb{1}(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities

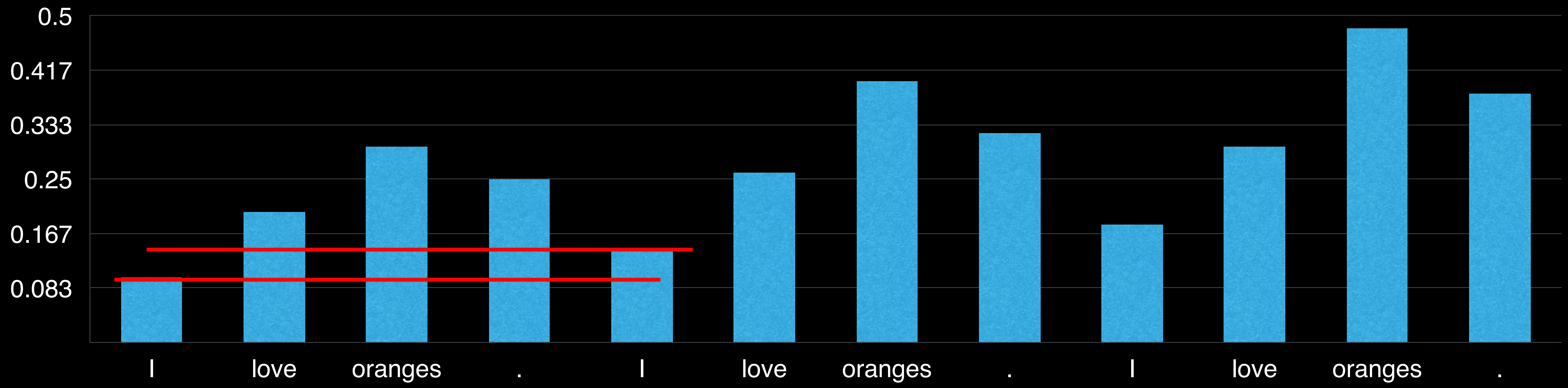


Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



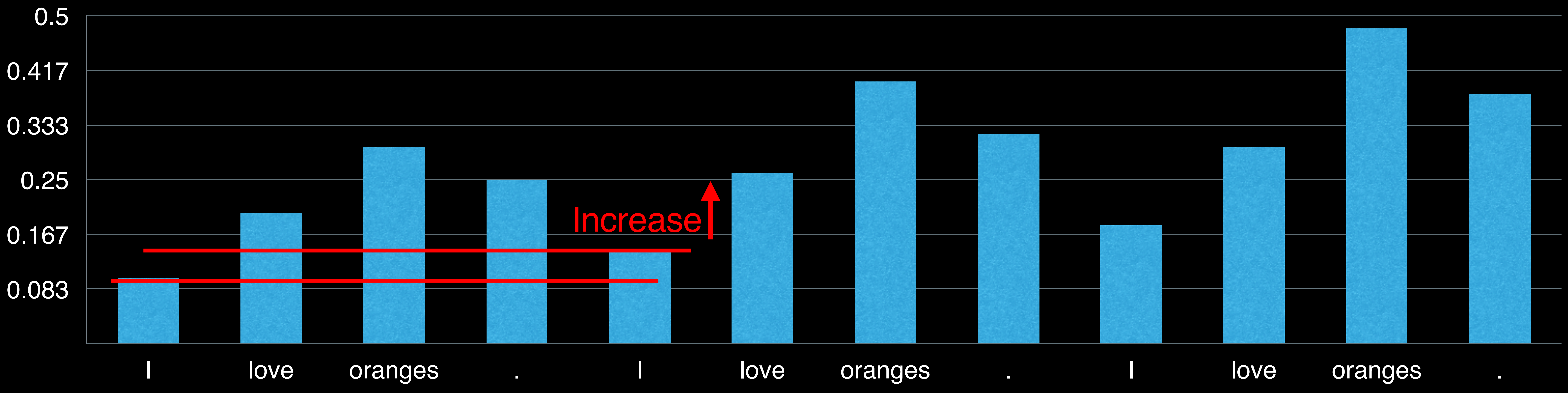
$$\mathbf{x} = (\underbrace{x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}}_{= \mathbf{s}^0}, \underbrace{x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}}_{= \mathbf{s}^1}, \underbrace{x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}}_{= \mathbf{s}^2})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s^n}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



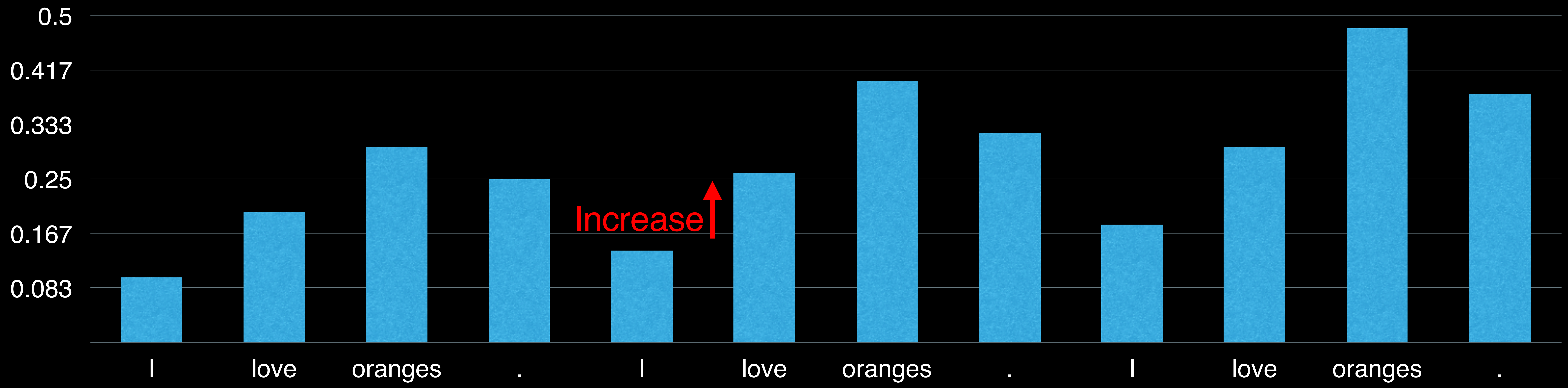
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s^2}})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\text{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{\mathbf{1}}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



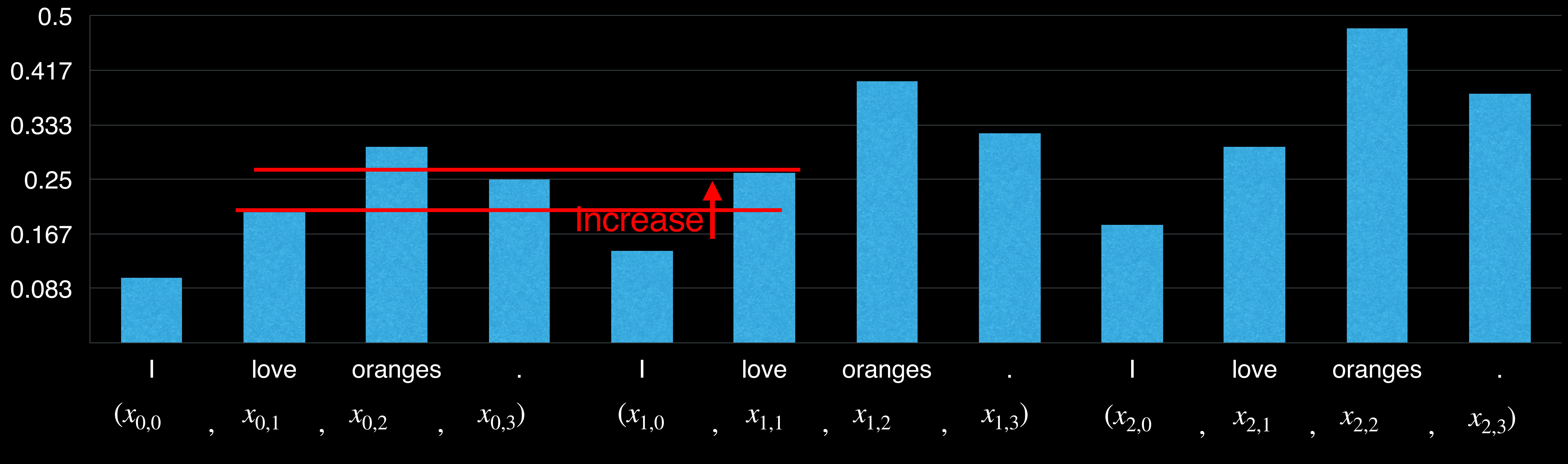
$$\mathbf{x} = (\underbrace{x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}}_{=\mathbf{s}^0}, \underbrace{x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}}_{=\mathbf{s}^1}, \underbrace{x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}}_{=\mathbf{s}^2})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\text{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{\mathbf{1}}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \,|\, \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \,|\, \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



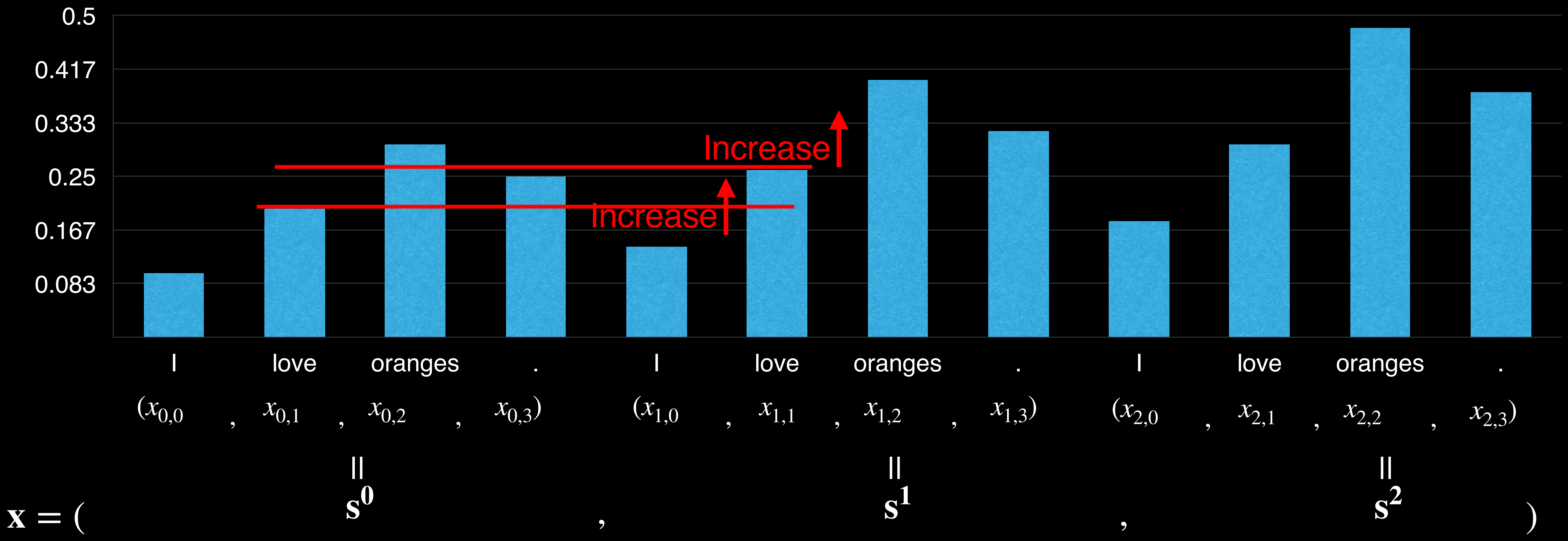
$$\mathbf{x} = (\mathbf{s}^0, \mathbf{s}^1, \mathbf{s}^2)$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} \mathbb{1}\left(P_{\theta}(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_{\theta}(x_{0,l} \mid \mathbf{x}_{<0,l})\right)$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities

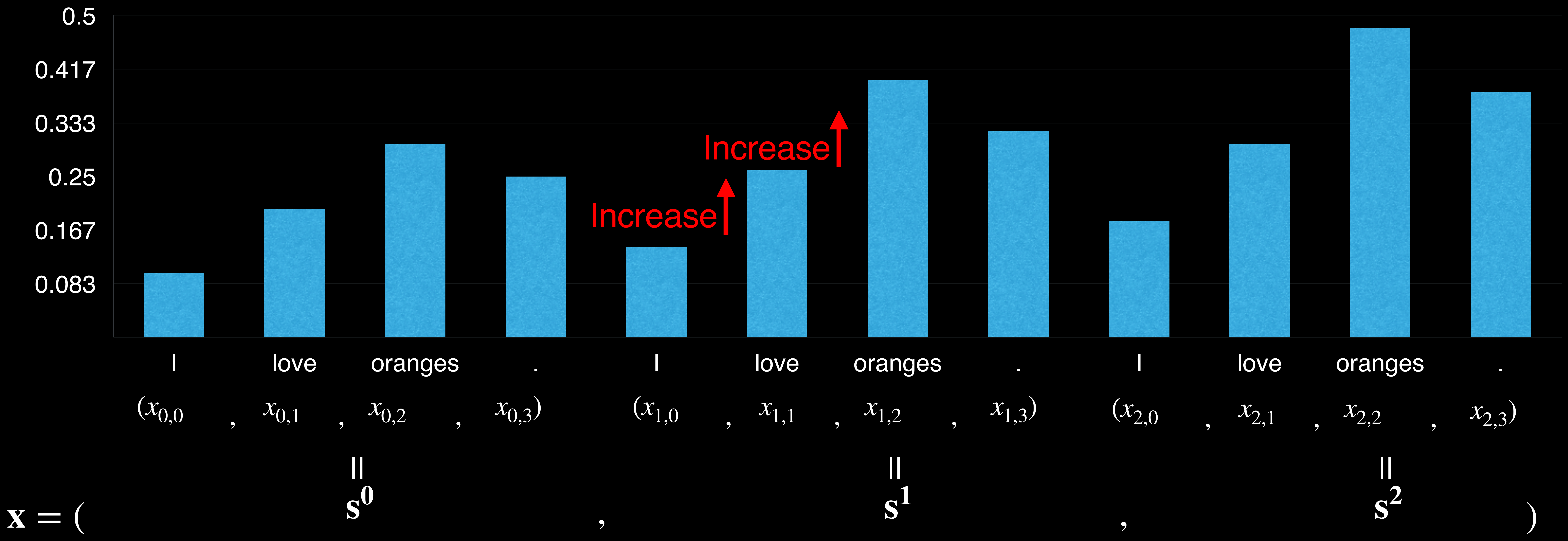


Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{\mathbf{1}}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities

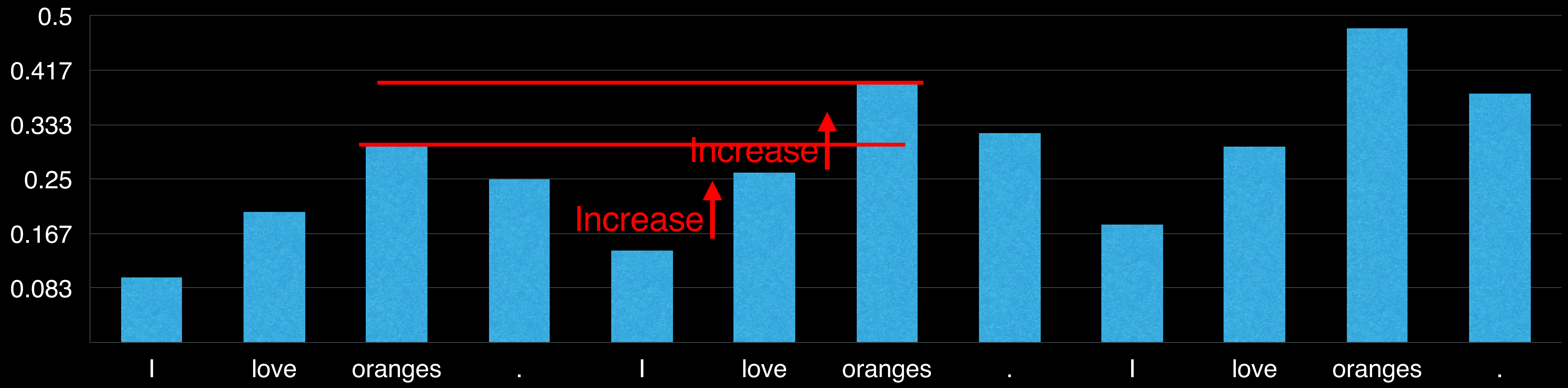


Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_s}\sum_{l=1}^{L_s} \mathbb{1}(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



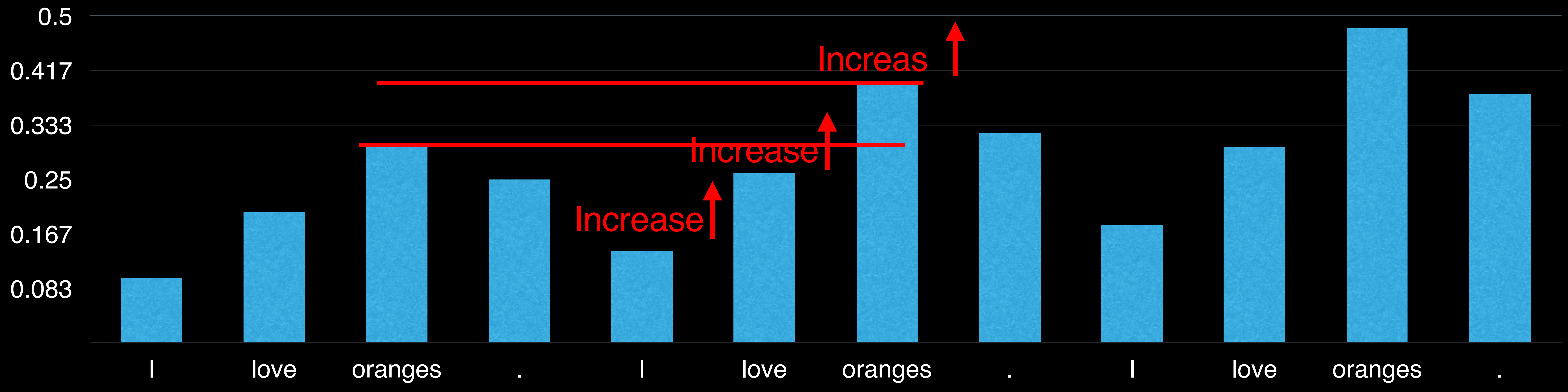
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{=\mathbf{s}^0}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{=\mathbf{s}^1}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{=\mathbf{s}^2})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} \mathbb{1}(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



$$(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}) \quad (x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}) \quad (x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})$$

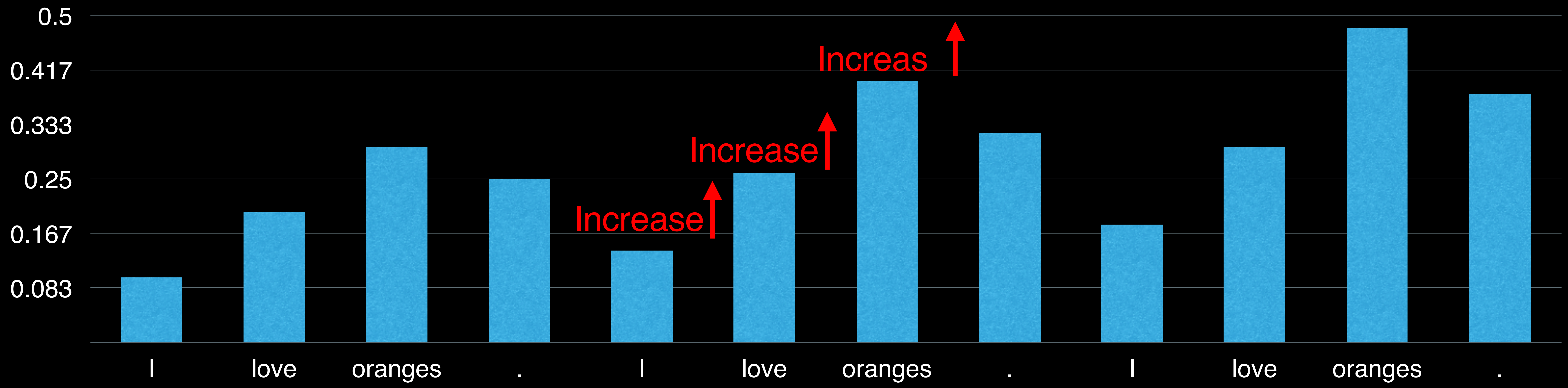
$$\mathbf{x} = (\mathbf{s}^{\mathbf{0}}, \mathbf{s}^{\mathbf{1}}, \mathbf{s}^{\mathbf{2}})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{\mathbf{1}}{L_{\mathbf{s}}} \sum_{l=1}^{L_s} \mathbb{1}(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



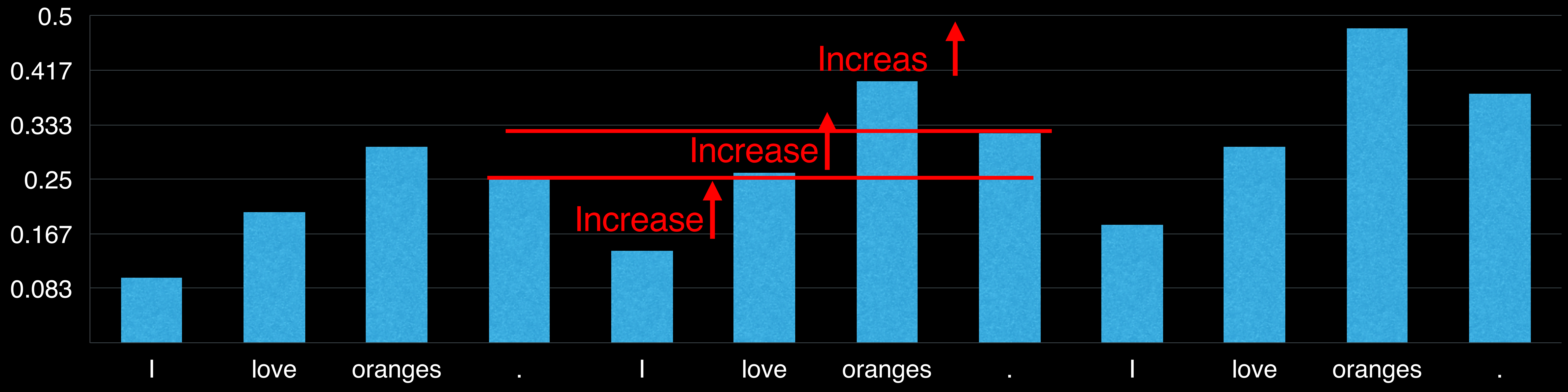
$$\mathbf{x} = (\underbrace{x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}}_{=\mathbf{s}^{\mathbf{0}}}, \underbrace{x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}}_{=\mathbf{s}^{\mathbf{1}}}, \underbrace{x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}}_{=\mathbf{s}^{\mathbf{2}}})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



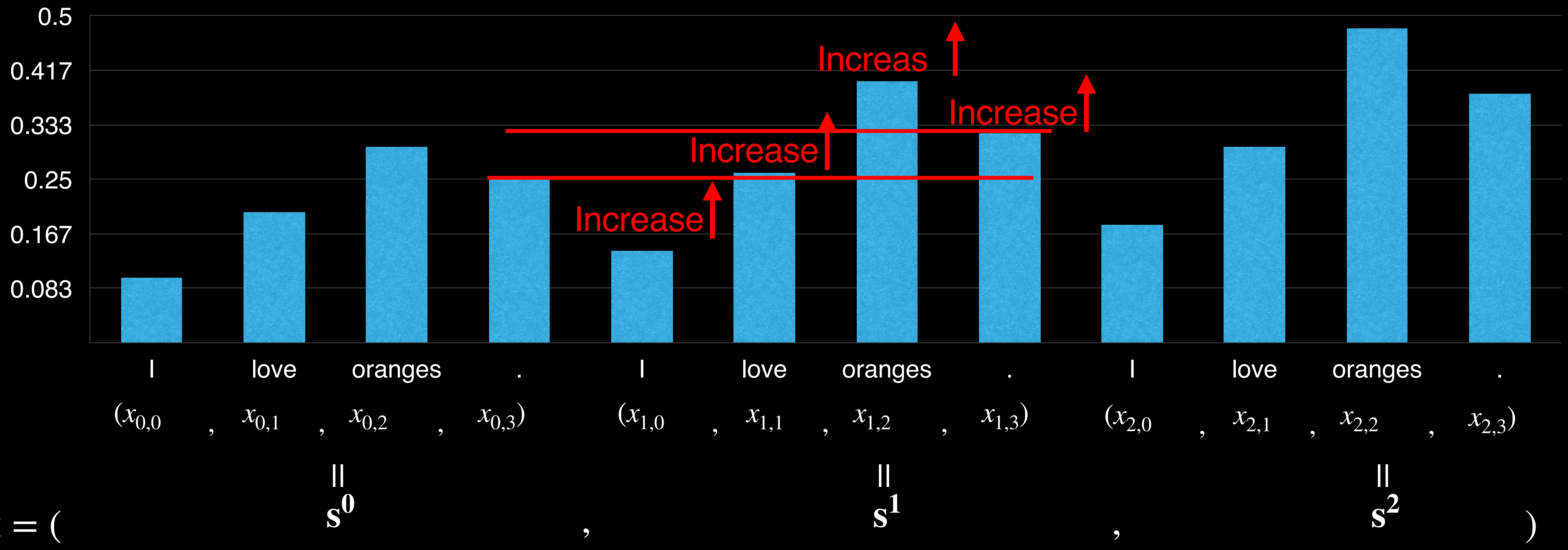
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s}^0}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s}^1}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s}^2})$$

Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s^n}) = \frac{\mathbf{1}}{L_\mathbf{s}} \sum_{l=1}^{L_s} 1(P_\theta(x_{n,l} \,|\, \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \,|\, \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities

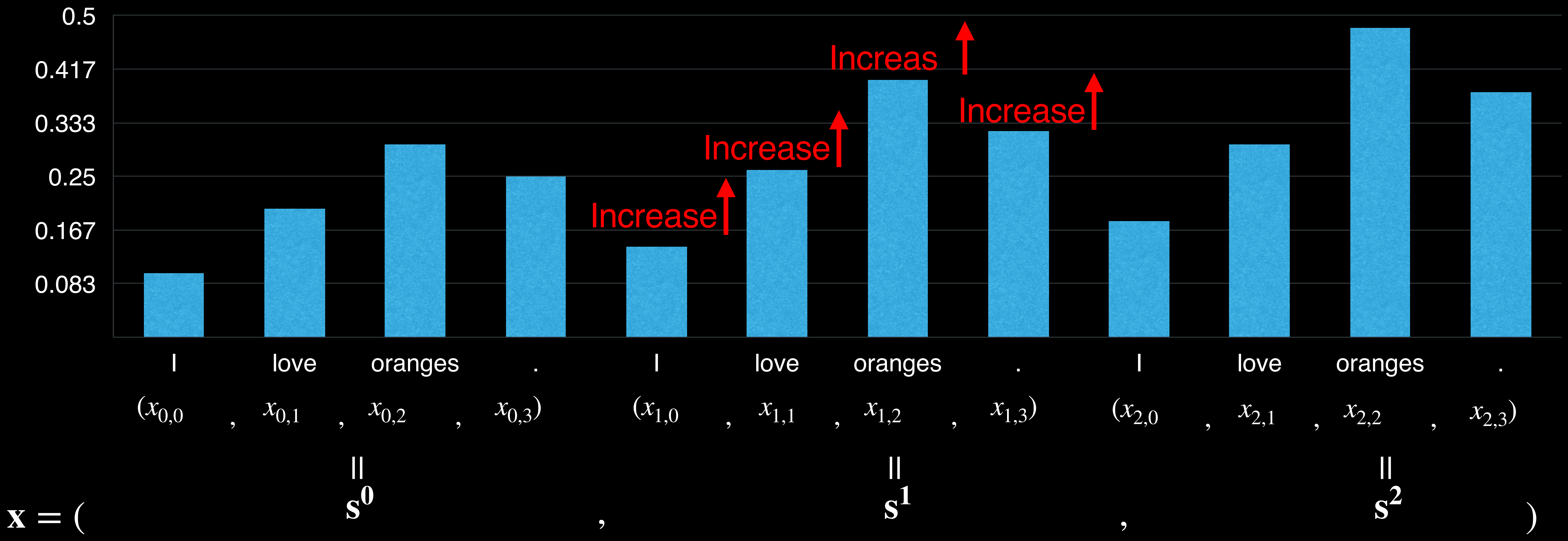


Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: **IP (Rate of Increased Token Probability)**

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{\mathbf{1}}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \,|\, \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \,|\, \mathbf{x}_{<0,l}))$$

  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities

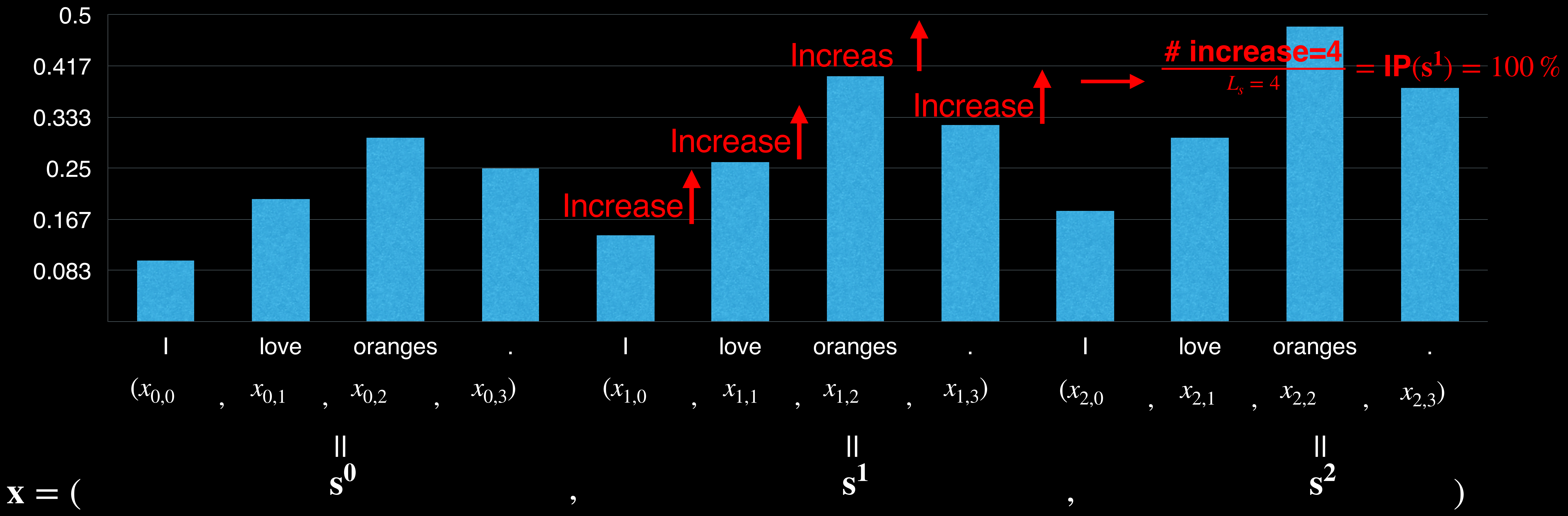


Why the First Sentence Repetition Occurs?

- Comparing prob of repetitive sentences to prob of initial sentence
  - **Metric**: IP (Rate of Increased Token Probability)

$$\mathrm{IP}(\mathbf{s}^{\mathbf{n}}) = \frac{\mathbf{1}}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l}))$$

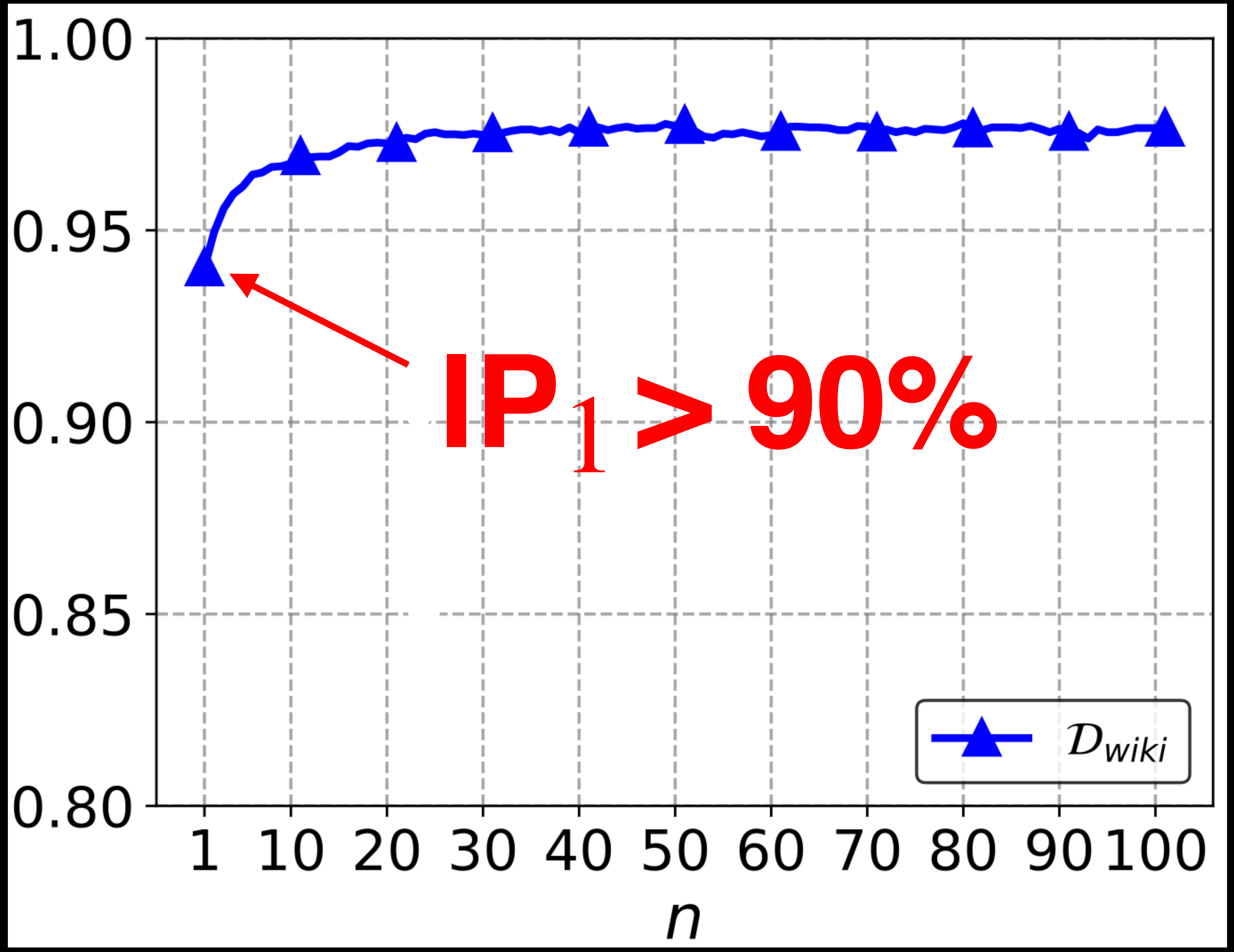
  - **Purpose**: Measure how many tokens' probabilities increase w.r.t initial probabilities



Why the First Sentence Repetition Occurs?

- Analyses
  - \> **90%** cases, probs of repeating the previous sentence **increase**
    - E.g., P('orange' | 'I love orange . I love') > P('orange' | 'I love')
  - **The model has a strong preference to repeat the previous sentence**

Y-axis: **IP (Rate of Increased Token Probability)**

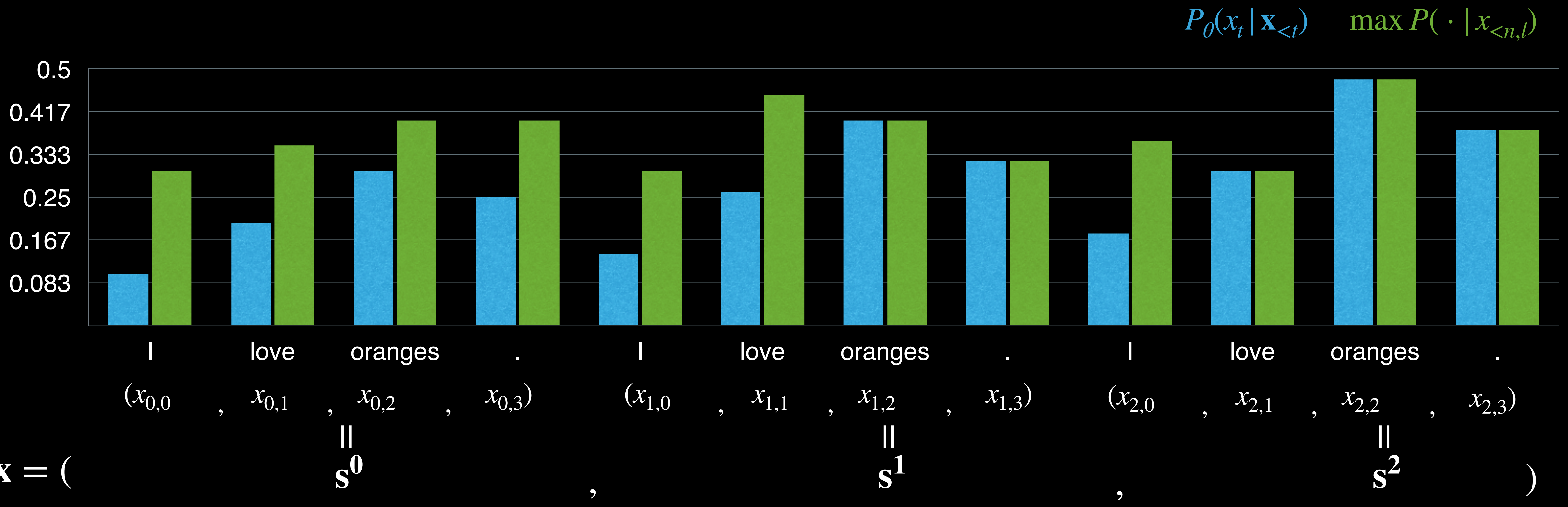


Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

   - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding

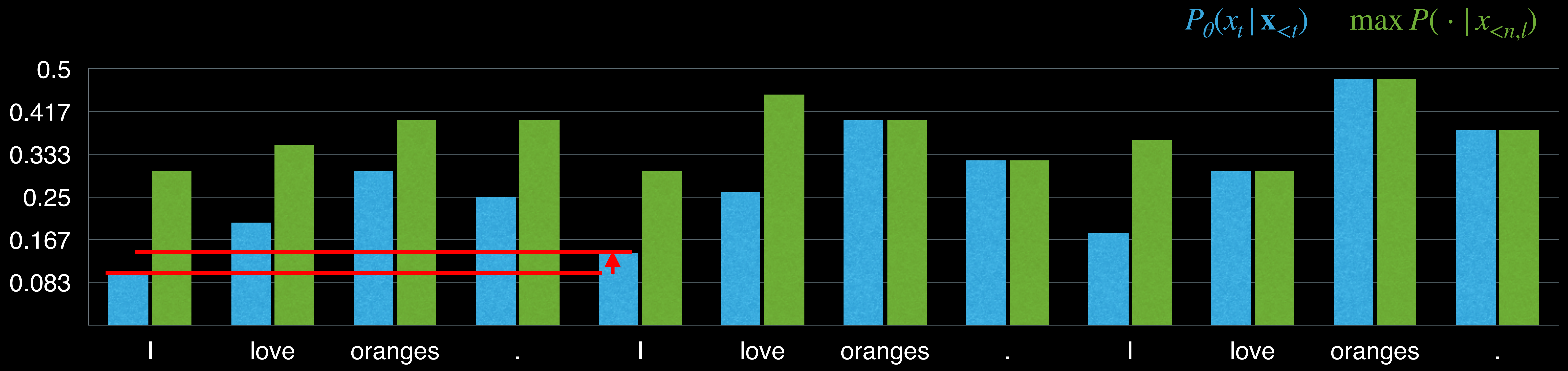


Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} 1(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} | \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} | \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot | x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



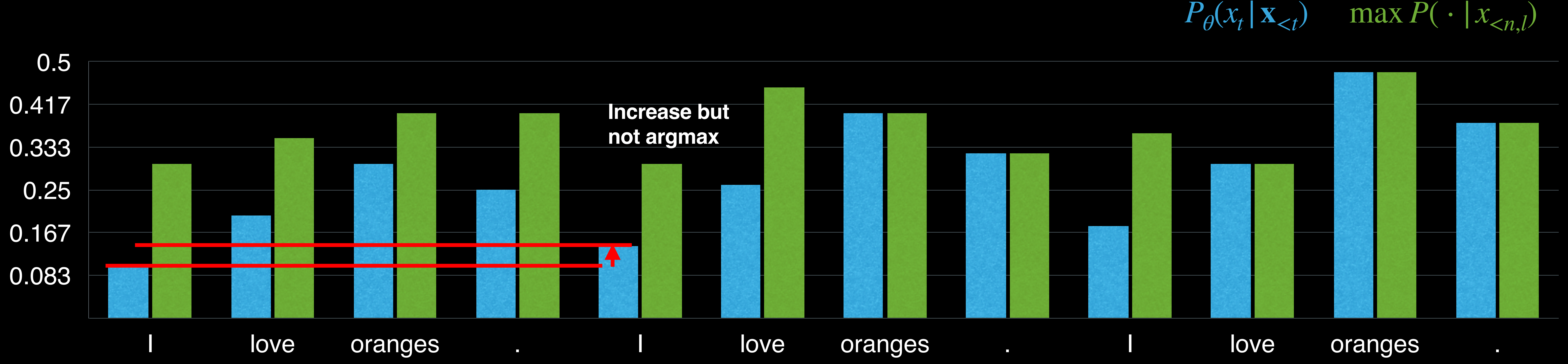
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} 1(x_{n,l} \text{ is a winner})$$

- $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l}|\mathbf{x}_{<n,l}) > P_\theta(x_{0,l}|\mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot|x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



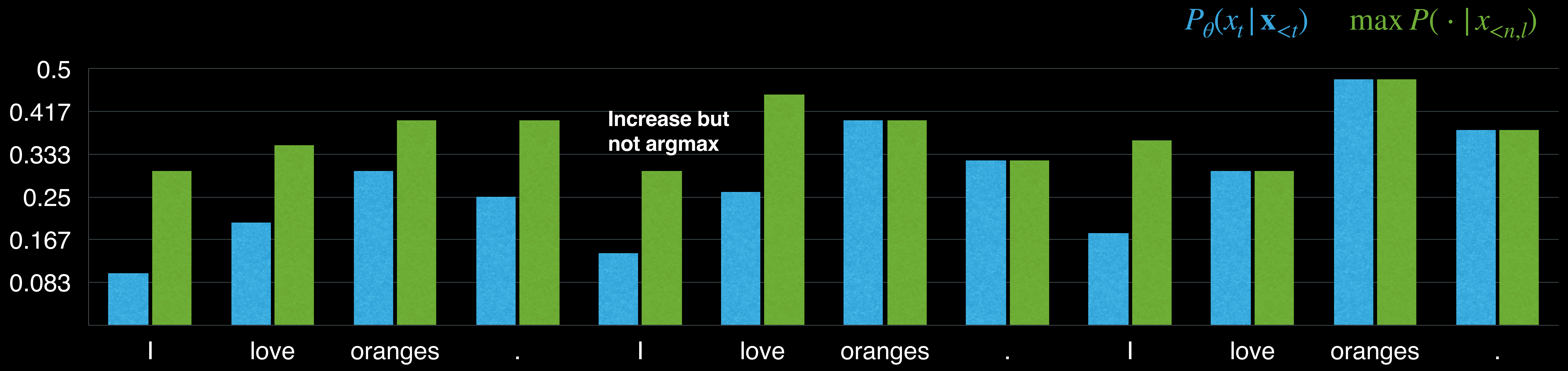
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\text{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} | \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} | \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot | x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



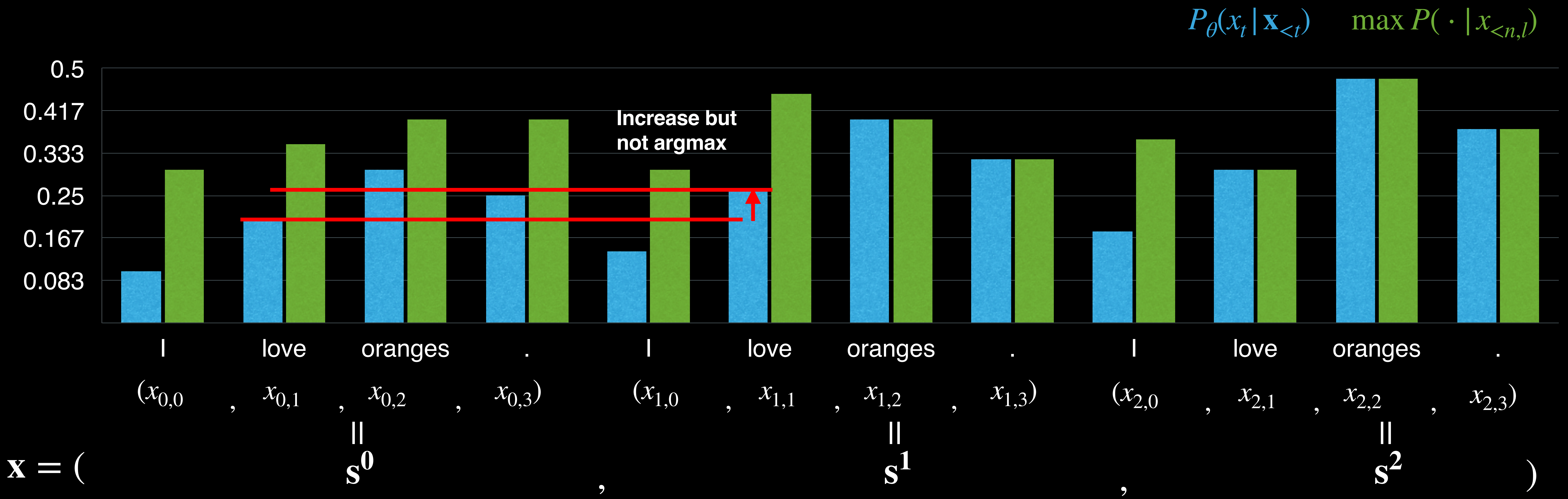
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\text{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding

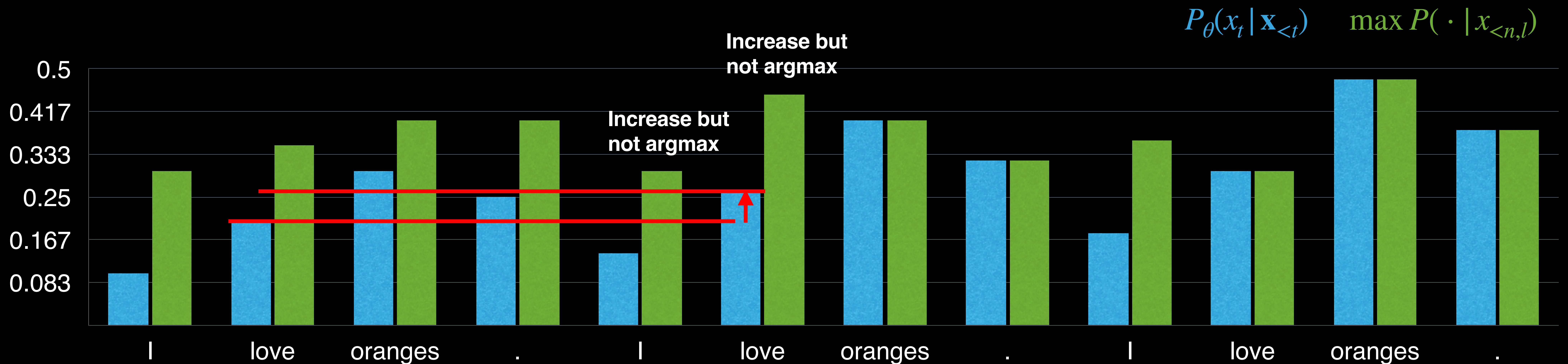


Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\text{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} 1(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} | \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} | \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot | x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



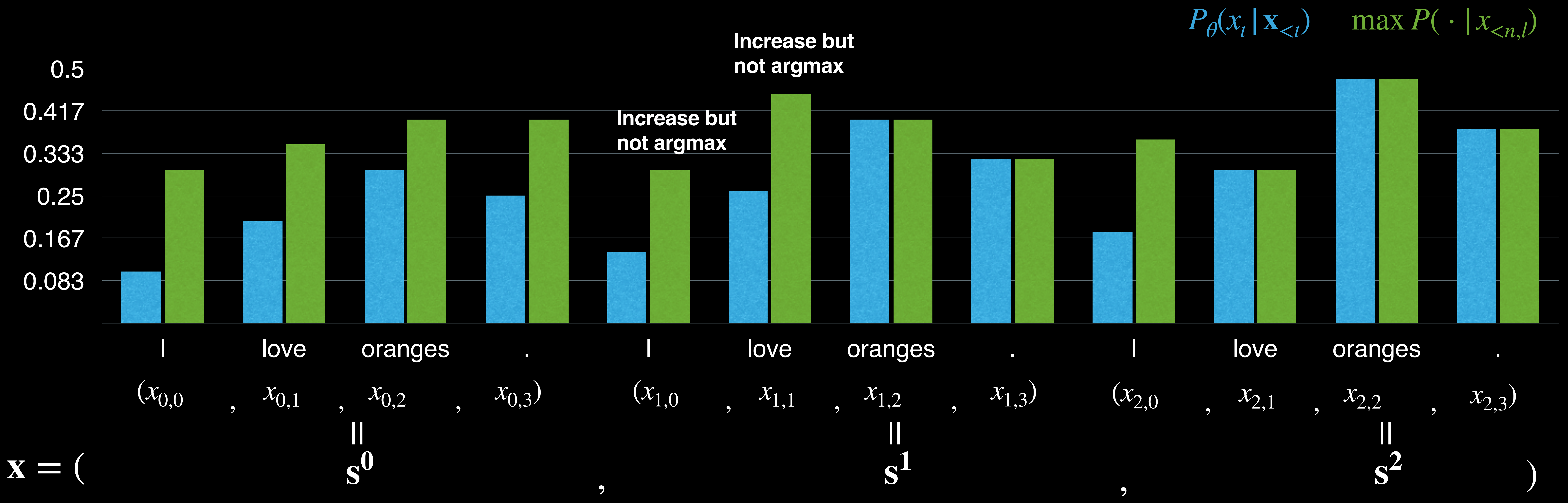
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{=\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{=\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{=\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} 1(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} | \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} | \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot | x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding

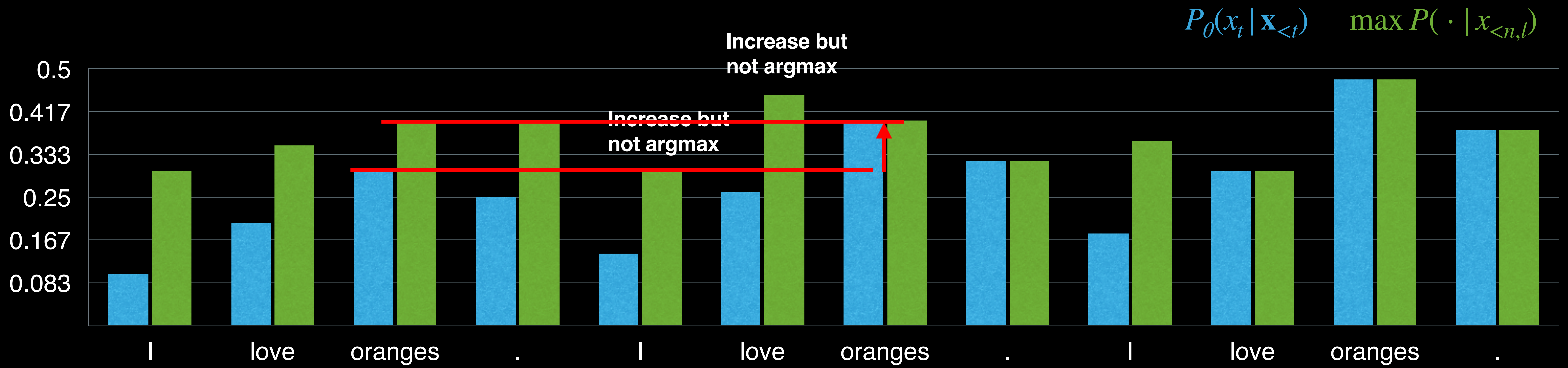


Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



$(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})\;(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})\;(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})$

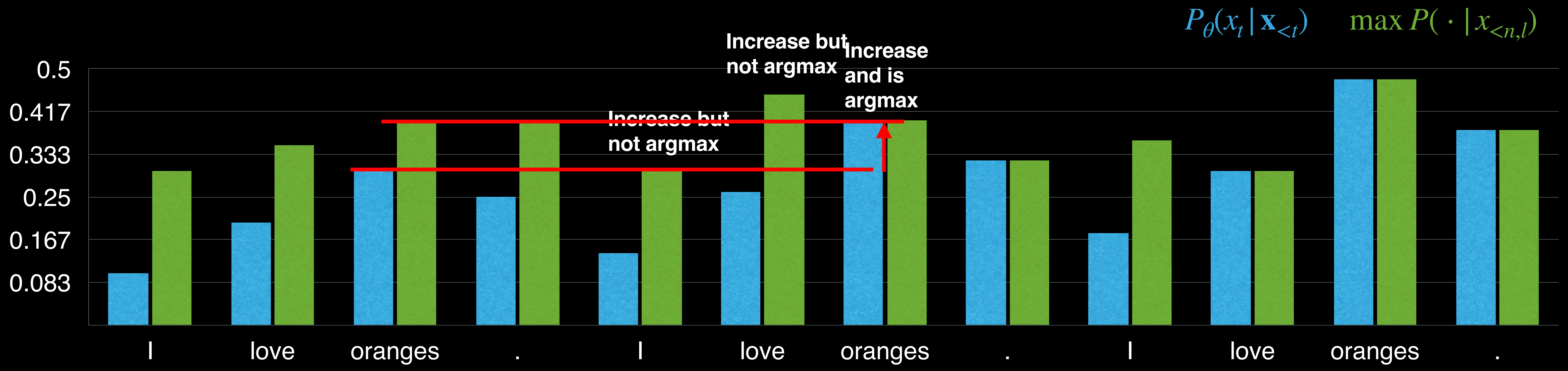
$\mathbf{x} = (\mathbf{s^0}, \mathbf{s^1}, \mathbf{s^2})$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} 1(x_{n,l} \text{ is a winner})$$

- $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



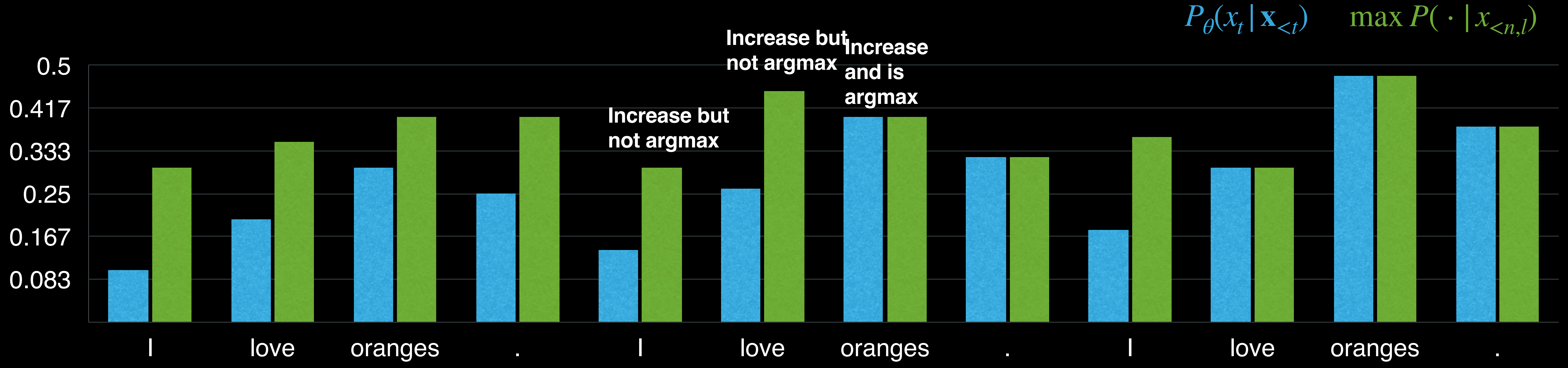
$$\mathbf{x} = (\underbrace{x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}}_{=\mathbf{s}^0}, \underbrace{x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}}_{=\mathbf{s}^1}, \underbrace{x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}}_{=\mathbf{s}^2})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\text{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} 1(x_{n,l} \text{ is a winner})$$

- $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} | \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} | \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot | x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



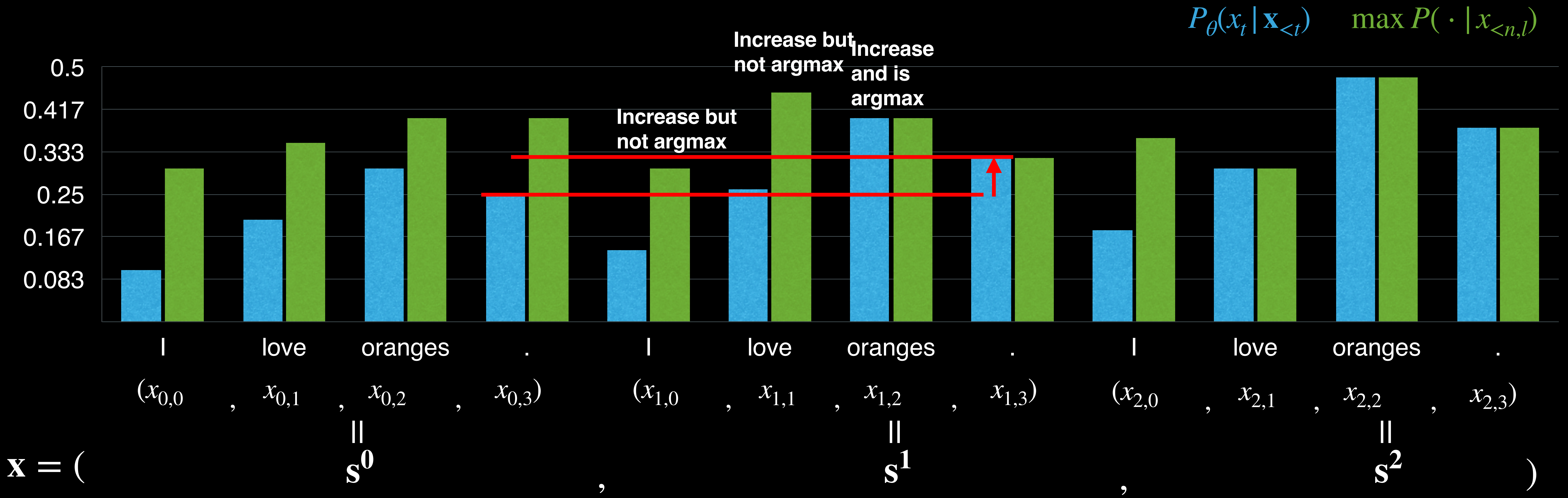
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{=\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{=\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{=\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding

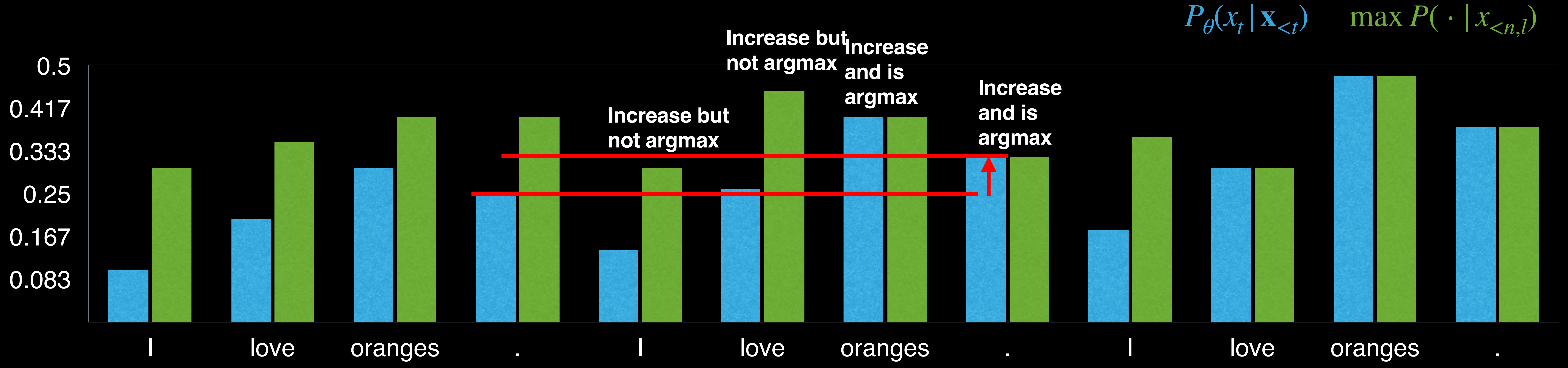


Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\text{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} | \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} | \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot | x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



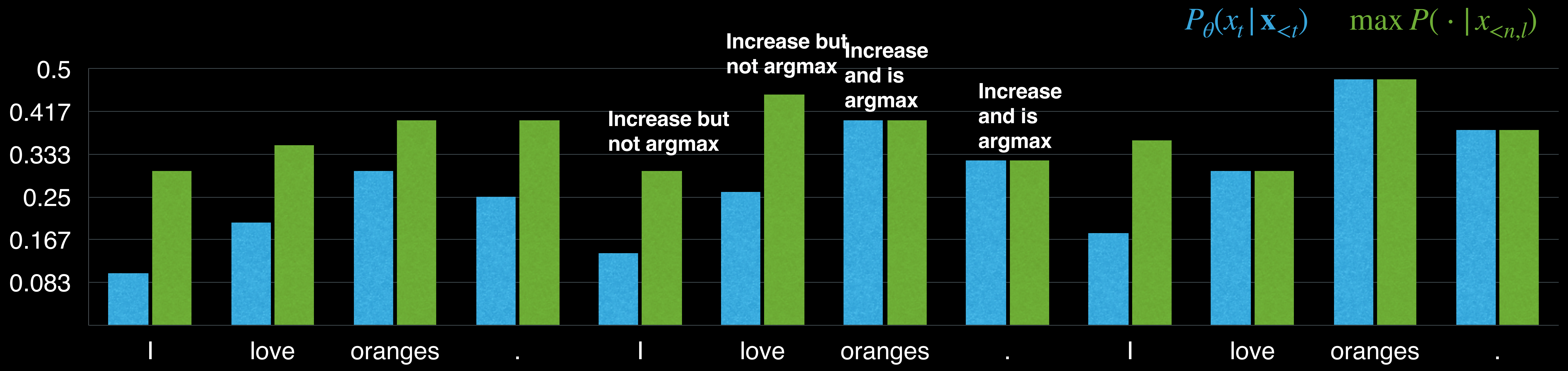
$$\mathbf{x} = (\underbrace{x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3}}_{\mathbf{s^0}}, \underbrace{x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3}}_{\mathbf{s^1}}, \underbrace{x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3}}_{\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\text{WR}(\mathbf{s^n}) = \frac{1}{L_\mathbf{s}} \sum_{l=1}^{L_\mathbf{s}} \mathbb{1}(x_{n,l} \text{ is a winner})$$

  - $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



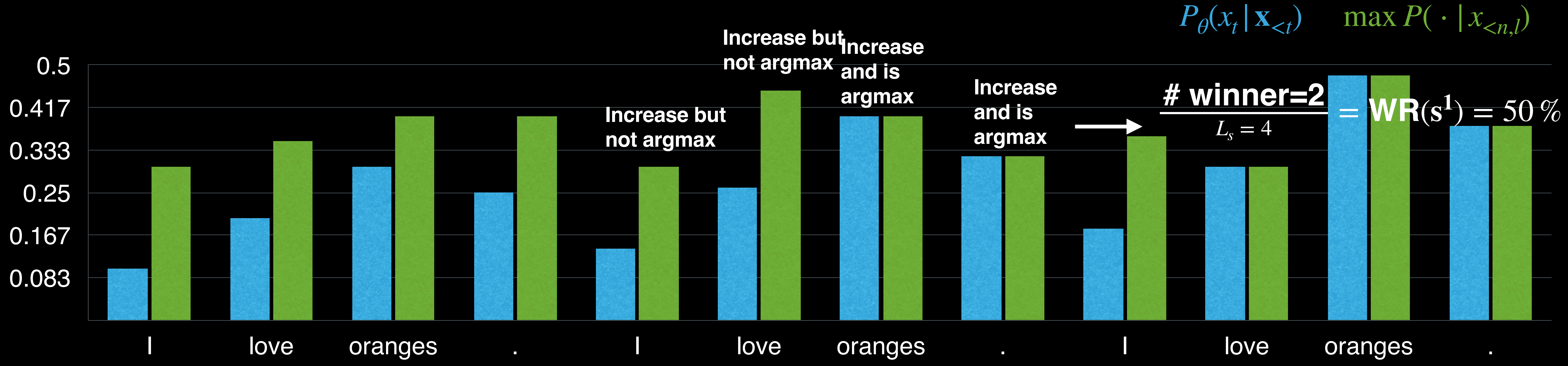
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

- Comparing the prob of repetitive sentences as number of repetition grows
  - **Metric**: **WR (Winner Rate)**

$$\mathrm{WR}(\mathbf{s^n}) = \frac{1}{L_s}\sum_{l=1}^{L_s} 1(x_{n,l} \text{ is a winner})$$

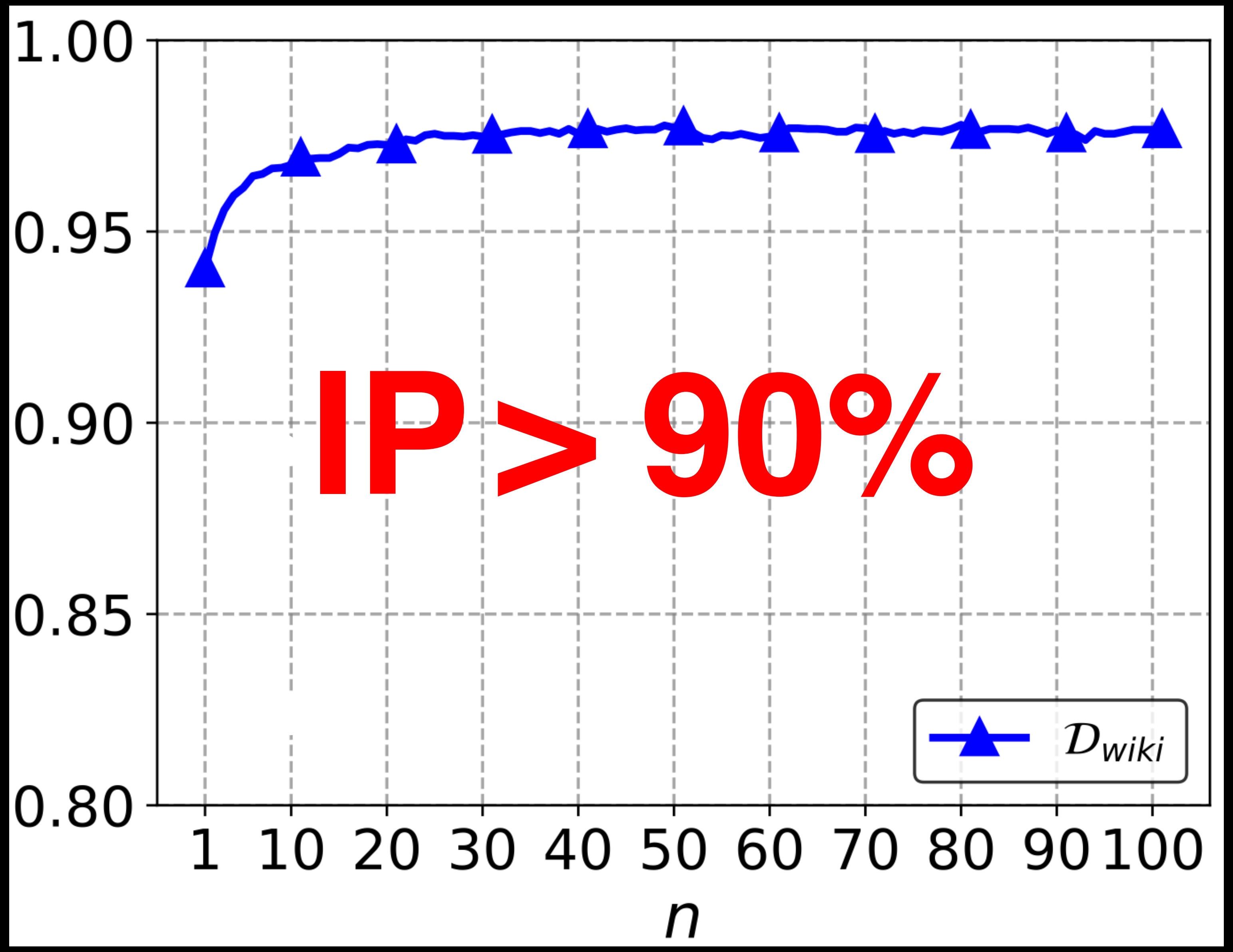
- $x_{n,l}$ is a *winner* if $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) > P_\theta(x_{0,l} \mid \mathbf{x}_{<0,l})$ and $x_{n,l} = \arg\max P(\cdot \mid x_{<n,l})$
  - **Purpose**: Measure how many of tokens are more likely to be **generated** by greedy decoding



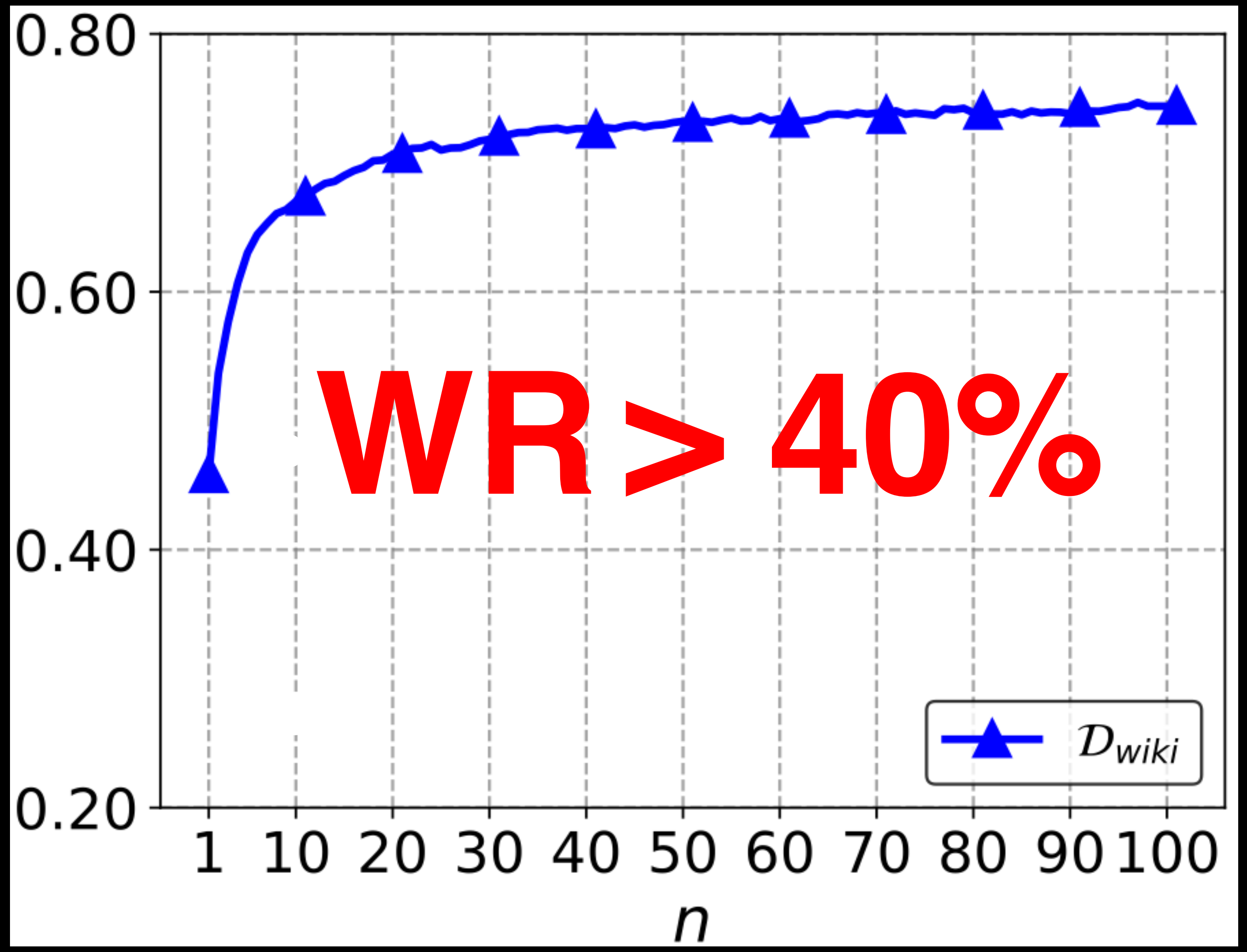
$$\mathbf{x} = (\underbrace{(x_{0,0}, x_{0,1}, x_{0,2}, x_{0,3})}_{\mathbf{s^0}}, \underbrace{(x_{1,0}, x_{1,1}, x_{1,2}, x_{1,3})}_{\mathbf{s^1}}, \underbrace{(x_{2,0}, x_{2,1}, x_{2,2}, x_{2,3})}_{\mathbf{s^2}})$$

Why Model Gets Stuck into the Sentence-level Loop?

***Y-axis*: IP (Rate of Increased Token Probability)**



***Y-axis*: WR (Winner Rate)**



- Analyses
  - \> **40%** cases, the first repetition occurs. That is, the **previous** sentence is repetitively generated with 40% probability.
  - **Self-reinforcement effect**: As number of repetitions grows, IP and WR **significantly** increase. In other words, more times repeating a sentence, higher probability continuing to generate that sentence.

What Kinds of Sentences are More Likely to be Repeated?

- Investigate sentences with different initial probabilities
  - **Metric**: **TP (Average Token Probabilities)**

$$\mathrm{TP}(\mathbf{s^n}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l})$$

  - **Purpose**: Measure the average token probability of the n-th sentence $\mathbf{s}^n$

What Kinds of Sentences are More Likely to be Repeated?

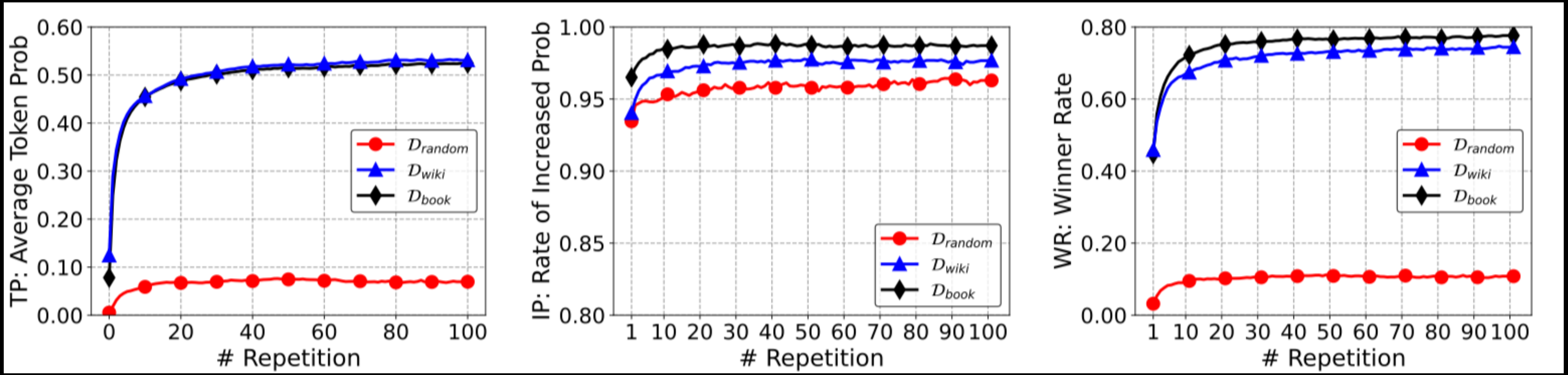
- Investigate sentences with different initial probabilities
  - **Metric**: **TP (Average Token Probabilities)**

$$\text{TP}(\mathbf{s}^{\mathbf{n}}) = \frac{1}{L_{\mathbf{s}}} \sum_{l=1}^{L_{\mathbf{s}}} P_{\theta}(x_{n,l} \,|\, \mathbf{x}_{<n,l})$$

  - **Purpose**: Measure the average token probability of the n-th sentence $\mathbf{s}^n$

- Investigate TP, IP and WR across different corpus
  - Random Sentences [D_{random}]: randomly sampled tokens
    - E.g., "fría backed rounds Manganiello Stansel Zemin compressus ."
  - Out-domain Sentences [D_{book}]: BookCorpus
  - In-domain Sentences [D_{wiki}]: dev set of Wikitext-103
- For each corpus, we calculate $[\mathbf{TP}_n, \mathbf{IP}_n, \mathbf{WR}_n]_{n=1}^{N}$

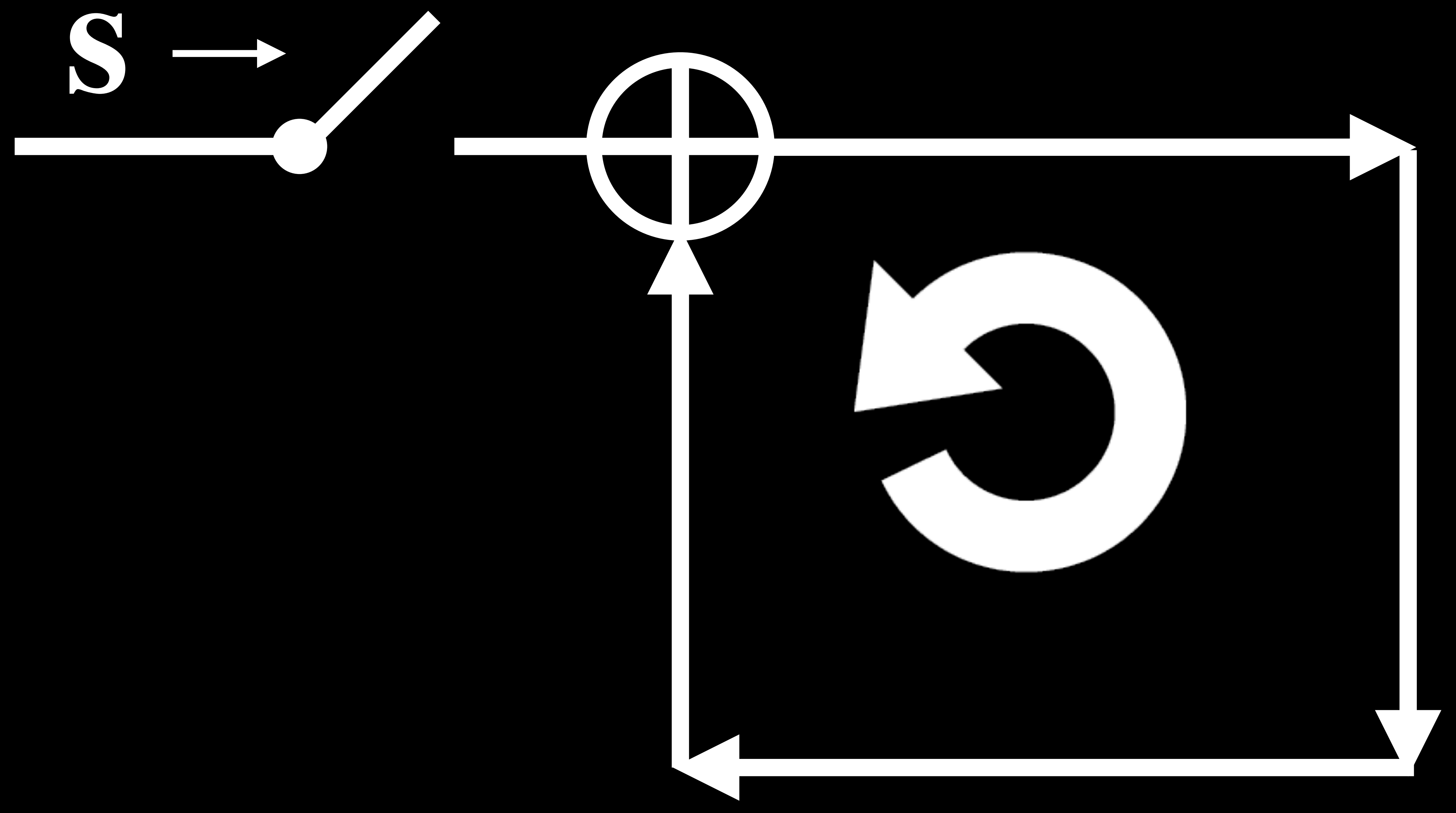
What Kinds of Sentences are More Likely to be Repeated?



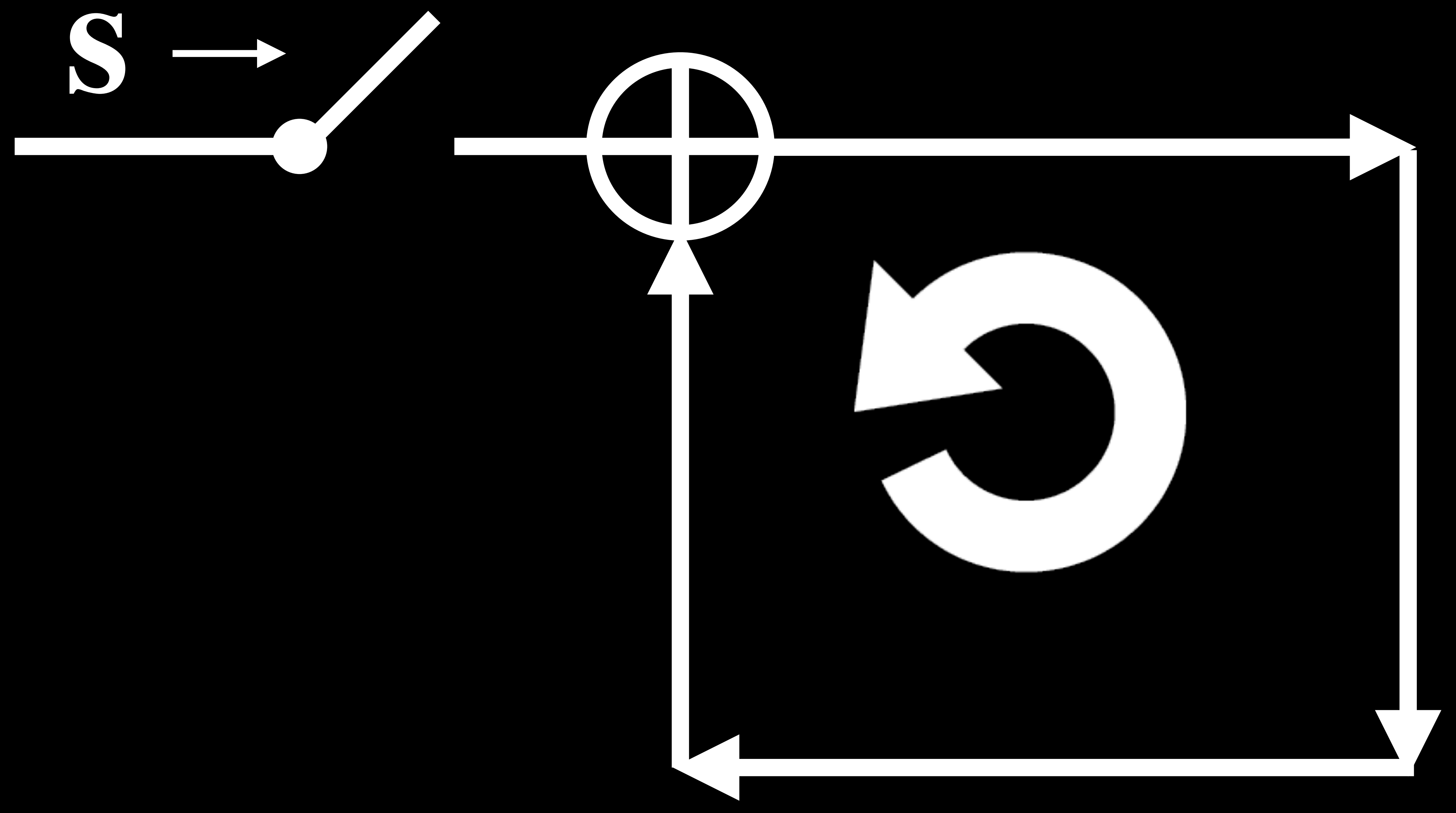
Analyses

- **Self-reinforcement effect exists even in random sentences**
- **High prob sentences are more likely to be repeated.**

Self-reinforcement Effect



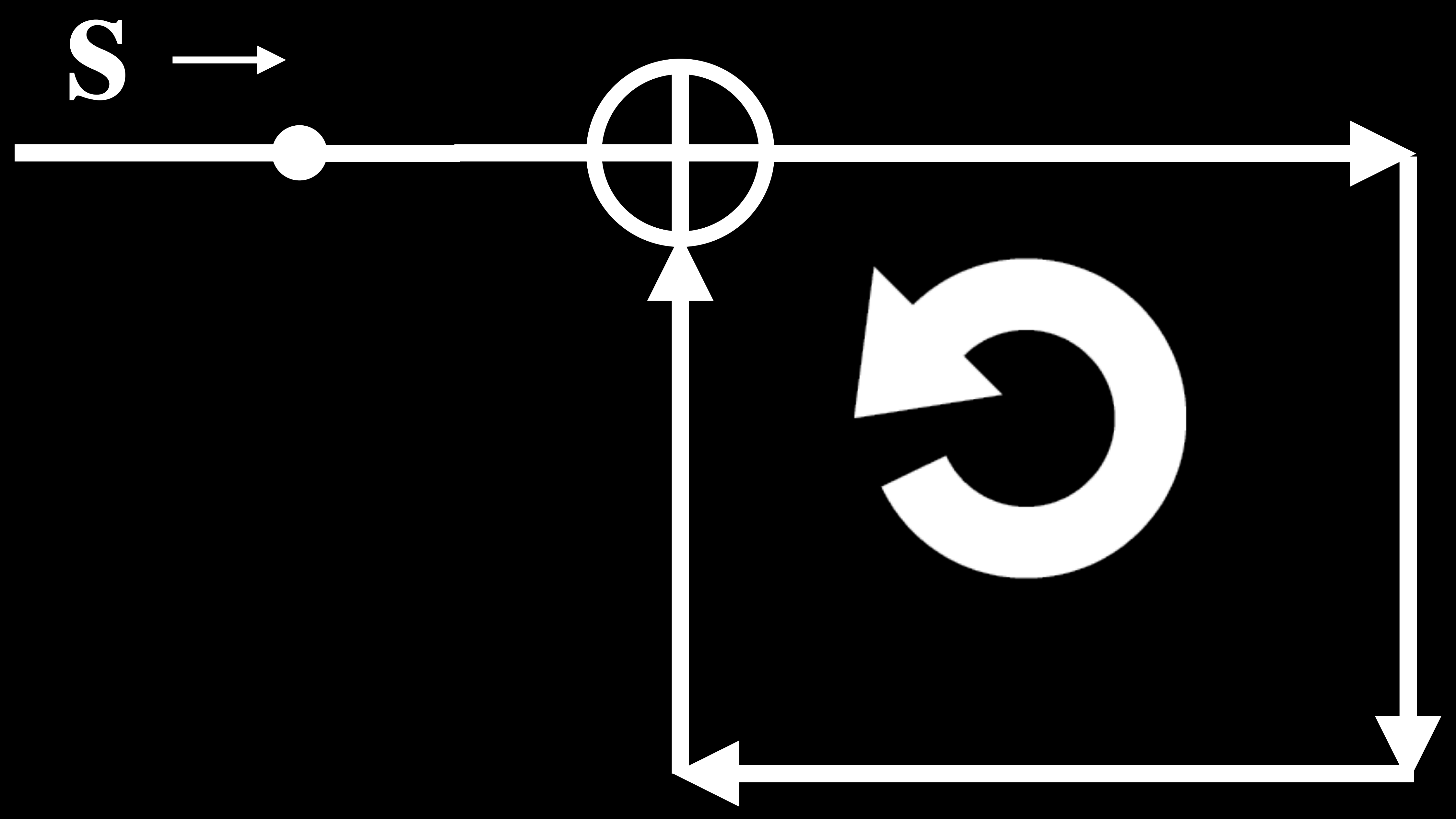
Self-reinforcement Effect



Enter

High likelihood sentences are more likely to go into the loop.

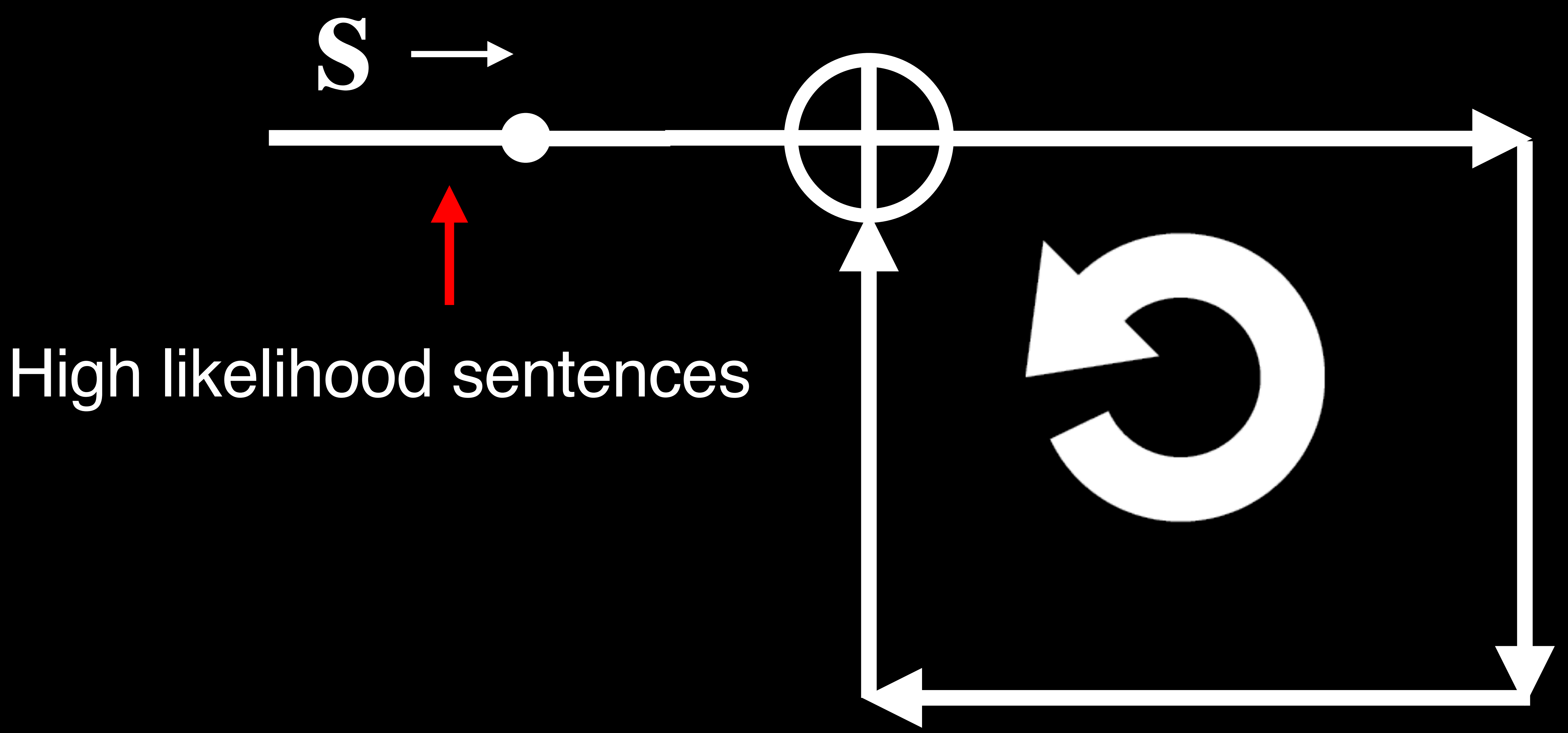
Self-reinforcement Effect



Enter

High likelihood sentences are more likely to go into the loop.

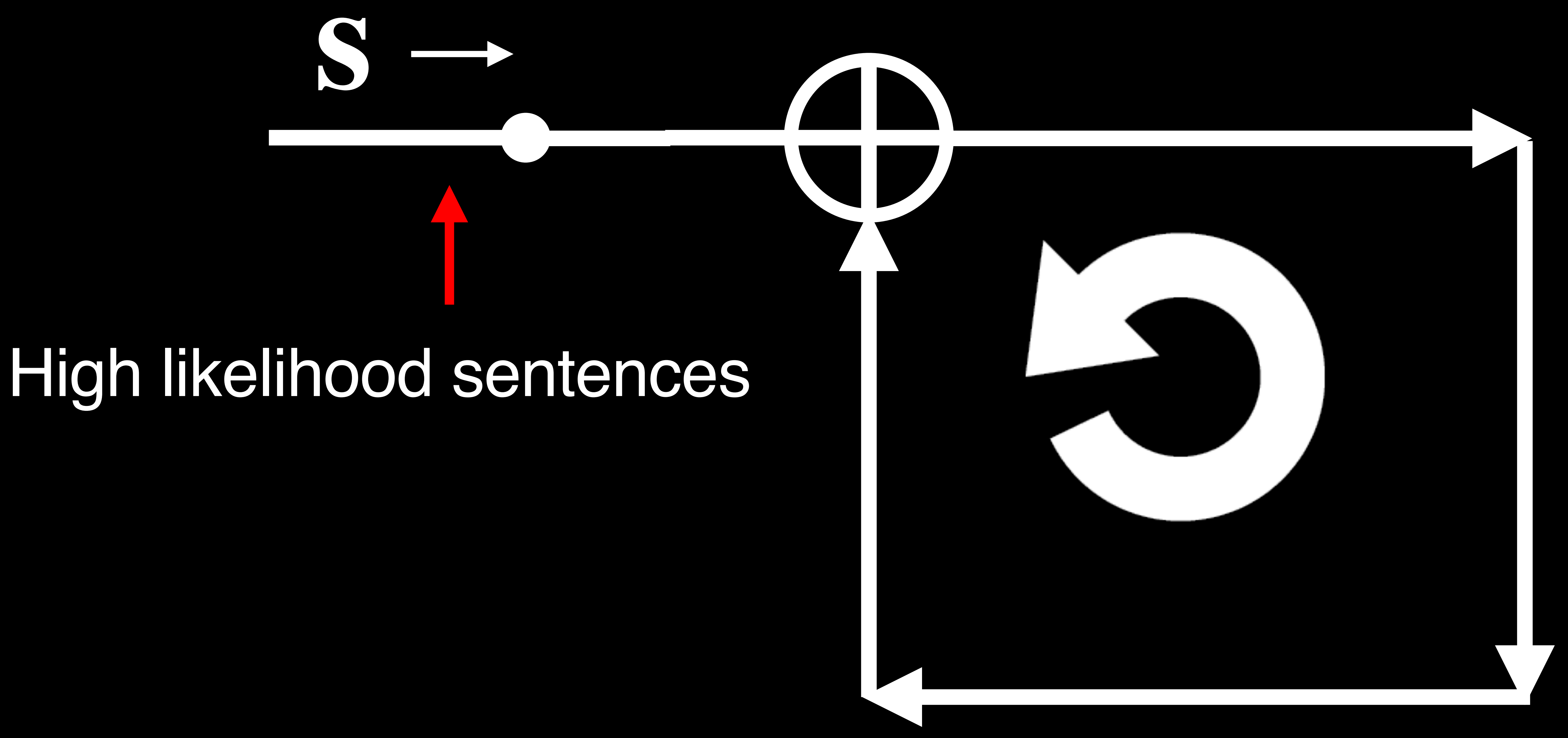
Self-reinforcement Effect



Enter

High likelihood sentences are more likely to go into the loop.

Self-reinforcement Effect



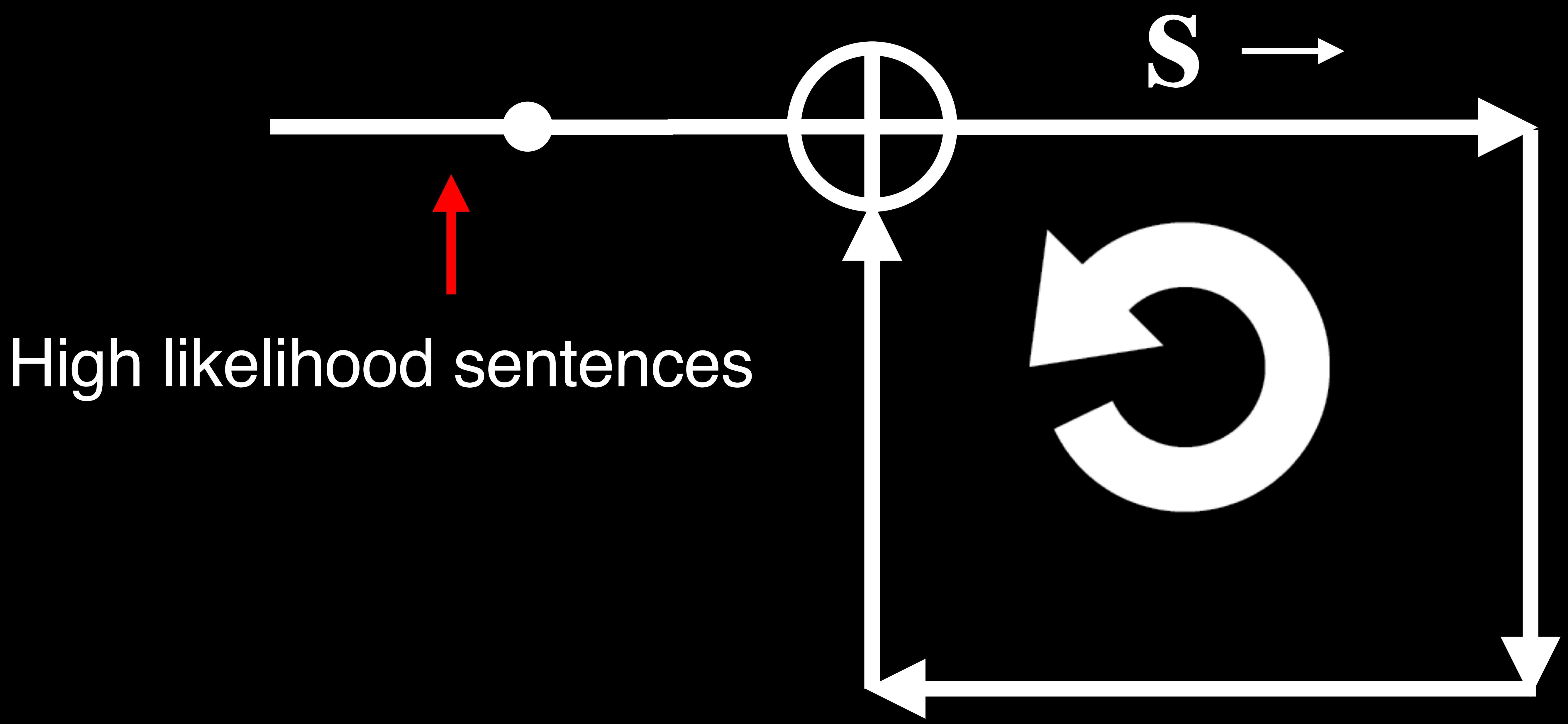
Enter

High likelihood sentences are more likely to go into the loop.

Enhance

At the first repetition, model prefers to further increase the prob of repeating the last sentence

Self-reinforcement Effect



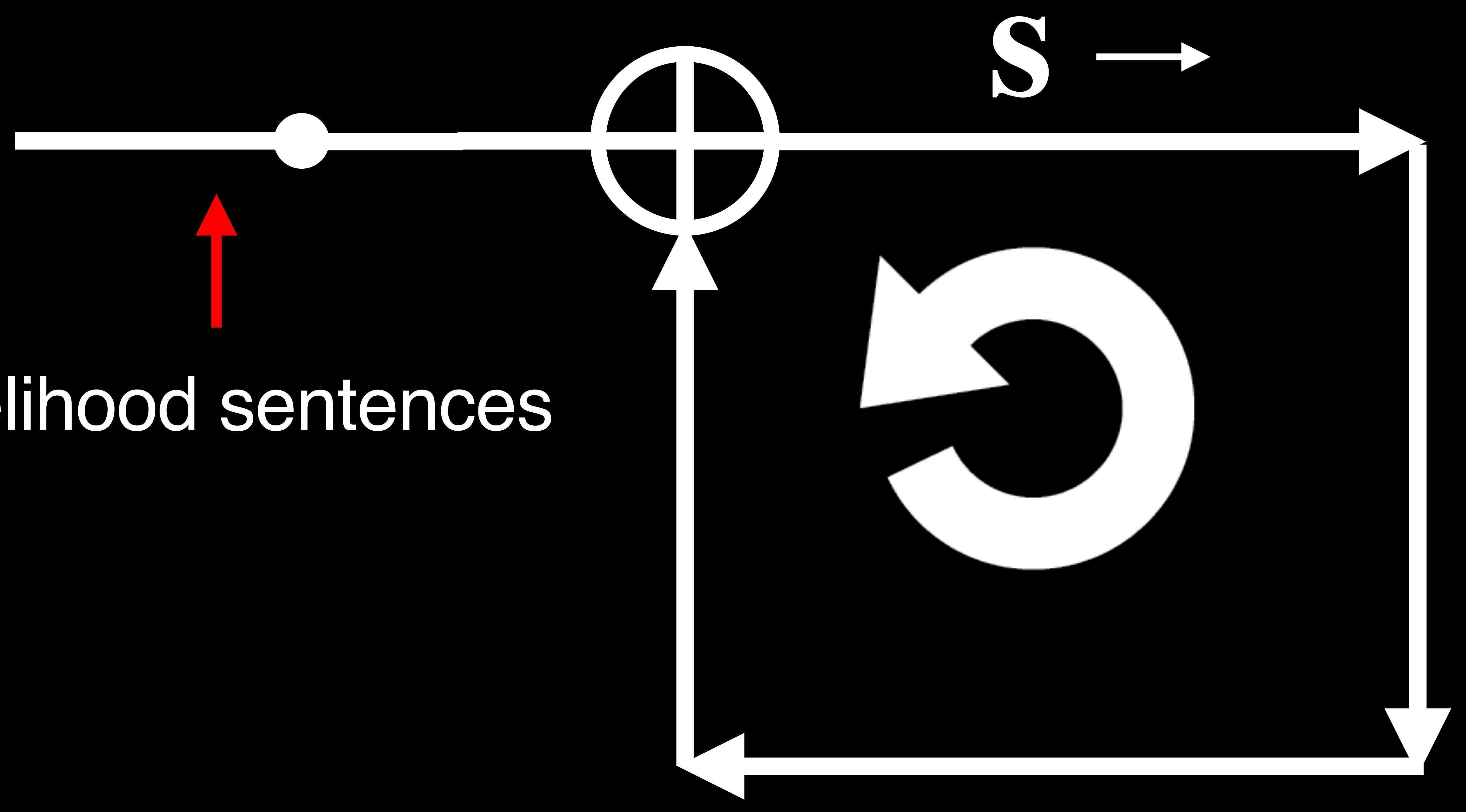
Enter

High likelihood sentences are more likely to go into the loop.

Enhance

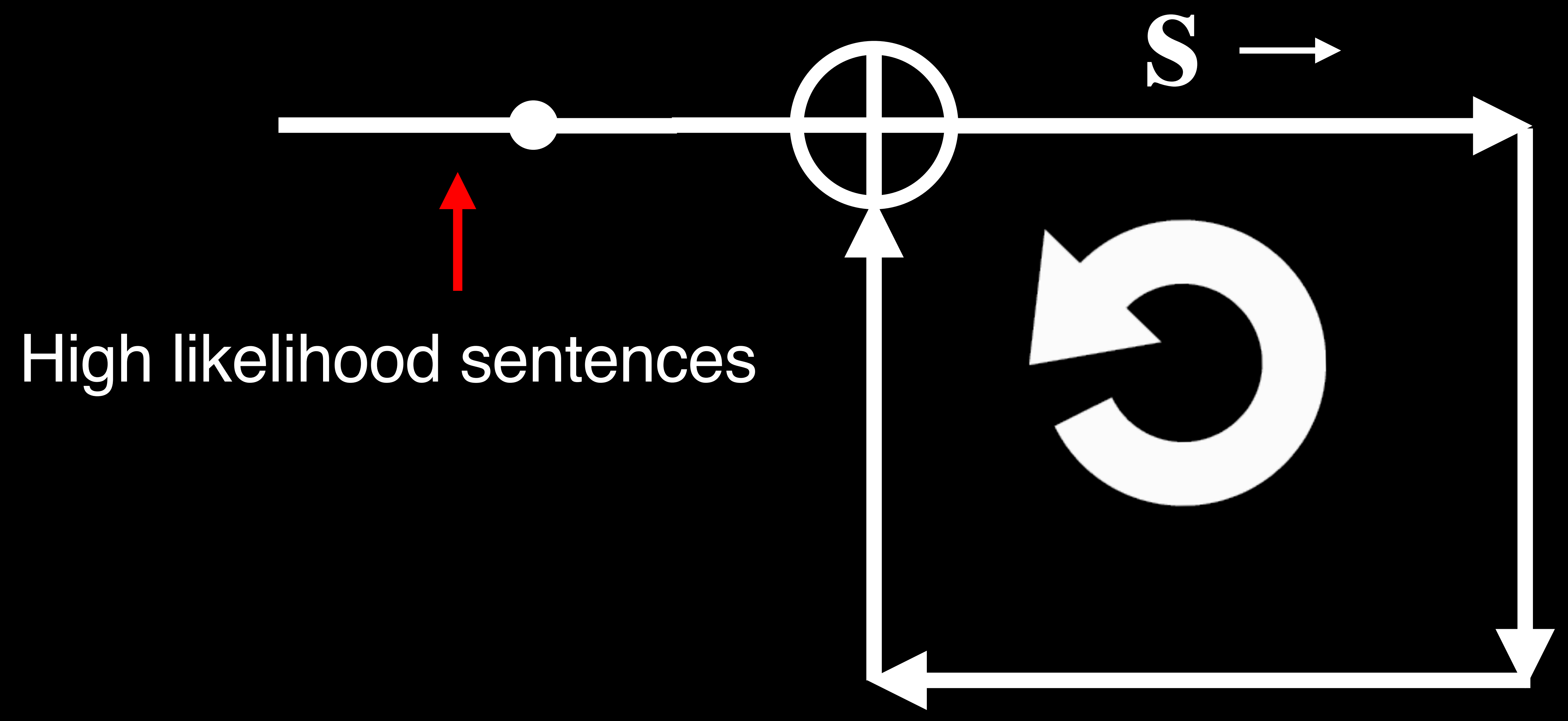
At the first repetition, model prefers to further increase the prob of repeating the last sentence

Self-reinforcement Effect



Enter	Enhance	Loop
High likelihood sentences are more likely to go into the loop.	At the first repetition, model prefers to further increase the prob of repeating the last sentence	When repeating the sentence for several times, it would get stuck in the sentence loop due to ***self-reinforcement effect***

Self-reinforcement Effect



Enter

High likelihood sentences are more likely to go into the loop.

Enhance

At the first repetition, model prefers to further increase the prob of repeating the last sentence

Loop

When repeating the sentence for several times, it would get stuck in the sentence loop due to ***self-reinforcement effect***

- Introduction
- Related Work
- Analyzing Repetition Problems
- **DITTO - a Method to Mitigate Repetitions**
- Experiments
- Future Work

DITTO - pseuDo-repetITion penalizaTiOn

- **Core issue**: **Self-reinforcement effect**
- **Reason**: Models don't know how to handle repetitive sentences
- **Motivation**: Let model train on repetitive sentence and learn to be averse to such repetitions
- **Method**
    - Positive Data: Ground-truth corpus
    - **Negativa Data: Pseudo Repetitive data**
        - Randomly pick a sentence $\mathbf{s}$ from the training corpus
        - Repeat $\mathbf{s}$ until they reaches the maximum input sequence length

$$\mathbf{x} = (\mathbf{s}^0, \cdots, \mathbf{s}^N) = (x_{0,0}, \cdots, x_{1,0}, \cdots, x_{N,0}, \cdots, x_{N,L_s})$$

    - Combine two kinds of data for training

DITTO

- **Sentence-level Repetition Penalization on Negative Data**
  - Per-step penalization loss for token $x \in \{x_{1,0}, \cdots, x_{N,L_s}\}$
  - Training objective for the l-th token in the n-th repetition

$$\mathbf{L}^{n,l}_{\text{DITTO}}(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l})) = -\log(1 - \left| P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) - \lambda \cdot P^*_\theta(x_{n-1,l} \mid \mathbf{x}_{<n-1,l}) \right|)$$

- $P*$ is excluded for gradient backpropgation and λ is the penalization factor
- Train the model by equally mixing $\mathbf{L}_{\text{DITTO}}$ update and normal MLE loss update.

DITTO

- **Sentence-level Repetition Penalization on Negative Data**
  - Per-step penalization loss for token $x \in \{x_{1,0}, \cdots, x_{N,L_s}\}$
  - Training objective for the l-th token in the n-th repetition

$$\mathbf{L}^{n,l}_{\text{DITTO}}(P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l})) = -\log\left(1 - \left| P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) - \lambda \cdot P^*_\theta(x_{n-1,l} \mid \mathbf{x}_{<n-1,l}) \right|\right)$$

  - $P*$ is excluded for gradient backpropgation and λ is the penalization factor
  - Train the model by equally mixing $\mathbf{L}_{\text{DITTO}}$ update and normal MLE loss update.
- **If λ = 1**
  - Loss minimized when $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) = P^*_\theta(x_{n-1,l} \mid \mathbf{x}_{<n-1,l})$
  - It can avoid repetition probability increase monotonically.
- **If λ < 1**
  - Loss minimized when $P_\theta(x_{n,l} \mid \mathbf{x}_{<n,l}) = \lambda \cdot P^*_\theta(x_{n-1,l} \mid \mathbf{x}_{<n-1,l})$ where λ<1.
  - The probability of tokens in repetitive sentence should decay ***exponentially*** with a factor of λ

- Introduction
- Related Work
- Analyzing Repetition Problems
- DITTO - a Method to Mitigate Repetitions
- **Experiments**
- Future Work

Examples of Open-ended Generations (Greedy)

Context: The reports generated considerable discussion in the press which ultimately led to a heated debate in Parliament on 4 May 1920 .

MLE (baseline): Sir Edward Cave , the Director of the Public Prosecutions , was quoted as saying that Cave had " no idea what the truth was , and that the truth was not the truth. But what the truth was , and that the truth was not the truth . But what the truth was , and that the truth was not the truth . But what the truth was , and that the truth was not the truth . </s> But what the truth was , and that the truth was not the truth .

DITTO: Sir Edward Cave , the Director of Public Prosecutions , was quoted as saying that " the Government has not been able to make a decision on the issue of the bodies of the dead . " </s> </s> = = = Public reaction = = = </s> </s> The public reaction to the killings was generally positive . The Times called the actions " a most appalling and appalling act " and the Daily Mail called for the police to be withdrawn .

Experiments of Open-ended Generations (Greedy Decoding)

Table 1: Results of different training-based methods on the test set of Wikitext-103 for open-ended generation. The results are reported based on three runs with different random seeds. The best value is **bolded** and the second best is <u>underlined</u>.

Model	MAUVE	Perplexity	Accuracy	Repetition-4	Repetition-Sen
MLE [26]	$0.34_{\pm 0.02}$	$25.68_{\pm 0.04}$	$0.39_{\pm 0.00}$	$44.20_{\pm 1.43}\%$	$14.50_{\pm 1.59}\%$
UL-token [32]	$0.57_{\pm 0.01}$	$26.98_{\pm 0.12}$	$0.39_{\pm 0.00}$	$28.30_{\pm 0.78}\%$	$7.40_{\pm 0.83}\%$
UL-token+seq [32]	$0.48_{\pm 0.03}$	$25.95_{\pm 0.08}$	$0.40_{\pm 0.00}$	$\mathbf{7.60_{\pm 0.46}\%}$	$\mathbf{0.05_{\pm 0.03}\%}$
SG [17]	$0.74_{\pm 0.01}$	$25.84_{\pm 0.06}$	$0.40_{\pm 0.00}$	$23.00_{\pm 0.28}\%$	$5.24_{\pm 0.75}\%$
DITTO (ours)	$\mathbf{0.77_{\pm 0.01}}$	$\mathbf{24.33_{\pm 0.04}}$	$\mathbf{0.42_{\pm 0.00}}$	$22.00_{\pm 0.31}\%$	$2.85_{\pm 0.74}\%$
Human	-	-	-	1.10%	0.01%

- **MAUVE** (Pillutla et al., 2021): MAVE is automatic metric to measure how close model generated-text is to human language
  - The large, the better
- **Repetition**: Portion of duplicate 4-grams/sentences in generated sequences
  - The closer to human, the better

DITTO achieve the highest MAUVE with lowest perplexity and highest accuracy.

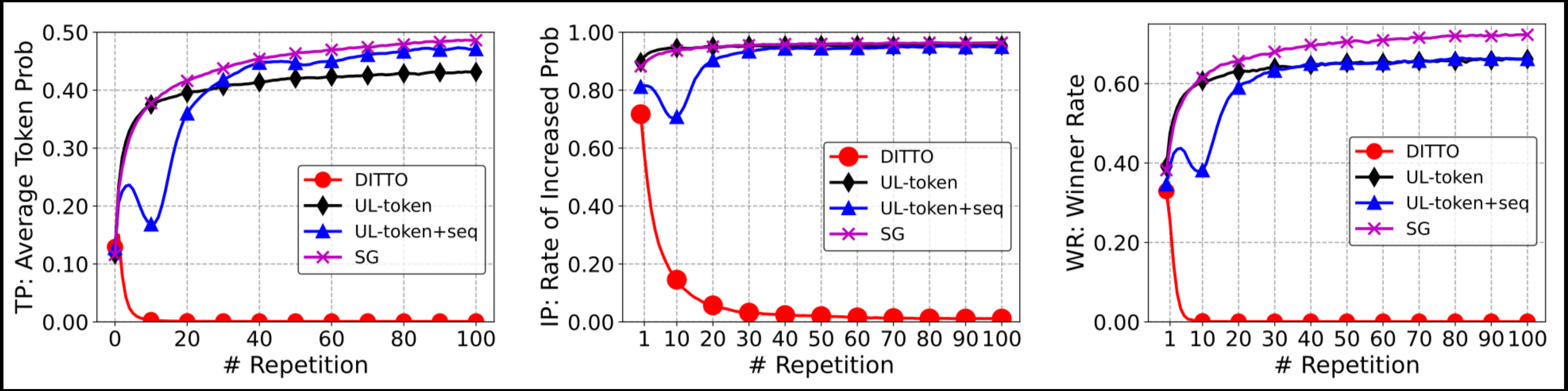
Experiments of Open-ended Generations (Stochastic Decoding)

Table 2: Results of different training-based methods on the test set of Wikitext-103 under different stochastic decoding algorithms. $k = 50$ and top-p ($p = 0.9$) for nucleus sampling. The numbers *closest* to *human scores* are in **bold** except for MAUVE [24].

Search	Model	MAUVE	Repetition-4	Repetition-Sen
Top-k	MLE [26]	$0.94_{\pm 0.00}$	$1.60_{\pm 0.09}\%$	$0.25_{\pm 0.06}‰$
	UL-token [32]	$0.95_{\pm 0.00}$	$0.70_{\pm 0.13}\%$	$0.00_{\pm 0.00}‰$
	UL-token+seq [32]	$0.93_{\pm 0.01}$	$0.09_{\pm 0.11}\%$	$0.06_{\pm 0.02}‰$
	SG [17]	$0.93_{\pm 0.01}$	$0.50_{\pm 0.19}\%$	$0.00_{\pm 0.00}‰$
	DITTO	$\mathbf{0.96_{\pm 0.00}}$	$\mathbf{1.00_{\pm 0.10}}\%$	$\mathbf{0.09_{\pm 0.01}}‰$
Nucleus	MLE [26]	$0.94_{\pm 0.00}$	$1.40_{\pm 0.08}\%$	$\mathbf{0.08_{\pm 0.01}}‰$
	UL-token [32]	$0.94_{\pm 0.00}$	$0.47_{\pm 0.08}\%$	$0.00_{\pm 0.00}‰$
	UL-token+seq [32]	$0.94_{\pm 0.01}$	$0.08_{\pm 0.05}\%$	$0.02_{\pm 0.02}‰$
	SG [17]	$0.93_{\pm 0.01}$	$0.40_{\pm 0.19}\%$	$0.06_{\pm 0.01}‰$
	DITTO	$\mathbf{0.96_{\pm 0.00}}$	$\mathbf{0.98_{\pm 0.09}}\%$	$\mathbf{0.08_{\pm 0.01}}‰$
	Human	-	1.10%	0.10‰

DITTO is compatible with different decoding strategies.

Experiments of Open-ended Generations (Self-reinforcement Effect)



- Other methods: cannot solve self-reinforcement effect
- DITTO: overcome the self-reinforcement effect

Experiments of Abstractive Summarization

Table 4: Abstractive summarization results on CNN/DailyMail.

Model	ROUGE-1	ROUGE-2	ROUGE-L
Pointer-generator + Coverage [29]	39.53	17.28	36.38
Mask Attention Network [7]	40.98	18.29	37.88
BertSum [18]	42.13	19.60	39.18
UniLM [5]	43.08	20.43	40.34
UniLM V2 [2]	43.16	20.42	40.14
ERNIE-GEN-large [33]	44.02	21.17	41.26
PEGASUS [34]	44.17	21.47	41.11
ProphetNet [25]	44.20	21.17	41.30
PALM [3]	44.30	21.12	41.14
BART-large w.t. MLE [15]	44.11±0.03	21.21±0.01	40.83±0.02
BART-large w.t. UL-token [32]	44.17±0.04	21.20±0.02	40.83±0.03
BART-large w.t. UL-token+seq [32]	44.13±0.07	21.15±0.11	40.71±0.09
BART-large w.t. SG [17]	44.18±0.06	21.17±0.07	40.89±0.05
BART-large w.t. DITTO	**44.41±0.03**	**21.45±0.01**	**41.16±0.02**

DITTO outperforms other methods with a large margin on summarization tasks.

Comments from NeurIPS Reviewers

"The paper **tackles a core challenge in NLG**. The 'loop' of the paper is **complete and convincing**."

- NeurIPS reviewer 95eQ

"I believe that this general method provides a **significant contribution for future work** beyond this specific use case: using an external set of negative samples which are easy to form and optimize."

- NeurIPS reviewer HDLP

"Though the community is aware of such problems, this is the **first** time I see such an analysis **systematically** showing the empirical results."

- NeurIPS reviewer tE1F

- Introduction
- Related Work
- Analyzing Repetition Problems
- DITTO - a Method to Mitigate Repetitions
- Experiments
- **Future Work**

Future Work

- **Why language models have “self-reinforcement effect” ?**
  - Model embedding
  - Model architecture
  - Intrinsic characteristics of language
- **High-quality negative datas**
- **Semantic repetitions**

Thanks!