

# CSCE 638 Natural Language Processing Foundation and Techniques

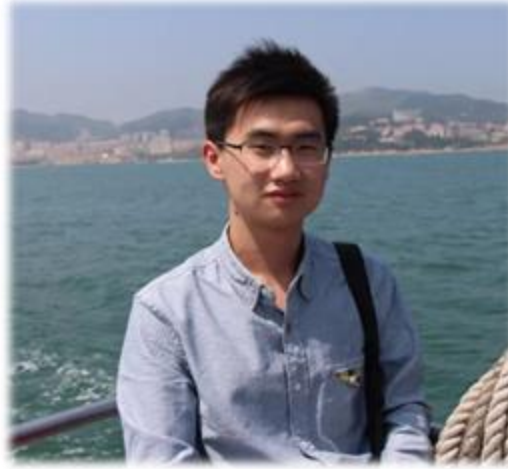
## Lecture 18: AI-Generated Text Detection

Kuan-Hao Huang

Spring 2025



# Invited Talk

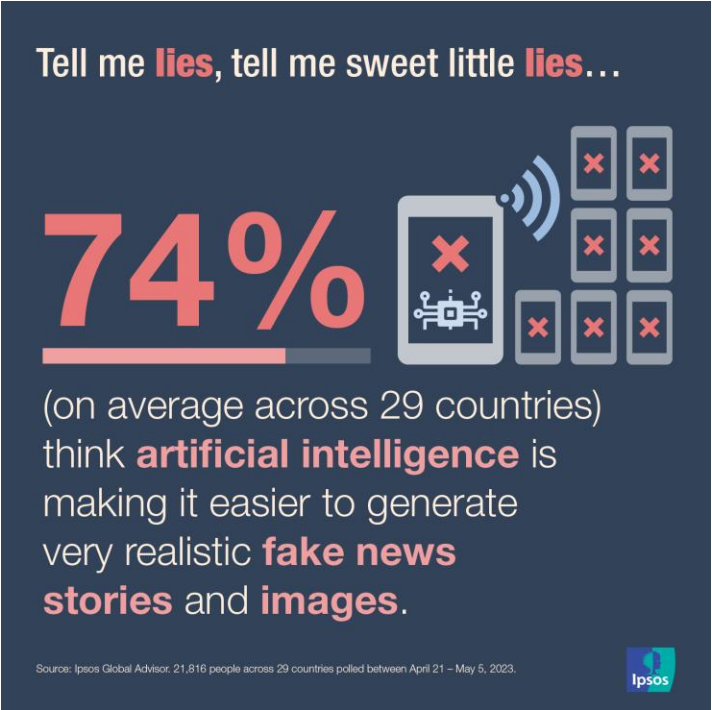


- **Speaker:** [Minhao Cheng](#), Assistant Professor at Pennsylvania State University
- **Title:** Beyond Generation: Enabling Detection and Traceability in Large Language Models through Watermarking
- **Date:** 3/31
- **Online @ Zoom:**
  - <https://tamu.zoom.us/my/khhuang?pwd=oAdWOKVOCGPApqDbJnVtktdW2AE6nb.1>

# Invited Talk

**Abstract:** The remarkable success of generative models, particularly large language models (LLMs), in producing natural and high-quality content across various domains is undeniable. Yet, their widespread use brings forth critical challenges concerning copyright, privacy, and security. To address these risks, the ability to reliably detect and, critically, trace the flow and potential misuse of machine-generated text is paramount for ensuring responsible LLM deployment. This talk will introduce various innovative techniques for embedding covert signals into generated content during its creation. These embedded signals will be algorithmically detectable and, significantly, will enable the tracing of the generated content even from brief token sequences, remaining imperceptible to human observers. Moreover, we will explore the specific hurdles in watermarking structured machine-generated data like code and present efficient strategies for integrating domain-specific knowledge into these watermarking frameworks to facilitate effective tracing.

# AI-Generated Text



Dupli Checker

Paraphrasing Tool Plagiarism Checker Reverse Image Search EN Login Free Tools Pricing

### AI Content Detector

Does your content sound to be written by an AI bot? Get to know the truth and check whether a piece of text is AI-generated with DupliChecker's online AI Detector for free!

Once upon a time in a quaint village nestled at the edge of an enchanted forest, there lived a curious and adventurous child named Amelia. With bright blue eyes full of wonder and a mop of unruly curls, she was always eager to explore the mysteries that lay beyond the village's boundaries.

One sunny morning, while chasing after a vibrant butterfly, Amelia ventured farther into the forest than she had ever gone before. Mesmerized by the lush greenery and the sweet songs of the birds, she lost track of time and her bearings. As the sun began to set, panic started to creep into her heart. She realized she was lost.

Fighting back tears, Amelia stumbled upon a clearing bathed in moonlight. Just as fear threatened to overwhelm her, a soft glow emerged from behind a tree trunk. With trembling steps, she approached the source of the light, her heart pounding in her chest.

Out of the shadows emerged a tiny figure, no taller than a daisy, with delicate wings shimmering like a kaleidoscope of colors. It was a fairy, her luminous presence casting a warm and comforting aura around the bewildered child.

#### Human Content Score

100%

Likely to be Human Generated

Human Written Content 100%

AI Written Content 0%

Pass AI Detection

[–] **Official Review of Paper3132 by Reviewer J57G**

ACL ARR 2024 February Paper3132 Reviewer J57G

28 Mar 2024, 05:01 ACL ARR 2024 February Paper3132 Official Review Readers: Program Chairs, Paper3132 Senior Area Chairs, Paper3132 Area Chairs, Paper3132 Reviewers Submitted, Paper3132 Authors [Show Revisions](#)

**Recommended Process Of Reviewing:** I have read the instructions above

**Paper Summary:**

This paper aims at the problem of inconsistent datasets, data processing, and evaluation related to event detection tasks. Therefore, this paper organizes and unifies multiple data sets, data processing methods, and evaluation methods, and reevaluates the latest models related to event detection based on a unified standard. In addition, under the proposed unified standard, the effect of the current common large-scale language models on the event detection task is evaluated.

**Summary Of Strengths:**

1. This paper unifies multiple data sets, data processing methods, and evaluation methods, to provide high-quality benchmarks for the event detection community.
2. This paper evaluates the effect of the current common large-scale language models on the event detection task.

**Summary Of Weaknesses:**

1. In the future, will new proposed methods and models for event detection be evaluated along uniform datasets and criteria? It's a little unlikely.
2. Do you really have the same data set and processing? What about subsequent new datasets?

---

# Defending Against Neural Fake News

---

**Rowan Zellers<sup>♦</sup>, Ari Holtzman<sup>♦</sup>, Hannah Rashkin<sup>♦</sup>, Yonatan Bisk<sup>♦</sup>**

**Ali Farhadi<sup>♦♥</sup>, Franziska Roesner<sup>♦</sup>, Yejin Choi<sup>♦♥</sup>**

<sup>♦</sup>Paul G. Allen School of Computer Science & Engineering, University of Washington

<sup>♥</sup>Allen Institute for Artificial Intelligence

<https://rowanzellers.com/grover>

# Is It Human-Written or Machine-Generated?

## Why Bitcoin is a great investment

June 6, 2019 - Paul Krugman

As most of my readers know, I'm an optimist.

This belief applies across my life, and to various investments as well. So I am intrigued by the success of cryptocurrencies, such as Bitcoin and Ethereum. The competition they are putting up against the gold standard looks insane, as Bitcoin goes off to the races.

There's no way to fully understand what's going on in the crypto world — and I am not even sure anyone could if you tried to. Still, I can tell you that Bitcoin's recent surge is really an opportunity to buy long-term real assets.

Cryptocurrencies are new and don't even have a useful underlying technology. They will probably fail, probably sooner than later. If people forget about them quickly, it is likely to be because the underlying technology will finally mature and win out. We don't even know whether that will happen in a generation or maybe a century, but it's still possible it might.

# Is It Human-Written or Machine-Generated?

## Link found between autism and vaccines

May 29, 2019 - Sofia Ojeda

Written by Sofia Ojeda, CNN

A paper published in the journal Genetics and Human Behavior claims to find a "significant link" between autism and a type of booster vaccine given in childhood.

The paper focuses on the mercury-based preservative thimerosal and considers the possibility that the injectable vaccine contributes to autism-related symptoms.

These include poor social skills, language delays and seizures, according to the paper.

Thimerosal, which is commonly used as a preservative in vaccines, was phased out of children's vaccines in 2010.

In 2010, the Food and Drug Administration declared that thimerosal was safe.

# Is It Human-Written or Machine-Generated?

## Founder Rowan Zellers raises 17M in Series A round for new AI startup offering “self-driving ice cream trucks”

May 29, 2019 - Kenneth Turan

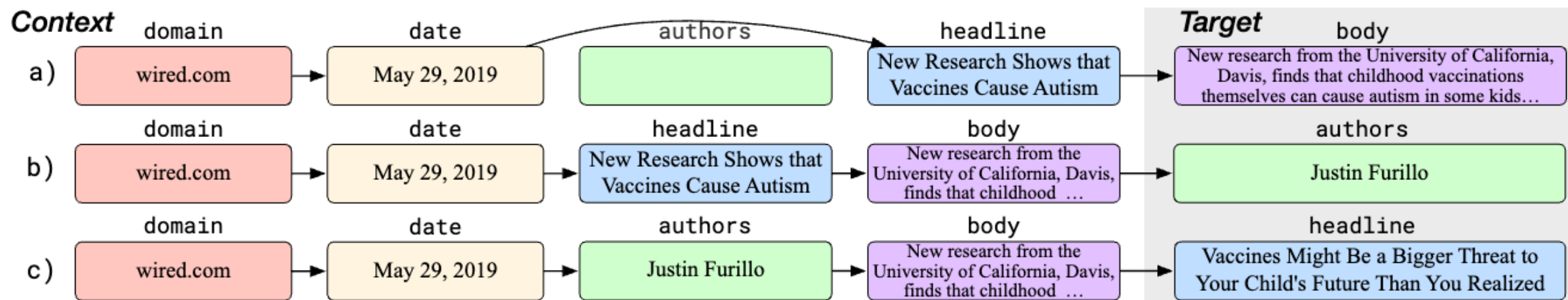
What the heck does ice cream have to do with artificial intelligence? Quite a lot, especially for a startup called Self-Realizing Ice Cream. Founder and CEO Rowan Zellers told me that the company’s tagline is “our mission is to bring ice cream to everyone and everywhere,” but he envisions a time not far in the future when trucks come to people to sell their ice cream, not only at a store, but on their own schedule, using AI.

After helping build his previous companies’ technology into smart homes for SkyKit and Aliance, Zellers came up with a new vision for his own ice cream trucks. They’d be like the autonomous vehicles he saw in Google Self Drive, but the level of intelligence would be better. He developed an artificial intelligence platform that would identify the ice cream flavors that people like (science, not taste), and then it’d recommend a new flavor based on their previous likes.

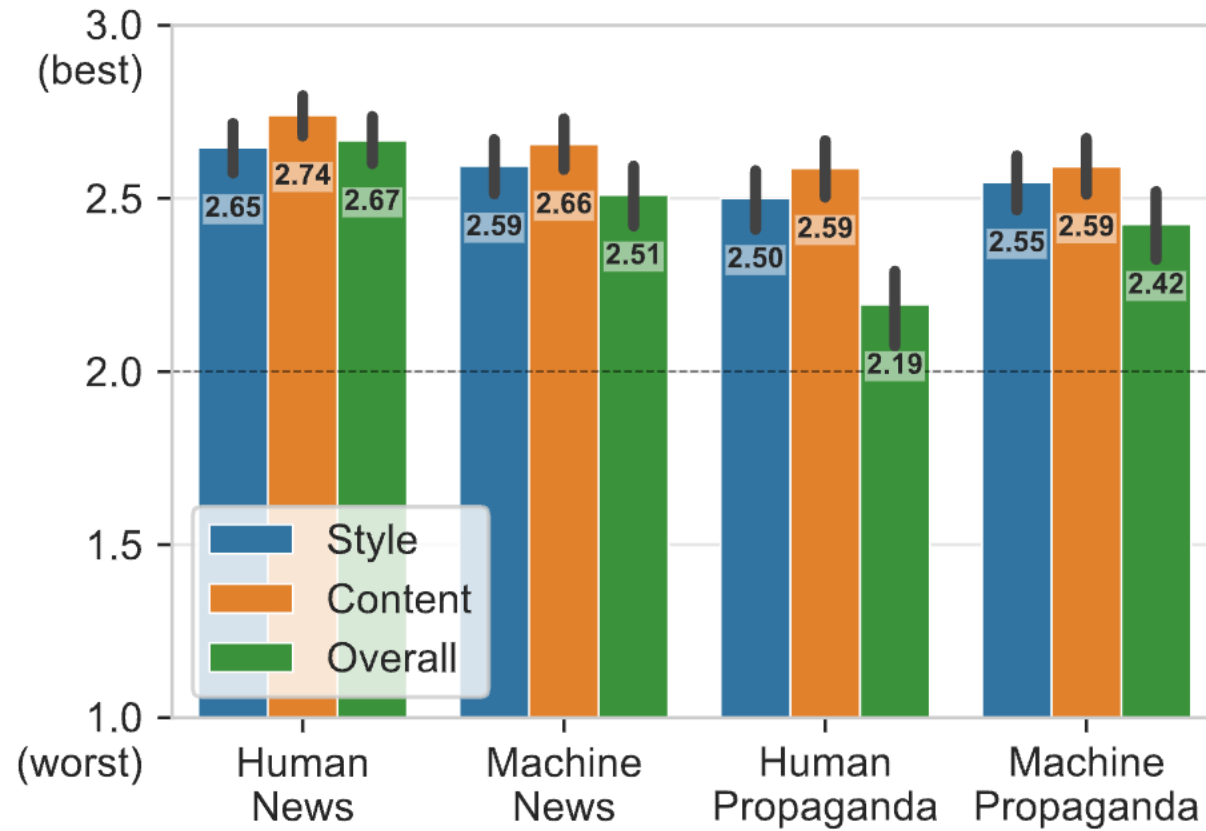


# Grover: Fake News Generator

$$p(\text{domain}, \text{date}, \text{authors}, \text{headline}, \text{body}).$$



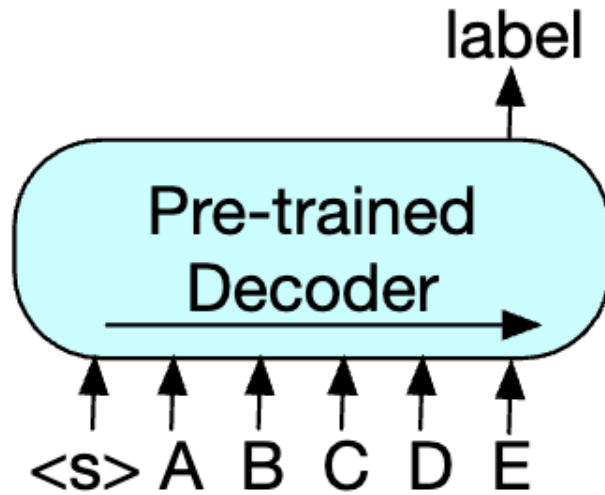
# Comparison to Human-Written Articles



# How About Fake News Detection?

- Train a binary classifier
  - Human-written news articles
  - Machin-generated news articles

# How About Fake News Detection?



		Unpaired Accuracy			Paired Accuracy				
		Generator size			Generator size				
		1.5B	355M	124M		1.5B	355M	124M	
Chance		50.0				50.0			
Discriminator size	1.5B	GROVER-Mega	<b>91.6</b>	<b>98.7</b>	<b>99.8</b>		<b>98.8</b>	<b>100.0</b>	<b>100.0</b>
	355M	GROVER-Large	<b>79.5</b>	<b>91.0</b>	<b>98.7</b>		<b>88.7</b>	<b>98.4</b>	<b>99.9</b>
		BERT-Large	68.0	78.9	93.7		75.3	90.4	99.5
		GPT2	70.1	77.2	88.0		79.1	86.8	95.0
	124M	GROVER-Base	<b>71.3</b>	<b>79.4</b>	<b>90.0</b>		80.8	88.5	<b>97.0</b>
		BERT-Base	67.2	75.0	82.0		<b>84.7</b>	<b>90.9</b>	96.6
		GPT2	67.7	73.2	81.8		72.9	80.6	87.1
	11M	FastText	63.8	65.4	70.0		73.0	73.0	79.0

# Machine-Generated Text Detection

- Grover: **supervised** machine-generated text detection
  - Require human-written and machine-generated examples
- **Zero-shot** machine-generated text detection
  - No access to human-written and machine-generated examples

The detector is model dependent!

# Some Simple Detection Methods

- Perplexity / Log-Likelihood  $\log p(x)$

Likelihood

$$P(\mathcal{X}) = \prod_{i=1}^n P(x_i)$$

Log-Likelihood

$$\log P(\mathcal{X}) = \sum_{i=1}^n \log P(x_i)$$

Per-Word Log-Likelihood

$$WLL(\mathcal{X}) = \frac{1}{W} \sum_{i=1}^n \log P(x_i)$$

Perplexity  $e^{-WLL(\mathcal{X})}$

$$WLL(\mathcal{X}) = \frac{1}{W} \sum_{i=1}^n \log P(x_i)$$

Language Models

$P(w_1)$	$P(w_2 w_1)$	$P(w_3 w_1w_2)$	$P(w_4 w_1w_2w_3)$
----------	--------------	-----------------	--------------------

This

is

a

cat

$-\frac{1}{N}$

# Some Simple Detection Methods

- Rank

Language Models

$R(w_1)$	$R(w_2)$	$R(w_3)$	$R(w_4)$
$P(w_1)$	$P(w_2 w_1)$	$P(w_3 w_1w_2)$	$P(w_4 w_1w_2w_3)$
This	is	a	cat

$$R(w) = \frac{1}{N} \sum R(w_i)$$

# Some Simple Detection Methods

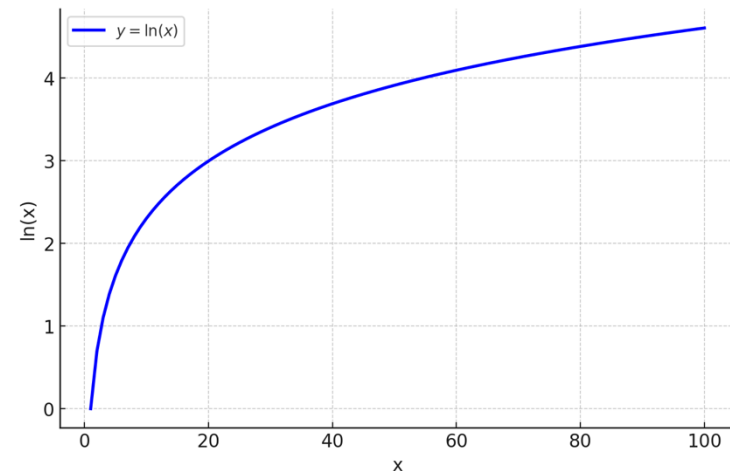
- Log-Rank

Language Models

$R(w_1)$	$R(w_2)$	$R(w_3)$	$R(w_4)$
$P(w_1)$	$P(w_2 w_1)$	$P(w_3 w_1w_2)$	$P(w_4 w_1w_2w_3)$

This is a cat

$$R(w) = \frac{1}{N} \sum \log R(w_i)$$





---

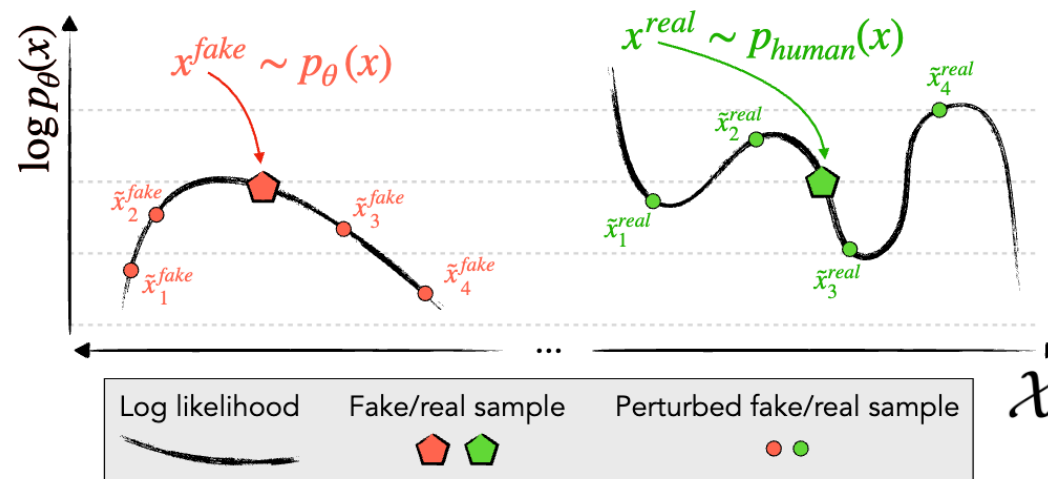
# **DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature**

---

**Eric Mitchell<sup>1</sup> Yoonho Lee<sup>1</sup> Alexander Khazatsky<sup>1</sup> Christopher D. Manning<sup>1</sup> Chelsea Finn<sup>1</sup>**

# Perturbation Discrepancy Gap Hypothesis

- Text generator  $p_\theta$
- Log probability of an example  $x$  is  $\log p_\theta(x)$
- Slightly perturbed example  $\tilde{x}$
- The difference  $\log p_\theta(x) - \log p_\theta(\tilde{x})$ 
  - Should be relatively **large** when example  $x$  is **machine-generated**
  - Should be relatively **small** when example  $x$  is **human-written**



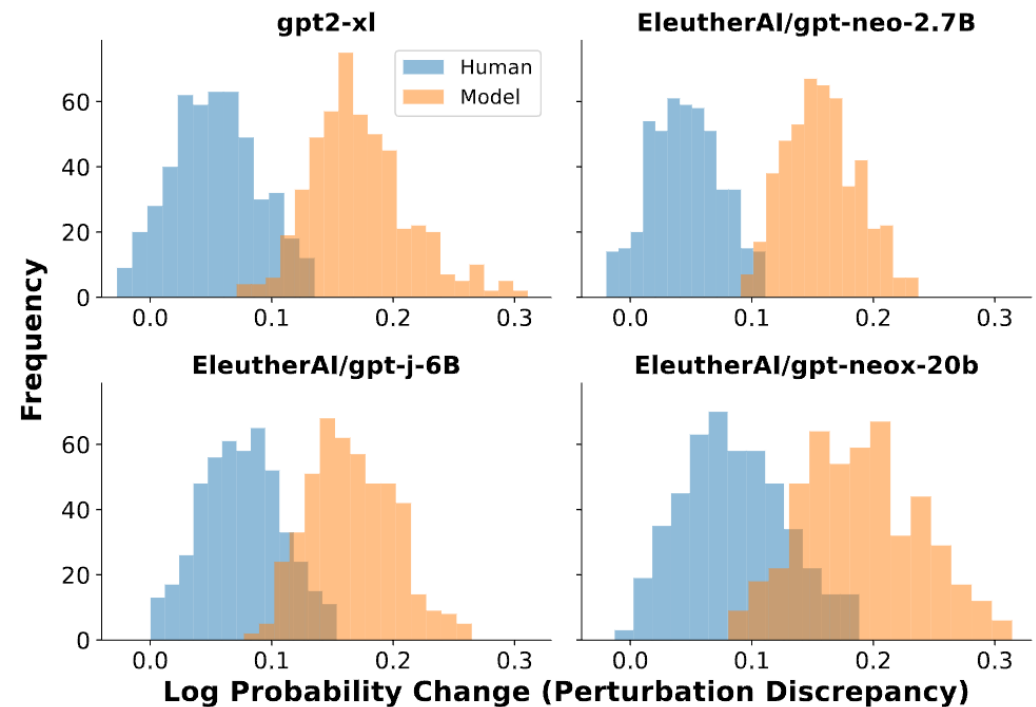
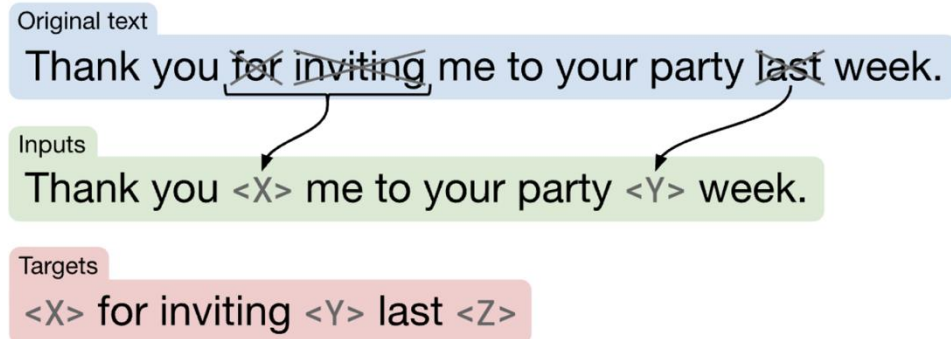
# Perturbation Discrepancy Gap Hypothesis

- Perturbation function  $q(\cdot | x)$
- Perturbation discrepancy

$$d(x, p_\theta, q) = \log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot | x)} \log p_\theta(x)$$

# Perturbation Discrepancy Gap Hypothesis

- Perturbation function  $q(\cdot | x)$ 
  - Samples from a mask-filling mode (e.g., T5)
- Perturbation discrepancy



$$d(x, p_\theta, q) = \log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot | x)} \log p_\theta(\tilde{x})$$

# Algorithm

---

**Algorithm 1** DetectGPT model-generated text detection

---

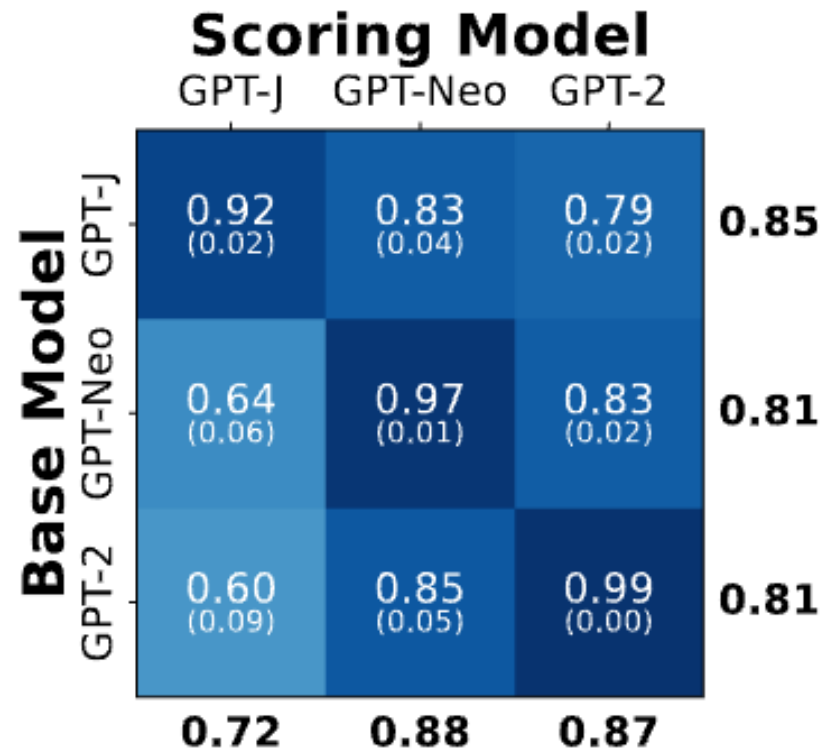
- 1: **Input:** passage  $x$ , source model  $p_\theta$ , perturbation function  $q$ ,  
number of perturbations  $k$ , decision threshold  $\epsilon$
  - 2:  $\tilde{x}_i \sim q(\cdot \mid x)$ ,  $i \in [1..k]$  // mask spans, sample replacements
  - 3:  $\tilde{\mu} \leftarrow \frac{1}{k} \sum_i \log p_\theta(\tilde{x}_i)$  // approximate expectation in Eq. 1
  - 4:  $\hat{\mathbf{d}}_x \leftarrow \log p_\theta(x) - \tilde{\mu}$  // estimate  $\mathbf{d}(x, p_\theta, q)$
  - 5:  $\tilde{\sigma}_x^2 \leftarrow \frac{1}{k-1} \sum_i (\log p_\theta(\tilde{x}_i) - \tilde{\mu})^2$  // variance for normalization
  - 6: **if**  $\frac{\hat{\mathbf{d}}_x}{\sqrt{\tilde{\sigma}_x}} > \epsilon$  **then**
  - 7:     **return** `true` // probably model sample
  - 8: **else**
  - 9:     **return** `false` // probably not model sample
-

# DetectGPT Results

Method	XSum						SQuAD					
	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	Avg.	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	Avg.
$\log p(x)$	0.86	0.86	0.86	0.82	0.77	0.83	0.91	0.88	0.84	0.78	0.71	0.82
Rank	0.79	0.76	0.77	0.75	0.73	0.76	0.83	0.82	0.80	0.79	0.74	0.80
LogRank	0.89*	0.88*	0.90*	0.86*	0.81*	0.87*	0.94*	0.92*	0.90*	0.83*	0.76*	0.87*
DetectGPT	<b>0.99</b>	<b>0.97</b>	<b>0.99</b>	<b>0.97</b>	<b>0.95</b>	<b>0.97</b>	<b>0.99</b>	<b>0.97</b>	<b>0.97</b>	<b>0.90</b>	<b>0.79</b>	<b>0.92</b>

# When Text Generator Is Not Accessible

- Use another generator to compute probability instead



# FAST-DETECTGPT: EFFICIENT ZERO-SHOT DETECTION OF MACHINE-GENERATED TEXT VIA CONDITIONAL PROBABILITY CURVATURE

**Guangsheng Bao**

Zhejiang University  
School of Engineering, Westlake University  
baoguangsheng@westlake.edu.cn

**Yanbin Zhao**

School of Mathematics, Physics and Statistics,  
Shanghai Polytechnic University  
zhaoyb553@nenu.edu.cn

**Zhiyang Teng**

Nanyang Technological University  
zhiyang.teng@ntu.edu.sg

**Linyi Yang, Yue Zhang\***

School of Engineering, Westlake University  
Institute of Advanced Technology, Westlake Institute for Advanced Study  
{yanglinyi, zhangyue}@westlake.edu.cn



# Issue of DetectGPT

$$d(x, p_\theta, q) = \log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot | x)} \log p_\theta(\tilde{x})$$

---

**Algorithm 1** DetectGPT model-generated text detection

---

Time-consuming

- 1: **Input:** passage  $x$ , source model  $p_\theta$ , perturbation function  $q$ , number of perturbations  $k$ , decision threshold  $\epsilon$
  - 2:  $\tilde{x}_i \sim q(\cdot | x), i \in [1..k]$  // mask spans, sample replacements
  - 3:  $\tilde{\mu} \leftarrow \frac{1}{k} \sum_i \log p_\theta(\tilde{x}_i)$  // approximate expectation in Eq. 1
  - 4:  $\hat{\mathbf{d}}_x \leftarrow \log p_\theta(x) - \tilde{\mu}$  // estimate  $\mathbf{d}(x, p_\theta, q)$
  - 5:  $\tilde{\sigma}_x^2 \leftarrow \frac{1}{k-1} \sum_i (\log p_\theta(\tilde{x}_i) - \tilde{\mu})^2$  // variance for normalization
  - 6: **if**  $\frac{\hat{\mathbf{d}}_x}{\sqrt{\tilde{\sigma}_x}} > \epsilon$  **then**
  - 7:     **return** true // probably model sample
  - 8: **else**
  - 9:     **return** false // probably not model sample
-

# Issue of DetectGPT

- This restaurant is extremely good, and I will give it a 5-star.
- This restaurant is **impressively** good, and I will rate it a 5-star.
- This restaurant is extremely **great**, and I will give it a 5-**score**.
- **The** restaurant is extremely good, and I **would** give it a 5-star.
- This restaurant is extremely good, **and** I will give it a 5-star.

We need to compute the probability for every single perturbed examples

# Issue of DetectGPT

- This restaurant is extremely good, and I will give it a 5-star.
- This restaurant is **impressively** good, and I will rate it a 5-star.
- This restaurant is extremely **great**, and I will give it a 5-**score**.
- **The** restaurant is extremely good, and I **would** give it a 5-star.
- This restaurant is extremely good, **and** I will give it a 5-star.

We need to compute the probability for every single perturbed examples

# Conditional Probability Function

$$p_{\theta}(\tilde{x}|x) = \prod_j p_{\theta}(\tilde{x}_j|x_{<j})$$

- This restaurant is [?]
- This restaurant is extremely good, and I will give it a 5-star.
- This restaurant is **impressively** good, and I will rate it a 5-star.

# Conditional Probability Function

$$p_{\theta}(\tilde{x}|x) = \prod_j p_{\theta}(\tilde{x}_j|x_{<j})$$

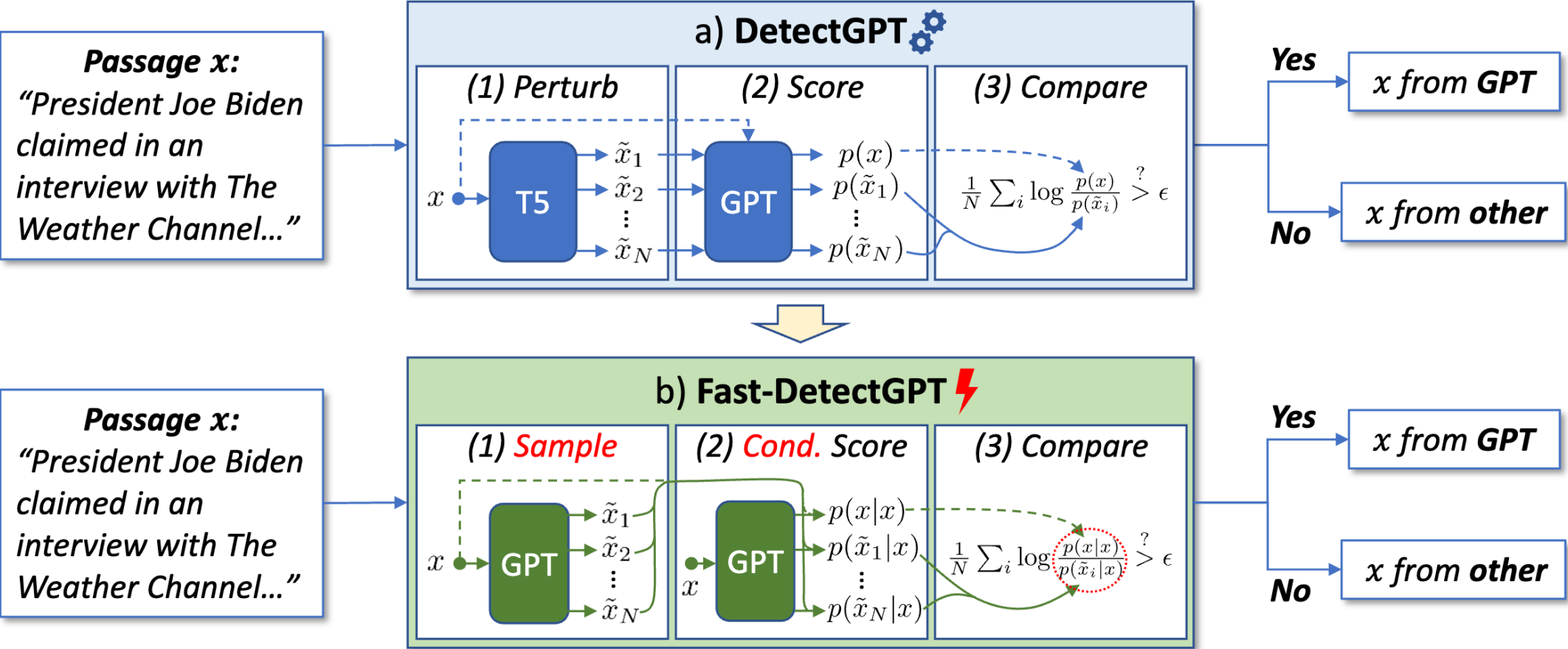
- This restaurant is extremely [?]
- This restaurant is extremely good, and I will give it a 5-star.
- This restaurant is extremely great, and I will give it a 5-score.

# Conditional Probability Function

$$p_{\theta}(\tilde{x}|x) = \prod_j p_{\theta}(\tilde{x}_j|x_{<j})$$

- This restaurant is extremely good, and I will give it a 5-[?]
- This restaurant is extremely good, and I will give it a 5-star.
- This restaurant is extremely good, and I will give it a 5-score.

# Fast-DetectGPT vs. DetectGPT



# Conditional Probability Curvature

$$\mathbf{d}(x, p_\theta, q_\varphi) = \frac{\log p_\theta(x|x) - \tilde{\mu}}{\tilde{\sigma}}$$

$$\tilde{\mu} = \mathbb{E}_{\tilde{x} \sim q_\varphi(\tilde{x}|x)} [\log p_\theta(\tilde{x}|x)] \quad \text{and} \quad \tilde{\sigma}^2 = \mathbb{E}_{\tilde{x} \sim q_\varphi(\tilde{x}|x)} [(\log p_\theta(\tilde{x}|x) - \tilde{\mu})^2]$$

Probability curvature proposed by DetectGPT

$$\mathbf{d}(x, p_\theta, q) = \log p_\theta(x) - \mathbb{E}_{\tilde{x} \sim q(\cdot|x)} \log p_\theta(\tilde{x})$$



# Algorithm

$$\mathbf{d}(x, p_\theta, q_\varphi) = \frac{\log p_\theta(x|x) - \tilde{\mu}}{\tilde{\sigma}}$$

---

**Algorithm 1** Fast-DetectGPT machine-generated text detection.

---

**Input:** passage  $x$ , sampling model  $q_\varphi$ , scoring model  $p_\theta$ , and decision threshold  $\epsilon$

**Output:** True – probably machine-generated, False – probably human-written.

```
1: function FASTDETECTGPT( $x, q_\varphi, p_\theta$ )  
2:    $\tilde{x}_i \sim q_\varphi(\tilde{x}|x), i \in [1..N]$                                 ▷ Conditional sampling  
3:    $\tilde{\mu} \leftarrow \frac{1}{N} \sum_i \log p_\theta(\tilde{x}_i|x)$                         ▷ Estimate the mean  
4:    $\tilde{\sigma}^2 \leftarrow \frac{1}{N-1} \sum_i (\log p_\theta(\tilde{x}_i|x) - \tilde{\mu})^2$       ▷ Estimate the variance  
5:    $\hat{\mathbf{d}}_x \leftarrow (\log p_\theta(x) - \tilde{\mu})/\tilde{\sigma}$                         ▷ Estimate conditional probability curvature  
6:   return  $\hat{\mathbf{d}}_x > \epsilon$ 
```

---

- This restaurant is extremely good, and I will give it a 5-star.

- This [?]

- This restaurant [?]

- This restaurant is [?]

- ...

White-box: sampled from text generator

Black-box: sampled from an alternative generator

# Results for White-Box Setting

Method	GPT-2	OPT-2.7	Neo-2.7	GPT-J	NeoX	Avg.
The White-Box Setting						
Likelihood	0.9125	0.8963	0.8900	0.8480	0.7946	0.8683
Entropy	0.5174	0.4830	0.4898	0.5005	0.5333	0.5048
LogRank	0.9385	0.9223	0.9226	0.8818	0.8313	0.8993
LRR	0.9601	0.9401	0.9522	0.9179	0.8793	0.9299
DNA-GPT $\diamond$	0.9024	0.8797	0.869	0.8227	0.7826	0.8513
NPR $\diamond$	0.9948 $\dagger$	0.9832 $\dagger$	0.9883	0.9500	0.9065	0.9645
DetectGPT (T5-3B/*) $\diamond$	0.9917	0.9758	0.9797	0.9353	0.8943	0.9554
Fast-DetectGPT (*/*)	<b>0.9967</b>	<b>0.9908</b>	0.9940 $\dagger$	<b>0.9866</b>	<b>0.9754</b>	<b>0.9887</b>

# Results for Black-Box Setting

Method	ChatGPT				GPT-4			
	XSum	Writing	PubMed	Avg.	XSum	Writing	PubMed	Avg.
RoBERTa-base	0.9150	0.7084	0.6188	0.7474	0.6778	0.5068	0.5309	0.5718
RoBERTa-large	0.8507	0.5480	0.6731	0.6906	0.6879	0.3821	0.6067	0.5589
GPTZero	<b>0.9952</b>	0.9292	0.8799	0.9348	<b>0.9815</b>	0.8262	0.8482	0.8853
Likelihood (Neo-2.7)	0.9578	0.9740	0.8775	0.9364	0.7980	0.8553	0.8104	0.8212
Entropy (Neo-2.7)	0.3305	0.1902	0.2767	0.2658	0.4360	0.3702	0.3295	0.3786
LogRank(Neo-2.7)	0.9582	0.9656	0.8687	0.9308	0.7975	0.8286	0.8003	0.8088
LRR (Neo-2.7)	0.9162	0.8958	0.7433	0.8518	0.7447	0.7028	0.6814	0.7096
DNA-GPT (Neo-2.7)	0.9124	0.9425	0.7959	0.8836	0.7347	0.8032	0.7565	0.7648
NPR (T5-11B/Neo-2.7)	0.7899	0.8924	0.6784	0.7869	0.5280	0.6122	0.6328	0.5910
DetectGPT (T5-11B/Neo-2.7)	0.8416	0.8811	0.7444	0.8223	0.5660	0.6217	0.6805	0.6228
Fast-Detect (GPT-J/Neo-2.7)	0.9907	<b>0.9916</b>	<b>0.9021</b>	<b>0.9615</b>	0.9067	<b>0.9612</b>	<b>0.8503</b>	<b>0.9061</b>

# Speed Improvement

Method	5-Model Generations ↑	ChatGPT/GPT-4 Generations ↑	Speedup ↑
DetectGPT	0.9554	0.7225	1x
Fast-DetectGPT	<b>0.9887</b> (relative↑ 74.7%)	<b>0.9338</b> (relative↑ 76.1%)	<b>340x</b>

## **Red Teaming Language Model Detectors with Language Models**

**Zhouxing Shi\*, Yihan Wang\*, Fan Yin\*, Xiangning Chen, Kai-Wei Chang, Cho-Jui Hsieh**

University of California, Los Angeles

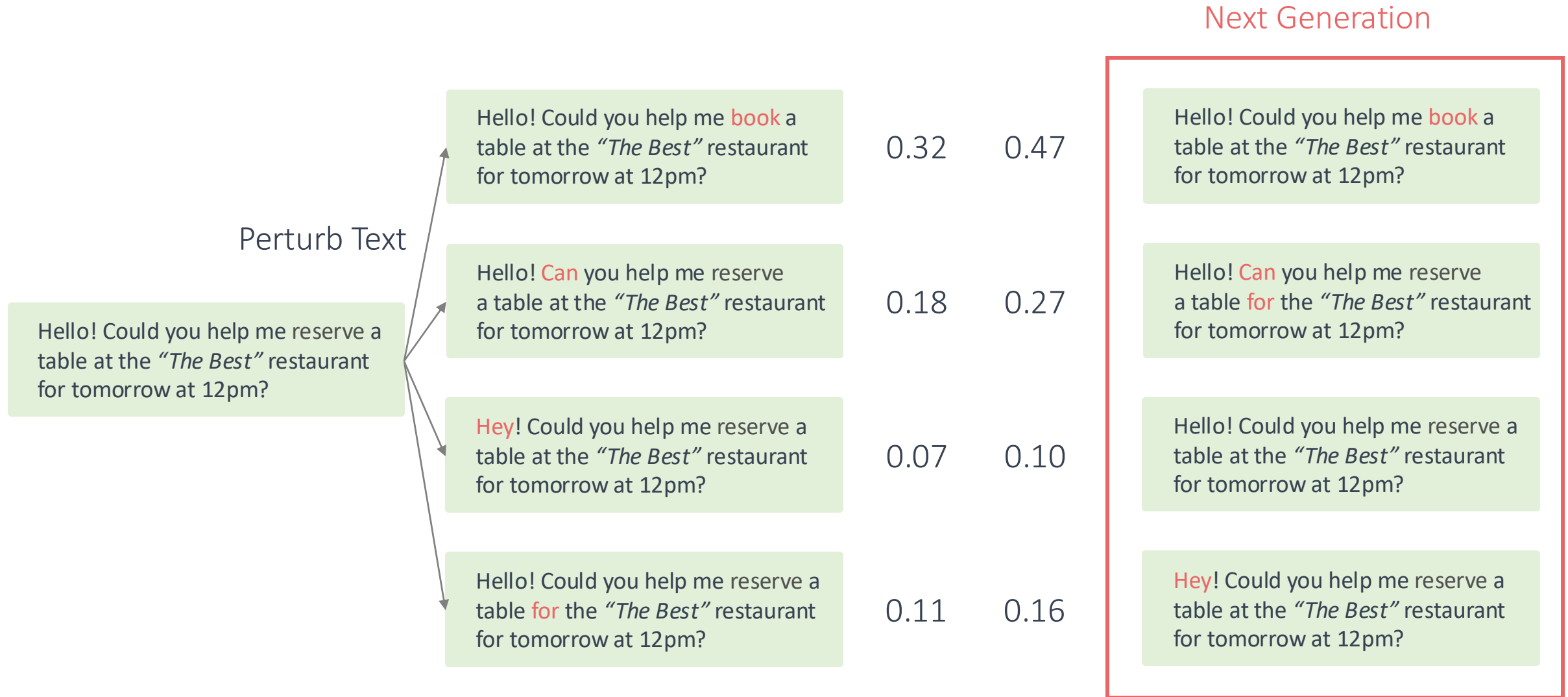
{zshi, yihanwang, fanyin20, xiangning, kwchang, chohsieh}@cs.ucla.edu

\*Alphabetical order

# Detectors Can Be Attacked

- Generate text by machines first
- Perturb machine-generated text
  - **Query-free** word replacement
  - **Query-based** word replacement
  - **Paraphrasing** text

# Recap: Genetic Algorithm for Word Replacement



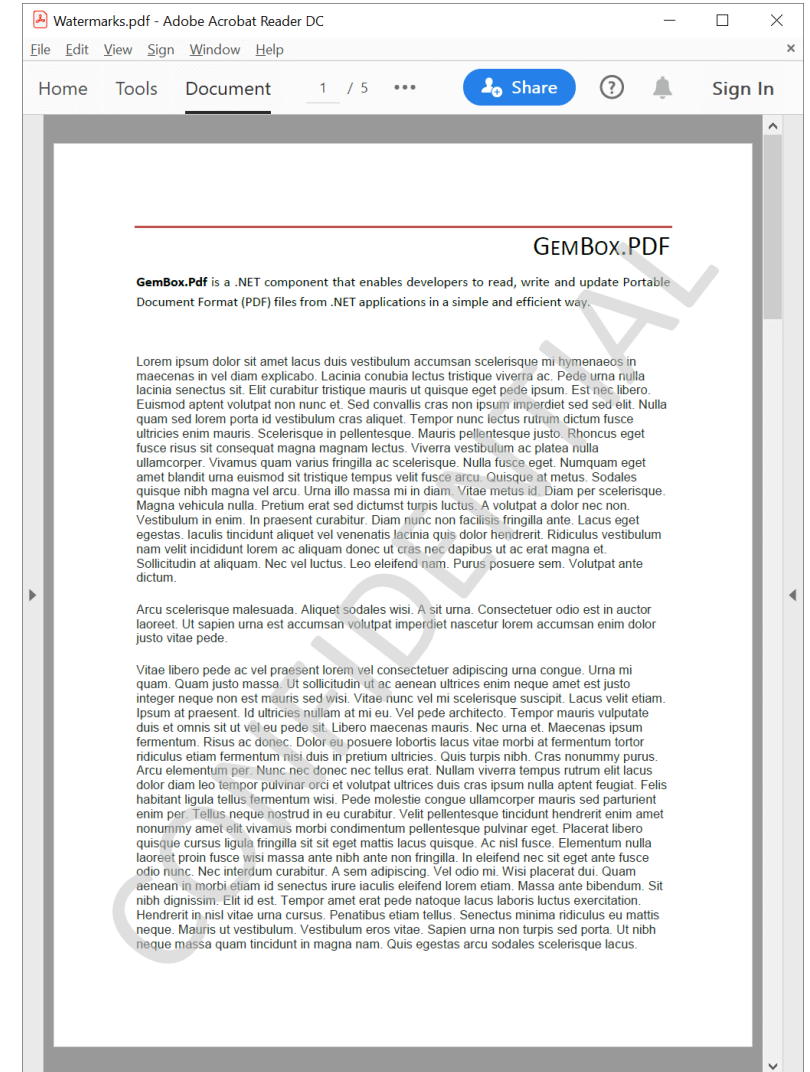
# Results

Generative Model	Dataset	Unattacked	Dipper Paraphrasing	Query-free Substitution	Query-based Substitution
GPT-2-XL	XSum	84.4	35.2	25.9	<b>3.9</b>
	ELI5	70.6	36.7	21.2	<b>3.8</b>
ChatGPT	XSum	56.0	34.6	25.6	<b>4.5</b>
	ELI5	55.0	39.5	12.2	<b>6.5</b>
LLaMA-65B	XSum	59.3	49.0	25.5	<b>9.9</b>
	ELI5	60.5	53.1	31.4	<b>18.6</b>



# Watermarking

- Post-detection can be hard
- Add **watermark** during training/generating
  - Watermark should not affect too much to the generation quality
  - Watermark cannot be too obvious
  - Watermark verification needs to be viable
  - Watermark cannot be removed easily



---

# **A Watermark for Large Language Models**

---

**John Kirchenbauer\* Jonas Geiping\* Yuxin Wen Jonathan Katz Ian Miers Tom Goldstein**  
**University of Maryland**

# Assumptions

- Add watermark when generating texts
- We have the access to the **vocabulary** of the model

# Watermarking Example

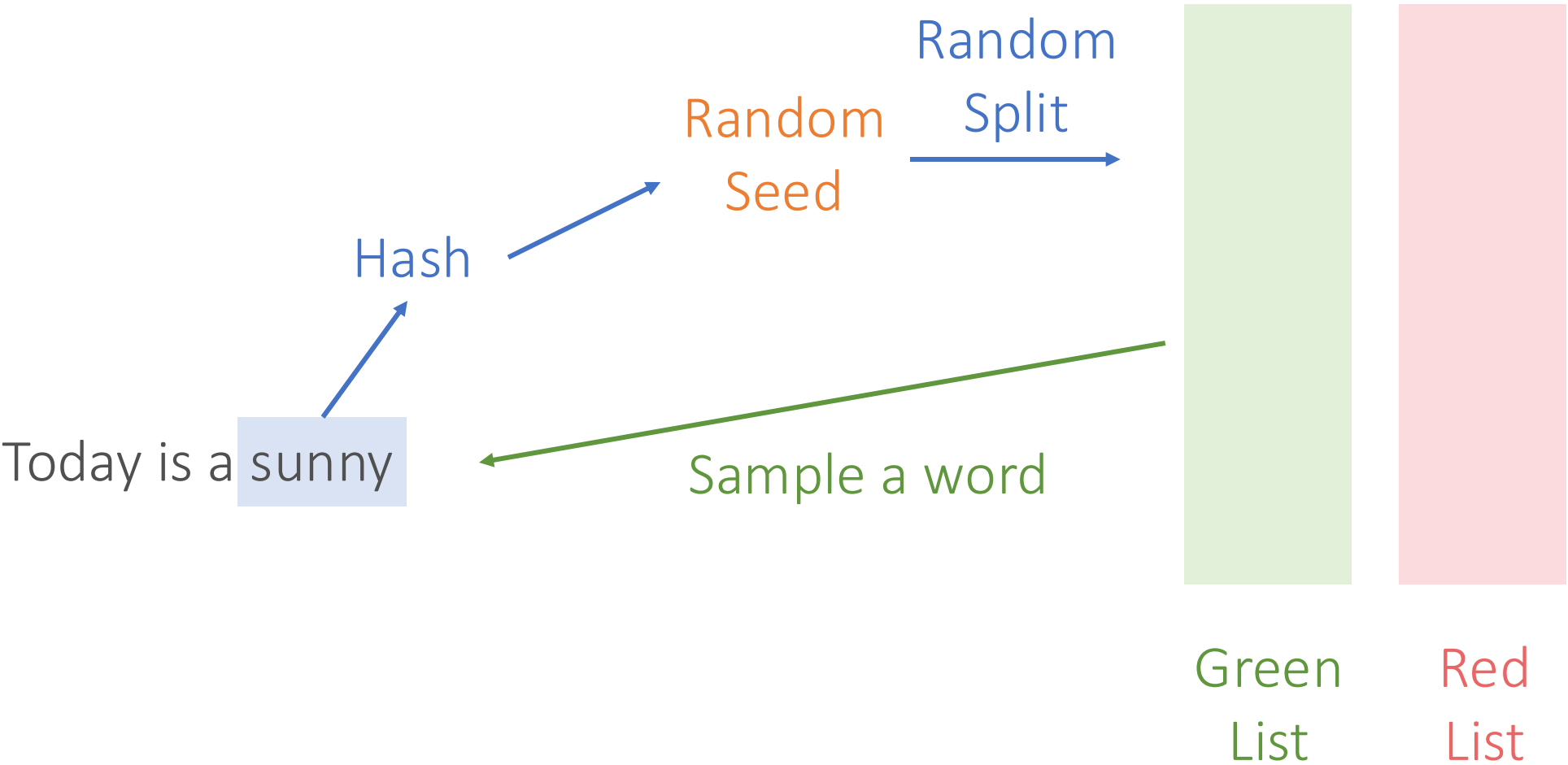
Prompt	Num tokens	Z-score	p-value
...The watermark detection algorithm can be made public, enabling third parties (e.g., social media platforms) to run it themselves, or it can be kept private and run behind an API. We seek a watermark with the following properties:			
<b>No watermark</b> Extremely efficient on average term lengths and word frequencies on synthetic, microamount text (as little as 25 words) Very small and low-resource key/hash (e.g., 140 bits per key is sufficient for 99.99999999% of the Synthetic Internet)	56	.31	.38
<b>With watermark</b> - minimal marginal probability for a detection attempt. - Good speech frequency and energy rate reduction. - messages indiscernible to humans. - easy for humans to verify.	36	7.4	6e-14

How to decide green/red words?

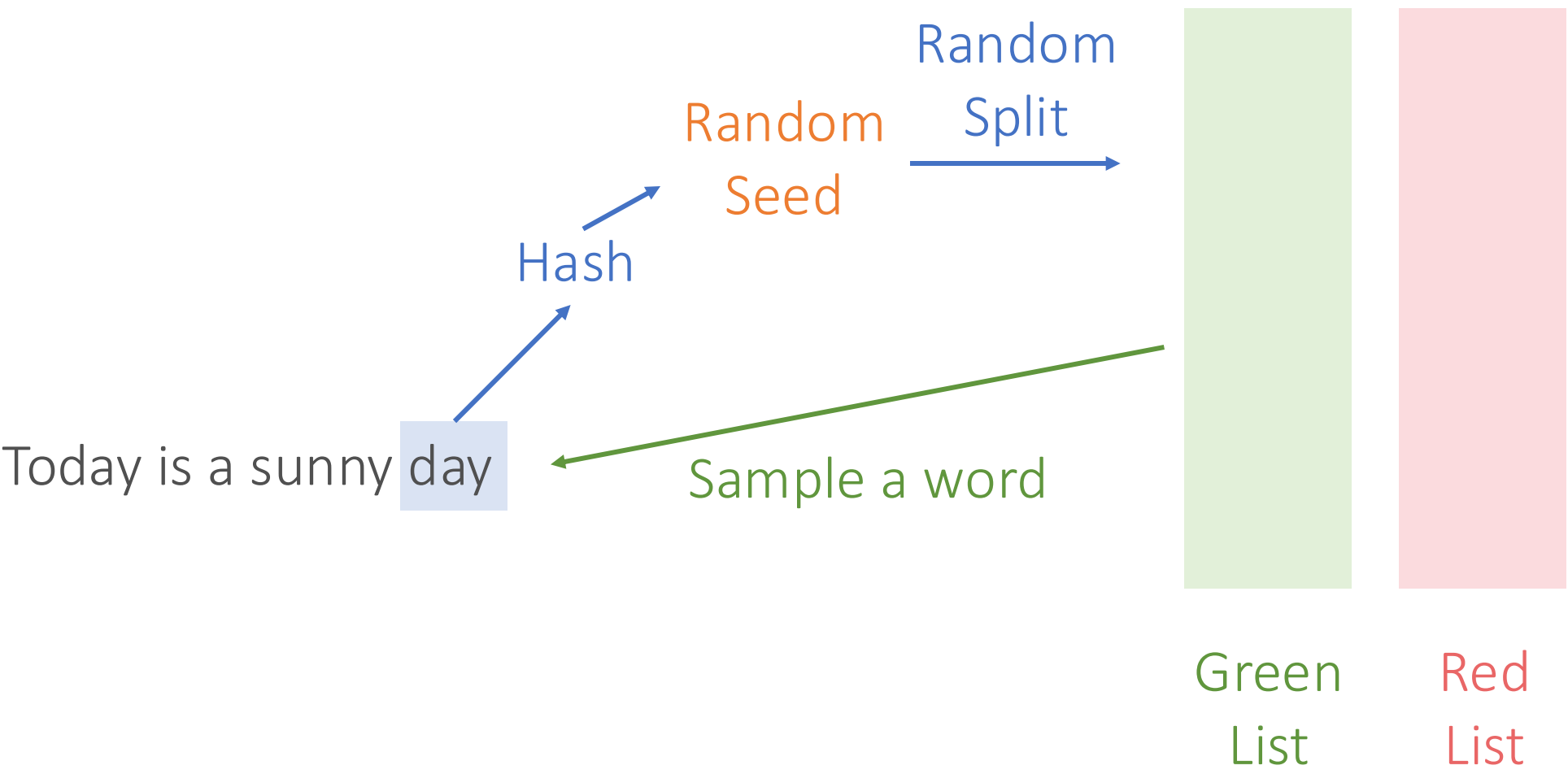
# Pre-Defined Green/Red List?

- Watermark should not affect too much to the generation quality (?)
- Watermark cannot be too obvious (x)
- Watermark verification needs to be viable (v)
- Watermark cannot be removed easily (v)

# Dynamically Define Red List



# Dynamically Define Red List



# Text Generation with Red List

- The chance of a random text has a valid watermark
  - $\left(\frac{1}{2}\right)^T$  for a length  $T$  text
- Watermark detection
  - Statistic way: one proportion z-test

$$z = 2(|s|_G - T/2)/\sqrt{T}.$$

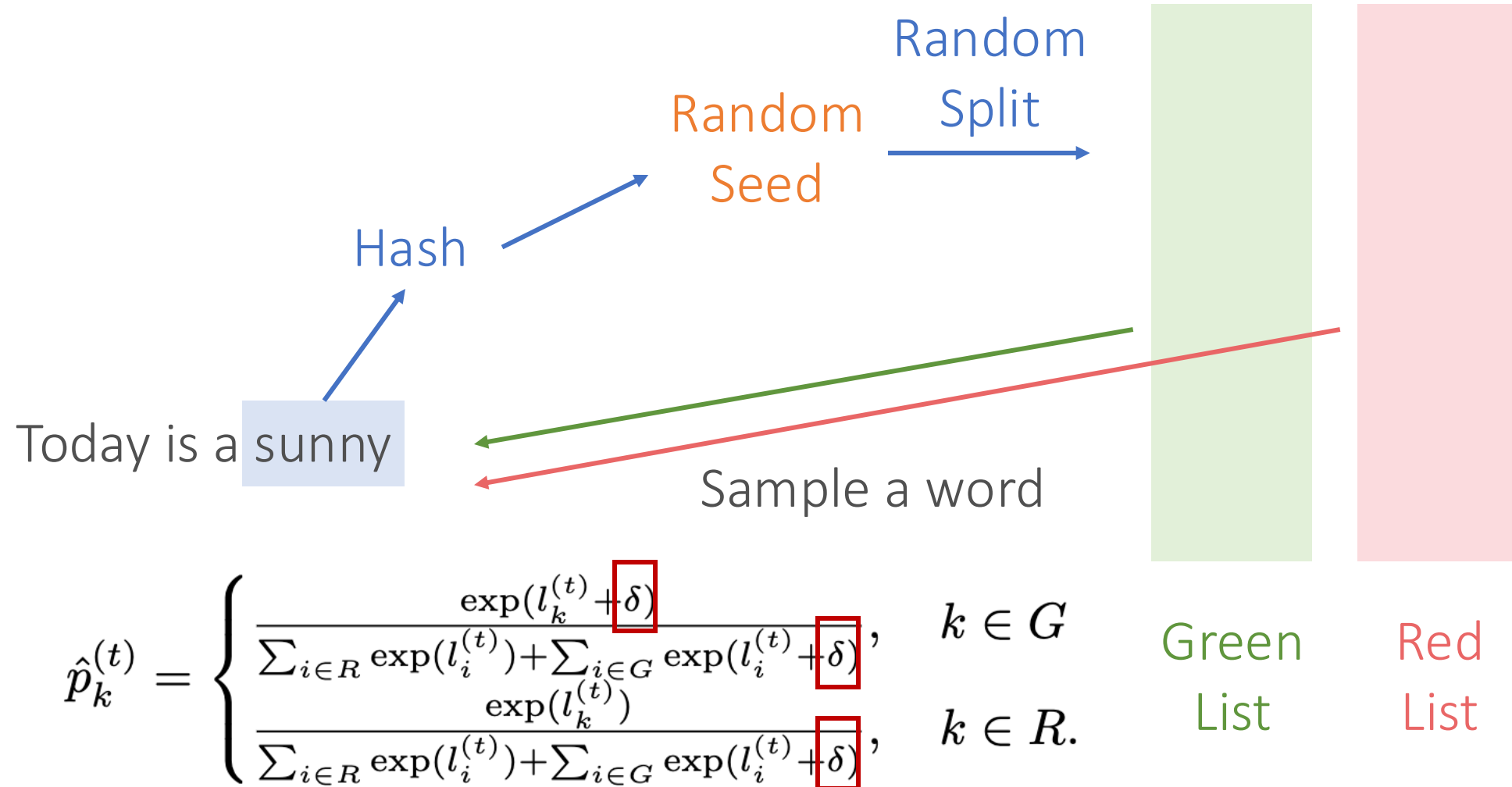
- If  $z > \text{threshold}$   $\rightarrow$  having watermark
- $z > 4$ , the probability of a false positive is  $3 \times 10^{-5}$



# Text Generation with Red List

- Generated texts can be not natural for certain cases
  - College Station
  - Los Angeles

# Dynamically Define Soft Red List



# Text Generation with Soft Red List

---

**Algorithm 2** Text Generation with Soft Red List

---

**Input:** prompt,  $s^{(-N_p)} \dots s^{(-1)}$

green list size,  $\gamma \in (0, 1)$

hardness parameter,  $\delta > 0$

**for**  $t = 0, 1, \dots$  **do**

1. Apply the language model to prior tokens  $s^{(-N_p)} \dots s^{(t-1)}$  to get a logit vector  $l^{(t)}$  over the vocabulary.

2. Compute a hash of token  $s^{(t-1)}$ , and use it to seed a random number generator.

3. Using this random number generator, randomly partition the vocabulary into a “green list”  $G$  of size  $\gamma|V|$ , and a “red list”  $R$  of size  $(1 - \gamma)|V|$ .

4. Add  $\delta$  to each green list logit. Apply the softmax operator to these modified logits to get a probability distribution over the vocabulary.

$$\hat{p}_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)} + \delta)}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)} + \delta)}, & k \in G \\ \frac{\exp(l_k^{(t)})}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)} + \delta)}, & k \in R. \end{cases}$$

5. Sample the next token,  $s^{(t)}$ , using the water-marked distribution  $\hat{p}^{(t)}$ .

**end for**

---

# Text Generation with Soft Red List

**Theorem 4.2.** *Consider watermarked text sequences of  $T$  tokens. Each sequence is produced by sequentially sampling a raw probability vector  $p^{(t)}$  from the language model, sampling a random green list of size  $\gamma N$ , and boosting the green list logits by  $\delta$  using Equation 4 before sampling each token. Define  $\alpha = \exp(\delta)$ , and let  $|s|_G$  denote the number of green list tokens in sequence  $s$ .*

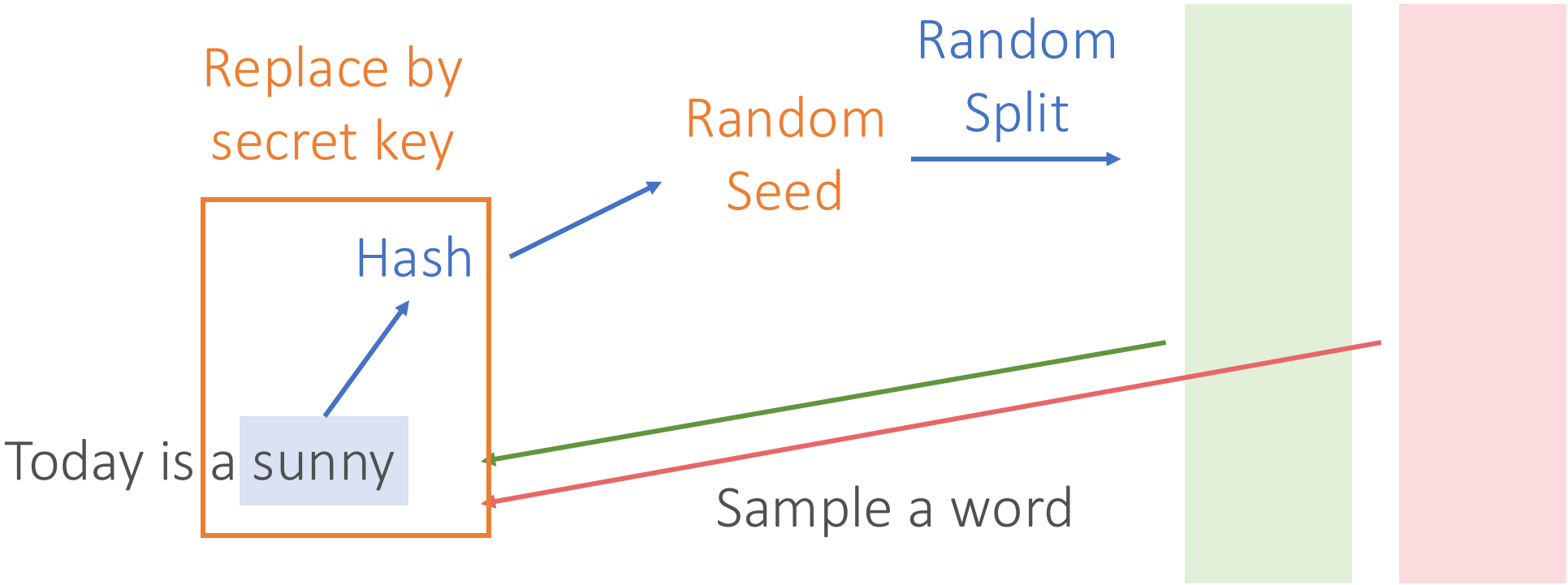
*If a randomly generated watermarked sequence has average spike entropy at least  $S^*$ , i.e.,*

$$\frac{1}{T} \sum_t S \left( p^{(t)}, \frac{(1 - \gamma)(\alpha - 1)}{1 + (\alpha - 1)\gamma} \right) \geq S^*,$$

*then the number of green list tokens in the sequence has expected value at least*

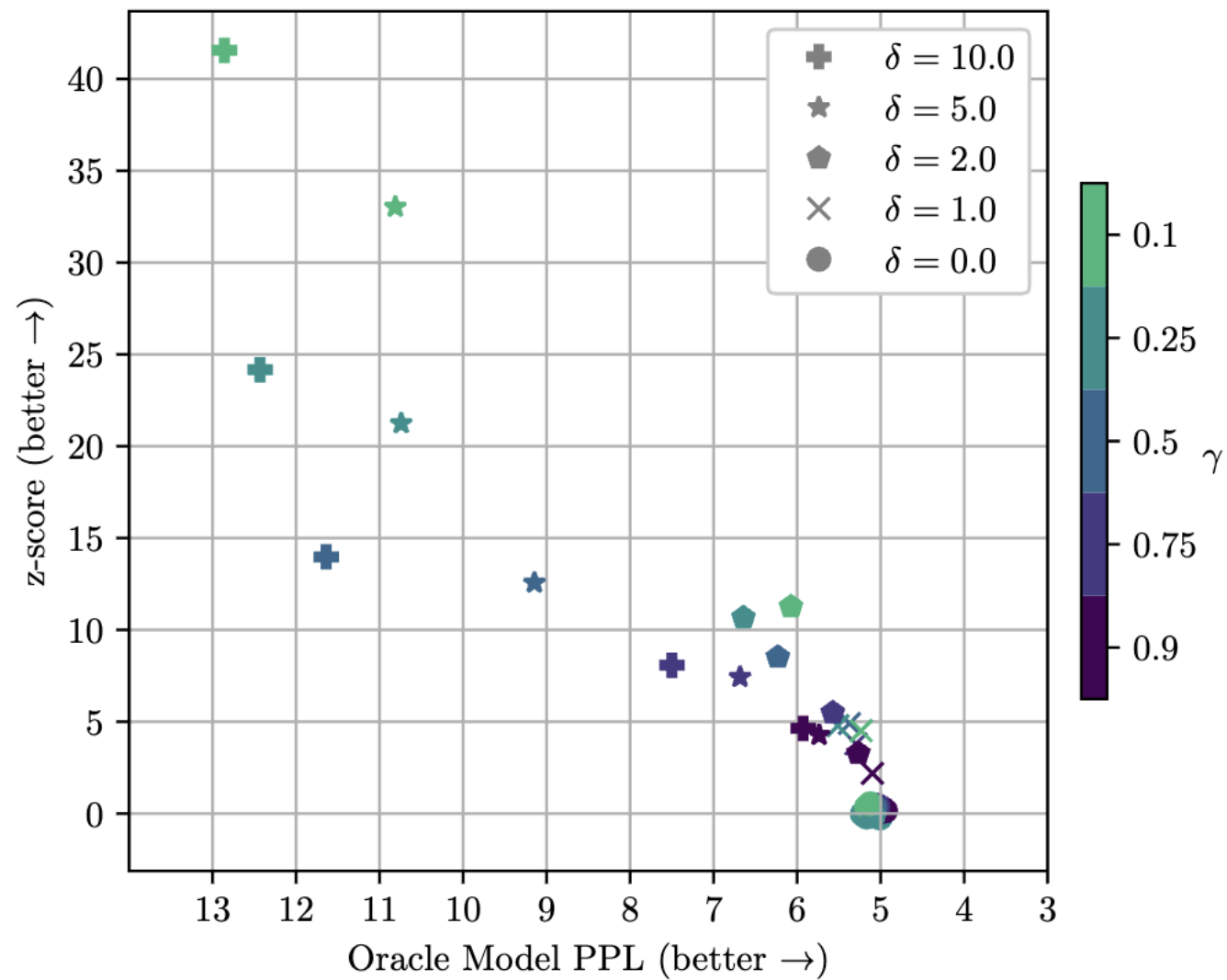
$$\mathbb{E} |s|_G \geq \frac{\gamma \alpha T}{1 + (\alpha - 1)\gamma} S^*,$$

# Private Watermarking



$$\hat{p}_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)} + \delta)}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)} + \delta)}, & k \in G \\ \frac{\exp(l_k^{(t)})}{\sum_{i \in R} \exp(l_i^{(t)}) + \sum_{i \in G} \exp(l_i^{(t)} + \delta)}, & k \in R. \end{cases}$$

# Text Generation Quality



# Text Generation Quality

prompt	real completion	no watermark (NW)	watermarked (W)
...tled out of court and publicly reconciled.\nIn the '80s the band's popularity waned in the United States but remained strong abroad. Robin released three solo albums, with limited success. The Bee Gees	returned with some moderate hits in the late 1990s and were inducted into the Rock and Roll Hall of Fame in 1997. With his brothers, Mr. Gibb won six Grammys.\nIn addition to his wife and his brother [...continues]	continued to tour, and Barry became a television producer.\nBut in the early '90s, the Bee Gees' popularity remained high. They scored a hit with "Don't Stop Believing" in 1990, and in 1992 the Bee Ge[...continues]	' 1990 album, "Spirits of the Century," was a mixed critical and commercial success.\nWhen the brothers were nominated for a Grammy Award in 1990, Mr. Gibb's "You Should Be Dancing" and "Massachusetts,[...continues]
... logged into their Google account and have verified profiles that match queries for the site.\nGoogle's John Mueller said there is no ranking benefit in using different Google Search Console and Google	Analytics accounts for each individual web site you manage. The topic came up before, as long as you are not spamming Google - there also is no down side to using the same accounts across multiple we[...continues]	Analytics data to calculate ranking sentiment.\nGoogle have a new feature called Keyword Difficulty Analysis that shows you the keywords your competitors are ranking for. It shows the demand curve as [...continues]	+ accounts to see different numbers.\nGoogle also released their own great blog post on the news algorithm. They give lots of great advice to help your site do better.\nFinally, at the end of September [...continues]

# Watermark Detection Results

				z=4.0				z=5.0			
sampling	$\delta$	$\gamma$	count	FPR	TNR	TPR	FNR	FPR	TNR	TPR	FNR
m-nom.	1.0	0.50	506	0.0	1.0	0.767	0.233	0.0	1.0	0.504	0.496
m-nom.	1.0	0.25	506	0.0	1.0	0.729	0.271	0.0	1.0	0.482	0.518
m-nom.	2.0	0.50	507	0.0	1.0	0.984	0.016	0.0	1.0	0.978	0.022
m-nom.	2.0	0.25	505	0.0	1.0	0.994	0.006	0.0	1.0	0.988	0.012
m-nom.	5.0	0.50	504	0.0	1.0	0.996	0.004	0.0	1.0	0.992	0.008
m-nom.	5.0	0.25	503	0.0	1.0	1.000	0.000	0.0	1.0	0.998	0.002
8-beams	1.0	0.50	495	0.0	1.0	0.873	0.127	0.0	1.0	0.812	0.188
8-beams	1.0	0.25	496	0.0	1.0	0.819	0.181	0.0	1.0	0.770	0.230
8-beams	2.0	0.50	496	0.0	1.0	0.992	0.008	0.0	1.0	0.984	0.016
8-beams	2.0	0.25	496	0.0	1.0	0.994	0.006	0.0	1.0	0.990	0.010
8-beams	5.0	0.50	496	0.0	1.0	1.000	0.000	0.0	1.0	1.000	0.000
8-beams	5.0	0.25	496	0.0	1.0	1.000	0.000	0.0	1.0	1.000	0.000



# How About Attacks?

- Generate text by machines first
- Perturb machine-generated text
  - **Query-free** word replacement
  - **Query-based** word replacement
  - **Paraphrasing** text

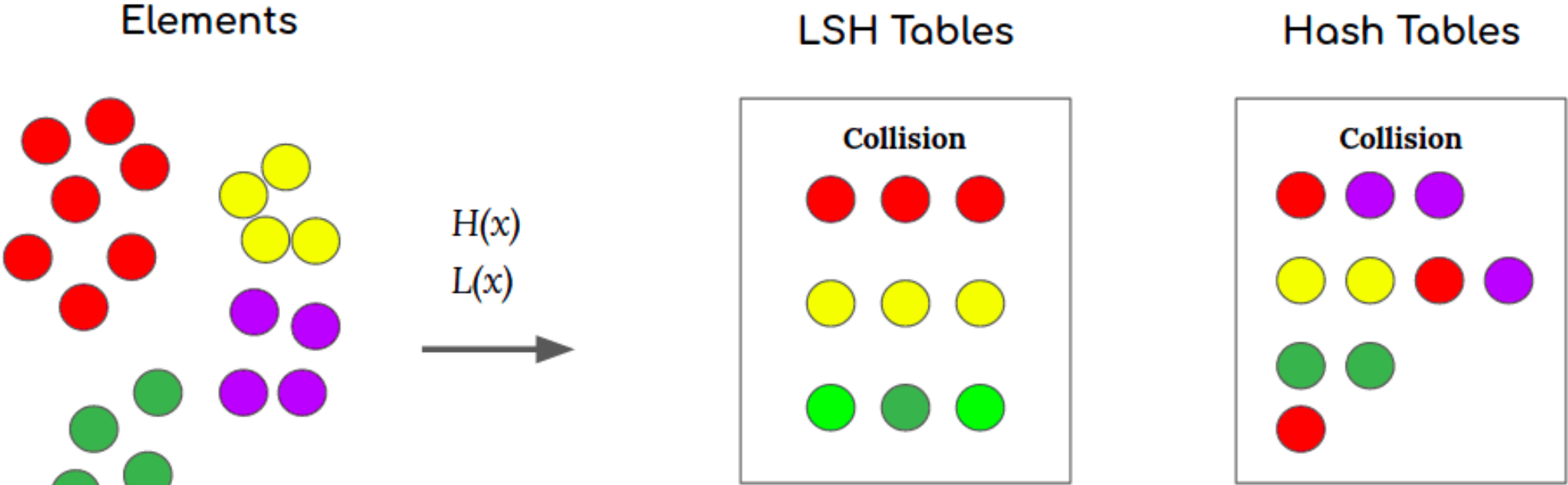
# Attacking Results

sampling	$\epsilon$	count	TPR@4.0	FNR@4.0	w/attck TPR@4.0	w/attck FNR@4.0	TPR@5.0	FNR@5.0	w/attck TPR@5.0	w/attck FNR@5.0
m-nom.	0.1	487	0.984	0.016	0.819	0.181	0.977	0.023	0.577	0.423
m-nom.	0.3	487	0.984	0.016	0.353	0.647	0.977	0.023	0.127	0.873
m-nom.	0.5	487	0.984	0.016	0.094	0.906	0.977	0.023	0.029	0.971
m-nom.	0.7	487	0.984	0.016	0.039	0.961	0.977	0.023	0.012	0.988
beams	0.1	489	0.998	0.002	0.834	0.166	0.998	0.002	0.751	0.249
beams	0.3	489	0.998	0.002	0.652	0.348	0.998	0.002	0.521	0.479
beams	0.5	489	0.998	0.002	0.464	0.536	0.998	0.002	0.299	0.701
beams	0.7	489	0.998	0.002	0.299	0.701	0.998	0.002	0.155	0.845

# SEMSTAMP: A Semantic Watermark with Paraphrastic Robustness for Text Generation

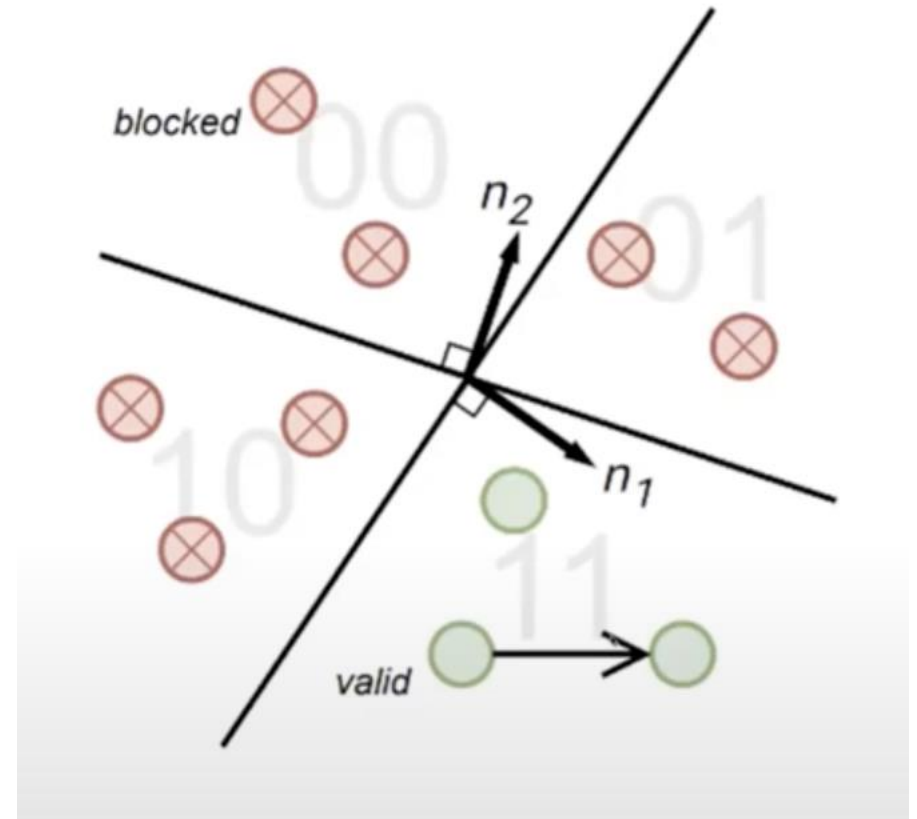
Abe Bohan Hou<sup>♣\*</sup>   Jingyu Zhang<sup>♣\*</sup>   Tianxing He<sup>♡\*</sup>  
Yichen Wang<sup>◇</sup>   Yung-Sung Chuang<sup>♠</sup>   Hongwei Wang<sup>‡</sup>   Lingfeng Shen<sup>♣</sup>  
Benjamin Van Durme<sup>♣</sup>   Daniel Khashabi<sup>♣</sup>   Yulia Tsvetkov<sup>♡</sup>  
<sup>♣</sup>Johns Hopkins University   <sup>♡</sup>University of Washington   <sup>◇</sup>Xi'an Jiaotong University  
<sup>♠</sup>Massachusetts Institute of Technology   <sup>‡</sup>Tencent AI Lab  
{bhou4, jzhan237}@jhu.edu   goosehe@cs.washington.edu

# Locality-Sensitive Hashing (LSH)



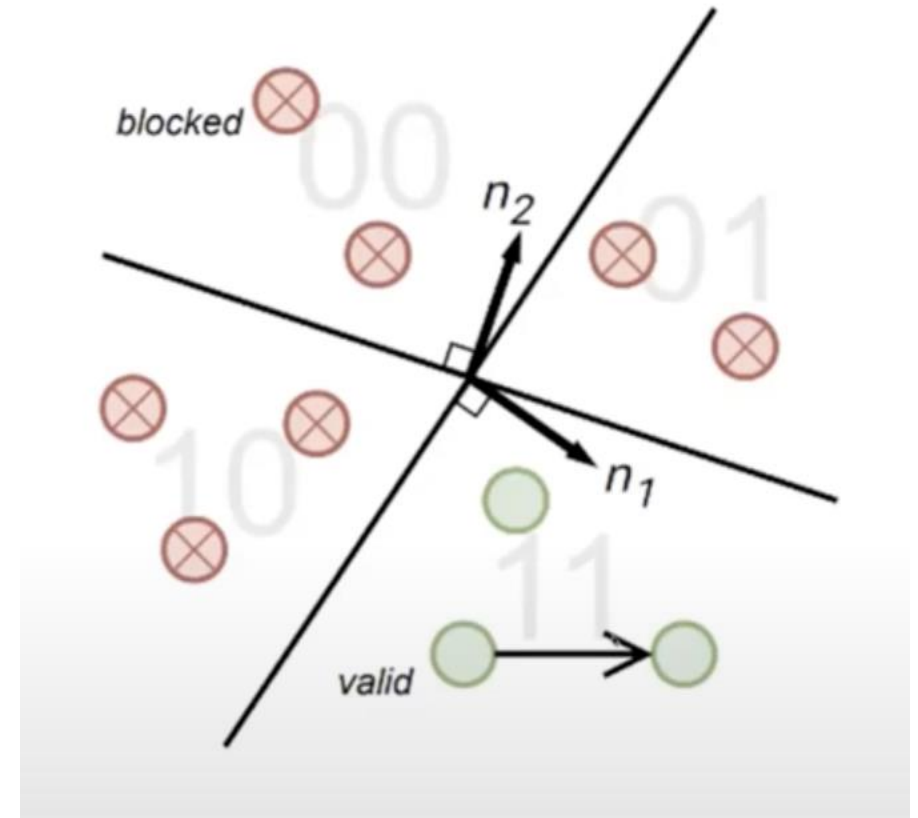
# Sentence Encoder

- Semantic encoder robust to paraphrasing
  - SentenceBERT, SimCSE, etc.



# Partition with LSH

- Each dot is a potential next sentence sampled from LM
- LSH partitions the semantic space through **random hyperplanes**
- Divide the semantic space into **valid** and **blocked** regions **by hashing on the previous sentence**

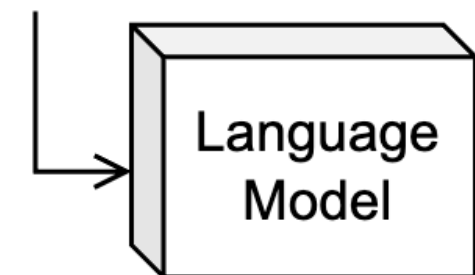


# Generation Overview

## ① Watermarked Generation

Lucy smiled.

*Rejection Sampling*

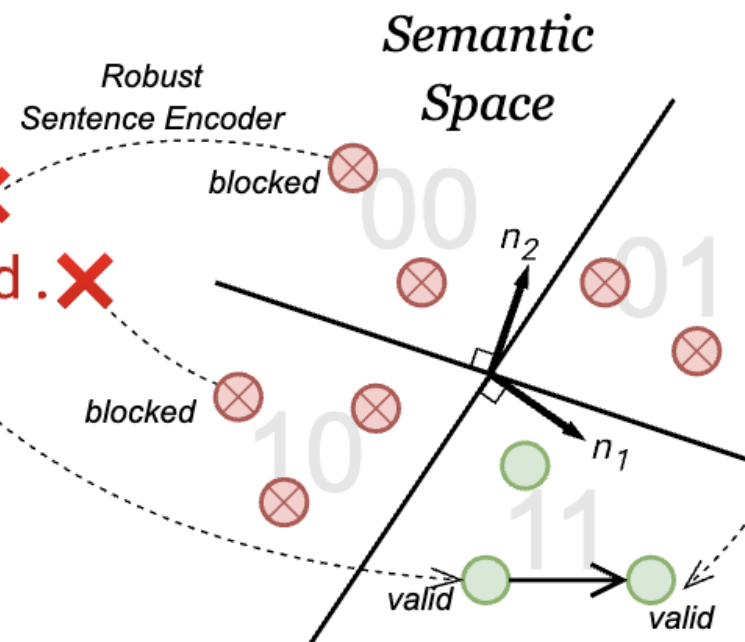


It was genuine.✗

Her eyes crinkled.✗

She was happy.✓

- LSH hyperplane
- LSH normal vector
- 01 LSH signature
- Valid region embedding
- ⊗ Blocked region embedding



# Paraphrase Attack

## ② Paraphrase Attack


*Watermark remains  
valid after paraphrase*

✓ She felt delighted.

## ③ watermark detecton

No Watermark

Today the company announced results for the third quarter of 2017. The company's board of directors also declared a quarterly cash dividend of \$0.23 per share. The dividend is payable to shareholders of record on November 14, 2017. Shareholders are invited to attend the company's annual meeting to propose and discuss a proposal to adopt a new long-term stockholder's plan. The meeting will be held on December 7, 2017.

z-test →  human written

SEMSTAMP

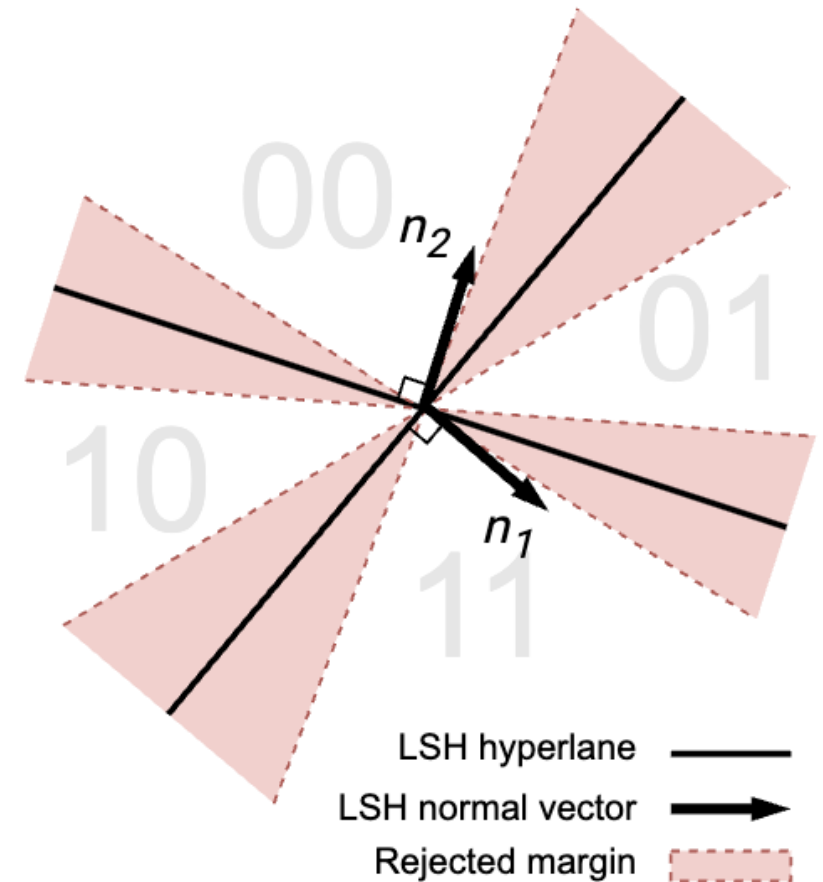
Today the company announced quarterly results for the period ending October 31, 2017. The company also provided an update on its ongoing Phase 3 clinical trial of the Phase 2/3 B-cell-derived T cell engager program. These results are included in a newly released Current Report on Form 8-K for the period ending September 30, 2017. You can read the full report at [www.curis.com](http://www.curis.com).

z-test →  machine written



# Consider Margin for Robustness

- Sentence encoder is not perfect
- Only accept sentences with distance larger than a margin



# Results

<i>Paraphraser</i>	<i>Algorithm</i>	RealNews   BookSum   Reddit-TIFU		
		<i>AUC</i> ↑	<i>TP@1%</i> ↑	<i>TP@5%</i> ↑
No Paraphrase	KGW	99.6   99.9   99.3	98.4   99.4   97.5	98.9   99.5   98.1
	SSTAMP	99.2   99.7   99.7	93.9   98.8   97.7	97.1   99.1   98.2
Pegasus	KGW	95.9   97.3   94.1	82.1   89.7   87.2	91.0   95.3   87.2
	SSTAMP	<b>97.8   99.2   98.4</b>	<b>83.7   90.1   92.8</b>	<b>92.0   96.8   95.4</b>
Pegasus-bigram	KGW	92.1   96.5   91.7	42.7   56.6   67.2	72.9   85.3   67.6
	SSTAMP	<b>96.5   98.9   98.0</b>	<b>76.7   86.8   89.0</b>	<b>86.0   94.6   92.9</b>
Parrot	KGW	88.5   94.6   79.5	31.5   42.0   22.8	55.4   75.8   43.3
	SSTAMP	<b>93.3   97.5   90.2</b>	<b>56.2   70.3   56.2</b>	<b>75.5   88.5   70.5</b>
Parrot-bigram	KGW	83.0   93.1   82.8	15.0   39.9   27.6	37.4   71.2   49.7
	SSTAMP	<b>93.1   97.5   93.9</b>	<b>54.4   71.4   71.8</b>	<b>74.0   89.4   82.3</b>
GPT3.5	KGW	82.8   87.6   84.1	17.4   17.2   27.3	46.7   52.1   50.9
	SSTAMP	<b>83.3   91.8   87.7</b>	<b>33.9   55.0   47.5</b>	<b>52.9   70.8   58.2</b>
GPT3.5-bigram	KGW	75.1   77.1   79.8	5.9   4.4   19.3	26.3   27.1   41.3
	SSTAMP	<b>82.2   90.5   87.4</b>	<b>31.3   47.4   43.8</b>	<b>48.7   63.6   55.9</b>

# ON THE RELIABILITY OF WATERMARKS FOR LARGE LANGUAGE MODELS

**John Kirchenbauer**<sup>\*1</sup>, **Jonas Geiping**<sup>\*2,3</sup>

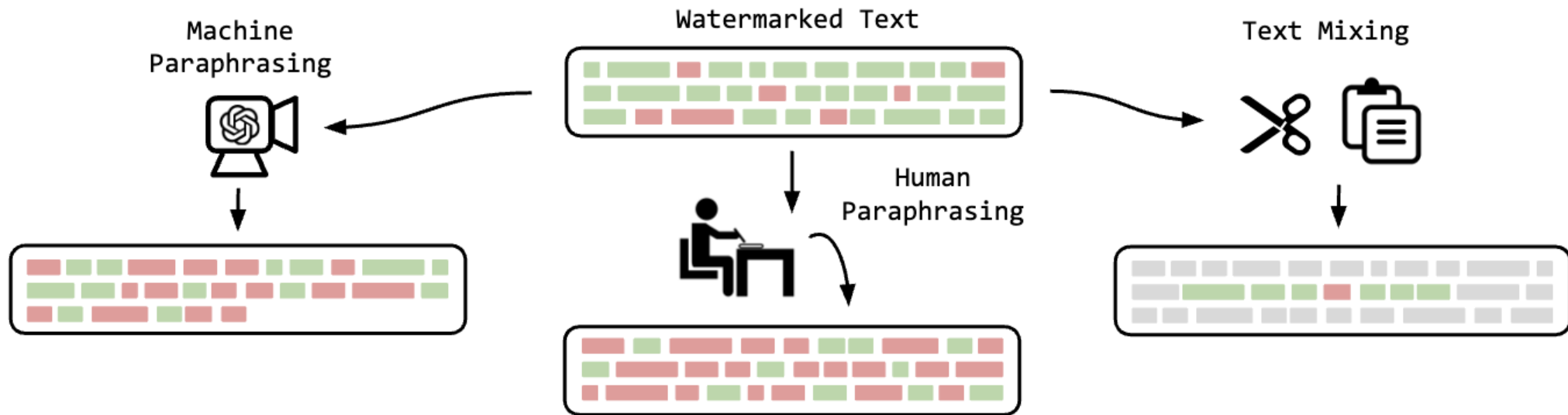
**Yuxin Wen**<sup>1</sup>, **Manli Shu**<sup>1</sup>, **Khalid Saifullah**<sup>1</sup>, **Kezhi Kong**<sup>1</sup>,  
**Kasun Fernando**<sup>4</sup>, **Aniruddha Saha**<sup>1</sup>, **Micah Goldblum**<sup>5</sup>, **Tom Goldstein**<sup>1</sup>

<sup>1</sup> University of Maryland

<sup>2</sup> ELLIS Institute Tübingen, <sup>3</sup> Max-Planck Institute for Intelligent Systems, Tübingen AI Center

<sup>4</sup> Scuola Normale Superiore di Pisa, <sup>5</sup> New York University

# More Study on Attacks for Token-Level Watermark



# Results

