清華大學 交叉信息研究院
Institute for Interdisciplinary Information Sciences, Tsinghua University

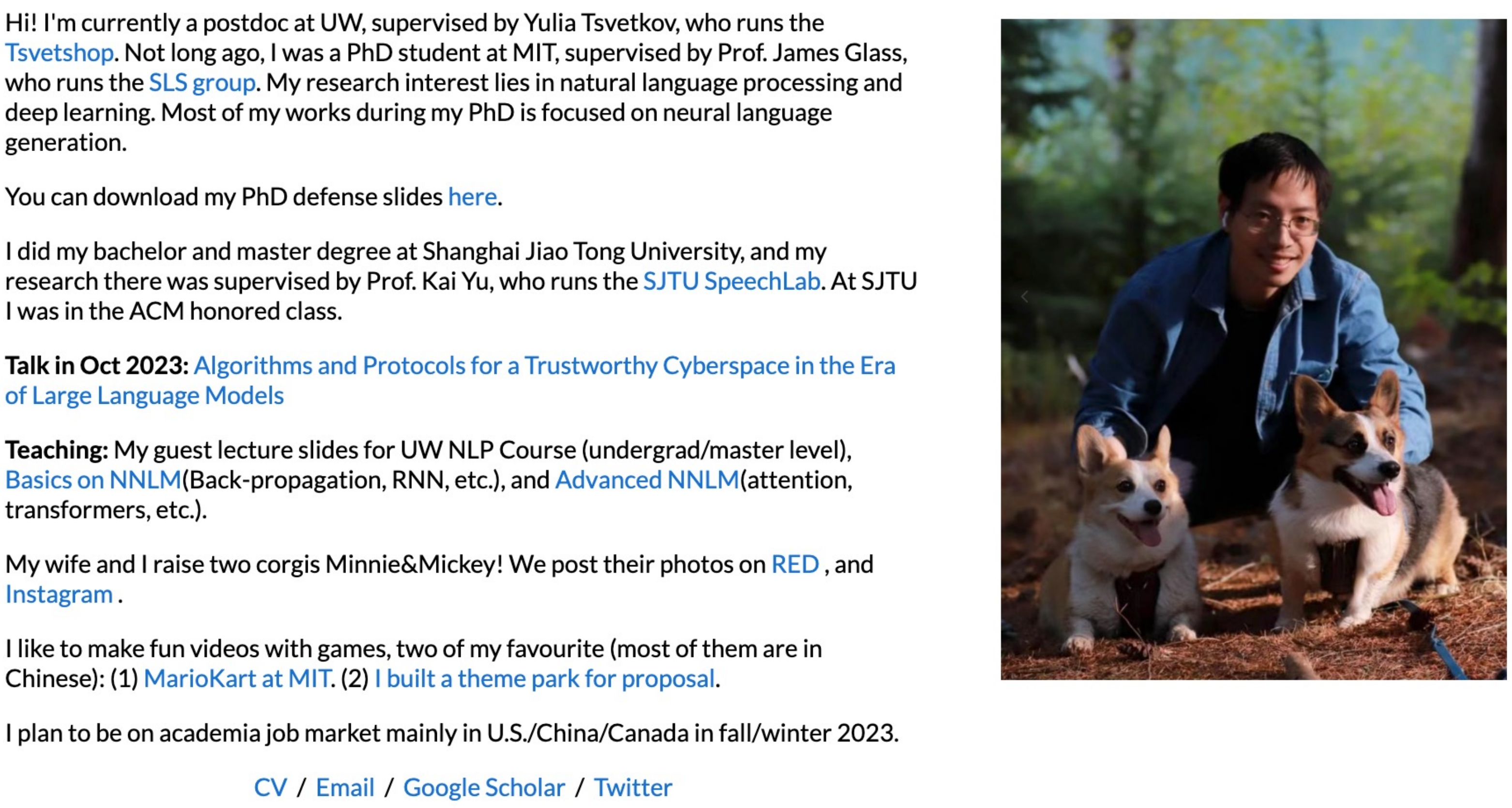
Algorithms and Protocols for a Trustworthy Cyberspace in the Era of Large Language Models

Tianxing He

University of Washington

Oct 2023

Tianxing He （贺天行）



Hi! I'm currently a postdoc at UW, supervised by Yulia Tsvetkov, who runs the Tsvetshop. Not long ago, I was a PhD student at MIT, supervised by Prof. James Glass, who runs the SLS group. My research interest lies in natural language processing and deep learning. Most of my works during my PhD is focused on neural language generation.

You can download my PhD defense slides here.

I did my bachelor and master degree at Shanghai Jiao Tong University, and my research there was supervised by Prof. Kai Yu, who runs the SJTU SpeechLab. At SJTU I was in the ACM honored class.

Talk in Oct 2023: Algorithms and Protocols for a Trustworthy Cyberspace in the Era of Large Language Models

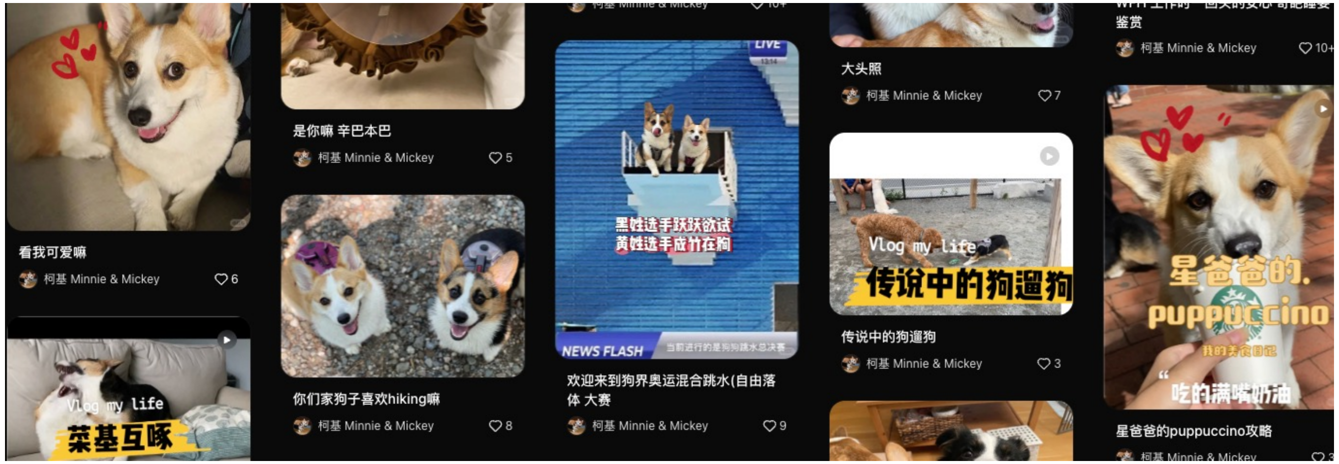
Teaching: My guest lecture slides for UW NLP Course (undergrad/master level), Basics on NNLM(Back-propagation, RNN, etc.), and Advanced NNLM(attention, transformers, etc.).

My wife and I raise two corgis Minnie&Mickey! We post their photos on RED , and Instagram .

I like to make fun videos with games, two of my favourite (most of them are in Chinese): (1) MarioKart at MIT. (2) I built a theme park for proposal.

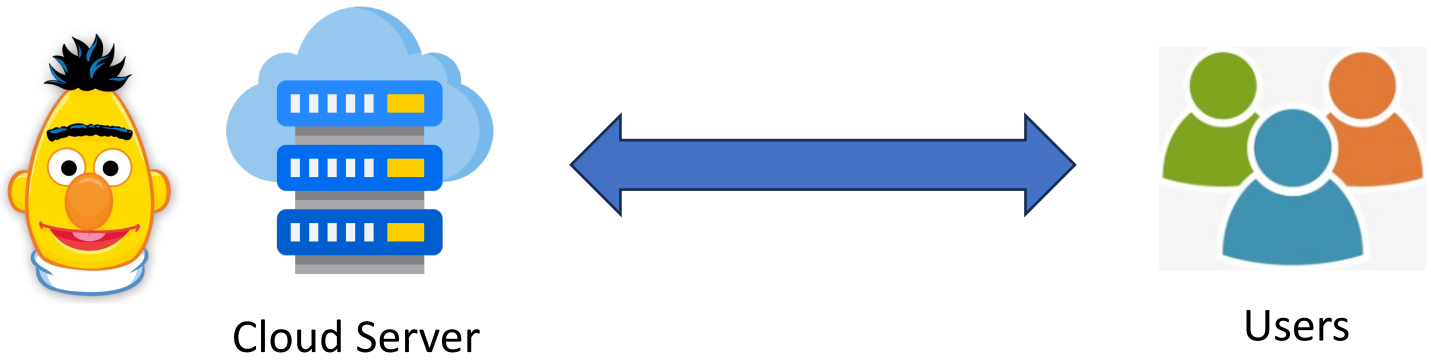
I plan to be on academia job market mainly in U.S./China/Canada in fall/winter 2023.

CV / Email / Google Scholar / Twitter



Towards a Trustworthy Cyberspace with LLMs

- The widening adoption of large language models (LLMs) on cloud brings urgent problems related to privacy and social engineering.
- How do we establish *trust* server and user ?

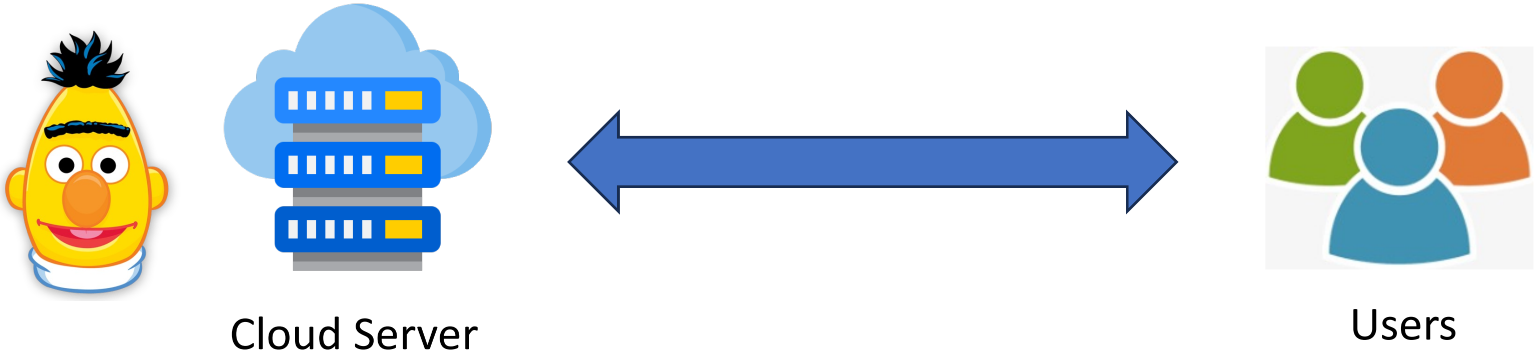


Social Engineering (propaganda, misinformation), Jailbreak, etc.

Privacy, Copyright, Factuality, Alignment, etc.

New challenges call for novel protocols/algorithms!

More specifically, my work focus on the generation aspect.



Cloud Server

Users

1. How can the server prevent malicious users from using the generation for misinformation?

SemStamp: A Semantic Watermark Algorithm

*H*Z*H*WCWSVKT, arXiv, 2023*

2. How can users hide prompt or generated text from the server (privacy-aware generation)?

LatticeGen: A Cooperative Protocol for Privacy-Aware Generation.

*Z*H*WMMCWT, arXiv, 2023*

Basic: Auto-Regressive Language Model

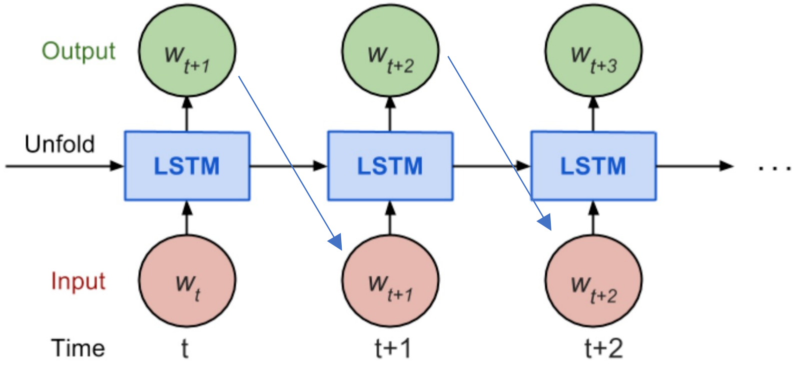
- LM assigns a probability $\mathrm{P}_\theta(W_{1:L})$ to a given sentence $W_{1:L}$
- Auto-regressive LMs predict the next token W_i given history $W_{1:i-1}$.

$$\log P_\theta(W) = \sum \log P_\theta(W_i | W_{1:i-1})$$

- The GPT series are all autoregressive LMs.
- Modeling: Recurrent Neural Network / LSTM / Transformer

→

Generation: sampled token is fed as input for next time-step

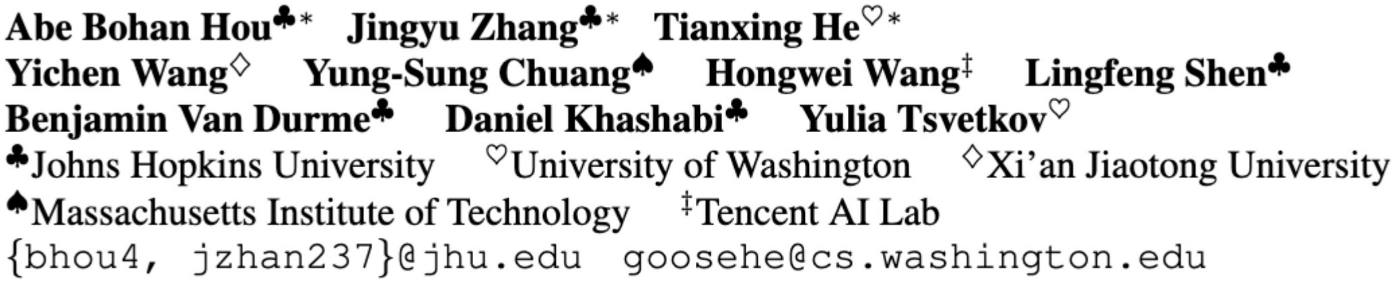


Our focus today is NOT about BERT, which is a <u>masked</u> language model.

Outline

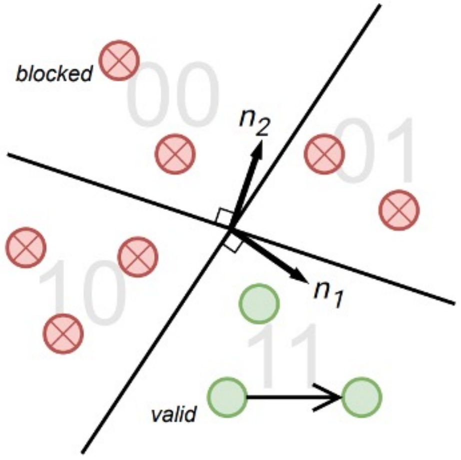
Questions welcomed during slide switches

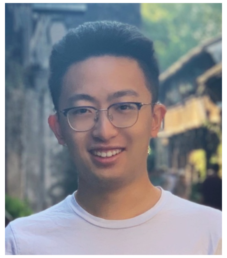
- **SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation**



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

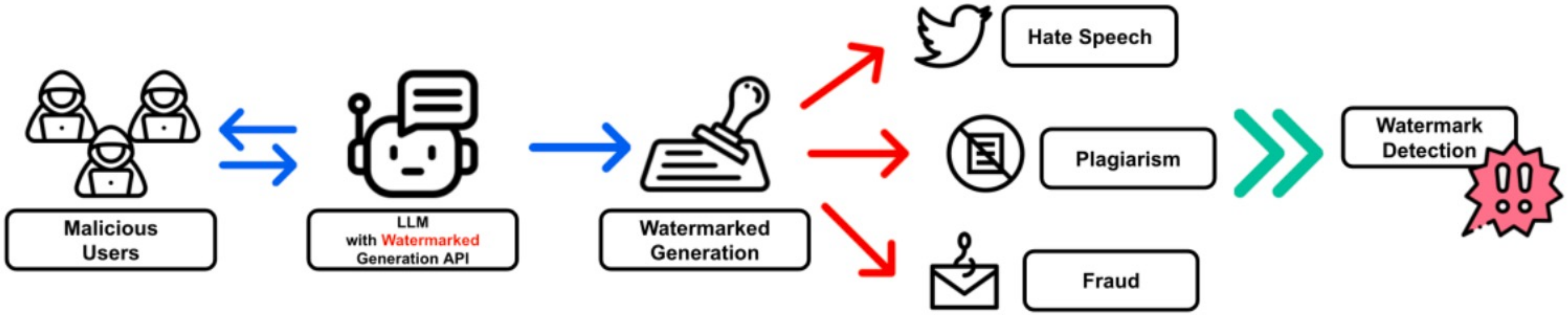
- **LatticeGen: A Cooperative Framework which Hides Generated Text in a Lattice For Privacy-Aware Generation on Cloud**
*Z*H*WMMCWT, arXiv, 2023*



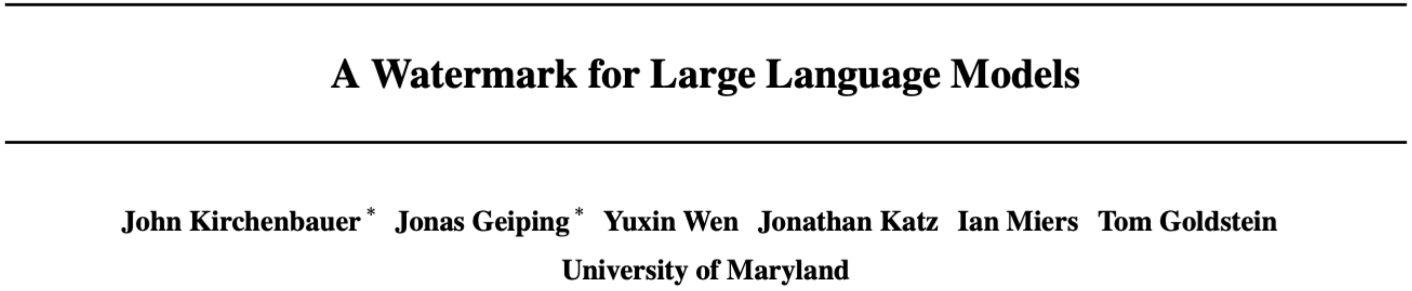


Watermarked Generation for LLM

- Watermarked generation: an approach which facilitates the detection of machine-generated text by adding algorithmically detectable signatures during LLM generation which are imperceptible to humans.



The Baseline Token-Level Algorithm

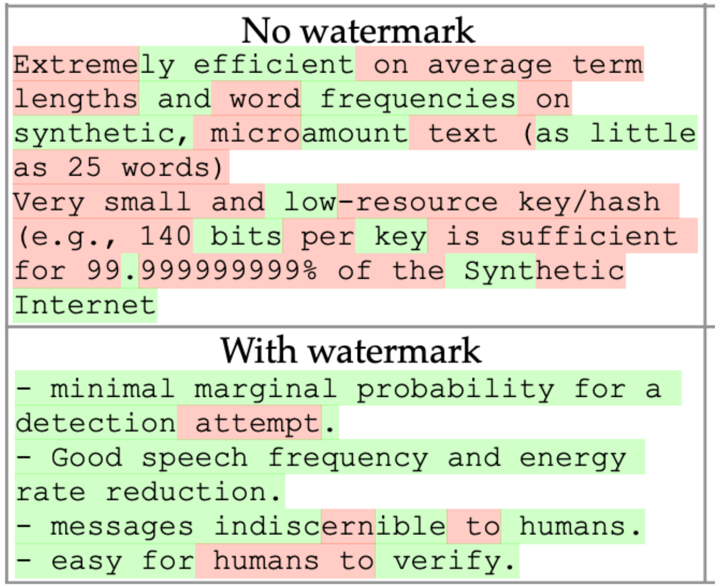


A Watermark for Large Language Models

John Kirchenbauer* Jonas Geiping* Yuxin Wen Jonathan Katz Ian Miers Tom Goldstein

University of Maryland

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



No watermark

```
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 99.999999999% of the Synthetic
Internet
```

With watermark

```
- minimal marginal probability for a
detection attempt.
- Good speech frequency and energy
rate reduction.
- messages indiscernible to humans.
- easy for humans to verify.
```

- The baseline algorithm operates by adding bias to a green-listed (G) subset of V.
- The green list is pseudo-randomly generated by using the previous token as the hash.
- The detection is determined by counting green-listed tokens in a given document.

The Baseline Token-Level Algorithm: Weakness

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

- Weakness:

(1) The token-level noise hurts quality. (damages PPL)

(2) Could be vulnerable to paraphrase attack. (Considering the hash is from the previous token)

Bi-gram Paraphrase Attack

- After beam-search (Pegasus) we get N(20) paraphrases.
- We select the beam with the lowest bigram overlap with the original sentence.
- This gives 3%~4% drop for the baseline alg.

Example Generation Sentence (SemStamp): It's not the same thing as a marketing campaign, but it is a good starting point.

Paraphrase: It isn't the same as a marketing campaign, but it is a good starting point.

Bigram Paraphrase: It's not a marketing campaign, but it's a good start.

SemStamp: A Semantic Watermark

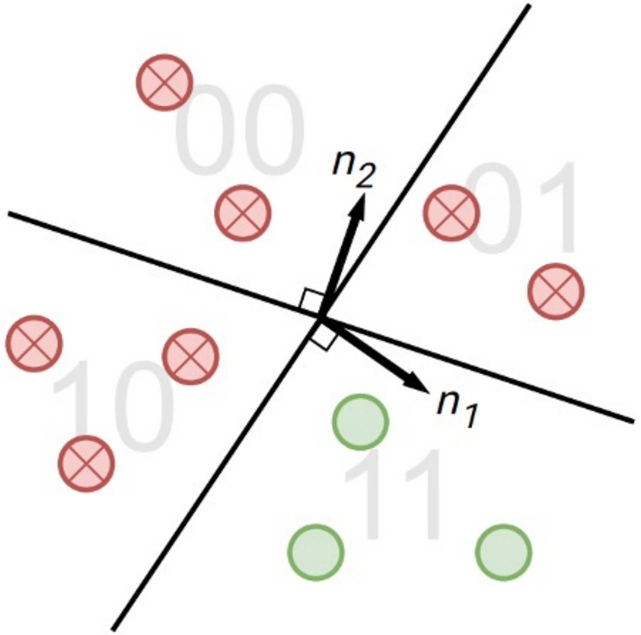
- We propose a sentence-level semantic watermark algorithm.

- We apply the masking on sentence-level "semantic space", instead of token-level.

- There are two core components:

(1) semantic encoder robust to paraphrasing (SentenceBert).

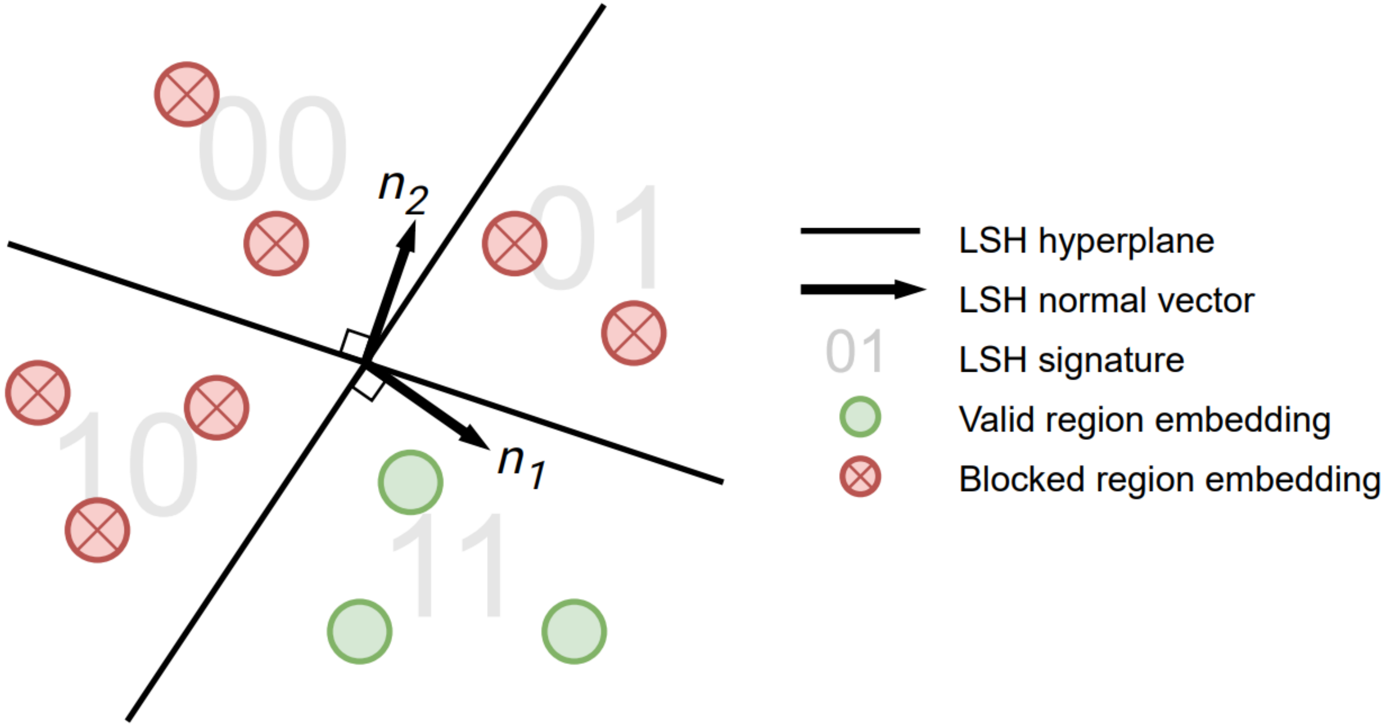
(2) Space partition and masking. (Locality-Sensitive Hashing, LSH)



Proposed: Semantic Watermark

- Each node represents a potential next sentence.

- LSH partitions the semantic space by random planes. We apply a watermark mask on randomly selected partitions (green).



Overview: Rejection-Sampling

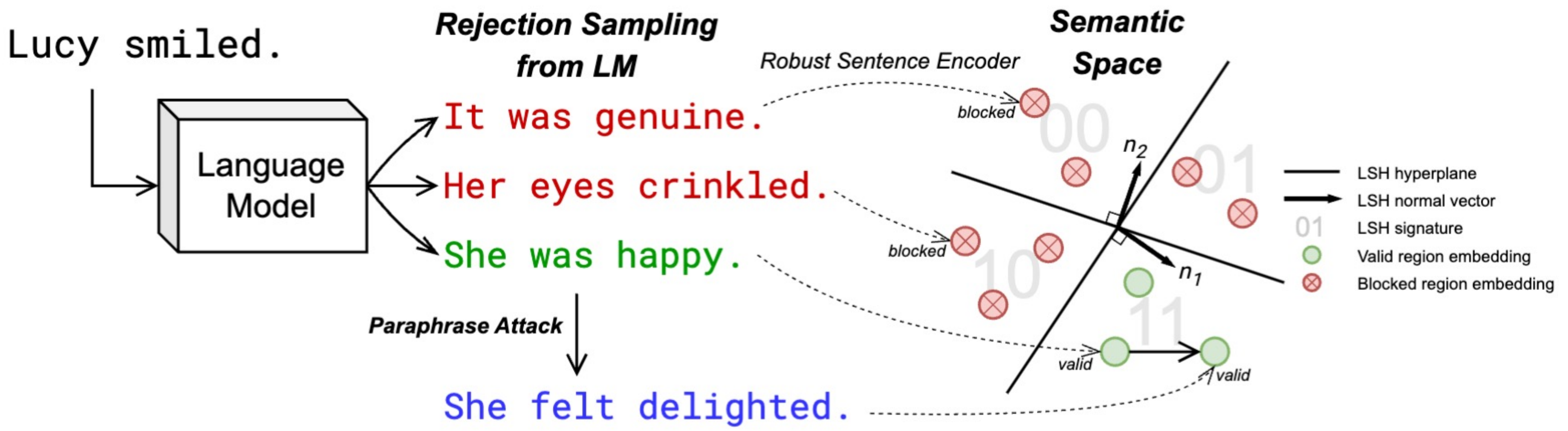
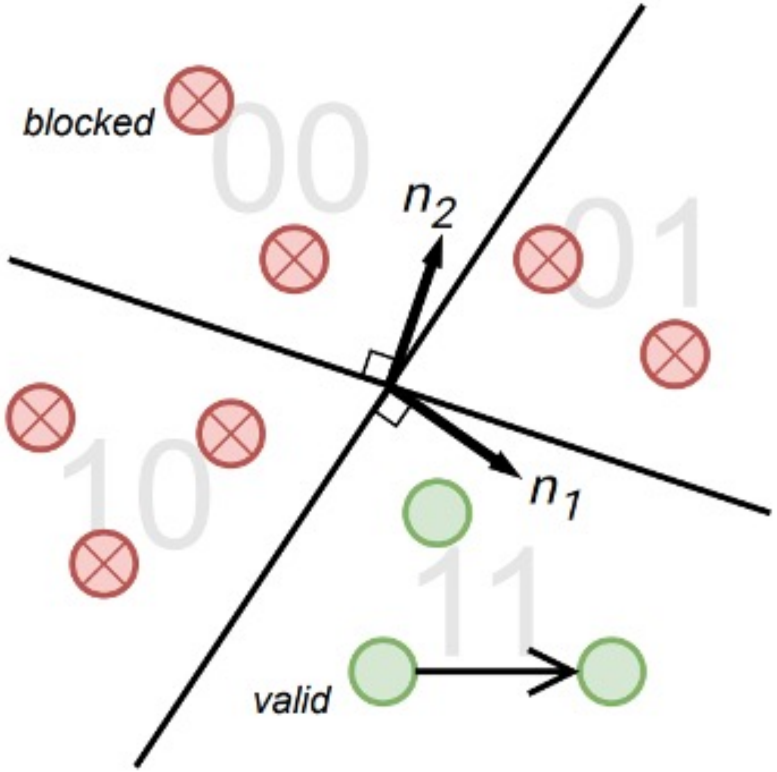


Figure 1: An overview of the proposed SEMSTAMP algorithm. The watermark is injected by mapping candidate sentences into embeddings through a robust sentence encoder, dividing the semantic space through locality-sensitive hashing, and rejection sampling from the LM to generate sentences with valid region embeddings.

- In our hyper-parameter setting, we need to sample around 14 sentences for a valid sample. We are trading speed for watermarking.

Robustness to Paraphrase Attack

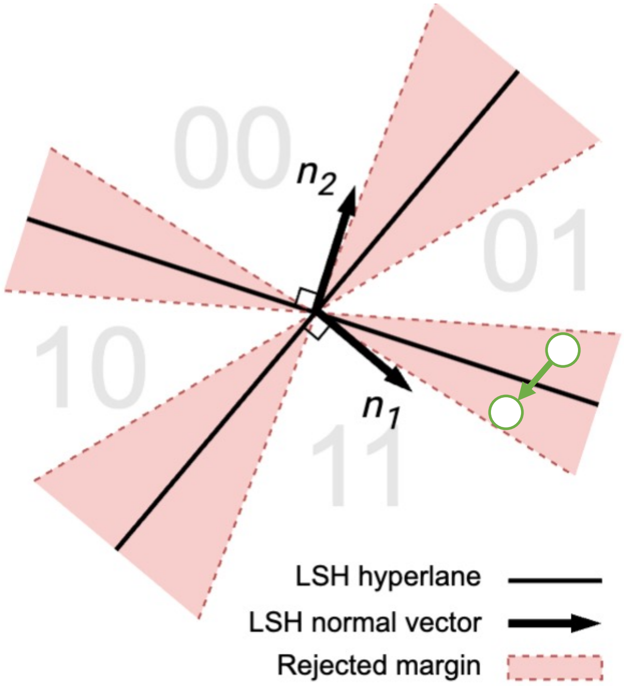
- Assuming the robustness of the embedder (enhanced by contrastive learning), the LSH signature of the paraphrased sentence does not change.



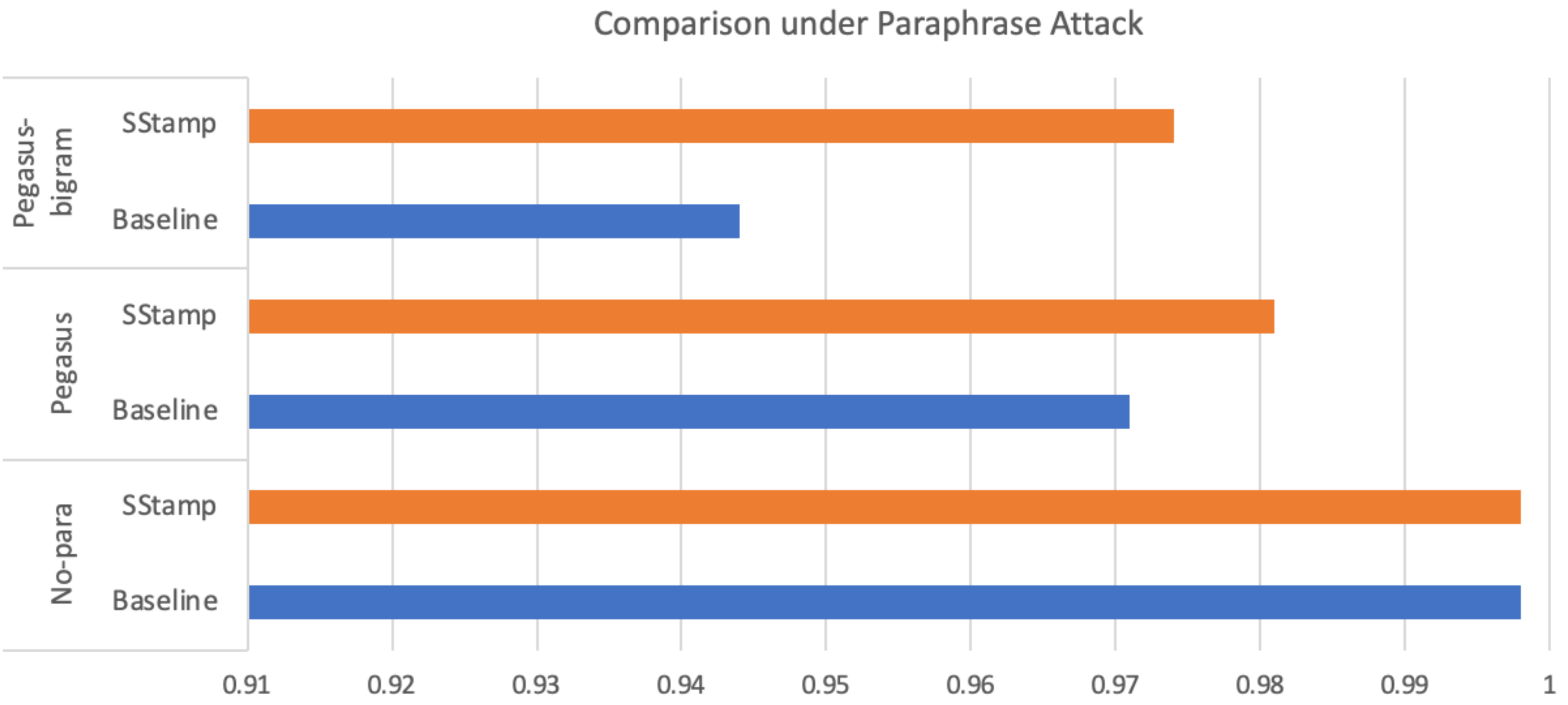
Trick: Reject Generation Close to LSH Boundary

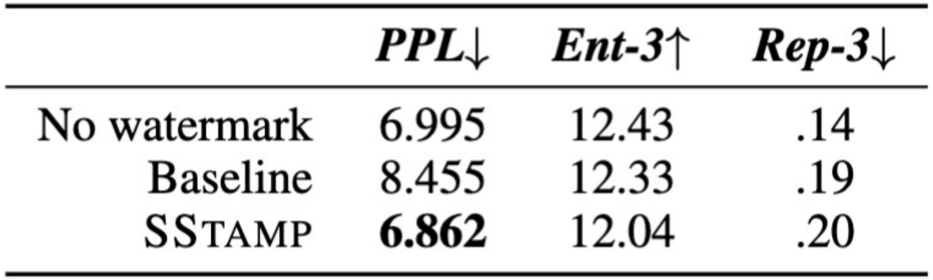
- In practice, we find that even after CL, the LSH code is not robust enough to paraphrasing (LSH accuracy under para. only ~70%).

- To alleviate this, we propose to add a rejection margin, and only accept sentences whose cos-sim with the normal vectors is larger than a margin (0.02).



Result: Improved Robustness



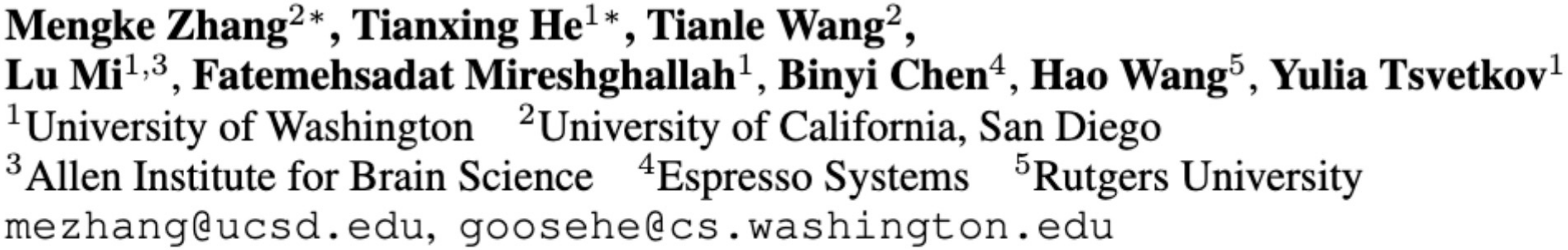


	PPL↓	*Ent-3*↑	*Rep-3*↓
No watermark	6.995	12.43	.14
Baseline	8.455	12.33	.19
SSTAMP	**6.862**	12.04	.20

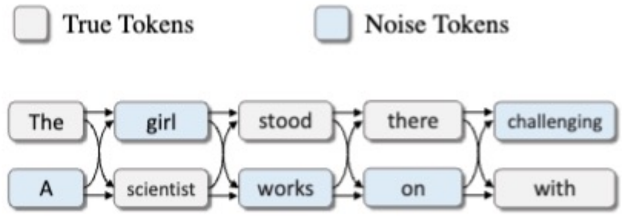
Outline

Questions welcomed during slide switches

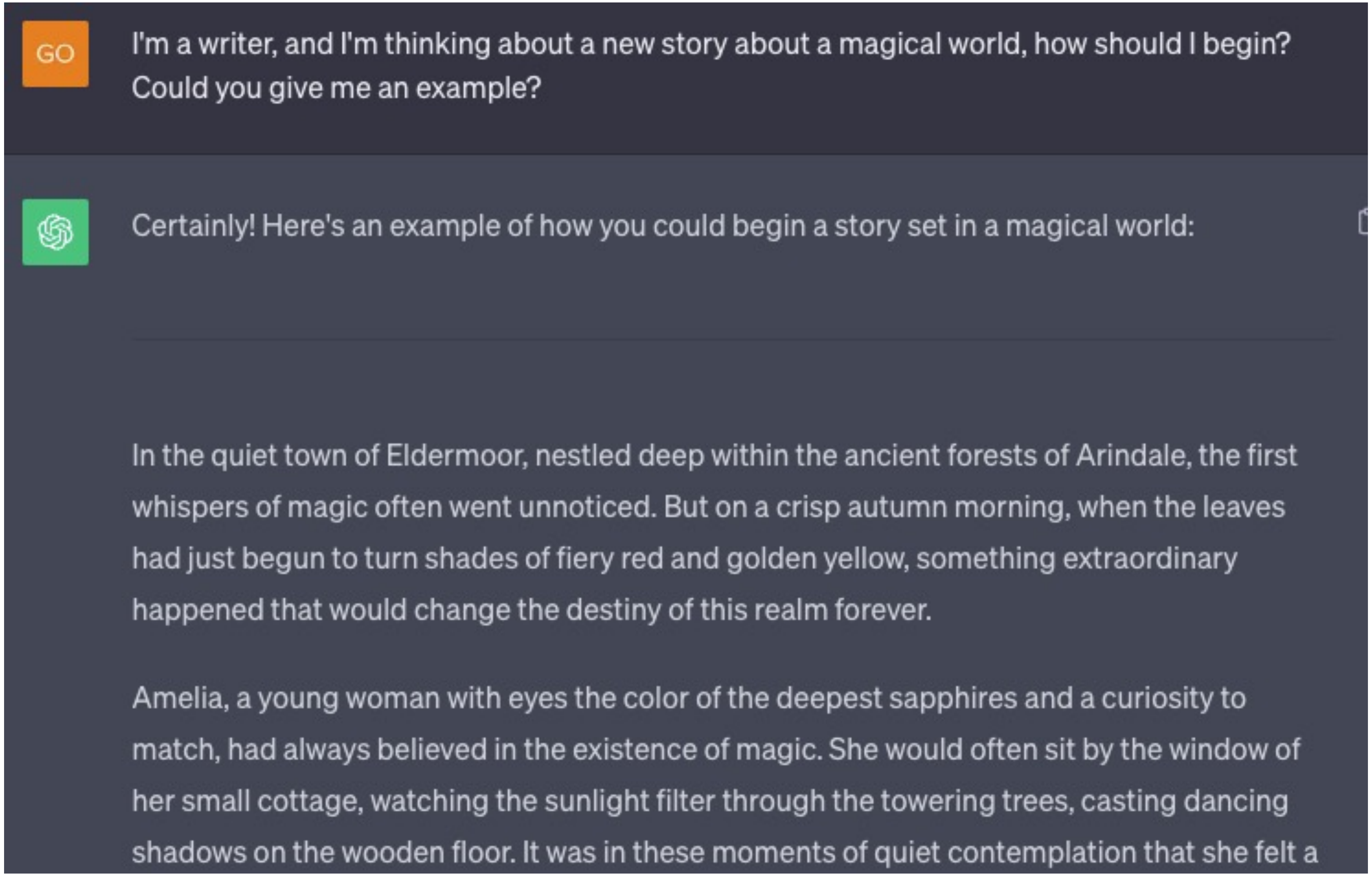
- **LatticeGen: A Cooperative Framework which Hides Generated Text in a Lattice For Privacy-Aware Generation on Cloud**



Mengke Zhang[2]*, Tianxing He[1]*, Tianle Wang[2],
Lu Mi[1,3], Fatemehsadat Mireshghallah[1], Binyi Chen[4], Hao Wang[5], Yulia Tsvetkov[1]
[1]University of Washington [2]University of California, San Diego
[3]Allen Institute for Brain Science [4]Espresso Systems [5]Rutgers University
mezhang@ucsd.edu, goosehe@cs.washington.edu



In the current prompted generation interface, the server has full control of the generation process, leaving zero option for users who want to keep the prompt or generated text to themselves.



I'm a writer, and I'm thinking about a new story about a magical world, how should I begin? Could you give me an example?

Certainly! Here's an example of how you could begin a story set in a magical world:

In the quiet town of Eldermoor, nestled deep within the ancient forests of Arindale, the first whispers of magic often went unnoticed. But on a crisp autumn morning, when the leaves had just begun to turn shades of fiery red and golden yellow, something extraordinary happened that would change the destiny of this realm forever.

Amelia, a young woman with eyes the color of the deepest sapphires and a curiosity to match, had always believed in the existence of magic. She would often sit by the window of her small cottage, watching the sunlight filter through the towering trees, casting dancing shadows on the wooden floor. It was in these moments of quiet contemplation that she felt a

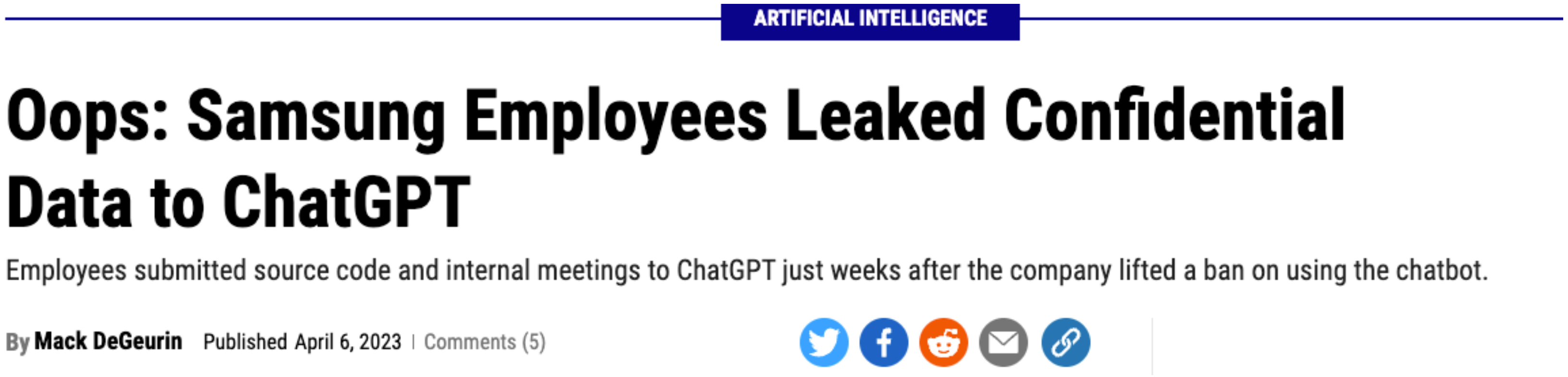
Motivation: Generated Text (also) Needs Obfuscation

- We argue that generated text also needs obfuscation because it affects the users’ real-life decisions.

- e.g., a customer is likely to go to the restaurant suggested by the LLM; an engineer could adopt the approach proposed by the LLM; a writer could take inspiration from outputs provided by the LLM, etc.

- Most work on NLP privacy (e.g., DP-SGD) focus on protecting the training data.

Motivation: Generated Text (also) Needs Obfuscation



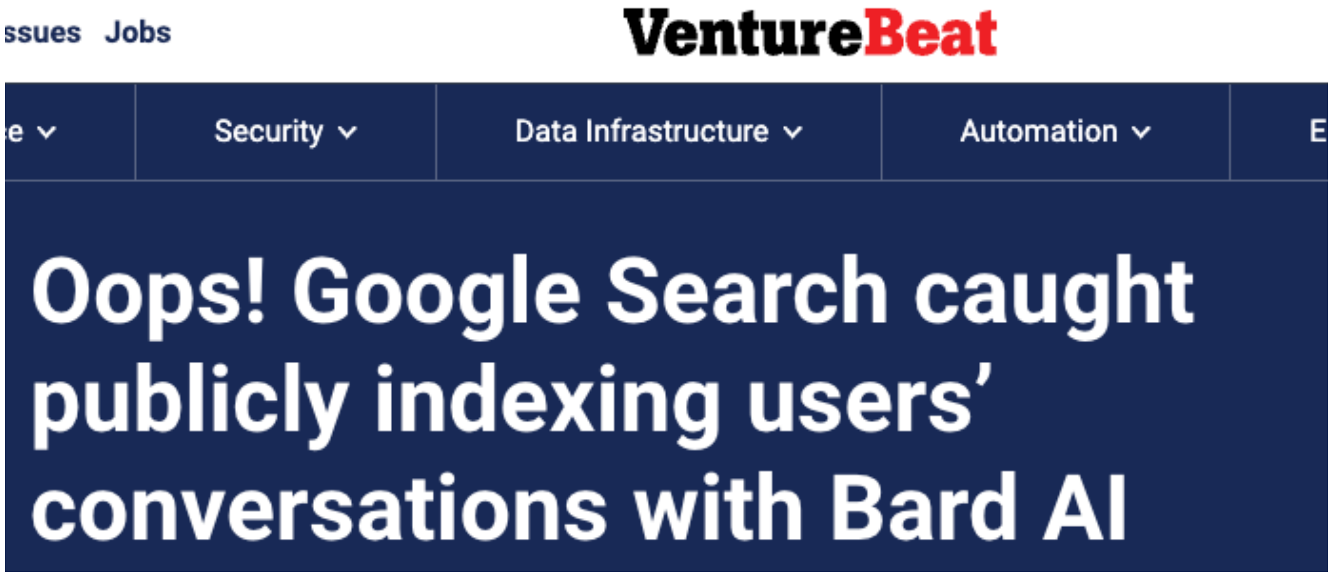
ARTIFICIAL INTELLIGENCE

Oops: Samsung Employees Leaked Confidential Data to ChatGPT

Employees submitted source code and internal meetings to ChatGPT just weeks after the company lifted a ban on using the chatbot.

By **Mack DeGeurin** Published April 6, 2023 | Comments (5)

Regulation maybe not enough! LatticeGen provides an algorithmic approach to protect user privacy.

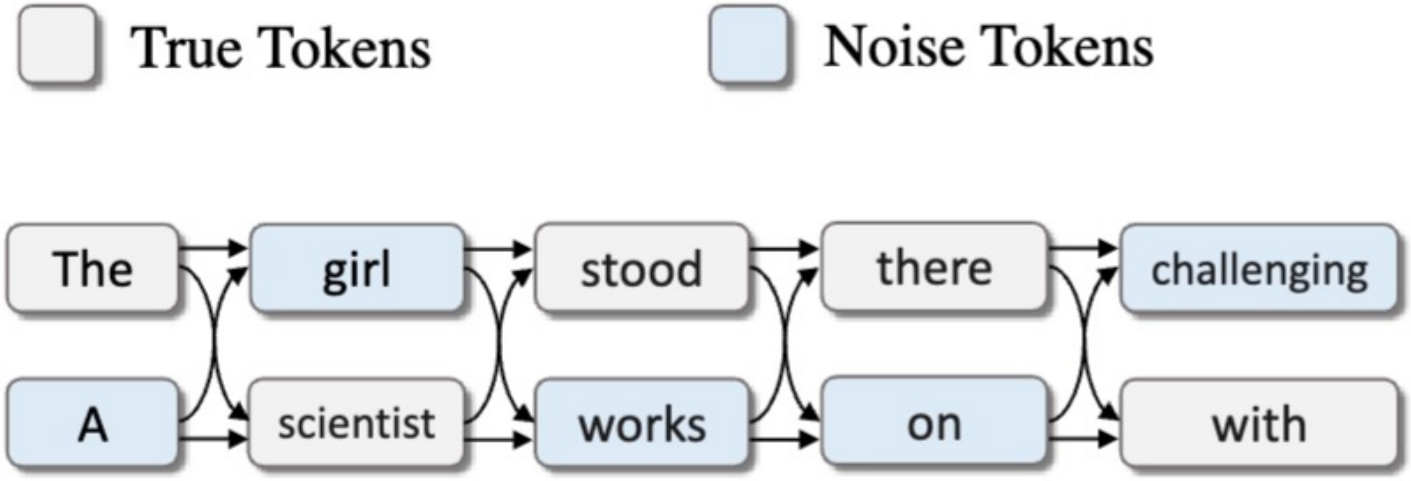


Oops! Google Search caught publicly indexing users' conversations with Bard AI

Intuition of LatticeGen:
Hiding Generated Text in a Lattice

Example Outcome:
This is a (shuffled) 2-lattice (N=2).

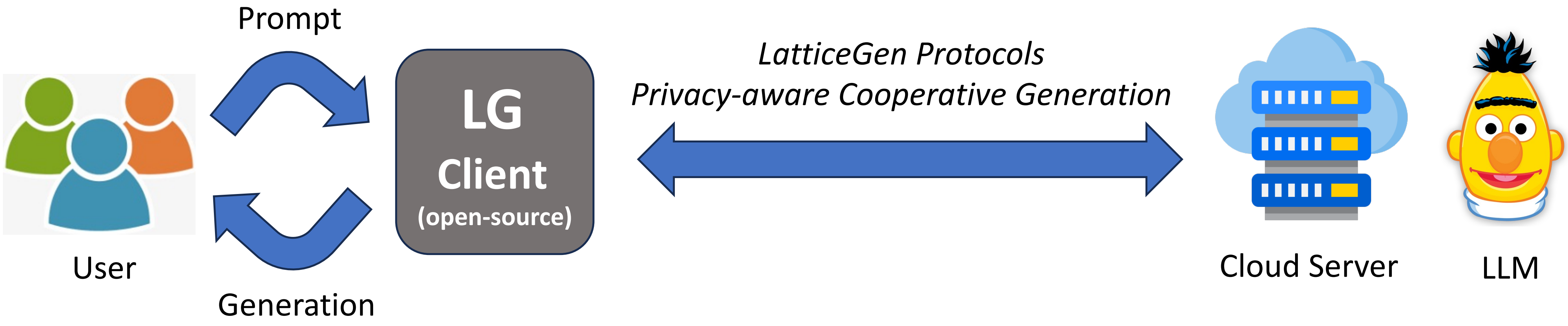
The user knows the true sequence, but the server does not.



The prompt can also be protected in the lattice, and is omitted in this figure.

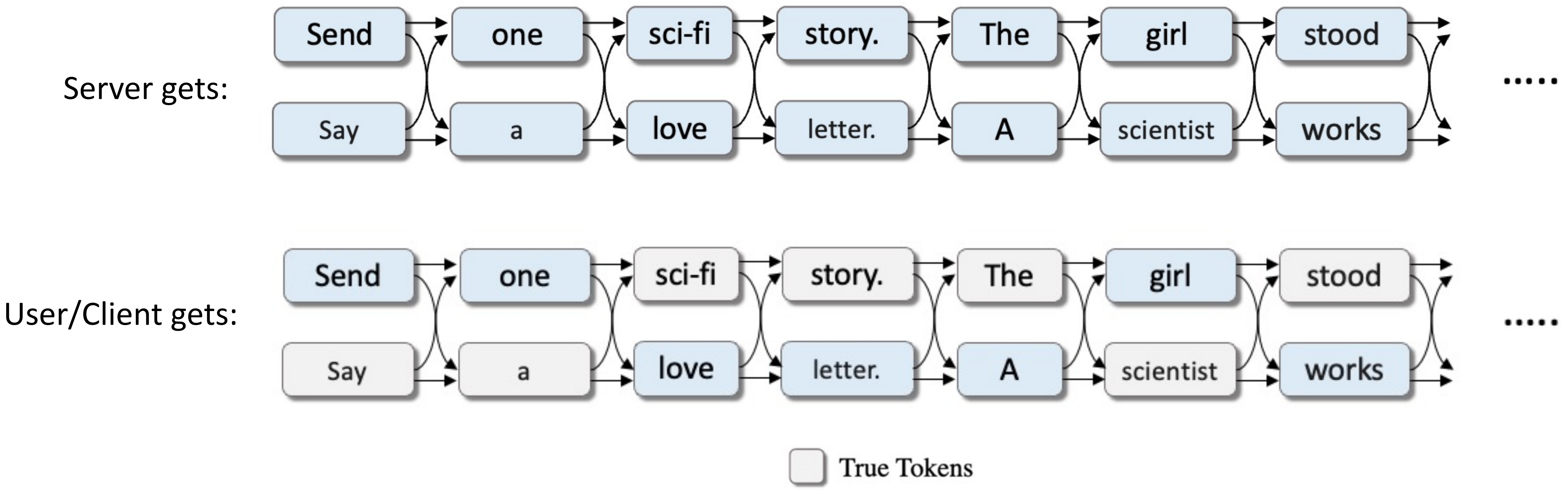
LatticeGen: High-Level Input & Output

- The user gives a prompt (e.g., "Say a sci-fi story.") to the LG client. The client only needs communication with the LLM server, and is supposed to protect user's privacy from the server. The client code could be open-sourced, and run by user on any laptop with a private config without involving a third-party.
- Assuming the server agrees to follow the LG protocol, the client handles the LG interactions with the cloud LLM server. (<-major focus of this project!)
- Finally, the client returns the generation to user.

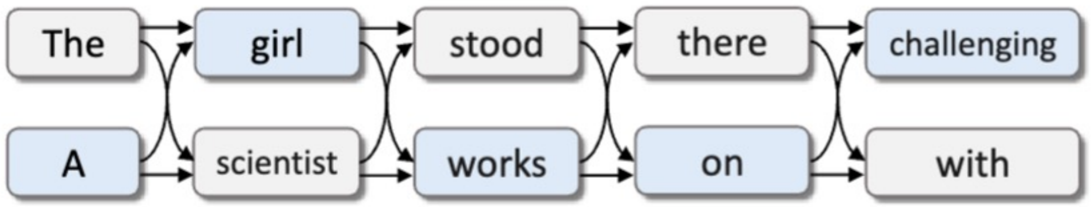


LatticeGen: High-Level Input & Output

- As a result, both the server and user gets the same noised lattice.
- The difference is that the user/client knows the true token sequence, while server does not.



Key Question: Why Not Just Generate Twice ?



VS

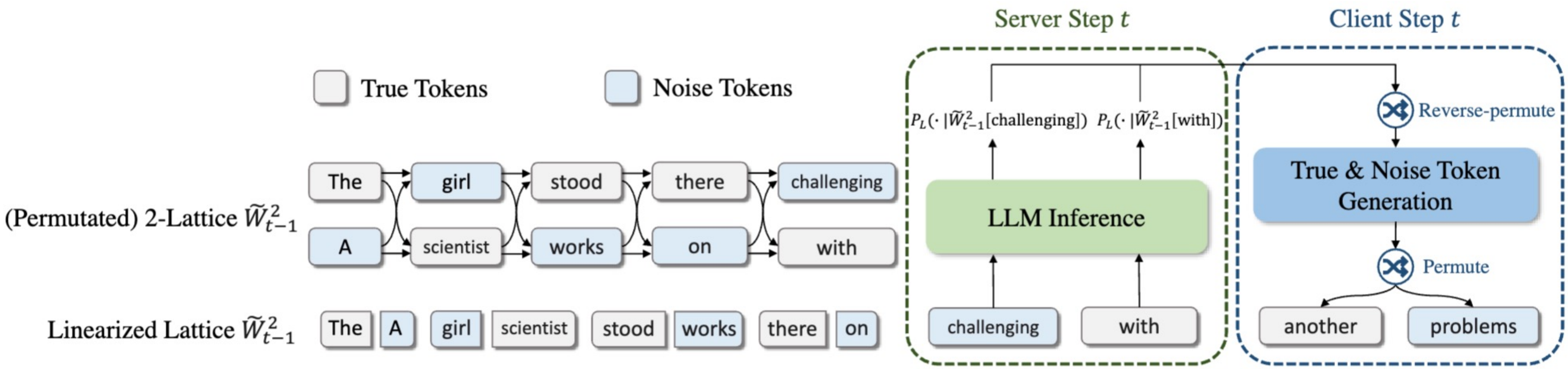
- 1st: The scientist stood there with…
- 2ed: A girl works on challenging…

The true sequence is in one of the 2^T (upperbound) possibilities.

The true sequence is in one of the two possibilities.

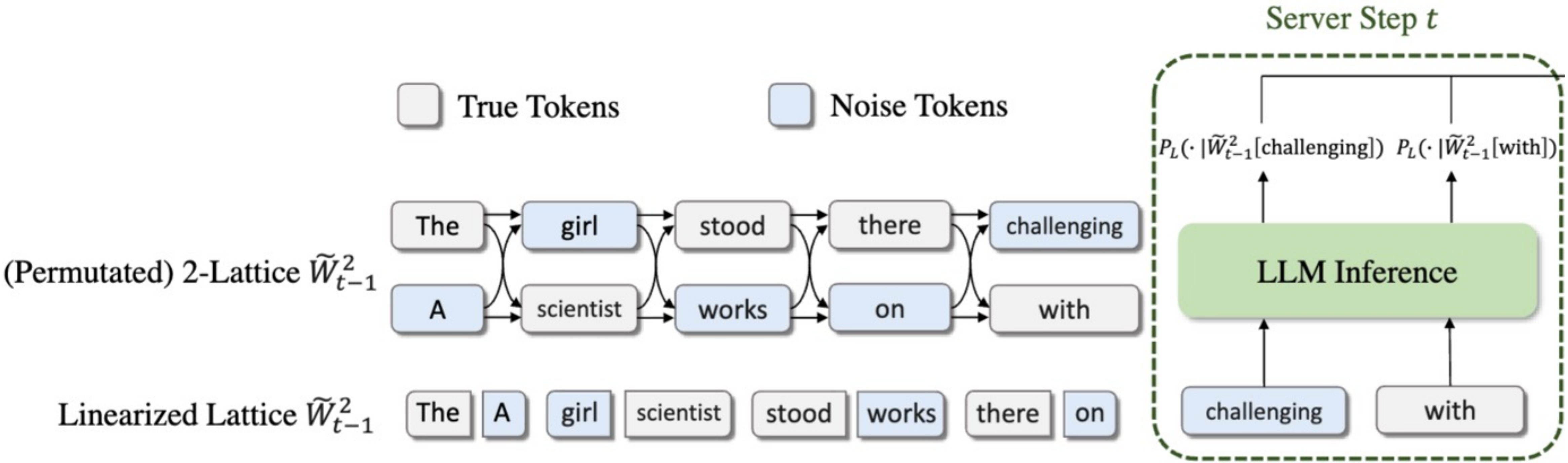
LatticeGen: Overview

- On each time-step, instead of inference/sample one token, the server and user cooperatively inference/sample N tokens. (e.g., N=2)


True Tokens
Noise Tokens
(Permutated) 2-Lattice $\widetilde{W}^2_{t-1}$
The
girl
stood
there
challenging
A
scientist
works
on
with
Linearized Lattice $\widetilde{W}^2_{t-1}$
The
A
girl
scientist
stood
works
there
on
Server Step t
$P_L(\cdot \mid \widetilde{W}^2_{t-1}[\text{challenging}])$
$P_L(\cdot \mid \widetilde{W}^2_{t-1}[\text{with}])$
LLM Inference
challenging
with
Client Step t
Reverse-permute
True & Noise Token Generation
Permute
another
problems

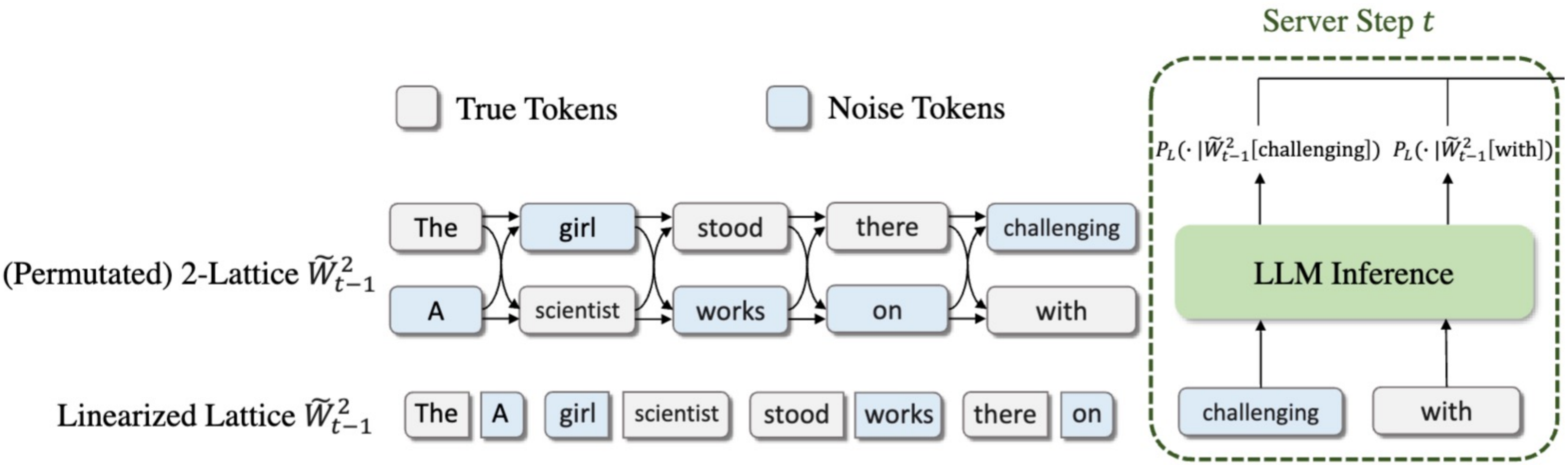
Prerequisite: Inference on a Linearized Lattice

- As a prerequisite of LG, we finetune the LLM to make next-word predictions on the *linearized lattice*.
- $P(\cdot\,|\widetilde{W}_T^2[\widetilde{w}_t^i])$ refers to the next-token prediction distribution on the position of $\widetilde{w}_t^i$.
- Please refer to our paper for how to finetune the LLM to accept this format.


Server Step t
True Tokens
Noise Tokens
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{challenging}])$
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{with}])$
(Permutated) 2-Lattice $\widetilde{W}_{t-1}^2$
The
girl
stood
there
challenging
A
scientist
works
on
with
LLM Inference
Linearized Lattice $\widetilde{W}_{t-1}^2$
The
A
girl
scientist
stood
works
there
on
challenging
with

Generation Protocol: Server Step at t

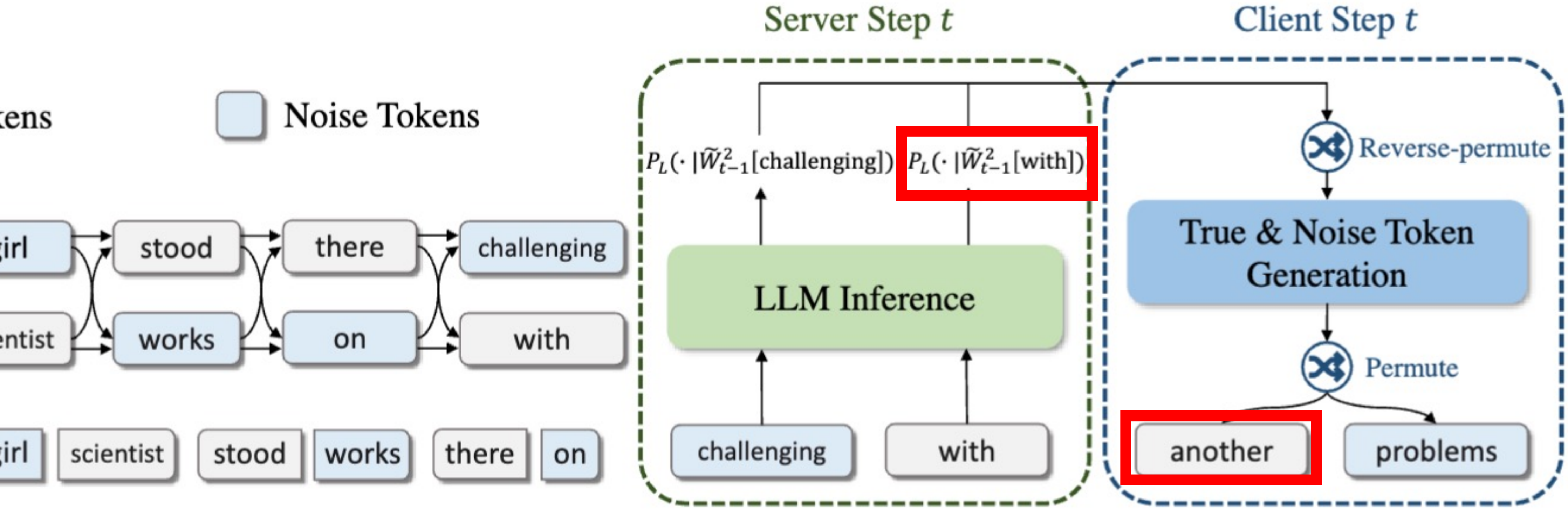
- The server makes inference on all of the N tokens from time-step t-1, and send the prediction distributions as len-|V| vectors to client.



True Tokens
Noise Tokens
Server Step t
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{challenging}])$
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{with}])$
(Permutated) 2-Lattice $\widetilde{W}_{t-1}^2$
The
girl
stood
there
challenging
A
scientist
works
on
with
LLM Inference
Linearized Lattice $\widetilde{W}_{t-1}^2$
The
A
girl
scientist
stood
works
there
on
challenging
with

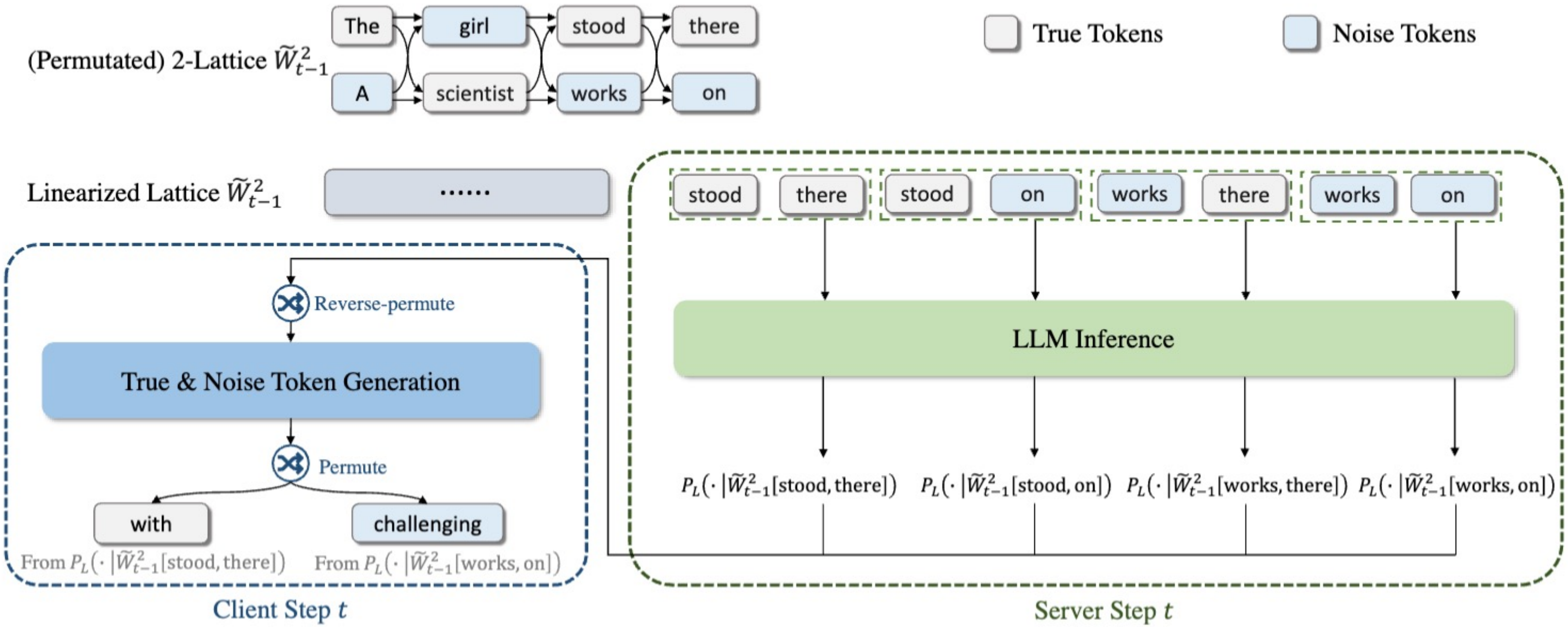
Generation Protocol: Client Step at t (Key Step)

- Upon receiving the two distributions, the user knows which of the previous token is the true one, and generates a true token from it.
- With a noise scheme (e.g., synonym), the client also generates a noise token.
- The user shuffles the two tokens, and send them to server for next time-step.



For Better Quality: Incorporating Bigram Units

The current inference unit is unigram, which degrades gen-quality a lot.
We can extend to bigram units (enumerate N^2combinations) to trade computation for quality.


(Permutated) 2-Lattice $\widetilde{W}_{t-1}^2$
The
girl
stood
there
A
scientist
works
on
True Tokens
Noise Tokens
Linearized Lattice $\widetilde{W}_{t-1}^2$
……
stood
there
stood
on
works
there
works
on
Reverse-permute
True & Noise Token Generation
Permute
with
challenging
From $P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{stood, there}])$
From $P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{works, on}])$
LLM Inference
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{stood, there}])$
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{stood, on}])$
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{works, there}])$
$P_L(\cdot\,|\widetilde{W}_{t-1}^2[\text{works, on}])$
Client Step t
Server Step t

Generation Quality Degradation

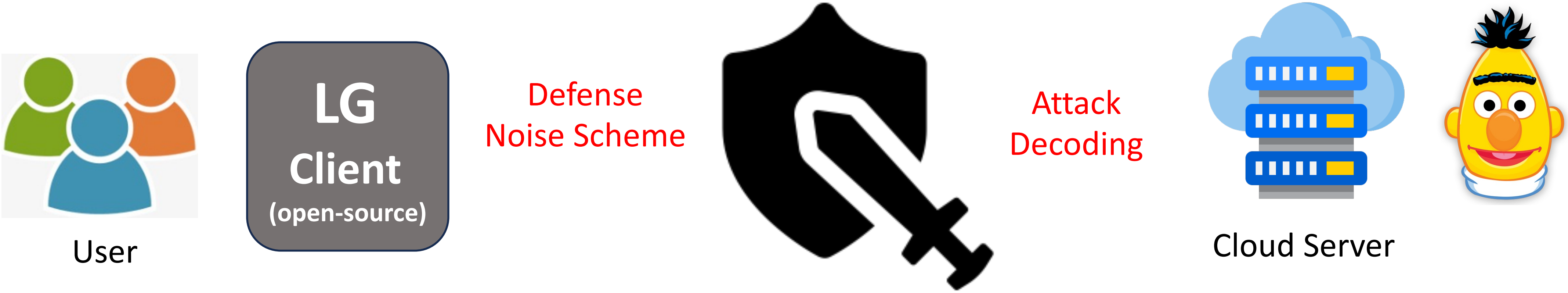
LatticeGen trades quality for protection (<- to be discussed soon).
Directly adding noise to text w.o. lattice would induce drastic degradation.

System	PPL	Protection
standard	28	None
synonym noise, w.o. lattice	229	N/A
LG, unigram, synonym noise	33	Poor
LG, unigram, mixing noise (to be discussed soon)	73	Good
LG, bigram, mixing noise	64	Good

Attack (Server) & Defense (Client): Overview

To defend against a hypothetically malicious server, we will begin a sequence of thought-adversarial game:

- What would a malicious server do to attack?
- How can client/user defense?



User

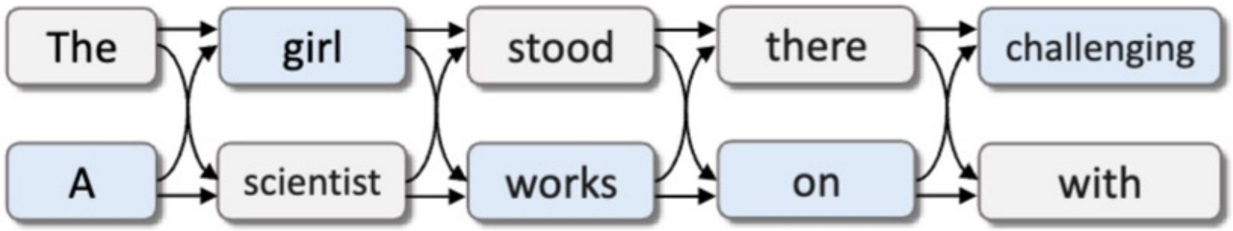
LG Client (open-source)

Defense Noise Scheme

Attack Decoding

Cloud Server

The Beam-Search Attack (Server)



The girl stood there challenging

A scientist works on with

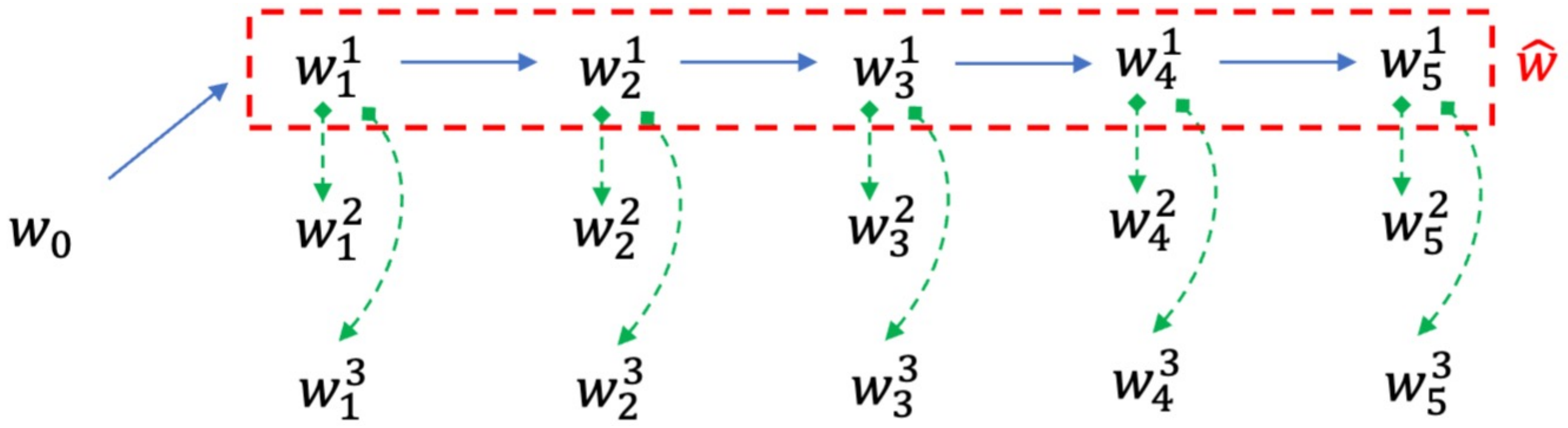
- Knowing that the true token is among the N tokens on each time-step, a nature attack objective is to find the maximizing-prob sequence:

$$\arg\max_{\hat{w}} \log P_L(\hat{w}|\tilde{W}_T^N) = \arg\max_{\hat{w}} \sum_{t=1}^{T} \log P_L(\hat{w}_t|\tilde{W}_{t-1}^N[\hat{w}_{t-1}]).$$

- This can be solved by a simple dynamic programming.

The Beam-Search Attack (Server)

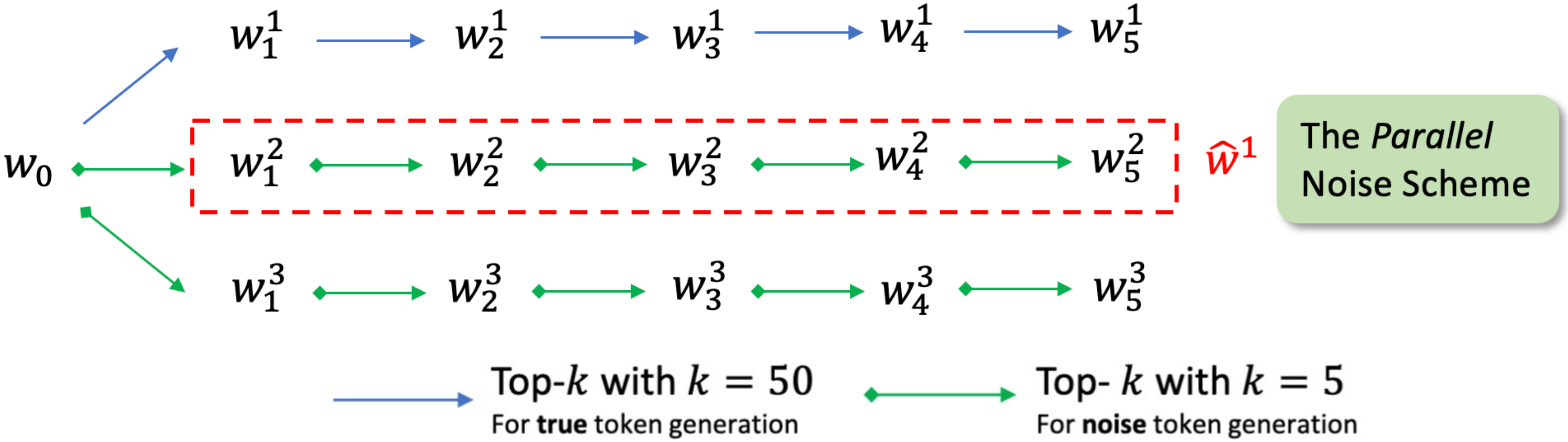
- The BS attack is very effective against the synonym scheme.



The *Synonym* Noise Scheme

The Parallel Noise Scheme (User)

- The user can evade BS by using a parallel scheme with radical sampling.


w_0
w_1^1 w_2^1 w_3^1 w_4^1 w_5^1
w_1^2 w_2^2 w_3^2 w_4^2 w_5^2
$\hat{w}^1$
w_1^3 w_2^3 w_3^3 w_4^3 w_5^3

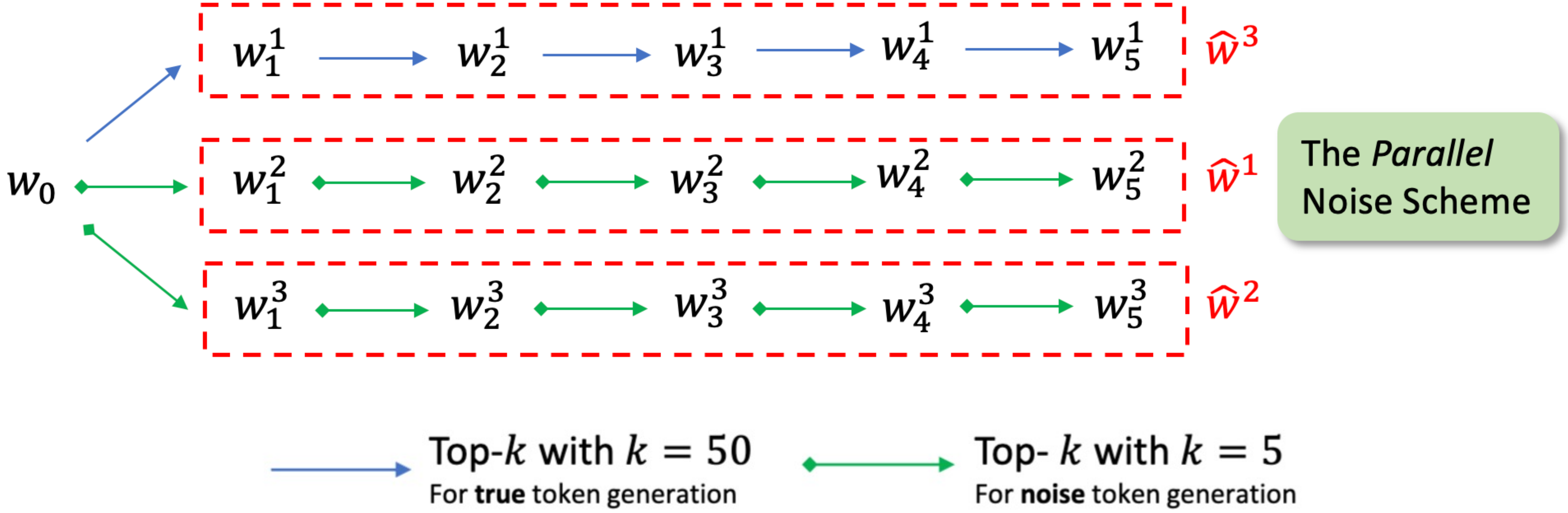
The *Parallel* Noise Scheme

Top-k with $k = 50$
For **true** token generation

Top- k with $k = 5$
For **noise** token generation

The Repeated Beam-Search Attack

- However, the server can repeatedly call beam-search and remove the hypothesis from each call.
- RBS is a stronger version of BS.


w_0
w_1^1 w_2^1 w_3^1 w_4^1 w_5^1
$\widehat{w}^3$
w_1^2 w_2^2 w_3^2 w_4^2 w_5^2
$\widehat{w}^1$
w_1^3 w_2^3 w_3^3 w_4^3 w_5^3
$\widehat{w}^2$
The *Parallel* Noise Scheme
Top-k with $k = 50$
For **true** token generation
Top- k with $k = 5$
For **noise** token generation

Metric of Protection

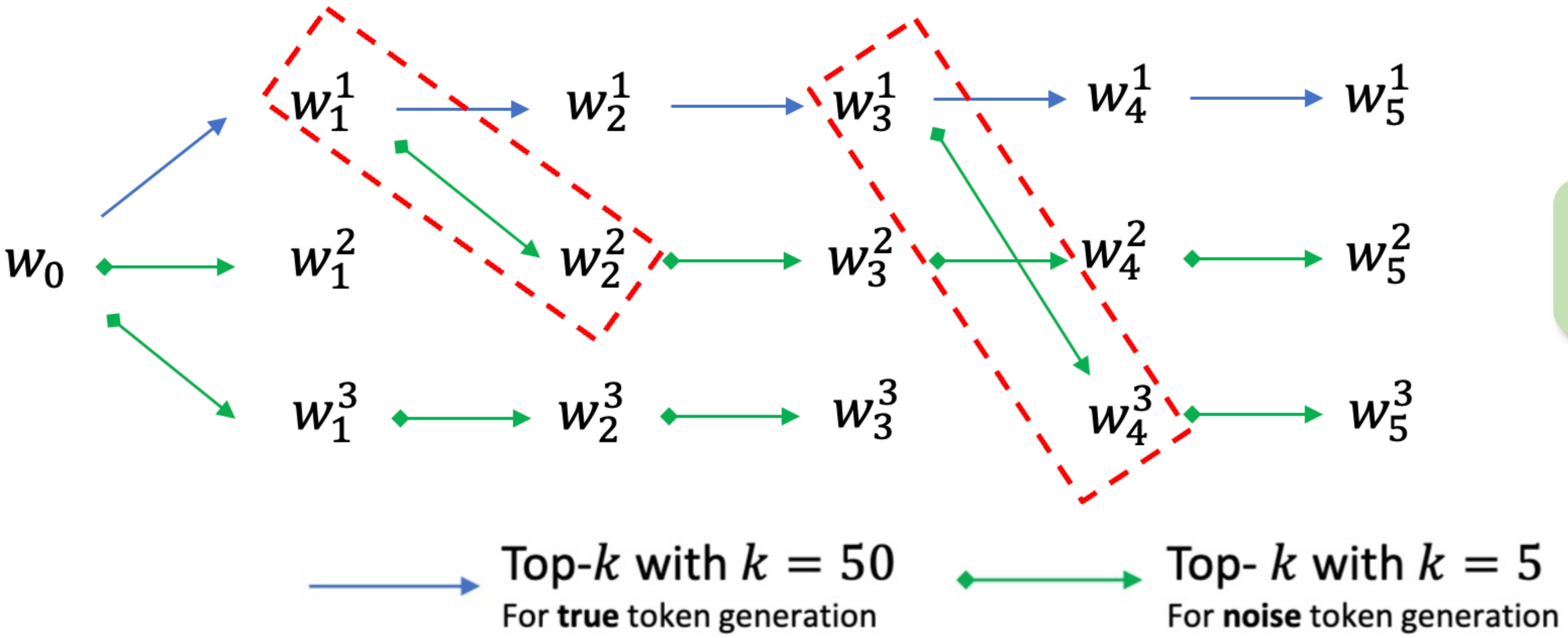
- After RBS, the attacker gets N hypotheses, and we care about the hypothesis with maximum overlap with the true sequence. (the average is always 1/N)

$$\text{max-true-ratio}(\{\hat{w}\}_{i=1}^{N}, w^{1}) = \max_{i} \frac{\sum_{t=1}^{T} \mathbb{1}_{\hat{w}_t^i = w_t^1}}{T}.$$

- The true-ratio only cares about exact match, we also have a BERTScore variant which measures the revealed semantic, which is defined in a similar manner.

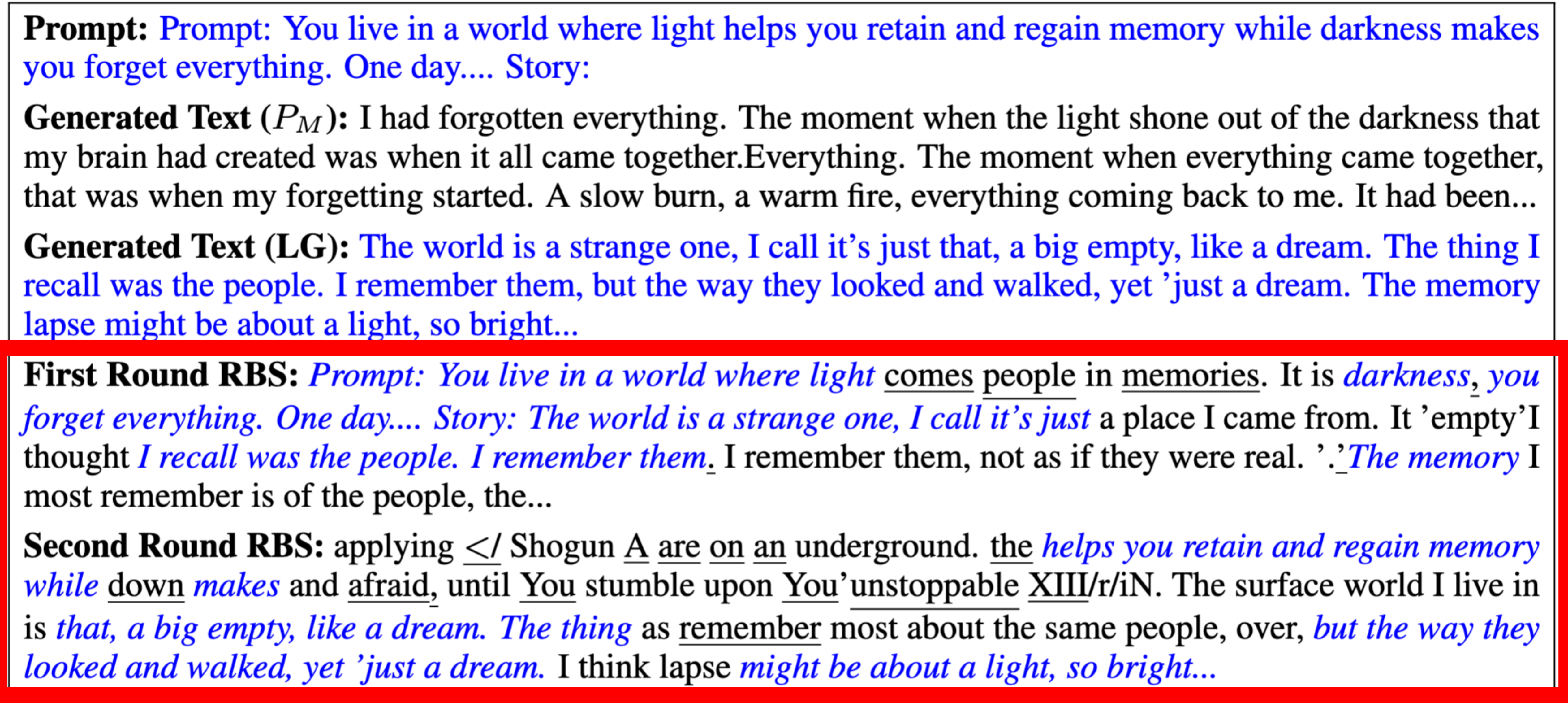
Defense Against RBS: The Mixing Scheme

- Under RBS, we realize that the true and noise sequence have to be mixed together.
- With a *mix-ratio (0.1),* we achieve this by randomly branching from the true sequence into the noise sequences.



The *Mixing* Noise Scheme

Mixing Scheme Example



Prompt: Prompt: You live in a world where light helps you retain and regain memory while darkness makes you forget everything. One day.... Story:

Generated Text (P_M): I had forgotten everything. The moment when the light shone out of the darkness that my brain had created was when it all came together.Everything. The moment when everything came together, that was when my forgetting started. A slow burn, a warm fire, everything coming back to me. It had been...

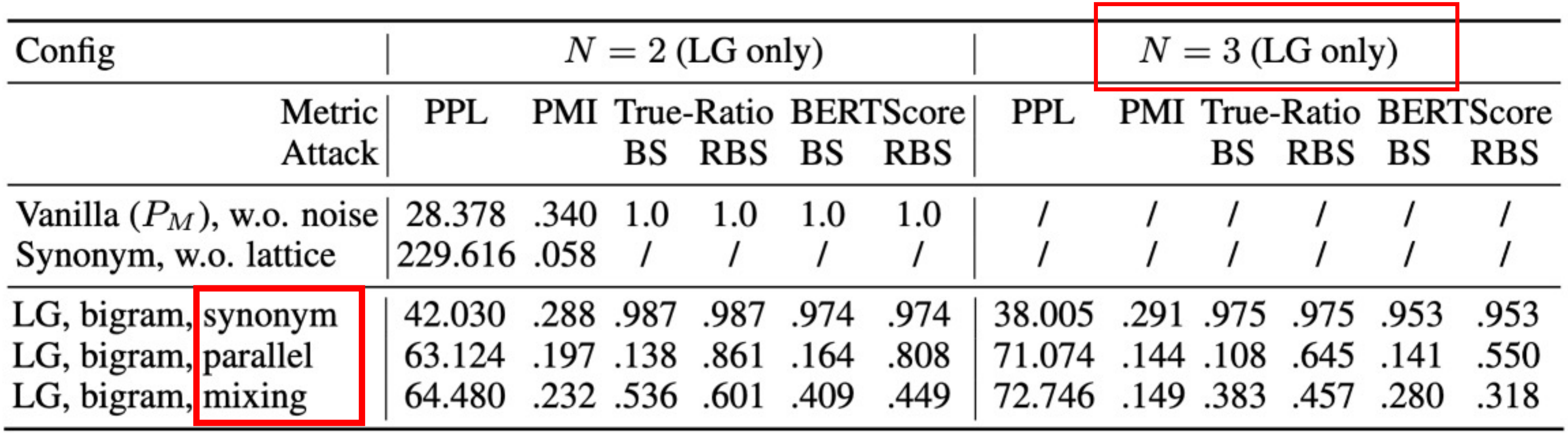
Generated Text (LG): The world is a strange one, I call it’s just that, a big empty, like a dream. The thing I recall was the people. I remember them, but the way they looked and walked, yet ’just a dream. The memory lapse might be about a light, so bright...

First Round RBS: *Prompt: You live in a world where light* comes people in memories. It is *darkness*, *you forget everything. One day.... Story: The world is a strange one, I call it’s just* a place I came from. It ’empty’I thought *I recall was the people. I remember them*. I remember them, not as if they were real. ’.’*The memory* I most remember is of the people, the...

Second Round RBS: applying </ Shogun A are on an underground. the *helps you retain and regain memory while* down *makes* and afraid, until You stumble upon You’unstoppable XIII/r/iN. The surface world I live in is *that, a big empty, like a dream. The thing* as remember most about the same people, over, *but the way they looked and walked, yet ’just a dream.* I think lapse *might be about a light, so bright...*

Results

- The synonym scheme is good for utility, but bad under BS/RBS.
- The parallel scheme is good for BS, but bad under RBS.
- The proposed mixing scheme achieves best protection.



Config	$N = 2$ (LG only)						$N = 3$ (LG only)					
Metric Attack	PPL	PMI	True-Ratio		BERTScore		PPL	PMI	True-Ratio		BERTScore	
			BS	RBS	BS	RBS			BS	RBS	BS	RBS
Vanilla (P_M), w.o. noise	28.378	.340	1.0	1.0	1.0	1.0	/	/	/	/	/	/
Synonym, w.o. lattice	229.616	.058	/	/	/	/	/	/	/	/	/	/
LG, bigram, synonym	42.030	.288	.987	.987	.974	.974	38.005	.291	.975	.975	.953	.953
LG, bigram, parallel	63.124	.197	.138	.861	.164	.808	71.074	.144	.108	.645	.141	.550
LG, bigram, mixing	64.480	.232	.536	.601	.409	.449	72.746	.149	.383	.457	.280	.318

Other Work and Interests

- Detection is hard! Especially in zero-shot cases [1] or under attacks (on-going).

- Designing stress tests for LLM-based NLG metrics (ACL 2023).

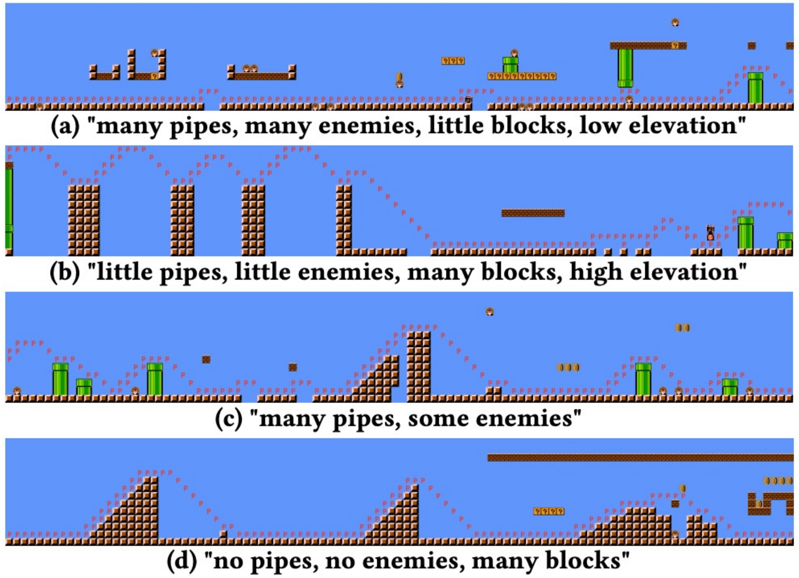
On the

Blind Spots

of Model-Based Evaluation Metrics for Text Generation

[1] On the Zero-Shot Generalization of Machine-Generated Text Detectors, (Sophia) Xiao Pu et al. EMNLP-Finding 2023

- Interest: Can powerful model help video game development (e.g., marioGPT)?



(a) "many pipes, many enemies, little blocks, low elevation"

(b) "little pipes, little enemies, many blocks, high elevation"

(c) "many pipes, some enemies"

(d) "no pipes, no enemies, many blocks"

Figure 1: Prompt-conditioned generations from a single seed block. MarioGPT is able to create diverse levels solely based on a text prompt in natural language.

Thanks! Questions?

goosehe@cs.washington.edu