Extending Logits-Based LLM Watermarking Schemes to Mitigate Stealing Attacks

Mikko Tripakis | CS 4973 Trustworthy Generative AI

Attributing LLM-Generated Content

‘Nobody is blind to it’: mass cheating
through AI puts integrity of Australi
universities at risk, academics claim

Washington Post
https://www.washingtonpost.com › ... › Opinion Columns

Opinion | AI has arrived. So have the AI charlatans.

Jan 29, 2024 — Then as now, regulators pursued cases of **fraud** and false claims
have been illegal wherever they happened in the economy but appeared ...

World's biggest music labels sue over AI copyright

25 June 2024

Share Save +

Almost all bad use of LLMs involves hiding the fact that an LLM was used

Another Reason for Attribution

AI-generated content is starting to fill up the internet

We want to avoid training on machine-generated text

Previous Work

SoA methods embed a hidden signal into LLM-generated text [1,2,3,4]

- For each generated token: aggregate and hash the last k tokens
- Seed PRF with hash+secret key, partition vocabulary into red/green lists
- Induce model to generate more green tokens and less red tokens
- Detect watermark through a statistical test on the ratio of green-list tokens in the output

The key to detecting the watermark is knowing the PRF seed

Previous work always uses a fixed context width k ← stealing attacks

1. J. Kirchenbauer et al., "A Watermark for Large Language Models," 2023, arXiv.
2. S. Dathathri et al., "Scalable watermarking for identifying large language model outputs," 2024, Nature, vol. 634, no. 8035.
3. X. Zhao, P. Ananth, L. Li, and Y.-X. Wang, "Provable Robust Watermarking for AI-Generated Text," 2023, arXiv.
4. "Watermarking of Large Language Models" - Talk by Scott Aaronson, UT Austin/OpenAI

Watermark Stealing - Threat Model

- Goal: extract watermark rules from an LLM
  - Enables spoofing attacks (fake the watermark)
  - And scrubbing attacks (remove the watermark)
- Knowledge: black-box on the LLM, white-box on the watermark
- Capabilities: query access to watermarked and unwatermarked LLM

N. Jovanović, R. Staab, and M. Vechev, “Watermark Stealing in Large Language Models,” 2024, arXiv

Watermark Stealing - Method

- Maintain empirical estimates of the conditional distributions of tokens for both watermarked and unwatermarked text
- Compute context scores to identify tokens that are likely to be in the green list
- Requires $O(10^4)$ LLM queries for high success rate

Claim: knowledge of context width k is important for attacker success

$$\frac{1}{z}\left[s(T,\{T_1,T_2,T_3\}) + w_1 \cdot s(T,\{T_{\text{min}}\}) + w_2 \cdot s(T,\{\})\right],$$

Attacker: assumes k=3

Proposed Mitigation: Variable Context Width

Intuition: make it harder to guess watermarking rules by adding randomness

- One idea: rotate secret keys
    - Increased overhead in detecting the watermark (need to try each key)
- Proposed idea:
    - Pseudorandomly vary the context width used to seed PRF

Algorithm 1: Text Generation with Variable-Context Watermark

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

Input: green list size, $\gamma \in (0, 1)$, hardness parameter, $\delta > 0$

Input: secret key ξ for the hash function

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. Pseudorandomly compute the context width k using ξ and the current tokens s: $k = PRF(s, \xi)$.
3. Compute a hash of the previous k tokens, $s^{(t-k)} \dots s^{(t)}$, and use it to seed a random number generator.
4. 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 $(1-\gamma)|V|$.
5. Add δ to each green list logit. Apply the softmax operator to these modified logits to get a probability distribution over the vocabulary:

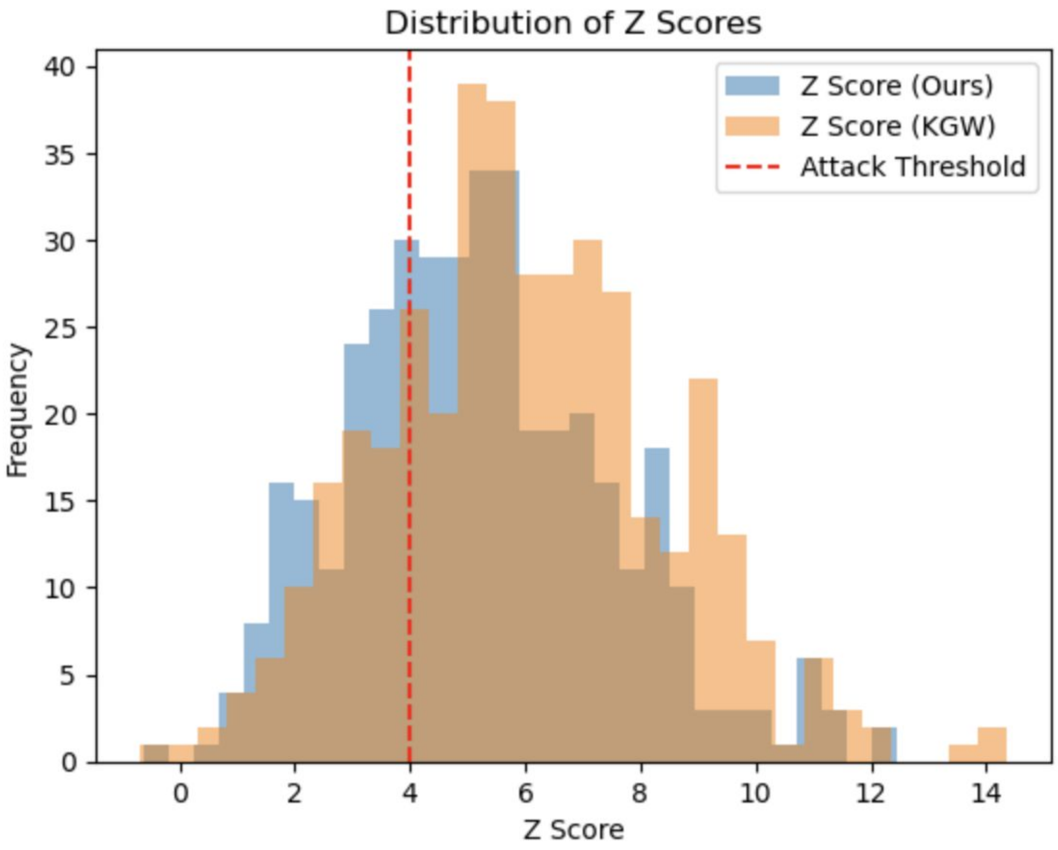
$$\hat{p}_k^{(t)} = \begin{cases} \frac{\exp(l_k^{(t)}+\delta)}{\sum_{i\in R}\exp(l_i^{(t)})+\sum_{i\in G}\exp(l_i^{(t)}+\delta)}, & k \in G \\ \frac{\exp(l_k^{(t)})}{\sum_{i\in R}\exp(l_i^{(t)})+\sum_{i\in G}\exp(l_i^{(t)}+\delta)}, & k \in R \end{cases}$$

6. Sample the next token, $s^{(t)}$, using the watermarked distribution $\hat{p}^{(t)}$.

Evaluation

- Dataset: C4-Real Newslike
- Watermarked Model: LLama-2 7B
- Stealing Attack:
  - Attempts to spoof watermark
  - Attacker model: Gemma 2B-Instruct
  - Makes 12,600 queries to watermarked model
- Metrics (+ Comparison to Prior Work)
  - Attack success rate (Z-scores of attack)
  - Perplexity of spoofed text
  - Perplexity of watermarked text

Results: Spoofing Attack



Results: Spoofing Attack

```
Mean Attack Success Rate (Ours): 68.43%
Mean Attack Success Rate (KGW): 76.77%
Worst Attack Success Rate (Ours): 62.63%
Worst Attack Success Rate (KGW): 74.75%
```

Ours

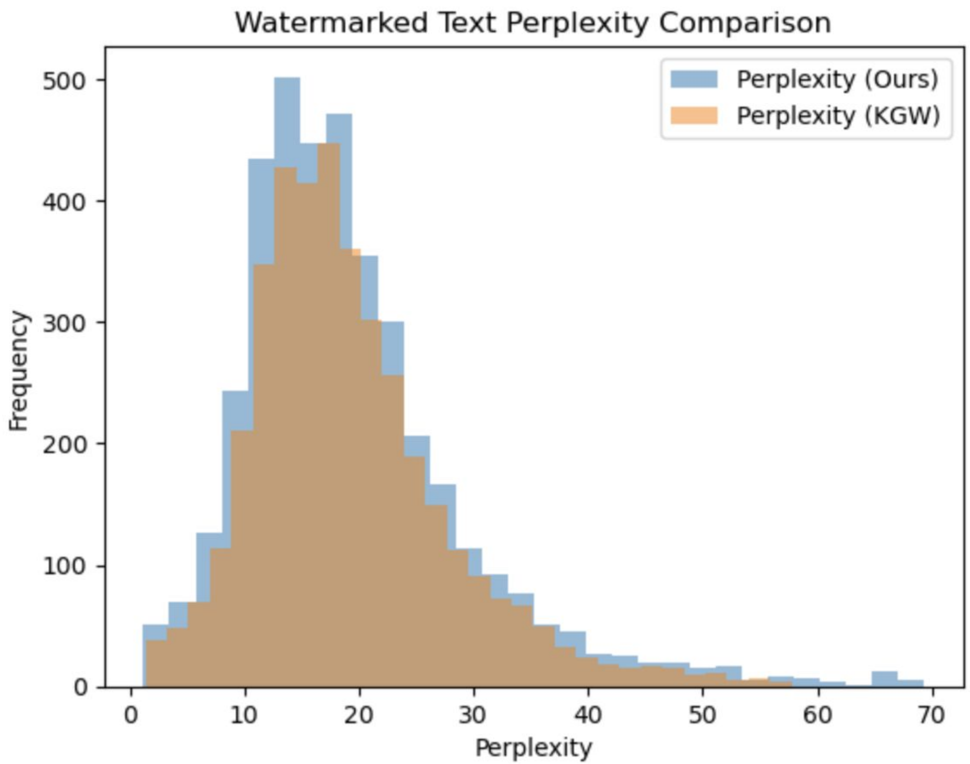
	Perplexity	Z_score
count	396.000	396.000
mean	10.910	5.233
std	9.835	2.301
min	3.170	-0.630
25%	6.810	3.590
50%	8.570	5.080
75%	11.860	6.730
max	107.310	12.440

KGW

	Perplexity	Z_score
count	396.000	396.000
mean	9.604	5.985
std	3.231	2.480
min	3.710	-0.700
25%	7.250	4.238
50%	9.170	5.825
75%	11.530	7.463
max	18.140	14.370

Ablation Study: Text Quality

```
Mean Perplexity (Ours): 19.35
Mean Perplexity (KGW): 19.10
```



Future Work

- More comprehensive evaluation
    - Different watermarks
    - More attacks
- How else can we beat stealing attacks?

References

J. Kirchenbauer et al., “A Watermark for Large Language Models,” 2023, arXiv. doi: 10.48550/ARXIV.2301.10226.

S. Dathathri et al., “Scalable watermarking for identifying large language model outputs,” Nature, vol. 634, no. 8035. Springer Science and Business Media LLC, pp. 818–823, Oct. 23, 2024. doi: 10.1038/s41586-024-08025-4.

X. Zhao, P. Ananth, L. Li, and Y.-X. Wang, “Provable Robust Watermarking for AI-Generated Text,” 2023, arXiv. doi: 10.48550/ARXIV.2306.17439.

“Watermarking of Large Language Models” - Talk by Scott Aaronson, https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17

N. Jovanović, R. Staab, and M. Vechev, “Watermark Stealing in Large Language Models,” 2024, arXiv. doi: 10.48550/ARXIV.2402.19361.
J. Kirchenbauer et al., “On the Reliability of Watermarks for Large Language Models,” 2023, arXiv. doi: 10.48550/ARXIV.2306.04634.

Z. Zhang et al., “Large Language Model Watermark Stealing With Mixed Integer Programming,” 2024, arXiv. doi: 10.48550/ARXIV.2405.19677.