A Nested Watermark for Large Language Models

Anonymous ACL submission

Abstract

The rapid development of large language models (LLMs) has raised concerns about the potential misuse of these models for generating fake news and misinformation. To mitigate this risk, watermarking techniques for auto-regressive language models have been proposed as a means of detecting text generated by LLMs. However, this method assumes that the target text, which is watermarked, contains a sufficient number of tokens, and the detection accuracy decreases as the number of tokens in the text becomes smaller. To address this issue, we introduce a novel nested watermark that embeds two watermarks in a nested structure. Our method ensures that high detection accuracy can be achieved even with fewer tokens compared to conventional approaches. Our experiments show that the nested watermark outperformed the single watermark in terms of embedding success ratio and text quality when dealing with short text.

1 Introduction

Large language models (LLMs) have made significant advancements in recent years, enabling the generation of high-quality text that is often indistinguishable from human-written content (Achiam et al., 2023). However, this remarkable ability has also raised concerns about the potential misuse of LLMs for creating and spreading fake news and misinformation (Crothers et al., 2023). To address this issue, researchers have proposed various methods to detect text generated by LLMs (Mitchell et al., 2023; Jawahar et al., 2020).

One such method is a watermark for LLMs, which embeds specific token patterns into the generated text, allowing for the identification of the text's source (Kirchenbauer et al., 2023a). This approach increases the probabilities of tokens included in a specific vocabulary, which is constructed based on a *key*. While this method has shown promise, its detection relies on statistical test, which assumes that the target text for watermarking contains a sufficient number of tokens. Therefore, as the number tokens in the text decreases, the detection accuracy of the watermark also declines, posing a challenge for short text (Sadasivan et al., 2023; Krishna et al., 2024).

To overcome this limitation, we propose a novel *nested watermark* that embeds two watermarks in a nested structure. Our method aims to achieve high detection accuracy even when the target text is short segments (from 50 to 100 tokens) commonly found in social media posts and other applications. By leveraging the nested structure, we can effectively embed watermarks in short text segments with less error rates, enhancing the efficiency of the watermarking process. Furthermore, we introduce a pseudo-instruction dataset that closely resembles real-world user prompts to assess the quality of the generated text under realistic input conditions. By evaluating the nested watermark using this dataset, we can accurately evaluate its performance in practical applications.

The main contributions of this paper are as follows:

- We introduce a novel nested watermark that mitigates the limitations of single watermarks in detecting LLM-generated text, particularly for short text segments.
- We demonstrate that our approach improves the *embedding success ratio* (ESR) for text segments under 100 tokens while preserving text quality, using a pseudo-instruction dataset that emulates real-world user prompts.
- Additionally, our nested watermark ensures that a portion of the source of the generated text can still be identified even if the first key is compromised, thereby enhancing the security and robustness of the watermarking process.

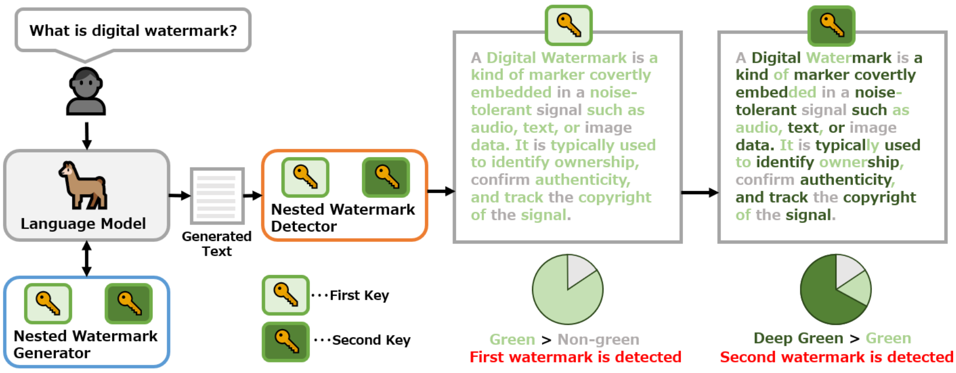

Figure 1: An overview of our nested watermark. The text on the right side of the figure demonstrates the detection of the first and second watermarks using the first and second keys, respectively. In the first text detected by the first key, the gray parts represent tokens classified as belonging to the token group without increased probabilities, while the light green parts indicate tokens classified as having increased probabilities. Furthermore, in the second text detected by the second key, the dark green parts signify tokens that belong to the group with increased probabilities during the embedding of the second watermark.

2 Related Work

The concept of embedding watermarks in text has been extensively explored long before the emergence of large language models (LLMs) (Kamaruddin et al., 2018; Atallah et al., 2001; Brassil et al., 1994). One of the key advantages of watermarks designed for LLMs is their high robustness against text tampering, as demonstrated by (Kirchenbauer et al., 2023b). However, despite their resilience, the detection accuracy of watermarks significantly deteriorates when subjected to paraphrase attacks. (Sadasivan et al., 2023).

(Zhu et al., 2024) proposed Duwak, a dual watermarking scheme for large language models that embeds secret patterns in both the token probability distribution and sampling scheme using two keys, similar to our method; however, our approach is distinctive in that it does not require access to the model parameters in detection for the second watermark.

3 Method

Figure 1 shows the overall structure of the proposed method when the number of nested watermarks is two. The proposed method consists of a nested watermark generator, nested watermark detector, and multiple different keys. In the nested watermark generator, while interacting with the language model that generates text according to the prompt, it embeds nested watermarks using multiple keys. The nested watermark detector receives the text generated by the language model and determines the presence or absence of each watermark from the multiple keys. In the following sections, we discuss the details of the nested watermark generator and nested watermark detector.

3.1 Nested Watermark Embedding

Let w_t be the t-th token in the text, and p_t^k be the probability of the k-th token in the vocabulary V at the t-th step. The probability p_t^k is calculated using the softmax function:

$$p_t^k = \frac{\exp(l_t^k)}{\sum_{i=1}^{|V|} \exp(l_t^i)} \quad (1)$$

where l_t^k is the logit of the k-th token in the vocabulary V at the t-th step.

We define a hash function, H, that map the concatenation of a token w_{t-n} at the $(t-n)$-th step and a secret key s_1 to a random number r_1, and the concatenation of a token w_{t-m} at the $(t-m)$-th step and a secret key s_2 to a random number r_2, where $m \neq n$:

$$r_1 = H(w_{t-n}, s_1) \quad (2)$$

$$r_2 = H(w_{t-m}, s_2) \quad (3)$$

The random numbers r_1 and r_2 are used to determine the token groups G_1 and G_2, respectively. G_1 is a subset of the vocabulary V, and G_2 is a subset of G_1. The ratio of the size of G_1 to the size of R_1 (the remaining tokens in the vocabulary) is $\gamma : (1-\gamma)$, where γ is a hyperparameter.

To embed the watermarks, we add biases δ_1 and δ_2 to the logits of the tokens in G_1 and G_2, respectively. The total sum of the exponential of the logits, D_{total}, is calculated as follows:

$$D_{total} = \sum_{i \in G_1, i \notin G_2} \exp(l_t^i + \delta_1) + \sum_{i \in R_1} \exp(l_t^i) + \sum_{i \in G_2} \exp(l_t^i + \delta_1 + \delta_2) \quad (4)$$

The adjusted probabilities for the tokens in G_1 and G_2 are then calculated as:

$$\hat{p}_t^k = \frac{\exp(l_t^k + \delta_1)}{D_{total}}, \quad k \in G_1, k \notin G_2 \quad (5)$$

$$\hat{p}_t^k = \frac{\exp(l_t^k + \delta_1 + \delta_2)}{D_{total}}, \quad k \in G_2 \quad (6)$$

3.2 Nested Watermark Detection

To detect the presence of the watermarks (G_1 and G_2) in the text, we calculate the counts c_1 and c_2 of the tokens belonging to G_1 and G_2, respectively. We then compute the z-scores z_1 and z_2 as follows:

For the first watermark:

$$z_1 = \frac{c_1 - \gamma T}{\sqrt{T\gamma(1-\gamma)}} \quad (7)$$

where T is the total number of tokens in the text.

For the second watermark:

$$z_2 = \frac{c_2 - \gamma c_1}{\sqrt{c_1\gamma(1-\gamma)}} \quad (8)$$

If the z-scores z_1 and z_2 exceed a predetermined threshold θ, we conclude that the watermarks are present in the text.

Following the detection method proposed by Zhu (Zhu et al., 2024) using Fisher's method, we also combine the p-values (P_1 and P_2) from the two independent tests for our nested watermarks into a single statistic that follows a chi-square (χ^2) distribution with $d = 4$ degrees of freedom:

$$-2(\ln(P_1) + \ln(P_2)) \sim \chi^2(4). \quad (9)$$

Furthermore, the resulting p-value P_F, derived from the chi-square distribution, is given as:

$$P_F = 1 - F_{\chi^2}(-2(\ln(P_1) + \ln(P_2)), 4), \quad (10)$$

where F_{χ^2} represents the cumulative distribution function (cdf) for the chi-square distribution, providing a unified statistical metric to evaluate the existence of watermarks in the text.

Bias(δ)	Win(%)	Lose(%)	Tie(%)	diff.
4.0	**19.00**	16.10	64.90	2.90
3.5	**18.70**	16.05	65.25	2.65
3.0	**17.70**	16.50	65.80	1.20

Table 1: Win, lose, and tie rates of the proposed method compared to the single watermark baseline for different values of the bias term (δ). The last column shows the difference between the win rate and the lose rate.

4 Experiment

4.1 Experimental Setup

To evaluate the effectiveness of the proposed nested watermark, we conducted experiments using Llama-2-7b-chat[1]. These experiments were performed with varying maximum output token counts, ranging from 50 to 100. For the nested watermark, we set the hyperparameters as follows: $\gamma = 0.5$, $\delta_1 = 1.5$, and $\delta_2 = 2.5$. The detection threshold θ was set to 4.0 as in (Kirchenbauer et al., 2023a). For the evaluation dataset, we generated 1,000 samples of an English instructions using GPT-4. This dataset consists of pseudo-prompts generated based on topics that reflect real-world use cases where LLMs are employed, such as news articles and social media posts. In contrast, previous work (Kirchenbauer et al., 2023a) focuses on text completion tasks, where the prompts used during inference are composed of fragmented texts sampled from C4 dataset. By employing our dataset, we can evaluate the proposed method in a setting that more closely resembles actual generation scenarios.

4.2 Evaluation Metric

Embedding Success Ratio (ESR) The detection accuracy of watermarks is commonly measured by Type II Error, which indicates the precision of watermark detection in a single embedding process. However, in practical applications, it is assumed that the detection is performed immediately after embedding, and if the embedding fails, the process is repeated until the detection succeeds, effectively reducing the Type II Error to zero. Based on this assumption, we introduce a new metric called the Embedding Success Ratio (ESR). ESR represents the proportion of successful watermark embeddings in a single attempt (ESR is equal to the reciprocal of Type II Error).

[1] https://huggingface.co/meta-llama/Llama-2-7b-chat

	Embedding success ratio (ESR %)					Text quality (win rate %)		
	Nested watermark (ours)			Single (baseline)				
Length	Watermark I	Watermark II	Unified	Watermark I	diff.	Nested	Single	diff.
100	84.6	56.8	**95.4**	93.3	2.1	27.30	**28.35**	-1.05
90	77.4	48.1	**92.7**	88.8	3.9	**27.40**	26.95	0.45
80	69.4	41.6	**87.4**	83.8	3.6	**25.70**	24.40	1.30
70	58.0	32.7	**82.7**	74.1	8.6	**23.15**	22.95	0.20
60	53.4	25.3	**75.4**	68.6	6.8	**20.95**	20.55	0.40
50	37.5	16.2	**63.5**	52.1	11.4	**18.70**	16.05	2.65

Table 2: Comparison of the proposed method and the baseline for varying text lengths. In the ESR comparison, the proposed method shows the individual accuracy of the first and second watermarks, as well as the accuracy when both detection results are combined using Fisher's method (Unified). For the text quality comparison, the win rates of each method are presented, excluding the instances judged as ties by GPT-4.

Text Quality To quantitatively evaluate the impact of watermark embedding on text quality, we employ the automatic evaluation method called LLM-as-a-judge (Zheng et al., 2024), which utilizes GPT-4 (gpt-4-1106-preview). By using LLM-as-a-judge, we can comprehensively assess not only the grammatical mistakes caused by watermarking but also how the watermarks affect the model's ability to provide semantically relevant responses to instructions. It is crucial to acknowledge that LLM-as-a-judge shows positional bias, influenced by the order of presented texts. To counteract this, we conduct two comparisons per example with swapped text orders and report the average result.

4.3 Preliminary Experiment

To determine the bias δ for the single watermark baseline, we conducted a preliminary experiment. As shown in Table 1, we compared text quality of the proposed method ($\delta_1 = 1.5$, and $\delta_2 = 2.5$) and single watermark baseline for three different values of bias δ, while maintaining the text length at 50. The experimental results show that when the bias term is high, such as $\delta = 3.5$ or 4.0, the text quality of the baseline significantly deteriorates compared to the case where $\delta = 3.0$. Even at $\delta = 3.0$, the proposed method slightly outperforms the baseline by 1.2%. However, considering that higher bias values lead to better ESR, for the remaining experiments, we adopt $\delta = 3.5$ as the baseline, where the proposed method's text quality is sufficiently superior.

4.4 Results

Table 2 presents a comparison of the embedding success ratio (ESR) and text quality between the proposed method and the baseline. In terms of ESR, the proposed method outperforms the baseline across all text lengths. The performance gap is most significant at the shortest length of 50, with a difference of more than 11 percentage points. On the other hand, as the length increases, the performance difference narrows. Within the proposed method, the first watermark achieves a higher ESR compared to the second watermark. When the length reaches 100, the second watermark alone enables detection in more than half of the samples. This finding indicates that the inclusion of a second watermark enhances the robustness and security of the watermarking scheme, providing a fallback mechanism even if the key for the first watermark is compromised.

Regarding text quality, the proposed method demonstrates performance on par with or superior to the baseline for all lengths, except for length of 100. The difference is most pronounced at a length of 50, with a 2.65 percentage point advantage for the proposed method. Similar to the observations in the ESR comparison, the quality difference tends to diminish as the length increases.

5 Conclusion

In this paper, we proposed a novel nested watermark which mitigates the limitations of single watermarks, particularly in scenarios involving short text segments. The nested watermark achieves a higher ESR while maintaining the quality of the generated text, as demonstrated through comprehensive experiments. Future research directions include investigating the performance of the nested watermark under adversarial settings, such as intentional attacks aimed at removing the watermarks.

6 Limitations

While our proposed nested watermark approach demonstrates promising results in terms of detection accuracy and text quality preservation, there are certain limitations to our study that should be acknowledged. Firstly, we employ the LLM-as-a-judge evaluation metric to assess the quality of the generated text. Based on this metric, our experimental results suggest that the proposed method achieves a higher ESR while maintaining text quality comparable to or better than the baseline. However, it is important to note that the evaluations performed by GPT-4 may not always align with human judgments. This discrepancy could potentially impact the reliability of the text quality assessment. Moreover, the text samples used in our experiments consist of extremely short token sequences and fragments truncated at a maximum length. This poses challenges in accurately evaluating the text quality, as the limited context may hinder the ability to make meaningful comparisons. This is evident in the case of length=50, where the tie rate is approximately 65% (tie rate indicates instances where the text quality cannot be clearly distinguished). This high tie rate suggests that clear differences in text quality are difficult to observe in such short sequences.

To address these limitations, future research should focus on conducting more rigorous evaluations of text quality. This can be achieved by involving multiple human evaluators and establishing clear evaluation criteria for different aspects of the text. By incorporating human judgments and defining specific evaluation dimensions, we can obtain a more comprehensive and reliable assessment of the text quality.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mikhail J Atallah, Victor Raskin, Michael Crogan, Christian Hempelmann, Florian Kerschbaum, Dina Mohamed, and Sanket Naik. 2001. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding: 4th International Workshop, IH 2001 Pittsburgh, PA, USA, April 25–27, 2001 Proceedings 4*, pages 185–200. Springer.

Jack Brassil, Steven H. Low, Nicholas F. Maxemchuk, and Lawrence O'Gorman. 1994. Electronic marking and identification techniques to discourage document copying. *Proceedings of INFOCOM '94 Conference on Computer Communications*, pages 1278–1287 vol.3.

Evan Crothers, Nathalie Japkowicz, and Herna L Viktor. 2023. Machine-generated text: A comprehensive survey of threat models and detection methods. *IEEE Access*.

Ganesh Jawahar, Muhammad Abdul-Mageed, and Laks Lakshmanan, V.S. 2020. Automatic detection of machine generated text: A critical survey. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2296–2309.

Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. 2018. A review of text watermarking: theory, methods, and applications. *IEEE Access*, 6:8011–8028.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 17061–17084.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the reliability of watermarks for large language models. *ArXiv*, arXiv preprint arXiv:2306.04634.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2024. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in Neural Information Processing Systems*, 36.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. 2023. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International Conference on Machine Learning*, pages 24950–24962. PMLR.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2023. Can ai-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Chaoyi Zhu, Jeroen Galjaard, Pin-Yu Chen, and Lydia Y. Chen. 2024. Duwak: Dual watermarks in large language models. *arXiv preprint arXiv:2403.13000*.