A Nested Watermark for Large Language Models

Koichi Nagatsuka
Hitachi, Ltd.
koichi.nagatsuka.mq@hitachi.com

Terufumi Morishita
Hitachi, Ltd.
terufumi.morishita.wp@hitachi.com

Yasuhiro Sogawa
Hitachi, Ltd.
yasuhiro.sogawa.tp@hitachi.com

Abstract

The rapid advancement of large language models (LLMs) has raised concerns regarding their potential misuse, particularly in generating fake news and misinformation. To address these risks, watermarking techniques for autoregressive language models have emerged as a promising means for detecting LLM-generated text. Existing methods typically embed a watermark by increasing the probabilities of tokens within a group selected according to a single secret key. However, this approach suffers from a critical limitation: if the key is leaked, it becomes impossible to trace the text's provenance or attribute authorship. To overcome this vulnerability, we propose a novel nested watermarking scheme that embeds two distinct watermarks into the generated text using two independent keys. This design enables reliable authorship identification even in the event that one key is compromised. Experimental results demonstrate that our method achieves high detection accuracy for both watermarks while maintaining the fluency and overall quality of the generated text.

1 Introduction

Large language models (LLMs), such as GPT-4 (Achiam et al., 2023), have achieved the ability to generate highly fluent text that is often indistinguishable from human-written content. As LLMs become increasingly widespread, they hold the potential to dramatically reduce the labor costs associated with tasks such as text composition, which have traditionally relied on human effort. However, concerns have also been raised regarding the misuse of LLMs for spreading misinformation and facilitating fraudulent activities (Crothers et al., 2023). To address such issues, significant research has focused on developing techniques to detect text generated by LLMs.

Detection methods for LLM-generated text can be broadly categorized into (i) post-hoc detection and (ii) proactive (watermark-based) detection (Kirchenbauer et al., 2023b). Post-hoc detection refers to methods applied after text generation, with typical approaches involving binary classification models that discriminate between human-written text and LLM-generated text (Jawahar et al., 2020; Mitchell et al., 2023). While these methods can be applied to outputs from arbitrary LLMs, their effectiveness fundamentally relies on a sufficient distributional gap between human- and machine-generated text. As LLMs continue to improve and this gap narrows, the performance of post-hoc detectors is expected to decline. In contrast, proactive detection methods embed detectable watermarks into the text during the generation process. Recent work has proposed watermarking techniques based on the autoregressive generation process of LLMs (Kirchenbauer et al., 2023a), achieving substantially higher detection accuracy compared to post-hoc methods.

In watermarking for LLM outputs, a specific token pattern is embedded as a watermark by modifying the generation probabilities of a subset of tokens at each decoding step, according to a single secret key. For detection, the same secret key is used to identify the proportion of probability-adjusted tokens, and statistical hypothesis testing is performed to accurately determine the text's origin. However, a critical limitation of this approach is that, if the secret key used for watermarking is leaked, anyone with access to the key can reproduce the watermark, rendering reliable source attribution virtually impossible.

To mitigate this vulnerability to key leakage, this paper proposes a novel nested watermarking method. In the proposed approach, two distinct secret keys are used to embed two independent watermarks into the generated text. This enables robust source attribution even if the first key is compromised, as the second watermark remains secure and verifiable. Experiments conducted on instruction

datasets constructed from prompt texts demonstrate that the proposed method achieves high detection accuracy for both watermarks. Furthermore, automatic evaluation experiments using GPT-4 confirm that our nested watermarking preserves the quality of generated text compared to single watermarking approaches.

2 Related Work

Even before the advent of LLMs, research on embedding watermarks into text has been actively pursued for a considerable period (Kamaruddin et al., 2018). Early studies on text watermarking focused on exploiting the format and structure of textual data (Brassil et al., 1994). Additionally, methods leveraging syntactic tree structures, which embed watermarks into text using keys, have also been proposed (Atallah et al., 2001).

With the recent proliferation of LLMs, the need to distinguish between human-written and machine-generated text has grown rapidly, positioning watermarking as one of the most promising techniques for this purpose. A notable advantage of watermarking for LLM outputs is its empirically demonstrated robustness against text modifications (Kirchenbauer et al., 2023b). On the other hand, vulnerabilities such as susceptibility to spoofing attacks have been identified, where adversarial edits are made to watermarked text without intentionally destroying the watermark itself (Sadasivan et al., 2023). Therefore, for watermarking and similar approaches to gain widespread adoption in real-world applications, it is essential to conduct ongoing research that considers a broader range of use cases.

(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.

Beyond watermarking, another proactive detection approach is the use of retrieval-based systems (Krishna et al., 2024). In these systems, outputs generated by LLMs are pre-registered in a database, allowing subsequent identification of the text's origin by reference to the stored entries. Although conceptually straightforward, this method is particularly effective as a countermeasure against the aforementioned spoofing attacks.

3 Method

Figure 1 illustrates the overall architecture of the proposed method in the case of two nested watermarks. The framework comprises a nested watermark generator, a nested watermark detector, and multiple distinct secret keys. Within the nested watermark generator, watermarks are embedded at different layers using multiple keys as the language model generates text in response to a prompt. The nested watermark detector subsequently analyzes the output text and determines the presence or absence of each watermark by utilizing the corresponding keys. The following sections provide a detailed description of both the nested watermark generator and the nested watermark detector.

3.1 Nested Watermark Generator

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:

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

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 maps the concatenation of the 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 $m \neq n$:

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

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

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 set of 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 exponentiated logits, D_{total}, is calculated as follows:

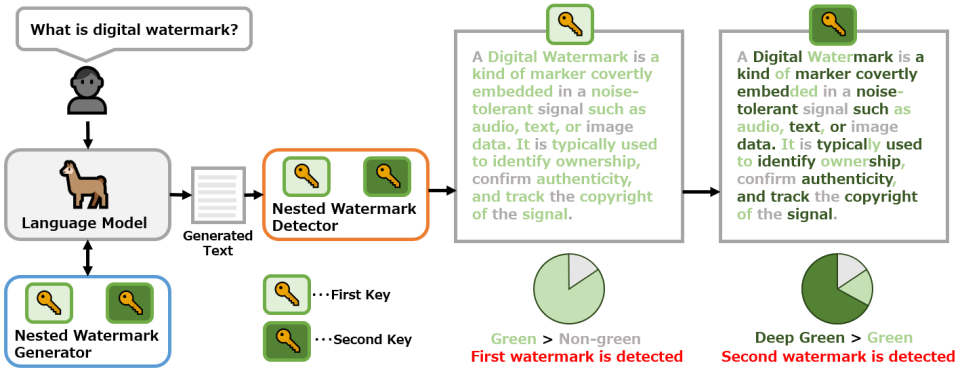

Figure 1: An overview of our nested watermark. The text on the right side of the figure demonstrates the detection of the first and second watermarks using the first and second keys, respectively. In the first text detected by the first key, the gray parts represent tokens classified as belonging to 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.

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

The adjusted probabilities for the tokens in G_1 and G_2 are then given by:

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

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

3.2 Nested Watermark Detector

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

For the first watermark:

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

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

For the second watermark:

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

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

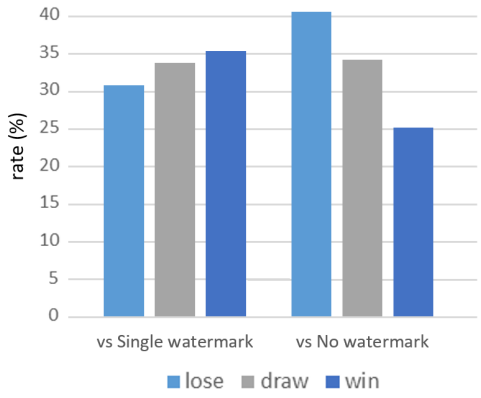

Figure 2: Comparison of text quality generated by each method. The vertical axis indicates the proportion of cases in which GPT-4 judged the text generated by the proposed method to be qualitatively superior (win), equivalent (draw), or inferior (lose) compared to the texts generated by each baseline.

4 Experiments

4.1 Experimental Setup

To evaluate the proposed method, we measure the detection accuracy of nested watermarking in terms of Type I error (false positives) and Type II error (false negatives). In addition, to assess the impact of nested watermark embedding on text quality, we conduct a quantitative evaluation using GPT-4 (gpt-4-1106-preview), following the LLM-as-a-judge framework (Zheng et al., 2024), which is an automatic evaluation method. For both detection accuracy and text quality assessments, we compare the proposed method with a baseline of single-key watermarking (single watermarking) (Kirchenbauer

Table 1: Comparison of detection accuracy between nested watermark and single watermark

	Nested watermark		Single watermark	
	First watermark	Second watermark	First watermark	First watermark
bias (δ)	1.5	2.5	1.5	4.0
z score (Avg.)	10.34	7.94	5.91	12.52
Type I error	0.000	0.001	0.000	0.000
Type II error	0.000	0.012	0.100	0.000

et al., 2023a). Notably, since LLM-as-a-judge is known to exhibit positional bias, where the text presented first tends to receive higher ratings, each comparison is conducted twice per example, swapping the order of the candidate texts, and the average score of both orders is reported.

For the evaluation dataset, we use 1,000 samples of English instruction data generated by gpt-4-1106-preview. This dataset consists of pseudo-prompts designed to reflect realistic use cases for LLMs (e.g., news articles and social media posts). In contrast, prior work (Kirchenbauer et al., 2023a) has focused on text completion tasks, where the prompts at inference time are constructed from fragmented texts sampled from C4 dataset. Therefore, utilizing our synthetic dataset allows us to evaluate the proposed method under a setting that is closer to real-world generation scenarios.

We employ Llama-2-7b-chat as the text generation model. The maximum output token length is set to 300; if the generated text exceeds this limit, generation is stopped at 300 tokens. The ratio γ of tokens with increased probabilities is set to 0.5 for embedding both the first and second watermarks. The increment applied to logits for nested watermarking is set to $\delta_1 = 1.5$ and $\delta_2 = 2.5$, respectively. The threshold for the z-score in statistical testing is set to 4.0. For decoding, beam search is used with a beam size of 8.

4.2 Experimental Results

4.2.1 Detection Accuracy of Nested Watermarking

Table 1 presents a comparison of detection accuracy between nested watermarking and single watermarking. For single watermarking, we evaluate two configurations: one where the bias is set equal to the first watermark in nested watermarking ($\delta_1 = 1.5$), and another where the bias equals the sum of $\delta_1 = 1.5$ and $\delta_2 = 2.5$, i.e., 4.0.

First, regarding the Type I Error (the proportion of cases where a watermark is incorrectly detected in non-watermarked text), all watermarking methods achieve an extremely low error rate of 0 or 0.1%. In contrast, for Type II Error (the proportion of cases where a watermark is not detected in watermarked text), the error rate for single watermarking with a bias of 1.5 exceeds 10%, indicating low detection performance. On the other hand, the Type II Error for the first watermark in the proposed method and for single watermarking with a bias of 4.0 is 0%. As evidenced by the relatively high average z-scores for these methods, increasing the bias sufficiently raises the z-score in the statistical test, thereby reducing the Type II Error rate.

It is worth noting that in the proposed method, the token set for the first watermark includes the token set for the second watermark as a subset. Therefore, the bias for the intersection is the sum of $\delta_1 = 1.5$ and $\delta_2 = 2.5$, making it 4.0 in total, which leads to improved detection accuracy for the first watermark. Additionally, the Type II Error for the second watermark in the proposed method is 1.2%. While there is room for improvement, this result confirms that the detection accuracy remains sufficiently high even in scenarios where the first key has been leaked.

4.2.2 Text Quality of Watermarked Outputs

Figure 2 presents the results of the text quality comparison for each method. When comparing the proposed method to the no watermark baseline, the win rate is 25.2%, while the lose rate is 40.6%. This indicates that the introduction of nested watermarking tends to reduce the overall quality of text generated by the language model. However, the sum of the win rate and draw rate exceeds 60%, suggesting that it remains reasonably feasible to generate fluent text even with nested watermarks applied.

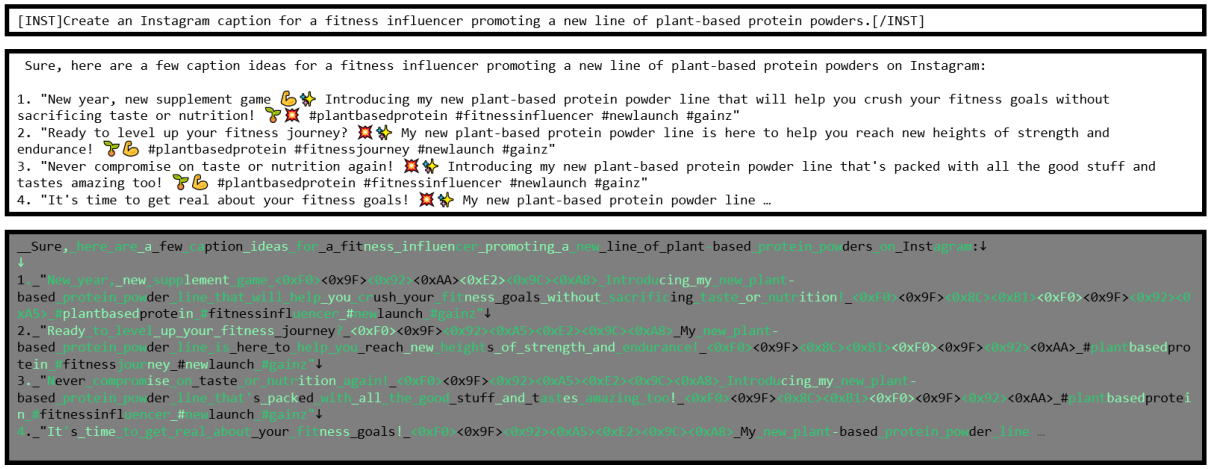

Figure 3: Example output of applying nested watermarking to Llama-2-7b-chat. The topmost box shows a prompt sampled from the instruction dataset. The second and third boxes display, respectively, the output text presented to the user and a visualization of the nested watermark in the text for detection purposes. In the third box, light green highlights indicate tokens marked by the first watermark, and dark green highlights indicate tokens marked by the second watermark. For readability, newline characters have been replaced with the symbol "↓" and line breaks have been processed accordingly.

Additionally, we conducted a comparison between the proposed method and the single watermark approach (single watermark) with a bias value of 4.0. Interestingly, relative to the single watermark, the proposed method achieved a higher win rate of 35.35% and a lower lose rate of 30.85%, indicating that our approach outperforms the single watermark method in terms of text quality. Intuitively, one might expect that a single watermark, which embeds a watermark only once, would have less impact on token generation probabilities compared to the nested watermarking approach, which embeds watermarks twice. However, considering the proportion of tokens affected by the bias, in the single watermark case, $\gamma \times |V|$ tokens are uniformly affected by a bias of 4.0. In contrast, for nested watermarking, $\gamma \times |V|$ tokens are first influenced by the initial bias (1.5), but only $\gamma^2 \times |V|$ tokens are subsequently affected by the second bias (2.5). Therefore, the number of tokens influenced by the maximum bias (4.0) is reduced by a factor of γ compared to the single watermark case. This difference likely contributes to the improved preservation of text quality observed with the proposed method.

5 Conclusion

In this paper, we proposed a nested watermarking scheme that enables source attribution even in cases where the first key is leaked. Experimental results demonstrated that the proposed method achieves high detection accuracy for both watermarks from two keys, as measured by Type I and Type II Errors. Furthermore, compared to conventional methods, our approach maintains text quality at an equal or higher level. Future work includes investigating the robustness of nested watermarking against text tampering and evaluating detection accuracy on shorter texts.

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*.