MACHINE NEVER SAID THAT: DEFENDING SPOOFING ATTACKS BY DIVERSE FRAGILE WATERMARK

Yuhang Cai, Yaofei Wang*, **Donghui Hu, Chen Gu**
Department of Computer Science and Information Engineering
HeFei University of Technology
HeFei, China
2023110524@mail.hfut.edu.cn,{wyf,hudh,guchen}@hfut.edu.cn

ABSTRACT

Misusing the large language models (LLMs) has intensified the need for robust generated-text detection through watermarking. Existing watermark methods prioritize robustness but remain vulnerable to spoofing attacks, where modified text retains detectable watermarks, falsely attributing malicious content to the LLM. We propose the Multiple-Sampling Fragile Watermark (MSFW), the first framework to integrate local fragile watermarks to defend against such attacks. By embedding context-dependent watermarks through a multiple-sampling strategy, MSFW enables two critical detection capabilities: (1) Modification detection via localized watermark fragility, where any modification disrupts adjacent watermark and reflected through localized watermark extraction; (2) Generated-text detection using unaffected global watermarks. Meanwhile, our watermarking method is unbiased and improves the diversity of the output by the multiple-sampling strategy. This work bridges the gap between robustness and fragility in LLM watermarking, offering a practical defense against spoofing attacks without compromising utility.

1 INTRODUCTION

Large language modelsOpenAI (2023); Touvron et al. (2023); DeepSeek-AI (2024) (LLMs) have greatly changed ways of text processing. Their output quality is improved to approach or surpass the human level in some fields, greatly reducing the difficulty of word processing and attracting great attention. However, these breakthroughs have raised another concern: the outputs of LLMs that are similar to human-written text but without effective supervision may be used for various malicious purposes, such as false message generation, chat fraud, etc. Pan et al. (2023); Kim et al. (2024)Therefore, determining whether a text is generated by AI becomes particularly important Chakraborty et al. (2024); Mitchell et al. (2023) and watermark is a promising method to reduce the risks of LLM abuse Kirchenbauer et al. (2023b); Li et al. (2024); Wu et al. (2023).

Current watermark methods often focus on the total watermark score to improve robustness Kirchenbauer et al. (2023a); Zhao et al. (2024); Wu et al. (2024); Hu et al. (2024). Modification in the text influences extracting watermark on nearby tokens but shows an inapparent impact on the total score, which means the watermark information can still be accurately extracted, preventing it from being easily removed and ensuring the effectiveness of generated-text detection by watermark. However, these watermarking methods ignore the necessity of local watermark detection. For example, as shown in Figure 1, the attacker modifies the text to make it contain malicious context. Although modification influences the extraction of the watermark, the generated-text detector based on the total watermark score still confirms that the modified text was generated by a large language model, making the real attacker hidden. This type of attack makes use of the characteristic of total watermark score and can cause great damage to the owners of large language models, as it forces large amounts of maliciously modified texts to be ”attributed” to the large language model through watermarks, triggering public questioning the large language model Gloaguen et al. (2024); Qi et al. (2024). We believe that although watermarks with total watermark score detection are ineffective

[*] Corresponding author Yaofei Wang

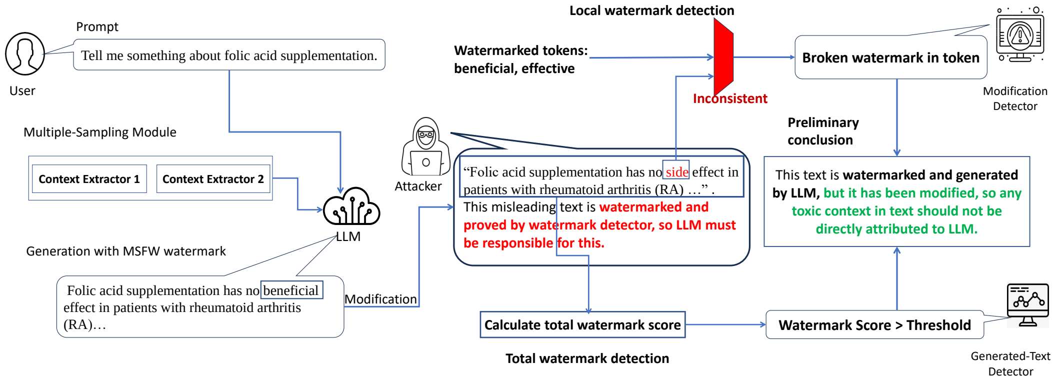
Prompt
Tell me something about folic acid supplementation.
User
Multiple-Sampling Module
Context Extractor 1
Context Extractor 2
LLM
Generation with MSFW watermark
Folic acid supplementation has no beneficial effect in patients with rheumatoid arthritis (RA)…
Modification
Attacker
"Folic acid supplementation has no side effect in patients with rheumatoid arthritis (RA) …" .
This misleading text is watermarked and proved by watermark detector, so LLM must be responsible for this.
Local watermark detection
Watermarked tokens: beneficial, effective
Inconsistent
Broken watermark in token
Modification Detector
Preliminary conclusion
This text is watermarked and generated by LLM, but it has been modified, so any toxic context in text should not be directly attributed to LLM.
Calculate total watermark score
Total watermark detection
Watermark Score > Threshold
Generated-Text Detector

Figure 1: The framework of MSFW method. We introduce the multiple-sampling strategy during generation to improve the diversity of output. If any malicious attackers modify the watermarked text and use it as evidence for spoofing attacks, MSFW method can reach modification detection by local watermark detection to find the modified tokens in watermarked text. Besides, the detector achieves generated-text detection by total watermark detection which proves the text is generated by LLMs. By combining these two detection results, we conclude that the text is generated by LLM but is modified, which resists spoofing attacks.

when facing watermark removal attacks, these methods are unable to defend against spoofing attacks because they ignore the necessity of modification detection by local watermark detection.

To reduce the risk of spoofing attacks, we propose the concept of the fragile watermark for LLMs. As shown in Fig.1, any tiny modification can cause an obvious influence of extracting the watermark on the next several tokens and diminish the local watermark score to an abnormal level that is enough to prove the existence of modification. Besides, the unaffected watermark can still be extracted to confirm the generated text. This kind of watermark can achieve modification detection and generated-text detection simultaneously, which is suitable for practical use especially when facing spoofing attacks. Then we propose the multiple-sampling fragile watermark (MSFW) method based on this concept. The method uses two context extraction strategies and randomly selects one to sample the next token from logits to embed the watermark. The watermarking process does not affect the output quality of the model, so the MSFW method is an unbiased watermark method. In addition, we design the detector based on the multiple-sampling strategy. If any token in context is modified, the detector fails to sample the same subsequent tokens as before and reflects the situation in the local watermark detection which functions as evidence for modification. On the other hand, the multiple-sample strategy of our method improves the diversity of generated text, which generates diverse text when using the same prompt rather than repetitive text. Experiments demonstrate that our fragile watermark method can achieve effective dual detection capabilities: modification and generated-text detection. Besides, our multiple-sample strategy for watermark is unbiased and improves the diversity of generation.

Our main contributions are summarized as follows:

1. We propose the concept of the fragile text watermark for the output of LLMs for the first time which introduces local watermark detection and enables detection of any tiny modifications in watermarked text to defend against spoofing attacks.

2. We propose a novel multiple-sampling fragile watermark method that achieves dual detection capabilities: modification and generated-text detection for the output of LLM by watermark.

3. We propose the multiple-sampling strategy for the watermarking method. It maintains the probability distribution of the original model output and improves the diversity of text generation. Experiments show that the MSFW method with the multiple-sampling strategy generates diverse text with the same prompt while maintaining unbiasedness, making the watermark more useful for practical situations.

2 Method

2.1 Preliminary

In LLM generation, P_M represents the probability distribution generated by pre-trained LLM, and $\mathcal{V}$ is the total vocabulary set. In a typical LLM generation task, LLM receives a prompt $x_{-n_p:0}$ and outputs a sequence $x_{1:n}$ according to the prompt and the generated tokens $x_{-n_p:i-1}$ by gradually generating the next token x_i. When generating the token x_i, the probability of the token in the vocabulary set $\mathcal{V}$ is given by the conditional probability distribution $P_M(x_i \mid x_{-n_p:i-1})$. An unbiased watermark requires that the expected value of the distribution after embedding the watermark is equal to the expected value of the original probability distribution, that is, given prompt $x_{-n_p:0}$ and key k if the watermark model $P_{M,w}$ satisfies

$$P_M(x_i \mid x_{-n_p:i-1}) = \mathbb{E}_{\theta_i \sim P_\Theta}\left[P_{M,w}\left(x_i \mid \boldsymbol{x}_{-n_p:i-1}, \theta_i\right)\right] \tag{1}$$

the above equation applies to any prompt p, any token x_i. If any generation step $0 < i <= n$ holds, then the watermarking method is unbiased for the original model P_M, meaning that the embedding of the watermark will not affect the expected output quality of the model.

2.2 Multiple-sampling Strategy

Previous watermarking methods often use the single-sampling strategy, which means they use the context extraction function with a fixed hyper-parameter to extract a certain number of contexts to sample watermarked tokens. For example, δ-reweight selects the previous n tokens in context, and samples watermark tokens by these tokens. The advantage of this strategy is that the sampling result remains the same in the same context. However, this strategy limits the diversity of watermarked text, especially when watermarked tokens are sparse in a token list and forces the LLM to output watermarked tokens like δ-reweight Hu et al. (2024). Therefore, we propose a multiple-sampling strategy, which introduces randomness in the context extraction process to generate diverse text. As shown in Figure 2, we use two sample functions with different hyper-parameters to extract from previous tokens. Then we randomly select one to extract the context and sample the watermarked tokens. Although only one context is used for sampling watermarked tokens at each time, the whole sentence has more possible tokens sampled by different contexts rather than a fixed context when generating tokens in a certain position. The multiple-sampling strategy increases the diversity of text compared with the single-sampling strategy, especially the diversity of the original method is poor.

2.3 Fragility and Local Watermark Detection

Former robust watermark methods often employ the total watermark detection strategy, focusing on extracting comprehensive watermark information from the entire text to determine whether it was generated by LLM. However, these methods ignore local watermark information. Taking the KGW method Kirchenbauer et al. (2023a) as an example, it encourages the model to produce green tokens. However, generating green tokens is not mandatory, and red tokens may still be generated regardless of watermark embedding. Therefore, the local red tokens are useless for both watermark and modification detection.

To achieve local watermark detection, we need to observe obvious changes in the local, but the robust watermark is so robust that modification is hard to influence the total watermark score, let alone the local watermark. So we propose the fragile text watermark for LLM, which is highly sensitive to modification and can reveal any possible modifications by detecting damaged watermarks. According to the concept of fragile text watermark, we identify a potential characteristic of fragility as shown in Figure 2: if the model is compelled to sample from specific watermarked tokens list during generation, then the probability of any tokens outside the watermarked tokens list in the detected text becomes 0, meaning it is impossible to generate such tokens when the watermark is embedded. In such cases, local watermark detection is meaningful because any abnormality (probability of detected token is 0) from the local watermark detection can be accurately determined as a modification, rather than the normal output of the watermark. Based on the characteristics of fragility during watermark detection, we introduce fragility into watermarking and implement local watermark detection, thereby achieving effective modification detection.

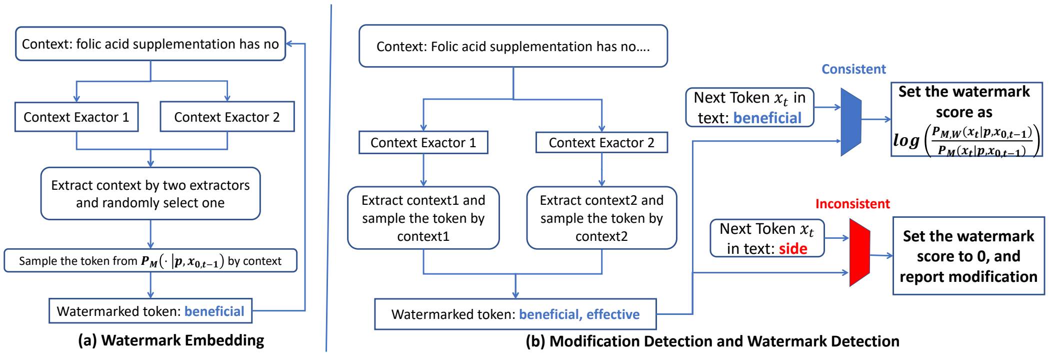
Context: folic acid supplementation has no
Context Exactor 1
Context Exactor 2
Extract context by two extractors and randomly select one
Sample the token from $P_M(\cdot\,|p, x_{0,t-1})$ by context
Watermarked token: beneficial
(a) Watermark Embedding
Context: Folic acid supplementation has no....
Context Exactor 1
Context Exactor 2
Extract context1 and sample the token by context1
Extract context2 and sample the token by context2
Watermarked token: beneficial, effective
Consistent
Next Token x_t in text: beneficial
Set the watermark score as $log\left(\frac{P_{M,W}(x_t|p,x_{0,t-1})}{P_M(x_t|p,x_{0,t-1})}\right)$
Inconsistent
Next Token x_t in text: side
Set the watermark score to 0, and report modification
(b) Modification Detection and Watermark Detection

Figure 2: The framework of MSFW method. (a) demonstrates the process of embedding watermark, which introduces two sample functions with different context extraction sets and randomly selects one from them to sample the watermarked token. (b) introduces the process of watermark detection which calculates the watermark score for the token “beneficial” that is consistent with watermarked tokens and the process of modification detection which reports modification for non-watermarked token “side” that is inconsistent with watermarked tokens .

2.4 MSFW Watermark Method

Algorithm 1 Multiple-sampling Fragile Watermark Method

```
Input: LLM $M$, prompt $p$, context extraction function $C1$ and $C2$ with different extraction hyper-parameter, sample function $F$, hash function $h$
Output: generated text $x_{0:i-1}$
  for Index $i = 1, 2, \ldots$ do
    Obtain probability distribution $P_M(\cdot \mid p, x_{0,i-1})$ from $M$
    Randomly select context extraction function: $C \leftarrow \text{RandomSelect}(C_1, C_2)$
    Extract context: $c_i \leftarrow C(x_{0:i-1})$
    Generate seed: $s_i \leftarrow h(c_i, \text{key})$
    Sample watermarked token: $x_i \leftarrow F(s_i)$
  end for
  return $x_{0:i-1}$
```

According to the multiple-sampling strategy, we introduce two sample functions with different context extraction sets and use a random selector to select one from them. As shown in Algorithm 1, we obtain the logits of LLM and perform the softmax function to get probability distribution. Then use the random selector to select one sample function. Next, generate a random seed based on context and private key. Finally, use the seed to sample one token in probability distribution and output it as the watermarked token. The watermark on tokens is fragile because the model is compelled to sample from a fixed watermarked tokens list during generation. If the token has been modified, the extraction of the watermark on this token is influenced.

2.5 Modification and Generated Text Detection

We described modification and generated-text in Algorithm 2. According to the embedding process, two sample functions were used throughout the generation process, so we constructed the same sampling function to restore the scene at the time of generation. Unlike the δ-reweight method Hu et al. (2024), we introduce the multiple-sampling strategy, which theoretically generates two possible watermarked tokens but outputs one each time. So we reproduce the watermark lists by two sampling settings. Based on the fragility of the watermark and local watermark detection strategy, we compare the watermarked tokens list with the current token to determine whether the token has been modified. If the current token is not in the watermarked tokens list, it should be marked as modified and the detector should turn to the next token because it is impossible to generate such token when the watermark is embedded. Otherwise, we compute the LLR score of each watermarked token by

Algorithm 2 Modification and Generated Text Detection

```
Input: LLM $M$, prompt $p$, generated text $x_{0:i-1}$, context extraction function $C1$ and $C2$ with different extraction hyper-parameter, sample function $F$, hash function $h$
Output: Average watermark detection score $LLR_{avg}$, Modification flag $M_{flag}$
Initialize $LLR_{total} \leftarrow 0$, $M_{flag} \leftarrow$ false
for Index $i = 1, 2, \ldots$ do
    Obtain probability distribution $P_M(\cdot \mid p, x_{0,i-1})$ from $M$
    Extract contexts: $c_i^1 \leftarrow C_1(x_{0:i-1})$, $c_i^2 \leftarrow C_2(x_{0:i-1})$
    Generate seeds: $s_i^1 \leftarrow h(c_i^1, \text{key})$, $s_i^2 \leftarrow h(c_i^2, \text{key})$
    Sample watermarked tokens: $x_i^1 \leftarrow F(s_i^1)$, $x_t^2 \leftarrow F(s_i^2)$
    if $x_i \in \{x_i^1, x_i^2\}$ then
        Compute log-likelihood ratio: $LLR(x_i) \leftarrow \log\left(\frac{1}{P_M(x_i|x_{0,i-1})}\right)$
        Accumulate score: $LLR_{total} \leftarrow LLR_{total} + LLR(x_i)$
    else
        Mark as modified: $M_{flag} \leftarrow$ true
    end if
end for
Compute average score: $LLR_{avg} \leftarrow \frac{LLR_{total}}{n}$
return $LLR_{avg}$, $M_{flag}$
```

$LLR(x_i) = \log \frac{P_{M,w}\left(x_i|\boldsymbol{x}_{-n_p:i-1},\theta_i\right)}{P_M\left(x_i|x_{-n_p:i-1}\right)}$. Traditional LLR score may result in $LLR(x_i) = -\infty$ when the token x_i is modified, then makes the detection fail. However, we detect the modification before calculating the LLR score and discard the modified tokens directly. So LLR score is meaningful for watermarked tokens in text. Finally, we calculate the sum of the watermark score and divide it by the total number of tokens to reflect the strength of the watermark. If the average watermark score is greater than the threshold, it is considered to carry a watermark and generated by LLM. Otherwise, it is considered not to carry one:

$$LLR_{avg} = \frac{1}{N} \sum_{N}^{i=1} LLR(x_i) \tag{2}$$

3 Experiments

3.1 Settings and Datasets

We use the model OPT-6.7B Zhang et al. (2022) and set the sampling method to top-K=50 to generate datasets.

Diversity We used the same datasets and detection metrics as Fu et al. (2024) to measure the diversity of results generated by different methods. Diversity is evaluated through test datasets based on the question dataset, which contains 20 high-end prompts that are repeated 50 times. We use four different watermark settings to generate test datasets with a maximum generated length of 128 to investigate the diversity changes of watermark methods after embedding the watermark.

Unbiasedness We follow the same evaluation process to show our method is unbiased. We assess the performance of MSFW with two seq2seq models: machine translation (MT) and text summarization (TS). For the TS task, our experimentation employs the BART-large model Liu et al. (2020) as the generator, and the CNN-DM Hermann et al. (2015) corpus as the test dataset. The MT task focuses on English-to-Romanian translation and employs the Multilingual BART (MBart) model Liu et al. (2020) on the WMT'14 En-Ro corpus dataset.

Modification Detection and Generated-text Detection We obtain 1,000 questions from the C4 dataset and use them as prompts to generate two datasets: one without watermark and the other with watermark, each containing 1000 pieces of text, and the max-length of each text is 20. We use a random perturbation parameter ϵ to create datasets subjected to different attack strengths and choose

Table 1: Results of modification detection under different perturbation strengths and attacks.

Datasets	KGW (δ=1)				MSFW			
	TPR	TNR	Recall	F1-score	TPR	TNR	Recall	F1-score
Addition (ϵ=0.1)	0.686	0.680	0.681	0.683	0.984	0.962	0.962	0.973
Addition (ϵ=0.2)	0.508	0.680	0.720	0.769	0.998	0.962	0.963	0.980
Replacement (ϵ=0.1)	0.727	0.680	0.694	0.710	0.923	0.962	0.960	0.941
Replacement (ϵ=0.2)	0.784	0.680	0.704	0.733	0.978	0.962	0.952	0.970
Deletion (ϵ=0.1)	0.725	0.680	0.693	0.709	0.979	0.962	0.962	0.970
Deletion (ϵ=0.2)	0.811	0.680	0.717	0.761	1.000	0.962	0.963	0.981

three types of attacks: addition, deletion, and replacement. For example, the addition dataset with ϵ = 0.1 means 10% of tokens are added as malicious attacks rather than generated.

3.2 Baseline and Evaluation Metrics

Modification Detection We implement the KGW method Kirchenbauer et al. (2023a) by counting the number of tokens in the red token set and Z-score which reflects the number of red tokens to detect modification. If $z - score < |threshold|$, then reports modified. We report watermarked and modified text as positive examples while reporting watermarked and non-modified text as negative examples Cai et al. (2025) and compute TPR, TNR, Recall, and F1-score to evaluate the ability of modification detection on different datasets.

Diversity and Unbiasedness We choose three different types of methods. For the KGW method, we set KGW with γ = 0.5 and δ in $\{1.0,2.0\}$. Meanwhile, we use two reweight methods and another method DiPmark Wu et al. (2024) with partition parameter α in 0.4,0.5 as typical unbiased methods. For diversity, we use Self-BLEU, Distinct 1-gram, and 2-gram to measure generation diversity. For unbiasedness, we use BLEU, BERTSCORE, ROUGLE-1, and Perplexity to assess the performance of the watermarking method in different situations.

Robustness We choose KGW and δ-reweight Hu et al. (2024) with the original LLR method maximin variant of the LLR (mmLLR) as watermark baselines. Specifically, we set KGW with γ = 0.5 and δ=1.0. For generated-text detection, we set KGW with γ = 0.5 and δ in $\{1.0,2.0\}$ while the hyper-parameter grid_ size of mmLLR is set to 10. We evaluate the performance of different watermarking methods by reporting the Area Under Curve (AUC) of watermark detection.

3.3 Modification Detection

Table 1 presents results of the modification detection method on different datasets. MSFW method exhibits better performance in detecting both modified text (TPR) and original text (TNR) than the KGW method. Compared to the robust KGW method, MSFW method is fragile by outputting the next token only from a specific watermarked tokens list during generation, then uses local watermark detection that is sensitive to broken watermark: any tokens outside the watermarked tokens list in the detected text are modified when the watermark is embedded. In short, MSFW method utilizes the characteristic of fragility and local watermark detection and then achieves effective modification detection.

3.4 Diversity

As shown in Table 2, after adding the watermark by δ-reweight method which able to achieve modification detection, the Self-Bleu score increases to 1, while the Dist-1 and Dist-2 scores significantly decrease. Although the MSFW method samples a fixed number of watermarked tokens each time like δ-reweight, our approach shows no significant changes in the indicators of Dist-1 and Dist-2, while the Self-Bleu score has a certain increase compared to another watermarking method like DipMark. The multiple-sampling strategy introduces randomness in watermarked tokens by using different context extraction settings and then increases the number of potential sentences that can be generated. Despite we only used two alternative settings, our method performs better than

Table 2: Evaluation of three diversity indexes for different watermark methods in the QA task. For KGW methods, we set the hyper-parameter in $\{1.0,2.0\}$. For DiPmark, we set the hyperparameter to 0.5.

Methods	Self-Bleu	Dist-1	Dist-2
No Watermark	0.094	0.262	0.760
KGW (δ=1.0)	0.096	0.254	0.750
KGW (δ=2.0)	0.105	0.247	0.732
DiPMark (α=0.5)	0.100	0.253	0.749
γ-reweight	0.100	0.253	0.749
δ-reweight	1.0	0.014	0.019
MSFW	0.229	0.241	0.707

Table 3: Performance of different watermarking methods on TS and MT

Methods	Machine Translation		Text Summarization		
	BERTScore	BLEU	BERTScore	ROUGE-1	Perplexity
No watermark	0.565±0.002	22.1±0.3	0.3267±0.0009	0.3865±0.0010	5.031±0.019
KGW (δ=1.0)	0.558±0.002	21.4±0.3	0.3232±0.0009	0.3820±0.0009	5.298±0.020
KGW (δ=2.0)	0.543±0.002	19.8±0.3	0.3120±0.0008	0.3713±0.0009	6.251±0.023
γ-reweight	0.566±0.002	22.3±0.3	0.3280±0.0009	0.3867±0.0010	5.011±0.019
DiPmark (α=0.4)	0.566±0.002	22.3±0.3	0.3268±0.0009	0.3861±0.0010	5.010±0.019
DiPmark (α=0.5)	0.566±0.002	22.3±0.3	0.3266±0.0009	0.3862±0.0010	5.000±0.019
MSFW	0.567±0.002	22.3±0.3	0.3246±0.0009	0.3861±0.0009	5.058±0.019

δ-reweight and even as well as other methods that are unable to achieve effective modification detection, which means we balance the relationship between modification detection and diversity by multiple-sampling strategy.

3.5 Unbiased

Table 3 shows the mean and variance of score per token for TS and MT. By comparing the KGW method with different δ parameters, our method achieves the same score distribution as well as other unbiased methods like δ-reweight. So our method is unbiased. As a comparison, the BLEU score of the KGW method decreases and the complexity increases when the δ parameter increases, which damages the quality of the output.

3.6 Generated Text Detection

Apart from modification detection, generated-text detection is a significant function for watermarks. So we show the result of AUC under different perturbation strength and perturbation methods in Table 4. The MSFW method achieves an AUC of 0.987 on the original watermarked dataset, surpassing both the KGW and δ-reweight (mmLLR) method, which means our method has a low type II error rate especially when the generated-text is short. Compared to the robust KGW method, the AUC of MSFW decreases when the attack strength increases because MSFW method extracts more tokens as context to diminish the probability of repetitive context, which influences the unbiasedness of the watermark. Yet it still achieves a certain level of robustness under modification attacks. Compared to the unbiased δ-reweight method, our method shows better performance when using the Top-K setting. Meanwhile, δ-reweight method with the mmLLR detection method does not perform well due to the probability of most tokens being set to 0 according to Top-K settings and the mmLLR method fails to make corresponding adjustments to this situation. So our MSFW method achieves dual detection capabilities for the output of LLM by effectively detecting modification in text and exhibiting certain robustness against modification.

Table 4: AUC of generated-text detection for different methods under different perturbation strength

Strength	Method	Addition	Replacement	Deletion
ϵ=0.0	KGW (δ=1)	0.808	0.808	0.808
	KGW (δ=2)	0.940	0.940	0.940
	δ-reweight (mmLLR)	0.760	0.760	0.760
	MSFW	0.987	0.987	0.987
ϵ=0.1	KGW (δ=1)	0.776	0.767	0.772
	KGW (δ=2)	0.922	0.914	0.912
	δ-reweight (mmLLR)	0.632	0.607	0.618
	MSFW	0.871	0.838	0.866
ϵ=0.2	KGW (δ=1)	0.753	0.720	0.736
	KGW (δ=2)	0.896	0.866	0.855
	δ-reweight (mmLLR)	0.580	0.562	0.566
	MSFW	0.778	0.658	0.702

4 Related Work

Watermarking for LLM There has been a recent emergence of watermarking large language models for AI detection Kamaruddin et al. (2018); Yoo et al. (2024); Yang et al. (2022). Kirchenbauer et al. (2023a) proposed the first LLM watermark algorithm, which uses a hash function to randomly divide the candidate word library into a red list and a green list, and by increasing the logits of the tokens in the green list, the output text comes more from the green list. To reduce the impact on the quality of text generation, Lee et al. (2024) considered the entropy of the text to adaptively modify logits. To enhance the robustness of the watermark, Zhao et al. (2024) improved the robustness by fixing the division of the red and green lists.

Unbiased Watermark The above methods will affect the quality of the text generated by the LLM. To avoid damaging the quality of text generation, Hu et al. (2024) and Wu et al. (2024) proposed unbiased watermark algorithms that can maintain probability distribution while embedding the watermark.

Fragile watermark Fragile watermark is a technique used primarily for digital image authentication and integrity verification, characterized by its sensitivity to modifications Shehab et al. (2018); Fridrich et al. (2000). It is designed to be easily altered or destroyed when the host image undergoes any modification, whether intentional or non-malicious. Thus, fragile watermarking plays a pivotal role in enhancing digital content integrity and is a key area of study in contemporary multimedia security research Lin et al. (2011). However, most existing research on fragile watermarking focuses on images, with limited exploration in the context of text.

5 Conclusion

We propose the concept of fragile watermark for the output of LLMs for the first time and a novel multiple-sample fragile watermark method (MSFW) which introduces local watermark detection and enables detection of any tiny modification in watermarked text. Experiments show that the MSFW method with the multiple-sampling strategy achieves generated-text detection and modification detection simultaneously. Besides, the method generates diverse text while maintaining unbiasedness, making the watermark more useful for practical situations.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62302146 and Open Fund of Anhui Province Key Laboratory of Cyberspace Security Situation Awareness and Evaluation under the grant No. CSSAE-2023-005.

References

Yuhang Cai, Yaofei Wang, Donghui Hu, and Gu Chen. Modification and generated-text detection: Achieving dual detection capabilities for the outputs of llm by watermark. *arXiv*, abs/2502.08332, 2025.

Souradip Chakraborty, Amrit Bedi, Sicheng Zhu, Bang An, Dinesh Manocha, and Furong Huang. Position: On the possibilities of AI-generated text detection. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6093–6115. PMLR, 21–27 Jul 2024.

DeepSeek-AI. Deepseek-v3 technical report. *arXiv*, abs/2412.19437, 2024.

J. Fridrich, M. Goljan, and A.C. Baldoza. New fragile authentication watermark for images. In *Proceedings 2000 International Conference on Image Processing (Cat. No.00CH37101)*, volume 1, pp. 446–449 vol.1, 2000. doi: 10.1109/ICIP.2000.900991.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5791–5808, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.315.

Thibaud Gloaguen, Nikola Jovanović, Robin Staab, and Martin Vechev. Discovering clues of spoofed lm watermarks. *arXiv preprint arXiv:2410.02693*, 2024.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Proceedings of the 29th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pp. 1693–1701, Cambridge, MA, USA, 2015. MIT Press.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. A review of text watermarking: Theory, methods, and applications. 6:8011–8028, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2796585. Conference Name: IEEE Access.

Hanna Kim, Minkyoo Song, Seung Ho Na, Seungwon Shin, and Kimin Lee. When llms go online: The emerging threat of web-enabled llms. *arXiv preprint arXiv:2410.14569*, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023a.

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

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? watermarking for code generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4890–4911, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, and Weijie J Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *arXiv preprint arXiv:2404.01245*, 2024.

Pei-Yu Lin, Jung-San Lee, and Chin-Chen Chang. Protecting the content integrity of digital imagery with fidelity preservation. *ACM Trans. Multimedia Comput. Commun. Appl.*, 7(3), Sep 2011. ISSN 1551-6857. doi: 10.1145/2000486.2000489.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742, 2020. doi: 10.1162/tacl_a_00343.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D. Manning, and Chelsea Finn. Detectgpt: zero-shot machine-generated text detection using probability curvature. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

OpenAI. Gpt-4 technical report. *arXiv*, abs/2303.08774, 2023.

Yikang Pan, Liangming Pan, Wenhu Chen, Preslav Nakov, Min-Yen Kan, and willian Wang. On the risk of misinformation pollution with large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 1389–1403. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-emnlp.97.

Pang Qi, Hu Shengyuan, Zheng Wenting, and Smith Virginia. No free lunch in llm watermarking: Trade-offs in watermarking design choices. *arXiv preprint arXiv:2402.16187*, 2024.

Abdulaziz Shehab, Mohamed Elhoseny, Khan Muhammad, Arun Kumar Sangaiah, Po Yang, Haojun Huang, and Guolin Hou. Secure and robust fragile watermarking scheme for medical images. *IEEE Access*, 6:10269–10278, 2018. doi: 10.1109/ACCESS.2018.2799240.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Junchao Wu, Shu Yang, Runzhe Zhan, Yulin Yuan, Derek F. Wong, and Lidia S. Chao. A survey on llm-generated text detection: Necessity, methods, and future directions. *arXiv preprint arXiv:2310.14724*, 2023.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. A resilient and accessible distribution-preserving watermark for large language models. In *Forty-first International Conference on Machine Learning*, 2024.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 11613–11621, 2022.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4031–4055. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.224.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*, 2022.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*, 2024.