Balanced Watermark: A Simple High-Imperceptibility Watermark for Large Language Models

Anonymous ACL submission

Abstract

In order to counteract the potential risks posed by increasingly intelligent Large Language Models (LLMs), several scholars attempt to apply watermark to the detection of LLM-generated text. Watermark researchers typically focus on detectability, robustness and invisible, but they tend to overlook the imperceptibility, which is crucial for preventing the watermark from being cracked. Watermarks with low imperceptibility are easily stolen and analyzed by malicious users, who can then forge watermarked text. To fill this research gap, we design Balanced Watermark (BW) by balancing the watermark strength across the vocabulary, achieving a fit to a non-watermarked LLM distribution to enhance imperceptibility. To effectively evaluate the imperceptibility of watermarks, we design a metric to evaluate for the first time. Our experiments prove that BW effectively improves imperceptibility and maintains high performance of the watermark in other features. We release our code[1] to the community for future research.

1 Introduction

With the rapid development of large language models (LLMs) (OpenAI, 2023; Touvron et al., 2023; AI@Meta, 2024), the text generated by LLMs increasingly resembles human-generated text and gradually fills every part of our lives, which poses several potential threats, including hallucinations (Alkaissi and McFarlane, 2023; Liu et al., 2024a), misinformation generation (Liu et al., 2024b; Zhang et al., 2024), and malicious use (OpenAI, 2023; Editorials, 2023). Therefore, detecting text generated by LLMs has become an emerging and critical issue.

Digital watermark (Atallah et al., 2001; He et al., 2022) is a promising method for detecting LLM-generated text, which embeds watermark information into the text during generation and determines whether the text is generated by the LLM by detecting the watermark information. A good watermark should possess the following five characteristics: (1) **Detectability**: The watermark adder can accurately distinguish between watermarked and non-watermarked text; (2) **Invisibility**: The quality of the watermarked text should not significantly degrade. (3) **Robustness**: The watermark should remain detectable when the watermarked text is subjected to attacks. (4) **Usability**: The time and resource consumption for adding and detecting watermarks should be acceptable. (5) **Imperceptibility**: It should be difficult for anyone other than the watermark adder to perceive the presence of the watermark in the text.

At present, there are many digital watermark frameworks for LLM-generated text (Abdelnabi

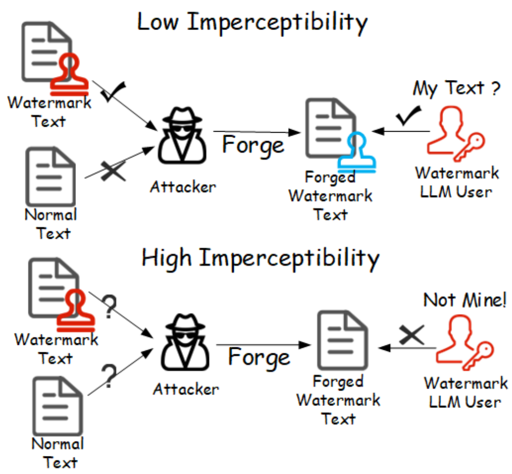


Figure 1: An explanatory diagram for watermark imperceptibility. Watermarked text with low imperceptibility can be easily detected by attackers, who can then summarize corresponding patterns and forge watermarked text. Text with high concealment can prevent attackers from forging watermarks, thereby preventing unauthorized users from mass-producing watermarked text.

[1] https://anonymous.4open.science/r/BalancedWatermark-6228

and Fritz, 2021; Yang et al., 2022; Yoo et al., 2023; Zhao et al., 2023b). Kirchenbauer et al. (2023) propose a simple and effective watermark framework, commonly referred to as KGW. KGW first generates a green list and a red list through the secret key and pre-text information at each step of the LLM generation process. Subsequently, KGW adds a fixed bias to the green list tokens to increases the probability of the LLM generating green list tokens. During detection, KGW analyzes the number of green tokens in the text to determine whether the text has been watermarked.

Despite numerous attempts to refine the watermarking approach based on KGW (Zhao et al., 2023a; Fairoze et al., 2023; Hou et al., 2023; Fu et al., 2024; Liu and Bu, 2024; Lu et al., 2024), we find that they primarily focus on improving invisibility and robustness, yet invariably overlook the imperceptibility of the watermark. This makes the watermarked text easily identifiable by malicious attackers, leading to potential attacks. To this end, we propose a novel watermark framework called Balanced Watermark (BW), aimed at enhancing the imperceptibility of watermarks while maintaining their invisibility and robustness. BW first divides the vocabulary into two lists based on the secret key. Subsequently, during the actual generation process, BW determines two signals with approximately equal occurrence probabilities based on word frequency and contextual information. The signal determine which of the two lists will be selected as the green list. BW ultimately adjusts the original probability distribution of the LLM according to the green list to embed watermark. We carry out a theoretical analysis to prove the better imperceptibility of BW. To empirically access the imperceptibility of watermarks, we further designe a rational metric to evaluate different watermark methods. Extensive experiments demonstrate that BW excels in imperceptibility and achieves competitive performance in other key aspects of watermark.

Our main contributions are as follows:

- We take imperceptibility as the starting point and propose Balanced watermark. BW balances multiple features of the watermark, possessing a certain level of competitiveness in each feature.

- We theoretically analyze how Balanced Watermark improves approximate probability unbias and imperceptibility of watermark.

- We empirically demonstrate the imperceptibility and effectiveness of BW across different datasets and LLMs.

2 Related Work

2.1 LLM-Genrated Text Watermark

In order to distinguish between texts generated by models and those composed in natural language, some scholars try to find a more accurate detector (Gehrmann et al., 2019; Guo et al., 2023; Mitchell et al., 2023; Rodriguez et al., 2022), while others decided to tackle the problem at the source, adding watermarks to the LLM-generated text. In the domain of watermark for LLM-generated text, there exist three predominant approaches: backdoor watermarks that modeify parameters of LLM (Adi et al., 2018; Peng et al., 2023); reweighting watermarks that add bias in the output probabilities of LLM (Kirchenbauer et al., 2023; Lu et al., 2024; Fu et al., 2024; Zhao et al., 2023a; Hu et al., 2023); text watermark, which is achieved through modifications made to the text itself (Yang et al., 2022; Li et al., 2023).

Reweighting watermark emerges as a focal point of current watermark research. Kirchenbauer et al. (2023) propose KGW, add a fixed value on the logits of green token in the LLM vocabulary. The definition of green token fully introduces randomness and the uniqueness of the secret key. This makes the output of LLM biased, and detection only needs to count the frequency of green token occurrence. Zhao et al. (2023a) only apply the uniqueness for the selection of green token, increases its detectability and robustness. Fu et al. (2024) explore the method for improving KGW in conditional text generation tasks.

2.2 Imperceptibility in Watermark

About imperceptibility in watermark, its function is to ensure that the watermark is imperceptible to observation by non-watermarking means. UW, starting from this perspective, proposed two novel reweighting methods to modify the output probabilities of LLMs, thereby realizing the embedding of watermarks (Hu et al., 2023). Additionally, UW introduce the concept of unbiased watermark, demonstrating an optimization goal for the imperceptibility. SIR has trained a logits bias generator to implement the addition of watermarks (Liu et al., 2023). The concept of unbiased watermark also introduced to it when training generator.

3 Methodology

3.1 Watermark and Imperceptibility

A LLM forms a complete piece of text by generating each token in a loop. For any input text $X = \{x_1, x_2, ..., x_{|X|}\}$, LLM will generate a probability distribution $p_\theta(t_i|X)$ over the vocabulary $\mathcal{V}$, θ represents the parameters of LLM and $t_i \in \mathcal{V}$. Subsequently, the LLM samples from $p_\theta(t_i|X)$ to obtain newly generated token.

We regard the act of embedding a watermark as a modification to $p_\theta(t_i|X)$. Given a watermark method w, the watermarking process can be viewed as:

$$p_{\hat{\theta}}(t_i|X) = p_\theta(t_i|X) + p_w(t_i|X) \quad (1)$$

$p_{\hat{\theta}}(t_i|X)$ denotes the probability distribution post-watermarking; $p_w(t_i|X)$ represents the probability bias introduced by the watermark, and $|p_w(t_i|X)|$ is regarded as the watermark strength at this point.

The imperceptibility of watermark can be described as the degree of change of probability distribution before and after watermarking. Thus, the perfect imperceptibility requires $p_w(t_i|X) = 0$, but this setting will prevent the embedding of a watermark. The discrete nature of the text allows us to relax the imperceptibility condition. For a input dataset D, the perfect imperceptibility over D is regarded as:

$$p_{\hat{\theta}}(t_i|D) = p_\theta(t_i|D) \quad (2)$$

It requires $p_w(t_i|D) = 0$ at this time. We further design the metric for measuring watermark imperceptibility as follows:

$$\mathcal{I}_w := min\{\frac{1}{|p_w(t_i|D)|}\}, t_i \in \mathcal{V} \quad (3)$$

A larger $\mathcal{I}_w$ indicates better imperceptibility of the watermark.

3.2 Balanced Watermark

BW achieves imperceptibility enhancement with appropriate design while keeping the watermark strength unchanged. BW consists of two steps: Word Frequency Green list Selection (WFGS) and Logits Bias (LB). WFGS determines how the watermark information is transformed into textual information, while LB dictates how the watermark information is embedded. The complete details for BW are shown in Algorithm 1.

Algorithm 1 Balanced Watermark

Input: Input sequence $X = \{x_1, x_2, ..., x_{|x|}\}$, Large Language Model LLM, secret key $\mathcal{K}$, logits bias $\delta > 0$.

Output: Watermarked text

Count word frequencies from large amounts of text generated by LLM;
Sort tokens on $\mathcal{V}$ by word frequencies and construct map function M;
Apply $\mathcal{K}$ as a random seed, randomly and uniformly partition the vocabulary $\mathcal{V}$ into lists $\mathcal{A}$ and $\mathcal{B}$.
for $i \leftarrow 1$ **to** ... **do**
    Based on the input sequence X and pre-output $y_{<i}$, LLM get a logits distribution $l^{(i)}$ on the vocabulary $\mathcal{V}$;
    if $M(y_{i-1}) = 1$ **then**
        $\mathcal{G} = \mathcal{A}, \mathcal{R} = \mathcal{B}$
    else if $M(y_{i-1}) = 0$ **then**
        $\mathcal{G} = \mathcal{B}, \mathcal{R} = \mathcal{A}$
    end if
    Add a fixed bias value δ to all green tokens logits, then obtain a new probability distribution $p_w^{(i)}$ over the vocabulary $\mathcal{V}$ through softmax;
    Sample the next token y_i from $p_w^{(i)}$.
end for

Step 1: Word Frequency Green list Selection. WFGS constructs a mapping function M to allocate the selection of the green list reasonably.

To construct M, we first obtain a large set of non-watermarked texts generated by the LLM. We then statistically analyzed the frequency of each token t in $\mathcal{V}$ across these texts. Based on the word frequency, we form an ordered list $\{t_1, t_2, ..., t_{|V|}\}$. According to this ordered list, we construct M as:

$$M(t_i) = \begin{cases} 1, & i\%2 = 0 \\ 0, & i\%2 \neq 0 \end{cases} \quad (4)$$

Prior to generation, we also need to prepare lists $\mathcal{A}$ and $\mathcal{B}$, which are obtained by randomly and evenly partitioning $\mathcal{V}$ according to a secret key $\mathcal{K}$.

The selection of the green list $\mathcal{G}$ when inputting X is:

$$\mathcal{G} = \begin{cases} \mathcal{A}, & M(x_{|X|}) = 1 \\ \mathcal{B}, & M(x_{|X|}) = 0 \end{cases} \quad (5)$$

We regard the other list that did not become $\mathcal{G}$ as the red list $\mathcal{R}$.

Step 2: Logits Bias. The purpose of LB is to enhance the probability of green list tokens appearing by $\mathcal{G}$. We implement this by adding a constant δ to the green token logits. The logits are the intermediate distributions obtained by the LLM when generating probability distributions, and the logits after the softmax are $p_\theta(t_i|X)$.

For the logits $l^{(i)}$ obtained at time i, the watermark probability distribution $p_w^{(i)}$ can be defined by the following formula:

$$p_w^{(i)} = \begin{cases} \frac{exp(l_k^{(i)}+\delta)}{\sum_{j\in\mathcal{R}} exp(l_j^{(i)})+\sum_{j\in\mathcal{G}} exp(l_j^{(i)}+\delta)}, & k \in \mathcal{G} \\ \frac{exp(l_k^{(i)})}{\sum_{j\in\mathcal{R}} exp(l_i^{(t)})+\sum_{j\in\mathcal{G}} exp(l_j^{(i)}+\delta)}, & k \in \mathcal{R} \end{cases} \tag{6}$$

Detection The detection of BW is straightforward. We simulate the process of WFGS to calculate the number of green tokens in a sentence. Then, we calculate z-statistic as the criterion for determining the existence of a watermark.

4 Theoretical Analysis

In this section, we prove that under the same watermark strength, the imperceptibility of BW is higher than UNIW.

To simplify formulations, we define U_D^t as follows:

$$U_D^t = \sum_{X\in D} p_w(t|X) = |D| \cdot p_w(t|D) \tag{7}$$

$p_w(t|X)$ is the probability of the watermarked LLM generating t upon input X, and D is the set of some possible X.

According to equation 3, the imperceptibility of a watermark only needs to pay attention to the token with the max probability bias. Considering solely this token t_m, the strength of a watermark is defined as:

$$\mathcal{S}_w := \sum_{X\in D} |p_w(t_m|X)| \tag{8}$$

$p_w(t_m|X)$ represents the probability bias for the token t_m when inputting X.

The imperceptibility of the watermark can be defined as:

$$\mathcal{I}_w := |\frac{1}{p_w(t_m|D)}| = \frac{|D|}{|U_D^{t_m}|} \tag{9}$$

Regardless of whether the watermark is UNIW or BW, there are only two scenarios for input X: assigning t_m to $\mathcal{G}$ or to $\mathcal{R}$. We form a dataset $D_\mathcal{G}$ consisting of all inputs X that assigning t_m to $\mathcal{G}$, and correspondingly, dataset $D_\mathcal{R}$ for R. The relationship between D, $D_\mathcal{G}$ and $D_\mathcal{R}$ can be represented by the following formula:

$$D_\mathcal{G} = D \setminus D_\mathcal{R} \tag{10}$$

Based on the principle to enhance the probability of green list tokens appearing, for any $X_\mathcal{G} \in D_\mathcal{G}$, $p_w(t_m|X_\mathcal{G}) > 0$. Similarly, $p_w(t_m|X_\mathcal{R}) < 0$.

Therefore, we transform Equation 8 into:

$$\mathcal{S}^{t_m} = |U_{D_\mathcal{G}}^{t_m}| + |U_{D_\mathcal{R}}^{t_m}| \tag{11}$$

According to equation 10, we can deduce:

$$U_D^{t_m} = U_{D_\mathcal{R}}^{t_m} + U_{D_\mathcal{G}}^{t_m} \tag{12}$$

UNIW employs a fixed green list $\mathcal{G}$, making that t_m is consistently assigned to the same list. Assuming t_m belongs to $\mathcal{G}$ in UNIW, we have a equation:

$$U_{D_\mathcal{R}}^{t_m} = 0 \tag{13}$$

Therefore, $\mathcal{I}_{UNIW}$ and $\mathcal{S}_{UNIW}$ have the following relationship:

$$\mathcal{I}_{UNIW} = \frac{|D|}{|U_{D_\mathcal{G}}^{t_m}| + |U_{D_\mathcal{R}}^{t_m}|} = \frac{|D|}{\mathcal{S}_{UNIW}^{t_m}} \tag{14}$$

In BW, we make the probability of t_m belonging to either $\mathcal{R}$ or $\mathcal{G}$ about 1/2 by WFGS. It is evident that we have a fundamental inference in BW:

$$|U_{D_\mathcal{G}}^{t_m}| + |U_{D_\mathcal{R}}^{t_m}| > |U_{D_\mathcal{R}}^{t_m} + U_{D_\mathcal{G}}^{t_m}| \tag{15}$$

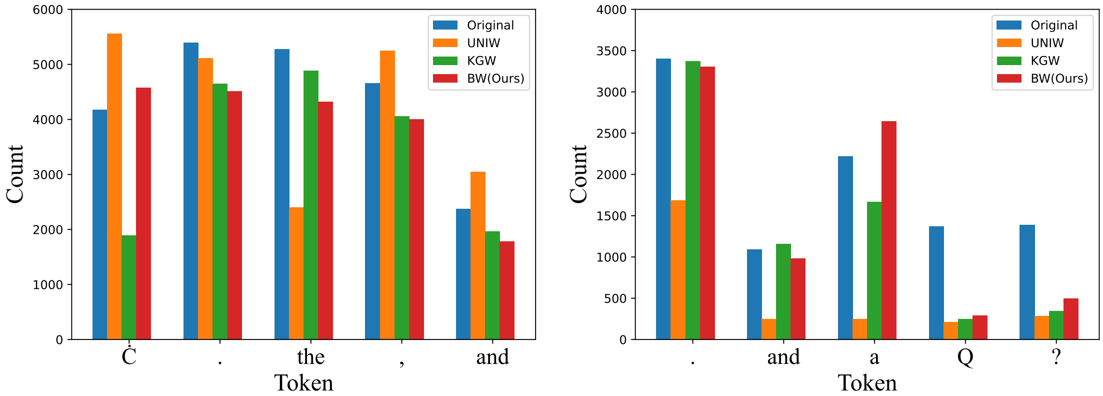
The relationship between the imperceptibility and watermark strength of BW is:

$$\mathcal{I}_{BW} = \frac{|D|}{|U_{D_\mathcal{G}}^{t_m} + U_{D_\mathcal{R}}^{t_m}|} > \frac{|D|}{\mathcal{S}_{BW}^{t_m}} \tag{16}$$

Assuming that BW and UNIW have the same watermark strength, that is, $S_{UNIW}^{t_m} = S_{BW}^{t_m}$. Under this assumption, based on Equations 14 and 16, we can deduce:

$$\mathcal{I}_{BW} > \frac{|D|}{\mathcal{S}_{BW}^{t_m}} = \frac{|D|}{\mathcal{S}_{UNIW}^{t_m}} = \mathcal{I}_{UNIW} \tag{17}$$

The equation 17 substantiates the conclusion we initially proposed in this section: under the same watermark strength, the imperceptibility of BW is higher than UNIW.



(a) OPT-2.7b + C4

(b) OPT-2.7b + LFQA

Figure 2: Comparisons of the words count produced in the corpus after adding different watermarks to OPT-2.7b with several high-frequency words under C4 and LFQA. The blue bars with shadows represent the original word frequencies that the watermark word frequencies need to fit.

5 Experiments

In this section, we conduct extensive experiments and answer the following questions: 1) How does BW perform in imperceptibility? We evaluate the imperceptibility of BW on various datasets using different models and compared it with other watermarking methods. 2) How does BW perform in terms of other features required for watermarking? We conduct extensive experiments to demonstrate that BW is equally excellent in other features required for watermarking. 3) What impact does different green list ratios have on the imperceptibility of BW? We conduct experiments with different green list ratios to analyze the changes in imperceptibility.

5.1 Implementation Details

Datasets To evaluate the performance of BW, we randomly select 500 texts from the news-like subset of the C4 dataset (Raffel et al., 2020)[2] and LFQA (Krishna et al., 2023)[3]. C4 is the dataset utilized in the KGW (Kirchenbauer et al., 2023), representing a general generation task. LFQA is the dataset employed by UNIW (Zhao et al., 2023a), which is a commonly used Question Answering (QA) task dataset. We extract the first 30 tokens from each text in C4 as the input. For LFQA, we extract the question portion of each example as the input.

Models We employ OPT-2.7b (Zhang et al., 2022) and Llama3-8b (AI@Meta, 2024) as the generative models. OPT-2.7b is a commonly utilized generative model adoped by KGW (Kirchenbauer et al., 2023), whereas Llama3-8b is a recently released Large Language Model. During each generation, we employ sampling as the decoding strategy and produce a maximum of 200 tokens. For BW, we generate corresponding frequency files for both models using the C4 dataset in the absence of watermarking.

Baselines We compare two watermark methods to test the performance of BW. UNIW (Zhao et al., 2023a) utilizes fixed green list, achieving optimal performance in multiple aspects, but greatly compromises imperceptibility. KGW (Kirchenbauer et al., 2023) enhances a certain degree of imperceptibility through a random green list, but it has a certain degree of randomness, while also undermining the invisibility and robustness of UNIW. In our conjecture, BW retains certain advantages of UNIW, thereby exhibiting better than KGW in these respects. We set the default parameters with the green list ratio γ set to 0.5 and the logits bias δ set to 2 for UNIW, KGW and BW.

5.2 Imperceptibility Comparison

A straightforward and feasible method to evaluate imperceptibility is to analyze the changes in word frequency, which we display in Figure 2. In Figure 2, we select high-frequency tokens from the OPT-2.7b vocabulary for display.

In Subfigure 2a, we find that the word frequency of BW often approaches the original word frequency more closely than that of UNIW. Particularly, for the token ***the***, UNIW is highly incon-

[2]https://huggingface.co/datasets/allenai/c4

[3]https://drive.google.com/drive/folders/1mPROenBB0fzLO9AX4fe71k0UYv0xt3X1

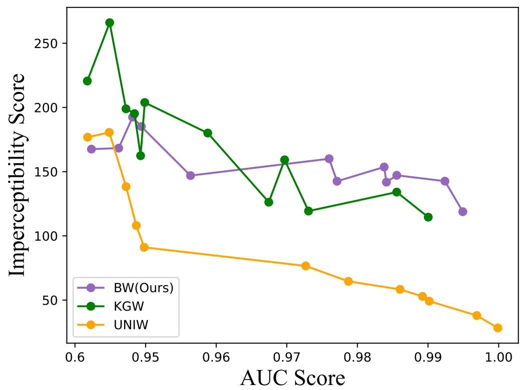


Figure 3: Imperceptibility of different watermarks corresponding to various AUC scores. Llama3-8b is the generative model and C4 is the dataset.

sistent with the original word frequency, making its imperceptibility very low. For KGW, although it is slightly closer to the original distribution on multiple tokens compared to BW, for the token $\dot{\boldsymbol{C}}$, the difference between KGW and the original distribution is even higher than that of UNIW.

In Subfigure 2b, the word frequency of BW is always closer to the original word frequency than UNIW. We also find that on the LFQA dataset, all watermarks cause severe differences in the word frequency of ***Q*** and ***?***. At this time, BW consistently approaches the original word frequency more closely, demonstrating superior imperceptibility.

For a clearer analysis the imperceptibility of BW, we examine the correlation trend between imperceptibility and detectability. We utilize the formula mentioned in Equation 3 as a clear numerical metric of the watermark imperceptibility. Due to the significant noise introduced by the low-frequency words in the corpus, we only account for the frequency changes of the top 20 most frequent words in the vocabulary. For detectability, we use the AUC Score for ROC curves, and we control it by setting different logits bias δ. The result is shown in Figure 3

We observe that under varying AUC scores, the imperceptibility of BW is relatively stable, showing no significant variation. At the same time, the imperceptibility of both KGW and UNIW declines as the AUC score increases.

At lower AUC scores, the imperceptibility of KGW is significantly higher than that of UNIW and BW. UNIW and BW both employ a fixed vocabulary partitioning, which results in a considerable degradation of imperceptibility once watermarking is introduced.

At AUC scores above 0.97, we observe that the imperceptibility of BW is consistently higher than that of KGW. The cause of this phenomenon may be: Although KGW employs a random setting to theoretically equate the probabilities of each token being classified as $\mathcal{G}$ or $\mathcal{R}$ during generation, it does not take into account the impact of word frequency. Therefore, at high AUC Score, the watermark information becomes more pronounced, and the resulting low imperceptibility due to this factor becomes increasingly evident.

We believe that high detectability is a necessary condition for the application of watermarks. It can be seen that BW is the only watermark that can maintain high imperceptibility under high detectability.

5.3 Watermark Features Comparison

In this section, we present the performance of BW in other watermark features, including detectability, invisibility, robustness, and usability. Ultimately, we demonstrate the superior comprehensive performance of BW.

Detectability and Invisibility The results with detectability and invisibility are presented in Table 1. For the detectability, we calculate the True Positive Rate (TPR) at False Positive Rates of 1% and 10%. Concurrently, we compute the AUC score for ROC curves for watermark detection. Following the work of Kirchenbauer et al. (2023), we employ perplexity (PPL) to assess invisibility, which means the quality of the watermarked text.

As shown in Table 1, the best performance in detectability metrics is either exhibited by UNIW or BW. This substantiates that the setting of BW does not significantly reduce the detectability of UNIW. At the same time, he random green list of KGW causes the watermark information to be added to the text without stability, resulting in detection performance slightly lower than that of UNIW and BW.

In terms of invisibility, UNIW consistently exhibits the best performance, whereas BW consistently outperforms KGW. This proves that a more stable green list will lead to better text quality.

It can be inferred that BW almost perfectly maintains the excellent detectability of a fixed green list, while also preserving the certain excellent invisibility.

Model	Method	C4				LFQA			
		1%FPR↑	10%FPR↑	AUC↑	PPL↓	1%FPR↑	10%FPR↑	AUC↑	PPL↓
OPT-2.7b	Original	✗	✗	✗	4.321	✗	✗	✗	7.280
	UNIW	0.942	0.984	0.995	6.160	0.818	0.960	0.981	14.651
	KGW	0.894	0.970	0.988	7.047	0.934	0.986	0.994	10.308
	BW(Ours)	0.954	0.982	0.992	6.610	0.954	0.990	0.996	9.081
Llama3-8b	Original	✗	✗	✗	3.293	✗	✗	✗	3.186
	UNIW	0.944	0.970	0.989	4.038	0.984	0.996	0.997	3.474
	KGW	0.808	0.924	0.965	5.262	0.760	0.960	0.981	4.608
	BW(Ours)	0.930	0.974	0.987	4.470	0.940	0.984	0.996	3.735

Table 1: The detectability and invisibility performance of various methods on different models for C4 and LFQA. ↑ means higher metrics are better. ↓ means lower metrics are better.

Metric	Method	Model	
		OPT-2.7b	Llama3-8b
$V(it/s)$	UNIW	771.81	775.8
	KGW	41.70	33.30
	BW(Ours)	50.00	45.40
$M(KiB)$	UNIW	877.60	8867.84
	KGW	877.60	8867.84
	BW(Ours)	1553.93	9051.67

Table 2: Comparisons of the detect speed on OPT-2.7b and Llama3-8b.V represents the detection speed, and M represents the additional memory required for detection. it/s signifies the number of texts detected per second.

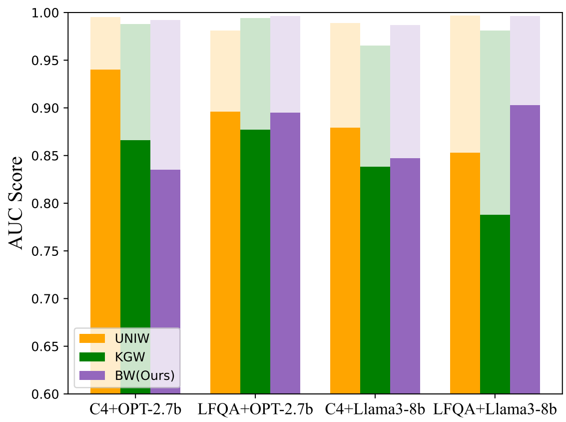


Figure 4: Results of paraphrasing various watermark texts by DIPPER. OPT-2.7b and Llama3-8b are generative models, with C4 and LFQA as the datasets. The transparent bars represent the AUC scores in the original state without any attack. The solid bars represent AUC scores after being subjected to a DIPPER attack.

Usability We test the detection speed and memory consumption of two models under different watermarks, with the results depicted in Table 2. The detection speed of BW is somewhat reduced compared to UNIW, yet it remains superior to that of KGW. In terms of memory consumption, BW occupies the most memory.

However, from a practical standpoint, both a detection speed of over 30 times per second and a memory consumption of less than 10 MB are acceptable to users.

Robustness To evaluate the robustness of four watermarks, we utilize DIPPER (Krishna et al., 2023) to paraphrase the watermarked texts, testing the extent of the decline in AUC scores. The result of robustness is shown in Figure 4.

As shown in the figure 4, BW performs better on the LFQA dataset, showing comparable robustness to UNIW when using OPT-2.7b, and demonstrating the best robustness when using Llama3-8b. Another point worth noting is that under the same dataset, BW exhibits better robustness when using Llama3-8b. Although BW has the poorest robustness under C4 and OPT-2.7b, it is more adaptable to complex generation conditions and LLMs, which makes it more competitive in practical applications.

Comprehensive Performance We comprehensively evaluate the three watermarks based on their five characteristics. The overall results are shown in Figure 5.

The imperceptibility of BW is the best, in contrast, UNIW is the worst. From the perspective of detection performance, the detectability of the three methods is close, but the random green list of KGW leads to instability, resulting in slightly worse detection performance. Robustness and invisibility are advantages of a fixed green list, so BW using the balanced green list is slightly worse than UNIW. The usability of BW and KGW may seem to be reduced significantly, but in reality, the usability of all three watermarks is acceptable to humans and practical.

5.4 Green List Ratio Analysis

In this experiment, we configure BW such that the ratio of the $\mathcal{A}$ and $\mathcal{B}$ lists derived from the

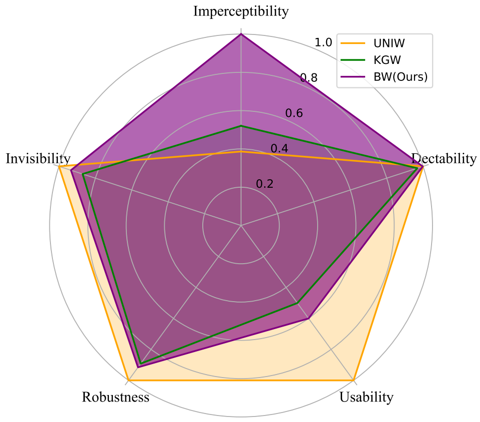


Figure 5: The comparative analysis of the comprehensive performance of BW and other watermarking techniques. The basis for the plotting is the results obtained from various indicators in our experimental section.

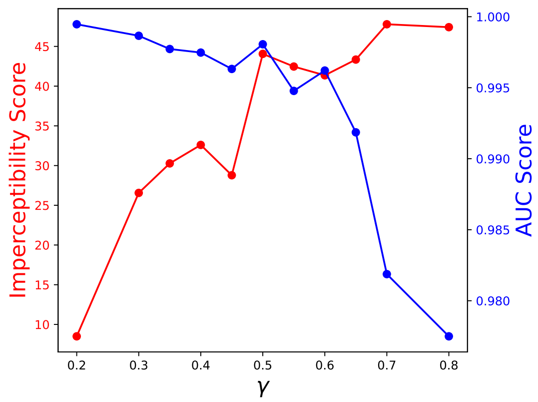


Figure 6: Comparison of imperceptibility and detectability of BW under different γ. OPT-2.7b is the generative model, and LFQA is the dataset. γ is a hyperparameter introduced in the KGW (Kirchenbauer et al., 2023), denotes the green list ratio.

vocabulary partition is consistently aligned with γ.

As illustrated in Figure 6, the imperceptibility of BW increases with the enhancement of γ. It is noteworthy that AUC score decreases at the same time.

At low γ, the use of a fixed logits bias leads to an extremely high variation in the probability of a few green list tokens, resulting in significant degradation of imperceptibility. The increase in γ results in the even distribution of logits bias across a greater number of tokens, thereby enhancing imperceptibility. From a certain perspective, an increase in γ leads to a reduction in watermark strength, which in turn results in a decrease in AUC scores.

Prompt	Q: How many hours of sleep should we get?\nA:		A:B	z-score	p-value
Original	Seven hours a night is the recommended maximum for adults. Children need about nine or 10 hours.				
UNIW	There is no set rule. Too much sleep, too little sleep, and an insufficient amount of sleep are all on the spectrum. Ideally, adults should get between 7 and 9 hours of sleep a night.	There is no set rule. Too much sleep, too little sleep, and an insufficient amount of sleep are all on the spectrum. Ideally, adults should get between 7 and 9 hours of sleep a night.	3	3.16	7e-4
BW	To get the most restful sleep, try to get 7-9 hours of sleep. In general, adults should aim for at least 8-9 hours of sleep.	To get the most restful sleep, try to get 7-9 hours of sleep. In general, adults should aim for at least 8-9 hours of sleep.	1.2	3.3	4e-4

Figure 7: An example output with Unigram watermark (UNIW) (Zhao et al., 2023a) and our proposed Balanced Watermark (BW) on a question in LFQA. UNIW divides the vocabulary into List A and List B, and selects List A as the green list. UNIW increases the green token probability and decreases the red token probability, thereby embedding the watermark. This results in an overall word frequency anomaly, reducing imperceptibility. BW ensures the preservation of detectability while balancing Lists A and B, thereby enhancing imperceptibility.

5.5 Case Study

As shown in Figure 7, when UNIW and BW use the identical $\mathcal{A}$ list and $\mathcal{B}$ list, the proportion of the A list to the B list in BW is noticeably more balanced. z-score and p-value are statistical measures obtained from the green list tokens. Analyzing these two statistical measures, UNIW and BW exhibit similar detectability.

6 Conclusion

In this paper, we propose a new watermark Balanced Watermark (BW) for LLM-generated text. BW substantially improves imperceptibility based on its original watermark while retaining certain performance attributes of the original, earning high marks in overall performance evaluation. To effectively evaluate imperceptibility, a metric for the assessment of imperceptibility is introduced for the first time. We corroborate the enhancement of BW in imperceptibility by comparing theoretical analysis, actual word frequency changes, and scores of imperceptibility metric. At the same time, for other watermarking feature, we demonstrate the superiority of BW through extensive experimentation.

7 Limitations

One limitation in our study is that we only use the most advanced watermarking attack method currently available to analyze the robustness of BW. We can try some other watermarking attack methods to analyze the robustness of BW in the future. Another limitation is that we do not test BW with models larger than 10B, only analyze OPT-2.7b and Llama3-8b due to computational power limitations. In the future, it would be possible to apply BW to models of different sizes to more effectively analyze the impact of model size on the watermarking effect. We suppose that watermark design is a game of trade-offs, where enhancing the performance of a single watermark feature inevitably leads to a decline in other watermark features. We hope that future watermark research can more comprehensively consider various performance aspects, leading to the design of watermarks with superior performance.

References

Sahar Abdelnabi and Mario Fritz. 2021. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *42nd IEEE Symposium on Security and Privacy, SP 2021, San Francisco, CA, USA, 24-27 May 2021*, pages 121–140. IEEE.

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pages 1615–1631.

AI@Meta. 2024. Llama 3 model card.

Hussam Alkaissi and Samy I McFarlane. 2023. Artificial hallucinations in chatgpt: implications in scientific writing. *Cureus*, 15(2).

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, IHW 2001, Pittsburgh, PA, USA, April 25-27, 2001, Proceedings*, volume 2137 of *Lecture Notes in Computer Science*, pages 185–199. Springer.

N Editorials. 2023. Tools such as chatgpt threaten transparent science; here are our ground rules for their use. *Nature*, 613(7945):612.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. 2023. Publicly detectable watermarking for language models. *IACR Cryptol. ePrint Arch.*, page 1661.

Yu Fu, Deyi Xiong, and Yue Dong. 2024. Watermarking conditional text generation for AI detection: Unveiling challenges and a semantic-aware watermark remedy. In *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 18003–18011. AAAI Press.

Sebastian Gehrmann, Hendrik Strobelt, and Alexander M. Rush. 2019. GLTR: statistical detection and visualization of generated text. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28 - August 2, 2019, Volume 3: System Demonstrations*, pages 111–116. Association for Computational Linguistics.

Biyang Guo, Xin Zhang, Ziyuan Wang, Minqi Jiang, Jinran Nie, Yuxuan Ding, Jianwei Yue, and Yupeng Wu. 2023. How close is chatgpt to human experts? comparison corpus, evaluation, and detection. *arXiv preprint arXiv:2301.07597*.

Xuanli He, Qiongkai Xu, Lingjuan Lyu, Fangzhao Wu, and Chenguang Wang. 2022. Protecting intellectual property of language generation apis with lexical watermark. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 10758–10766. AAAI Press.

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. Semstamp: A semantic watermark with paraphrastic robustness for text generation. *CoRR*, abs/2310.03991.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased watermark for large language models. *CoRR*, abs/2310.10669.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Yuhang Li, Yihan Wang, Zhouxing Shi, and Cho-Jui Hsieh. 2023. Improving the generation quality of watermarked large language models via word importance scoring. *CoRR*, abs/2311.09668.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2023. A semantic invariant robust watermark for large language models. *CoRR*, abs/2310.06356.

Fang Liu, Yang Liu, Lin Shi, Houkun Huang, Ruifeng Wang, Zhen Yang, and Li Zhang. 2024a. Exploring and evaluating hallucinations in llm-powered code generation. *CoRR*, abs/2404.00971.

Xuannan Liu, Peipei Li, Huaibo Huang, Zekun Li, Xing Cui, Jiahao Liang, Lixiong Qin, Weihong Deng, and Zhaofeng He. 2024b. Fakenewsgpt4: Advancing multimodal fake news detection through knowledge-augmented lvlms. *CoRR*, abs/2403.01988.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. *CoRR*, abs/2401.13927.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. 2024. An entropy-based text watermarking detection method. *CoRR*, abs/2403.13485.

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, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 24950–24962. PMLR.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Wenjun Peng, Jingwei Yi, Fangzhao Wu, Shangxi Wu, Bin Zhu, Lingjuan Lyu, Binxing Jiao, Tong Xu, Guangzhong Sun, and Xing Xie. 2023. Are you copying my model? protecting the copyright of large language models for eaas via backdoor watermark. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 7653–7668. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21:140:1–140:67.

Juan Diego Rodriguez, Todd Hay, David Gros, Zain Shamsi, and Ravi Srinivasan. 2022. Cross-domain detection of gpt-2-generated technical text. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1213–1233.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971.

Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. 2022. Tracing text provenance via context-aware lexical substitution. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11613–11621. AAAI Press.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. 2023. Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2092–2115. Association for Computational Linguistics.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. Opt: Open pre-trained transformer language models. *Preprint*, arXiv:2205.01068.

Yizhou Zhang, Karishma Sharma, Lun Du, and Yan Liu. 2024. Toward mitigating misinformation and social media manipulation in LLM era. In *Companion Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, Singapore, May 13-17, 2024*, pages 1302–1305. ACM.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023a. Provable robust watermarking for ai-generated text. *CoRR*, abs/2306.17439.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. 2023b. Protecting language generation models via invisible watermarking. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 42187–42199. PMLR.

A Robustness

We present the specific numerical details in Figure 4 using Table 4. Same as Figure 4, there is a noticeable decrease in the robustness of UW. The robustness of BW and KGW both keep the AUC score above 0.75. UNIW always has the best robustness.

δ	AUC Score			$\mathcal{I}$		
	KGW	UNIW	BW(Ours)	KGW	UNIW	BW(Ours)
0.2	0.661	0.663	0.681	220.60	176.83	167.56
0.4	0.772	0.770	0.817	266.03	180.60	168.31
0.6	0.853	0.853	0.885	198.96	138.38	192.47
0.8	0.895	0.904	0.929	195.18	108.07	185.13
1.0	0.925	0.943	0.956	162.42	91.06	146.94
1.2	0.946	0.973	0.976	203.84	76.61	160.05
1.4	0.959	0.979	0.977	180.07	64.60	142.53
1.6	0.967	0.986	0.984	126.22	58.33	153.62
1.8	0.970	0.989	0.984	156.21	52.84	141.83
2.0	0.973	0.990	0.986	119.33	49.14	147.16
3.0	0.986	0.997	0.992	134.08	37.98	142.60
5.0	0.990	1.000	0.995	114.55	28.31	118.81

Table 3: In the C4 dataset, using the Llama3-8b model, the AUC scores and imperceptibility of different watermarks at different δ.

Dataset	Method	Normal	Attack
C4	UNIW	0.989	0.879
	KGW	0.965	0.838
	UW	0.974	0.587
	BW(Ours)	0.987	0.847
LFQA	UNIW	0.997	0.853
	KGW	0.981	0.788
	UW	0.956	0.592
	BW(Ours)	0.996	0.903

Table 4: The robustness details of two datasets on the Llama3-8b. Normal represents the AUC score in the absence of attacks. Attack indicates the AUC score after being subjected to a DIPPER attack.

Dataset	Method	Normal	Attack
C4	UNIW	0.995	0.940
	KGW	0.988	0.866
	UW	0.991	0.619
	BW(Ours)	0.992	0.835
LFQA	UNIW	0.981	0.896
	KGW	0.994	0.877
	UW	0.986	0.546
	BW(Ours)	0.996	0.895

Table 5: The robustness details of two datasets on the OPT-2.7b. Normal represents the AUC score in the absence of attacks. Attack indicates the AUC score after being subjected to a DIPPER attack.

For further analysis, we also conduct experiments on the OPT-2.7b. The result is shown in Table 5.

We find that under the same dataset, the robustness of BW is very stable. Another noteworthy point is that KGW exhibits the greatest fluctuation in robustness, similar to its performance in other watermarking characteristics.

B Hyper-Parameters

B.1 δ Analysis

We demonstrate in Figure 3 the impact of different δ on the imperceptibility of three watermarks. The numerical details are shown in Table 3.

We find that the AUC scores of BW perform better than KGW and UNIW at low δ values. This is an unintended good effect, we speculate that the reason for this phenomenon lies in the fact that the green list selected by BW consists of two completely opposing lists. We consider the magnitude of watermark imperceptibility as the disruption of the watermark to the overall LLM distribution. While ensuring the same level of imperceptibility, KGW performs well at low watermark strengths, whereas BW performs better as the watermark strength increases. It can be observed that the imperceptibility of BW is more stable. This makes the design of BW more practically significant.

C γ Analysis

We demonstrate the impact of the green list ratio γ on the detectability and imperceptibility of BW in Figure 6. The numerical details are shown in Table 6.

When γ is low, BW exhibits a high level of de-

γ	AUC Score	$\mathcal{I}$
0.2	0.999	8.504
0.3	0.997	26.568
0.35	0.998	30.272
0.4	0.997	32.599
0.45	0.996	28.782
0.5	0.998	44.060
0.55	0.995	42.465
0.6	0.996	41.360
0.65	0.992	43.343
0.7	0.981	47.787
0.8	0.977	47.422

Table 6: The detail of imperceptibility and detectability of BW under different γ. OPT-2.7b is the generative model, and LFQA is the dataset.

tectability and a lower level of imperceptibility. Low γ confines the fixed watermark strength to a few tokens, severely undermining imperceptibility. When γ is higher, the detectability of BW is somewhat reduced, while imperceptibility increases. The increase in γ does not change the watermark strength, but the amplification of the green list makes the detection difference between watermarked text and non-watermarked text smaller.