Watermarking with Low-Entropy POS-Guided Token Partitioning and Z-Score-Driven Dynamic Bias for Large Language Models

He Li[1,2,3], Xiaojun Chen[*1,2,3], Zhendong Zhao[1,2,3], Yunfei Yang[1,2,3], Xin Zhao[1,2,3], Jingcheng He[1,2,3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]State Key Laboratory of Cyberspace Security Defense, Beijing, China
[3]School of Cyber Security,University of Chinese Academy of Sciences, Beijing, China
{lihe2023,chenxiaojun,zhaozhendong,yangyunfei,zhaoxin,hejingcheng}@iie.ac.cn

Abstract

Texts generated by large language models (LLMs) are increasingly widespread online. Due to the lack of effective attribution mechanisms, the enforcement of copyright and the prevention of misuse remain significant challenges in the context of LLM-generated content. LLMs watermark emerges as a crucial technology to trace the source of AI-generated content. However, most existing watermarking methods reduce the fidelity of semantics. To address this issue, this paper introduces a novel watermarking framework. To enhance the fidelity of semantics, we propose low-entropy POS-guided token partitioning mechanism and z-score-driven dynamic bias mechanism. Moreover, to enhance the robustness against potential bias sparsity exploitation attack, we propose a relative position encoding (RPE) mechanism, which can uniformly distribute bias in the generated text. Evaluated across 6 baselines, 4 tasks, and 5 LLMs under 8 attacks, compared to the KGW, our watermark improves semantic fidelity by 24.53% (RC-PPL) and robustness by 3.75% (F1). Our code is publicly available, facilitating reproducibility in LLM watermarking research.

1 Introduction

In recent years, Large Language Models (LLMs) have made rapid progress, exemplified by models such as LLaMA-3 (Grattafiori et al., 2024), GPT-4 (OpenAI, 2023), Qwen2.5 (Qwen et al., 2025), and DeepSeek-R1 (DeepSeek-AI, 2025b,a). These models have been widely applied across diverse natural language processing (NLP) tasks, including text generation (Raffel et al., 2020), long-form question answering (Hudeček and Dusek, 2023), and story extension (Mostafazadeh et al., 2016). As LLMs continue to advance in these NLP domains, critical research topics have emerged concerning copyright (Gillotte, 2020; Megías et al., 2021), privacy (Patil et al., 2023; Perez et al., 2022), and security (Bender et al., 2021; Mirsky et al., 2023), which have become prominent areas of investigation. Watermarking (Chang et al., 2024; Guo et al., 2024; Lau et al., 2024) has emerged as a pivotal technology for tracing the origin of text outputs (Liu et al., 2024a; Crothers et al., 2023).

Table 1: Comparison of state-of-the-art watermarking methods.

	Quality	Robustness	Detectability	Detection
KGW (Kirchenbauer et al., 2023)	(50.79%)	(0.91)	(1.00)	Only Text
SWEET (Lee et al., 2024)	(36.67%)	(0.89)	(1.00)	Only Text
EWD (Lu et al., 2024)	(41.39%)	(0.91)	(1.00)	Only Text
Unigram (Zhao et al., 2023)	(46.69%)	(0.95)	(1.00)	Only Text
DiPMark (Wu et al., 2023)	(23.57%)	(0.77)	(1.00)	Only Text
EXP (Aaronson and Kirchner., 2022)	(186.69%)	(0.85)	(1.00)	Only Text
SynthID (Dathathri et al., 2024)	(174.32%)	(0.83)	(1.00)	Only Text
SIR (Liu et al., 2023)	(31.86%)	(0.83)	(0.99)	Only Text
Unbiased (Hu et al., 2023)	(18.71%)	(0.77)	(1.00)	Model+Text
Ours	(22.97%)	(0.98)	(1.00)	Only Text

Note: ◔◓◑● represent a gradual increase in performance. Higher ACC and Avg F1, lower RC-PPL and inference Time indicate superior performance. Evaluated on Qwen-2.5 with 300 tokens at temperature 1.0. Quality, robustness, and detectability are measured by RC-PPL, AVG F1, and ACC, respectively.

However, the existing watermarking methods (as shown in Fig.1) do not consider the determinacy of different parts of speech, and parts of speech with high determinacy (low entropy) should often not be modified because the core semantics of a sentence are generally determined by these words. Consider KGW (Kirchenbauer et al., 2023), a mainstream watermarking approach that categorizes vocabulary into a red-green list and significantly modifies token sampling probabilities. However, this method overlooks the determinacy disparities among different parts of speech, thus declines in semantic fidelity. The recent SIR framework (Liu et al., 2023) attempts to enhance semantic fidelity by aligning watermark logits with semantic embeddings generated by an auxiliary LLM. Nevertheless, it fails to modulate vocabulary selection based on the entropy values of distinct parts of speech.

In this paper, we introduce a watermarking

[*]Xiaojun Chen is the corresponding author.

Findings of the Association for Computational Linguistics: EMNLP 2025, pages 4840–4859
November 4-9, 2025

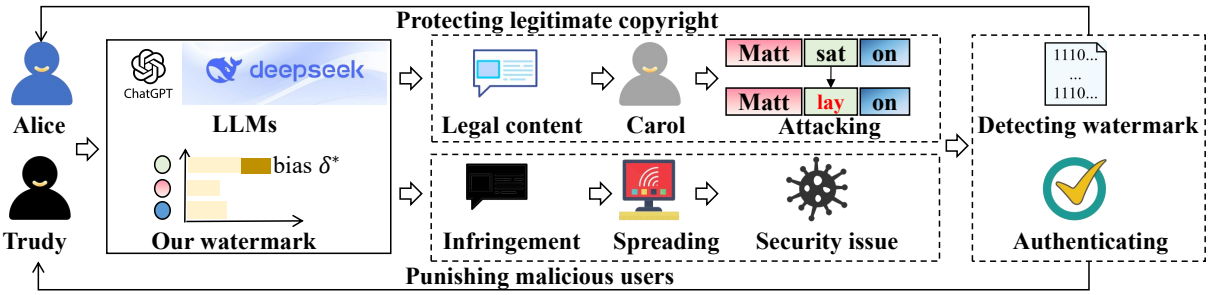


Figure 1: The application scenarios of our watermark.

framework with low-entropy POS-guided token partitioning and z-score-driven dynamic bias for large language models (LLMs). As illustrated in Fig. 1, our method is designed for applications where it can protect legitimate copyright and deter malicious users through the implementation of our watermark. The key contributions of our work are as follows:

• **Low-Entropy POS-Guided Token Partitioning:** We utilize entropy statistics based on part-of-speech (POS) to partition vocabulary list. By excluding blue list tokens from logit perturbations, it mitigates grammatical distortions.

• **Z-Score-Driven Dynamic Bias:** We propose a z-score-driven dynamic bias for high-entropy POS, adaptively adjusting bias intensity to avoid over-distorting token distributions.

• **Comprehensive Evaluation and Generalization:** We conduct extensive experiments involving 6 watermarking methods, 4 tasks, and 5 LLMs under 8 types of adversarial perturbations. Experimental results demonstrate that compared to the KGW method, our framework achieves a 24.53% (PC-PPL) improvement in semantic fidelity and a 3.75% (F1) enhancement in robustness.

2 Background

2.1 Text Generation of LLMs

As shown in Fig.2(a), LLMs compute the logits as $logits_{LLM}(\cdot|x_{0:t-1})$ based on the input text sequence $x_{0:t-1}$. The probability distribution is obtained by applying the $Softmax(\cdot)$ operation to $logits_{LLM}(\cdot|x_{0:t-1})$, and the next token x_t is sampled according to probability distribution p:

$$p = Softmax\left(logits_{LLM}\left(\cdot|x_{0:t-1}\right)\right) \quad (1)$$

2.2 KGW Watermark Generation

Divide Red-Green List. As shown in Fig.2(b), given a vocabulary of size $|V|$ (typically $|V| \geq 50000$), a prefixed hash H_p of the previous token x_{t-1} seeds a random number generator. This generator partitions the vocabulary into a "green list" G ($\gamma|V|$ tokens) and a "red list" R ($(1-\gamma)|V|$ tokens).

Introduce Bias into Green Tokens. For tokens in the green list, their logits are adjusted by a fixed bias δ:

$$logits^w(\cdot|x_{0:t-1}) = logits(\cdot|x_{0:t-1}) + \delta \quad (2)$$

Here, $logits^w$ represents the watermarked logits.

Token Sampling. The watermarked token x_t^w is sampled from the watermarked probability distribution $\hat{p}$:

$$\hat{p} = Softmax\left(logits^w\left(\cdot|x_{0:t-1}\right)\right) \quad (3)$$

2.3 KGW Watermark Detection

As shown in Fig.2(c), given a text sequence $x_{0:T}$, count the number of green tokens $|s|_G$, the detection statistic is computed as:

$$\text{Z-score} = \frac{|s|_G - \gamma T}{\sqrt{T\gamma(1-\gamma)}} \quad (4)$$

γ denotes the proportion of the green token. The text is classified as watermarked if $z_{\text{score}} >$ threshold, and non-watermarked otherwise.

3 Related Work

Existing LLM watermarking methods [1] can be broadly classified into two categories: logits perturbation-based methods and token sampling-based methods, as systematically summarized in Table 1. The KGW framework (Kirchenbauer et al., 2023), a representative approach in this field, employs a Red-Green list vocabulary partitioning

[1]Our work builds upon the MarkLLM toolkit (Pan et al., 2024), an open-source framework for constructing extensible watermarking architectures.

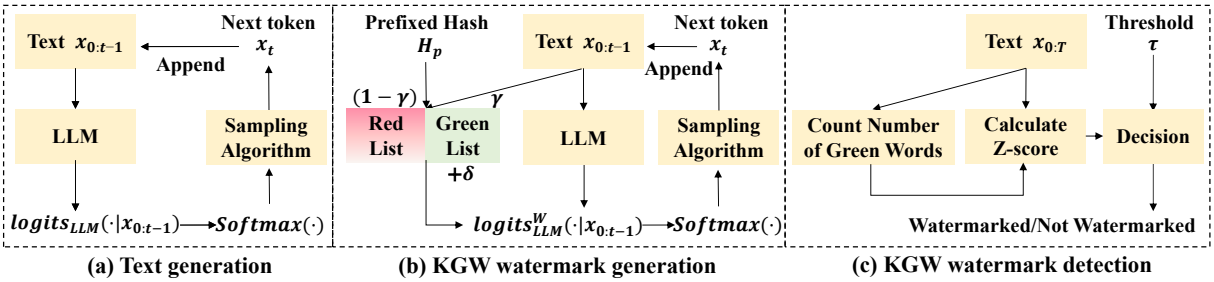


Figure 2: The overview of KGW watermark for LLM.

mechanism, significantly altering token sampling probabilities. However, this aggressive perturbation often results in substantial semantic fidelity degradation. Moreover, its dynamic partitioning mechanism, which relies on preceding token states, makes any alteration of upstream text can disrupt the token distribution balance, thereby reducing detection robustness.

Logits Perturbation-Based Methods. To address detectability challenges in low-entropy domains like code generation, methods such as SWEET (Lee et al., 2024) and EWD (Lu et al., 2024) introduce adaptive watermark embedding strategies that condition on local entropy estimates. However, their reliance on task-specific entropy thresholds limits generalizability across diverse NLP tasks. The SIR model (Liu et al., 2023) aligns watermark logits with semantic embeddings generated by external language models to preserve semantic fidelity, yet it fails to maintain robustness against syntactic transformations. DIP (Wu et al., 2023) adopts context-aware logit reweighting to maintain output distribution consistency, but its perturbation mechanism lacks the structural resilience necessary to withstand sophisticated adversarial manipulations.

Token Sampling-Based Method. Early sampling-based approaches like Christ (Christ et al., 2024) leverage pseudo-random binary selection sequences to embed watermarks, but their strict binary constraints and lack of semantic awareness limit practical applicability in real-world generation tasks. The EXP framework (Aaronson and Kirchner., 2022) improves usability by introducing exponential score boosting to bias token selection, though this aggressive sampling mechanism often leads to unnatural text outputs and significant quality degradation. These limitations highlight the need for a more adaptive watermarking framework that can maintain robust detection performance while preserving output quality across task contexts.

4 Proposed Method

4.1 Overview

Typically consisting of two stages—watermark injection and detection—the general framework of watermarking methods operates as follows: during each generation step, a watermark key is derived to partition vocabulary into green/red lists, after which a bias is applied to token probabilities to promote sampling from the green list. Following this framework, we propose (1) **low-entropy POS-guided token partitioning** (Sec. 4.2), which splits vocabulary into red-green-blue lists to optimize the distribution of POS; (2) **z-score-driven dynamic bias** (Sec. 4.3), which adaptively adjusts bias based on z-score to improve semantic fidelity.

Our framework (as systematically illustrated in Fig. 3) first employs the entropy statistics to obtain sort of POS for splitting the Low-entropy POS-guided token partitioning. Then we calculate dynamic bias according to z-score. Finally, we introduce bias into the green tokens and obtain the watermarked text.

4.2 Low-Entropy POS-Guided Token Partitioning

Step 1: Entropy statistics based on part of speech. Linguistic analysis indicates that part-of-speech (POS) categories exhibit distinct entropy characteristics in natural language generation. As supported by sparse watermarking research (Hoang et al., 2024), nouns and adjectives consistently demonstrate significantly lower entropy values compared to verbs and adverbs, as empirically validated in Fig. 3(a) step 1.

Step 2: Token partitioning based on entropy. Leveraging this observation, we introduce a low-entropy POS-guided token partitioning mechanism: low-entropy POS tokens (nouns, adjectives) are as-

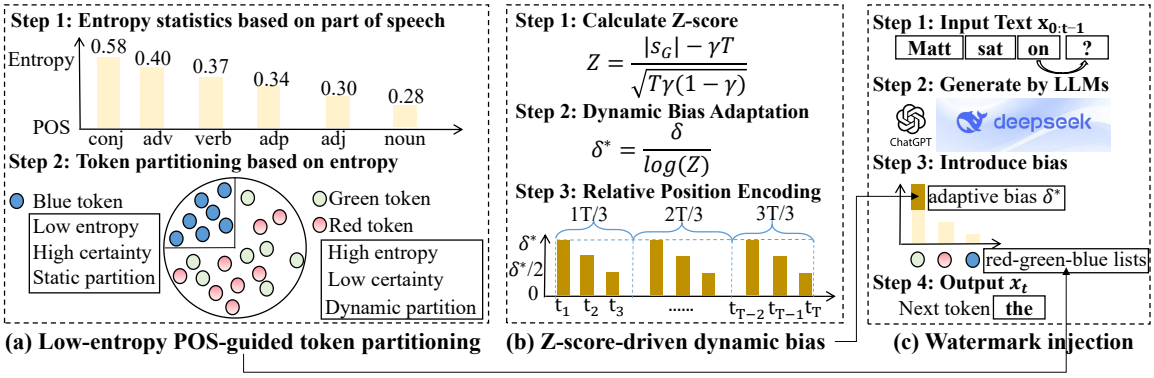


Figure 3: Overall architecture of the our watermark framework.

signed to a static *blue list* that remains unperturbed during watermarking, while higher-entropy POS tags (verbs, adverbs, etc.) form the dynamic *red-green list* subject to probabilistic bias adjustments (Fig. 3(a) step 2). This partitioning mechanism explicitly separates tokens into three mutually exclusive sets using a hybrid hashing mechanism, formalized as:

$$V = V_{\text{Blue}} \cup V_{\text{Red}} \cup V_{\text{Green}}, V_i \cap V_j = \emptyset \, (i \neq j) \quad (5)$$

In this context, V_{Blue} comprises static low-entropy tokens identified through global Part-of-Speech (POS) analysis. In order to facilitate watermark detection, the proportion of tokens in the green list is fixed at half of the total vocabulary. Meanwhile, the tokens in $V_{\text{Red/Green}}$ are subject to dynamic classification relying on contextual hashing.

4.3 Z-Score-Driven Dynamic Bias

To address the excessive distributional distortion caused by KGW's global fixed bias, our framework introduces a z-score-driven dynamic bias mechanism that dynamically adjusts perturbation intensity based on cumulative watermark signal strength, formalized through the following stages:

Step 1: Calculate Z-score. The z-score quantifies the watermark signal strength in the current text, thus we calculate it to determine whether to adjust the bias intensity. We can obtain the z-score of previous tokens according to eq. 6 (Fig. 3(b) step 1):

$$\text{Z-score} = \frac{|s|_G - \gamma T}{\sqrt{T\gamma(1-\gamma)}} \quad (6)$$

where γ denotes the green list proportion in the vocabulary and T is the text length (number of tokens). The number of green tokens denoted as $|s|_G$.

Step 2: Dynamic bias adaptation. During the initial generation phase, green tokens receive a fixed bias δ to establish a detectable watermark signal:

$$\text{logits}_t = \text{logits}(x_t|x_{0:t-1}) + \delta \cdot \mathbb{I}(x_t \in V_{\text{Green}}) \quad (7)$$

where $\mathbb{I}(\cdot)$ is the indicator function ensuring bias application only to green list tokens (Fig. **??** middle). Once the cumulative z-score z—measuring the deviation of green token frequency from expected values—exceeds a threshold τ, the bias intensity decays logarithmically to mitigate distributional distortion:

$$\text{logits}_t^{\text{W}} = \text{logits}(x_t|x_{0:t-1}) + \frac{\delta}{\log(z)} \cdot \mathbb{I}(x_t \in V_{\text{Green}}) \quad (8)$$

This adaptive decay balances watermark detectability and semantic fidelity by reducing perturbation as the watermark signal becomes sufficiently strong (Fig. 3(b) step 2).

Step 3: Relative Position Encoding. Unlike global fixed bias of KGW (Fig. 4(a)), to counter potential bias sparsity exploitation by attackers, we integrate relative position encoding (RPE) (Fig. 3(b) step 3), dividing the generation process into non-overlapping intervals and applying dynamic bias uniformly across all intervals. This RPE mechanism prevents localized perturbation accumulation, making targeted watermark removal significantly more challenging. This mechanism of applying biases evenly across the entire scope prevents bias concentration in local areas, which could otherwise render the watermark vulnerable to targeted attacks (Pang et al., 2024; Liu et al., 2024b).

4.4 Workflow of Watermark Injection and Detection

For watermark injection, LLMs generate logits by using input text $x_{0:t-1}$ as shown in Fig. 3(c) step 1

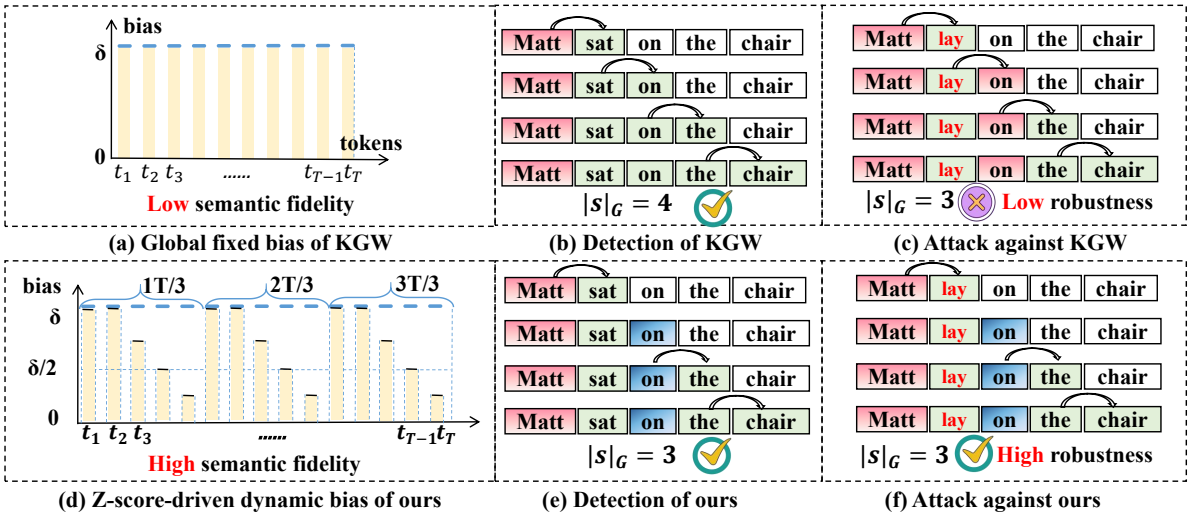


Figure 4: Qualitative comparisons of robustness against typical attacks between our watermark and KGW.

and 2. We utilize low-entropy POS-guided token partitioning mechanism to splits vocabulary into Low-entropy POS-guided token partitioning and utilize z-score-drive dynamic bias mechanism to obtain adaptive bias δ^*. Then, we introduce adaptive bias δ^* into green tokens. Finally, we get next token x_t.

In watermark detection, given a text sequence, we first derive a watermark key for each token from its contextual information and partition the vocabulary into green/red lists consistent with the watermark injection procedure to determine token membership in the green list. We then count the number of green tokens (denoted as $|s|_G$) and compute the z-score statistic for watermark presence verification (eq. 6). A higher z-score indicates greater confidence in watermark presence. Detailed detection derivation and procedure are provided in Appendix A.3 and Algorithm 2, respectively.

5 Theoretical Foundation

5.1 Theoretical Proof of Detectability

We introduce the concept of *Sharpness Entropy* to characterize the distributional perturbation induced by watermarking, serving as a generalization of KGW's "Spike Entropy" (Kirchenbauer et al., 2023) with enhanced sensitivity to token distributional changes.

Definition 1 (Sharpness Entropy)**.** *The Sharpness Entropy of a watermarked probability distribution $\hat{p}(x_t)$ with modulus θ is defined as:*

$$E_s(\hat{p}(x_t), \theta) = -\sum_{x_t \in V} (1 + \theta \hat{p}(x_t)) \log_2 \hat{p}(x_t)$$

where $\theta \geq 0$ is a hyperparameter controlling the sensitivity to high-certainty POS. This metric quantifies the trade-off between distributional sharpness and watermark signal strength.

Theorem 1 (Entropy Lower Bound)**.** *For a watermarked sequence of length T with average Sharpness Entropy, there exists a non-trivial lower bound E_s^* such that: $\frac{1}{T}\sum_{t=1}^{T} E_s(\hat{p}(x_t), \theta) \geq E_s^*$, where E_s^* is the infimum (greatest lower bound) determined by the low-entropy POS-guided token partitioning mechanism and z-score-driven dynamic bias mechanism.*

Theorem 2 (Green Token Expectation)**.** *The expected number of green list tokens $\mathbb{E}[|s|_G]$ in a watermarked sequence satisfies: $\mathbb{E}[|s|_G] \geq \gamma T E_s^*$ where $\gamma = |V_{Green}|/|V|$ is the green list proportion, and $|s|_G$ denotes the count of green tokens in sequence s. This lower bound ensures sufficient watermark signal accumulation for reliable detection.*

Theorem 3 (Green Token Variance)**.** *The variance of green token counts is upper-bounded by: $Var[|s|_G] \leq T\gamma(1-\gamma)$ This bound follows from the independence of token selections under the our watermark framework, enabling efficient hypothesis testing via z-score statistics.*

5.2 Theoretical Proof of Semantic Fidelity

We formalize semantic fidelity loss using the expected PPL score discrepancy between original and watermarked sequences: $Q = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} \text{PPL}(y_t, \hat{y}_t)\right]$ where y_t and $\hat{y}_t$ denote the original and watermarked tokens at position t. The optimization problem balances

quality loss Q against a minimum detection accuracy constraint using Lagrange multipliers: $\min_{\delta,\alpha} Q \quad \text{s.t.} \quad \mathcal{A}(\rho) \geq \mathcal{A}_0$ The optimal bias δ^* decreases with the square root of sequence length and inversely with Sharpness Entropy, ensuring minimal distributional distortion while maintaining detectability.

5.3 Theoretical Proof of Robustness

Static Blue List Invariance. By selecting low-entropy POS tags (e.g., nouns, adjectives) to form the static blue list B:$B = \arg\min_{c \in \text{POS}} E_{\text{POS}}(c)$ where $E_{\text{POS}}(c)$ denotes the entropy of POS category c, the retention probability of blue list tokens under deletion/replacement attacks is significantly enhanced. For an attacked sequence $\tilde{y}_{0:T}$, the conditional retention probability of a blue token at position t is:$P(y_t \in B \mid \tilde{y}_{<t}) = \prod_{i=1}^{h}(1-\rho_i)$ where ρ_i is the attack success rate at position i. Due to the context-independent nature of B, this probability decays as $O(1/\sqrt{T})$, in contrast to dynamic list methods whose retention probability decays exponentially with context changes.

Adaptive Bias Perturbation Control. The KL-divergence between watermarked and original distributions is decomposed as: $D_{\text{KL}}(\hat{p}\|p) = \sum_{x_t \in G} \hat{p}(x_t) \log \frac{\hat{p}(x_t)}{p(x_t)} + \sum_{x_t \in R \cup B} \hat{p}(x_t) \log \frac{\hat{p}(x_t)}{p(x_t)}$

Under the Z-Score-Driven Dynamic Bias mechanism, when the cumulative z-score exceeds the detection threshold ($z_{t-1} > \tau$), the bias decays as $\delta/\log(z)$, leading to a second-order approximation: $D_{\text{KL}}(\hat{p}\|p) \leq \frac{\delta^2}{2} \sum_{x_t \in G} \left.\frac{\partial^2 p(x_t)}{\partial \delta^2}\right|_{\delta=0} + o(\delta^2)$

This inequality shows that SAB limits distributional perturbation to a quadratic term in δ, ensuring that semantic fidelity is preserved while maintaining non-trivial detection signals against adversarial attacks.

Examples. Under word substitution attacks (e.g., replacing "sat" with "lay" in Fig. 4(b)(c)), the replacement of preceding words can induce misclassification of subsequent tokens in the red-green list framework. For instance, "on" might shift from the green list to the red list due to altered contextual hashing H_p, thereby degrading watermark detection robustness. Unlike KGW's context-dependent partitioning, our static blue list assignment—using a global hash function H_f—ensures consistent token categorization regardless of preceding token modifications. For example (Fig. 4(e)(f)), substituting "sat" with "lay", our framework maintains "on" as a blue token, significantly enhancing watermark robustness against substitution attacks.

6 Experimental Setting

We conduct a comprehensive evaluation of **6 watermarking methods** across **4 NLP tasks**, **4 datasets**, and **5 LLMs** under **8 types of adversarial attacks**. Experiments are performed on two NVIDIA A800 (80GB) GPUs, using datasets with over 500 samples each, taking up a total of around 100 GPU hours. This section details task configurations, evaluation metrics, and parameter settings.

6.1 Tasks and Datasets

Following previous works (Kirchenbauer et al., 2023; Wu et al., 2023; Aaronson and Kirchner., 2022; Liu et al., 2023), the evaluation encompasses four representative tasks, spanning diverse NLP scenarios:

Text Generation: Leveraging the C4 corpus (Raffel et al., 2020) with OPT-1.3B (Zhang et al., 2022) model for general text generation tasks.

News Generation: Utilizing CNN/Daily-Mail dataset (Hermann et al., 2015) with Llama-3-8B-Instruct model (Grattafiori et al., 2024) for news generation tasks.

Story Extension: Employing the ROCStories dataset (Mostafazadeh et al., 2016) with Qwen2.5 (Qwen et al., 2025) to extend narrative sequences.

Long-Form QA: Using the ELI5 dataset (Fan et al., 2019) with DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025a) for detailed question answering.

6.2 Evaluation Metrics

Following previous works (Kirchenbauer et al., 2023; Wu et al., 2023; Aaronson and Kirchner., 2022; Liu et al., 2023), quantitative metrics are defined to assess performance across dimensions:

Semantic Fidelity: Perplexity (PPL) and Relative Change in Perplexity (RC-PPL). Log Diversity (LD) and Relative Change in Log Diversity (RC-LD), formulation as following:

$$\text{RC-X} = \frac{\text{X}^w - \text{X}^{nw}}{\text{X}^{nw}} \times 100\%$$

where X is PPL or LD.

Detectability: Best F1-score and Accuracy (ACC) for binary watermark presence classification.

Robustness: F1-score against the following attacks: word deletion (Del), word substitution

(Sub), keyboard-based typo attack (Typo), expansion (Exp) attack (e.g., expanding "don't" to "do not"), position-based swap attack (Swap), and all-lowercase conversion (LC) attacks (Liu et al., 2024b). For robustness evaluation, we compute the average F1-score (AVG) across these eight attack scenarios.

Higher values for LD, RC-LD, P, R, F1, and ACC indicate superior performance, while lower RC-PPL signifies better semantic fidelity preservation.

6.3 Baselines

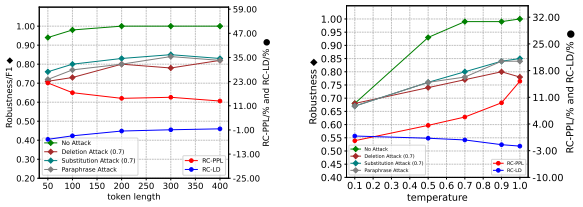
Our baselines consist of the following watermark methods: KGW (Kirchenbauer et al., 2023), SWEET(Lee et al., 2024), EWD(Lu et al., 2024), DiPMark(Wu et al., 2023), SIR.

6.4 Parameter Configuration

Key hyperparameters are optimized via grid search (Fig. 5 and App. C.2), including:

Blue List POS Selection: static to low-entropy parts of speech (nouns and adjectives), comprising 36% of tokens in the model of LLAMA(Touvron et al., 2023), ensuring grammatical integrity.

Temperature Robustness : Temperature of model in main experiments is 1.0, and additional results in Table 8 and Fig. 5b cover lengths from 0.1 to 1.0.



(a) Influence of token length. (b) Influence of temperature.

Figure 5: Influence of hyper-parameters of our watermark (Soft). The y-axis lines with ● and ♦ are right and left, respectively.

Length Scalability: Token lengths in main experiments are 500, and additional results in Table 7 and Fig. 5a cover lengths from 50 to 400.

7 Experimental Results

7.1 Detectability Performance

Our watermark achieves perfect detectability (F1=1.00, ACC=1.00) across all tested models (Table 2), matching the KGW method's detection logic.

Table 2: The semantic fidelity and detectability of different methods.

Tasks/ Datasets/ Models	Methods	Semantic Fidelity Perplexity		Semantic Fidelity Diversity		Detectability Performance	
		PPL↓	RC-PPL↓	LD↑	RC-LD↑	F1↑	ACC↑
Stroy Extension/ ROCStories/ Qwen2.5	KGW	14.72	50.79%	7.59	-0.61%	**1.00**	**1.00**
	SWEET	13.34	36.67%	7.47	-2.28%	**1.00**	**1.00**
	EWD	13.80	41.39%	7.47	-2.19%	**1.00**	**1.00**
	DiPMark	12.06	23.57%	7.71	0.89%	**1.00**	**1.00**
	EXP	27.98	186.69%	8.93	16.82%	**1.00**	**1.00**
	SIR	12.87	31.86%	7.12	-6.75%	**1.00**	**1.00**
	Ours	**11.62**	**19.06%**	**8.95**	**17.15%**	**1.00**	**1.00**
Text Generation/ C4/ OPT	KGW	13.45	28.83%	7.44	-2.46%	**1.00**	**1.00**
	SWEET	12.70	21.63%	7.48	-1.99%	**1.00**	**1.00**
	EWD	13.51	29.39%	7.58	-0.72%	**1.00**	**1.00**
	DiPMark	12.00	14.97%	7.84	2.81%	**1.00**	**1.00**
	EXP	23.62	126.28%	8.44	10.62%	**1.00**	**1.00**
	SIR	13.60	30.30%	7.55	-1.08%	0.99	0.99
	Ours	**11.60**	**11.11%**	**8.46**	**10.88%**	**1.00**	**1.00**
News Generation/ CNN-Daily-Mail/ LLAMA3	KGW	6.75	26.56%	7.26	0.31%	0.99	0.99
	SWEET	6.46	21.24%	7.15	-1.21%	0.99	0.99
	EWD	6.81	27.81%	7.38	1.94%	**1.00**	**1.00**
	DiPMark	5.61	5.22%	7.35	1.51%	0.96	0.97
	EXP	9.46	77.51%	7.19	-0.68%	0.99	0.99
	SIR	6.64	24.61%	7.12	-1.66%	0.94	0.94
	Ours	**5.59**	**4.88%**	**7.73**	**6.77%**	**1.00**	**1.00**
Long-form Question Aswer/ ELI5/ DeepSeek-R1	KGW	5.92	28.50%	7.40	0.77%	**1.00**	**1.00**
	SWEET	5.34	15.88%	7.28	-0.82%	**1.00**	**1.00**
	EWD	5.79	25.52%	7.27	-0.96%	0.99	**1.00**
	DiPMark	4.85	5.14%	7.36	0.23%	0.97	0.97
	EXP	6.52	41.42%	7.45	1.51%	**1.00**	**1.00**
	SIR	5.59	21.36%	7.15	-2.66%	0.94	0.94
	Ours	**4.68**	**1.52%**	**7.59**	**3.41%**	**1.00**	**1.00**

note: 1. ↑ denotes that the larger the value and ↓ means the opposite. 2. Temperature is 1.0 and token length is 300. 3. Bold indicates first place ranking.

This consistency is enabled by maintaining the standard z-score calculation framework, ensuring compatibility with existing detection pipelines.

7.2 Semantic Fidelity Analysis

As shown in Table 2, our watermark demonstrates superior semantic fidelity preservation. On the DeepSeek-R1 model, it achieves an RC-PPL of +1.52%, significantly lower than KGW's +28.50% and SIR's +21.36%. The RC-LD of +3.41% indicates slight improvement to token distribution diversity, attributed to the z-score-driven dynamic bias mechanism that dynamically adjusts logit perturbations to avoid over-distortion.

7.3 Robustness Evaluation

Our watermark retains an F1-score of AVG ≥ 0.89 on different tasks (Table 3), outperforming KGW ($\geq$0.84) and SIR ($\geq$0.77). The static blue list design contributes to 36% higher information retention under word replacement attacks, as static low-entropy POS tokens are less susceptible to semantic-preserving modifications.

7.4 Ablation Experiments

Core component validation (Table 4) demonstrates the critical role of individual mechanisms in our framework:

Table 3: The robustness of different methods.

Tasks/ Datasets/ Models	Methods	Robustness against Attacks								
		Del	Sub	CS	DP	Typo	AE	Swap	LC	AVG
		F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑
Stroy Extension/ ROCStories/ Qwen-2.5	KGW	0.86	0.87	**1.00**	0.90	0.69	**1.00**	**0.98**	**1.00**	0.91
	SWEET	0.76	0.89	0.98	0.87	0.67	**1.00**	0.96	**1.00**	0.89
	EWD	0.81	0.91	0.99	0.89	0.67	**1.00**	0.97	**1.00**	0.91
	DiPMark	0.67	0.68	0.79	0.68	0.67	**1.00**	0.69	0.99	0.77
	EXP	0.66	0.81	0.94	0.78	0.66	**1.00**	0.92	**1.00**	0.85
	SIR	0.66	0.79	0.97	0.87	0.68	**1.00**	0.66	0.98	0.83
	Ours	**0.92**	**0.95**	**1.00**	**0.92**	**0.72**	0.99	**0.98**	**1.00**	**0.94**
Text Generation/ C4/ OPT	KGW	0.86	0.94	**1.00**	0.90	0.69	**1.00**	0.98	**1.00**	0.92
	SWEET	0.81	0.97	0.99	0.93	0.67	**1.00**	0.97	**1.00**	0.92
	EWD	0.85	0.96	**1.00**	0.94	0.67	**1.00**	0.98	**1.00**	0.92
	DiPMark	0.68	0.70	0.81	0.68	0.67	**1.00**	0.71	0.97	0.78
	EXP	0.67	0.86	0.96	0.75	0.67	**1.00**	0.88	0.99	0.85
	SIR	0.69	0.74	0.96	0.84	**0.79**	0.99	0.87	0.97	0.85
	Ours	**0.90**	**0.98**	**1.00**	**0.95**	**0.79**	0.99	**0.99**	**1.00**	**0.95**
News Generation/ CNN-Daily-Mail/ Llama-3	KGW	0.74	0.85	0.92	0.78	0.67	0.99	0.84	0.93	0.84
	SWEET	0.71	0.82	0.91	0.74	0.67	0.99	0.80	0.94	0.82
	EWD	0.76	0.87	0.94	0.76	0.67	**1.00**	0.88	0.95	0.85
	DiPMark	0.67	0.68	0.70	0.68	0.67	0.96	0.67	0.81	0.73
	EXP	0.67	0.79	0.89	0.75	0.67	0.99	0.76	0.94	0.81
	SIR	0.66	0.75	0.88	0.80	0.73	0.94	0.69	0.87	0.79
	Ours	**0.80**	**0.91**	**0.96**	**0.83**	**0.75**	0.99	**0.90**	**0.96**	**0.89**
Long-form Question Aswer/ ELI5/ DeepSeek-R1	KGW	0.78	0.90	0.93	0.79	0.73	**1.00**	0.88	**0.99**	0.87
	SWEET	0.73	0.85	0.93	0.80	0.70	0.99	0.86	0.98	0.86
	EWD	0.80	0.87	0.95	0.84	0.68	0.99	0.90	0.98	0.88
	DiPMark	0.66	0.67	0.72	0.66	0.66	0.96	0.67	0.86	0.73
	EXP	0.68	0.73	0.89	0.70	0.67	**1.00**	0.73	0.98	0.80
	SIR	0.67	0.68	0.86	0.76	0.69	0.93	0.67	0.88	0.77
	Ours	**0.85**	**0.93**	**0.97**	**0.87**	**0.75**	0.96	**0.92**	**0.99**	**0.91**

note: 1. ↑ denotes that the larger the value and ↓ means the opposite. 2. Temperature is 1.0 and token length is 300. 3. Bold indicates first place ranking.

Removal of Low-Entropy POS-Guided Token Partitioning Omitting the low-entropy POS-guided token partitioning mechanism leads to a significant degradation in semantic fidelity, manifested as a 9.7% increase in RC-PPL. This outcome underscores the necessity of stabilizing low-entropy parts of speech (POS), as their perturbation directly disrupts core sentence semantics. Notably, log diversity (LD) exhibits a marginal increase of 2.31%, likely due to the absence of POS constraints allowing broader token sampling, though this comes at the cost of semantic coherence.

Removal of Z-Score-Driven Dynamic Bias Excluding the z-score-driven dynamic bias mechanism results in a 2.89% increase in RC-PPL and a 12.7% reduction in attack F1-score, highlighting the indispensable role of adaptive bias in balancing semantic fidelity and robustness. The fixed bias scheme (replaced by static perturbation) overdistorts token distributions, leading to unnatural text generation and reduced resilience against adversarial attacks such as word substitution and deletion.

Combined Removal of Both Mechanisms The simultaneous removal of both mechanisms causes a synergistic decline in performance: RC-PPL increases by 12.39%, and attack F1-score drops by 3%. This drastic degradation confirms the complementary nature of the low-entropy POS partitioning (preserving semantic integrity) and dynamic bias (maintaining distributional consistency), emphasizing their indispensable roles in achieving robust and semantically faithful watermarking.

These findings validate that the proposed mechanisms are not only individually critical but also exhibit strong synergistic effects, collectively optimizing the trade-off between watermark detectability, semantic fidelity, and robustness against adversarial manipulations.

Table 4: Comparison with ablation experiment.

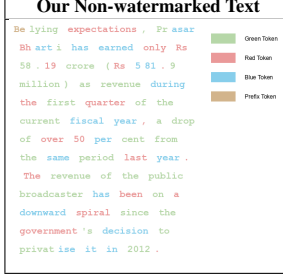
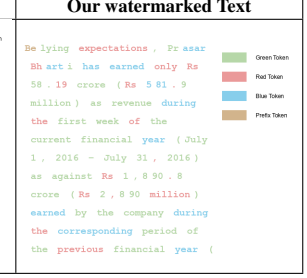
Method	Semantic Fidelity				Robustness			Detectability	
	Perplexity		Diversity		Del	Sub	DP	Performance	
	PPL↓	RC-PPL↓	LD↑	RC-LD↑	F1↑	F1↑	F1↑	F1↑	ACC↑
w/o L	12.17	+21.26%	7.50	-1.69%	0.81	0.88	0.85	0.99	0.99
w/o Z	11.49	+14.45%	7.59	-0.46%	0.73	0.76	0.76	0.99	0.99
w/o Both	12.44	+23.95%	7.49	-1.77%	0.81	0.88	0.85	0.98	0.98
Ours	**11.20**	**+11.56%**	**7.67**	**+0.54%**	**0.84**	**0.92**	**0.87**	**1.00**	**1.00**

Note: 1.↑ denotes that the larger the value and ↓ means the opposite. 2."w/o L" is without "Low-entropy POS-guided token partitioning"; "w/o Z" is without "Z-score-driven dynamic bias". 3.Temperature is 1.0 and task is text generation.

7.5 Qualitative Examples

Table 5 illustrates watermarked text outputs compared to KGW. our watermark generates grammatically correct sentences with lower RC-PPL (22.97% vs. KGW's 50.79%). Visualized token distributions show tighter alignment with non-watermarked text, demonstrating semantic fidelity.

Table 5: Watermarked Examples

KGW Non-watermarked Text	KGW Watermarked Text
	
Our Non-watermarked Text	**Our watermarked Text**
	

8 Conclusion

This paper introduces a novel watermark framework, which contains two innovations: (1) a low-entropy POS-guided token partitioning mechanism using low-entropy POS tags to fix critical tokens, enhancing semantic fidelity; (2) a z-score-driven dynamic bias mechanism with relative position encoding for adaptive logit adjustments, maintaining distributional consistency while enabling efficient detection. The framework establishes a new benchmark for trustworthy AI content by enabling provenance tracing in different scenarios. Its generalizable design supports seamless integration into heterogeneous LLM ecosystems, offering solutions for copyright protection and accountability in generative AI.

9 Limitation and Future Work

Shorter Token Lengths. While the framework demonstrates strong performance across diverse settings, it exhibits a minor decline in detection accuracy for token lengths shorter than 100, primarily due to insufficient watermark signal density. This limitation motivates future research into context-aware watermark embedding for short-form texts, such as incorporating syntactic dependency modeling or adaptive thresholding based on sequence length.

Multi-Modal Input. Additionally, exploring the integration of multi-modal input encoding—beyond text-only scenarios—to enhance watermark resilience in image-to-text and audio-to-text generation tasks represents an exciting avenue for expansion. These efforts will further solidify the framework's utility in real-world applications where content brevity and modality diversity are common challenges.

Acknowledgments

This work was supported in part by the Beijing Municipal Science Technology Commission New generation of information and communication technology innovation Research and demonstration application of key technologies for privacy protection of massive data for large model training and application (Z231100005923047).

Ethical considerations

This work focuses on watermarking techniques for large language models (LLMs) with the goal of enabling reliable attribution of AI-generated content, thereby supporting copyright protection and mitigating misuse. Our methodology does not involve human subjects, and therefore did not require IRB review. All experiments were conducted using publicly available datasets (e.g., C4, CNN/DailyMail, ROCStories, ELI5) under their intended research use licenses.

The watermarking framework is designed to be minimally invasive, preserving semantic fidelity and grammatical integrity while embedding detectable signals. The risks associated with our method are no greater than those inherent in the underlying LLMs themselves, as the watermark does not introduce new generative capabilities or alter model behavior beyond controlled logit adjustments.

We note that watermarking technologies can be dual-use: while they enable positive applications such as content provenance and accountability, they could also be misused in contexts of censorship or false attribution. To mitigate such risks, we advocate for transparent deployment and clear communication regarding the presence and purpose of watermarks.

All code and models used in this study are intended for research purposes only, and we encourage responsible use in alignment with ethical AI guidelines.

References

S. Aaronson and H. Kirchner. 2022. https://www.scottaaronson.com/talks/watermark.ppt.

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. 2021. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610–623, New York, NY, USA. Association for Computing Machinery.

Yapei Chang, Kalpesh Krishna, Amir Houmansadr, John Frederick Wieting, and Mohit Iyyer. 2024. PostMark: A robust blackbox watermark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8969–8987, Miami, Florida, USA. Association for Computational Linguistics.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1125–1139. PMLR.

Evan Crothers, Nathalie Japkowicz, and Herna Viktor. 2023. Machine generated text: A comprehensive survey of threat models and detection methods. *Preprint*, arXiv:2210.07321.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, taylan. cemgil, Zahra Ahmed, Kitty Stacpoole, and 5 others. 2024. Scalable watermarking for identifying large language model outputs. *Nature*, 634:818 – 823.

DeepSeek-AI. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. ELI5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567, Florence, Italy. Association for Computational Linguistics.

Jessica Gillotte. 2020. Copyright infringement in ai-generated artworks. *Intellectual Property: Other eJournal*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yuxuan Guo, Zhiliang Tian, Yiping Song, Tianlun Liu, Liang Ding, and Dongsheng Li. 2024. Context-aware watermark with semantic balanced green-red lists for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22633–22646, Miami, Florida, USA. Association for Computational Linguistics.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Preprint*, arXiv:1506.03340.

Duy C. Hoang, Hung T. Q. Le, Rui Chu, Ping Li, Weijie Zhao, Yingjie Lao, and Khoa D. Doan. 2024. Less is more: Sparse watermarking in llms with enhanced text quality. *Preprint*, arXiv:2407.13803.

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

Vojtěch Hudeček and Ondrej Dusek. 2023. Are large language models all you need for task-oriented dialogue? In *Proceedings of the 24th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 216–228, Prague, Czechia. Association for Computational Linguistics.

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

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. 2024. Waterfall: Scalable framework for robust text watermarking and provenance for LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 20432–20466, Miami, Florida, USA. Association for Computational Linguistics.

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

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.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong,

and Philip Yu. 2024a. A survey of text watermarking in the era of large language models. *ACM Comput. Surv.*, 57(2).

Zesen Liu, Tianshuo Cong, Xinlei He, and Qi Li. 2024b. On evaluating the performance of watermarked machine-generated texts under adversarial attacks. *Preprint*, arXiv:2407.04794.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. 2024. An entropy-based text watermarking detection method. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11724–11735, Bangkok, Thailand. Association for Computational Linguistics.

David Megías, Minoru Kuribayashi, Andrea Rosales, and Wojciech Mazurczyk. 2021. Dissimilar: Towards fake news detection using information hiding. In *Signal Processing and Machine Learning. In The 16th International Conference on Availability, Reliability and Security (Vienna, Austria)(ARES 2021). Association for Computing Machinery, New York, NY, USA, Article*, volume 66.

Yisroel Mirsky, Ambra Demontis, Jaidip Kotak, Ram Shankar, Deng Gelei, Liu Yang, Xiangyu Zhang, Maura Pintor, Wenke Lee, Yuval Elovici, and Battista Biggio. 2023. The threat of offensive ai to organizations. *Computers Security*, 124:103006.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

OpenAI. 2023. Gpt-4 technical report.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2024. MarkLLM: An open-source toolkit for LLM watermarking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71, Miami, Florida, USA. Association for Computational Linguistics.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. 2024. Attacking llm watermarks by exploiting their strengths. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Vaidehi Patil, Peter Hase, and Mohit Bansal. 2023. Can sensitive information be deleted from llms? objectives for defending against extraction attacks. *CoRR*, abs/2309.17410.

Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, Amelia Glaese, Nat McAleese, and Geoffrey Irving. 2022. Red teaming language models with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3419–3448, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

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(1).

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. *ArXiv*, abs/2302.13971.

Yihan Wu, Zhengmian Hu, Junfeng Guo, Hongyang Zhang, and Heng Huang. 2023. A resilient and accessible distribution-preserving watermark for large language models. In *International Conference on Machine Learning*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. 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. *CoRR*, abs/2205.01068.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable robust watermarking for ai-generated text. *arXiv preprint arXiv:2306.17439*.

A Variations of Our Method

The dynamic bias driven by z-score in our method can be modified according to actual needs. We will refer to the methods in the main text as "soft". Next, we will introduce two variants, "hard" and "wavy". The algorithm of watermark generation is shown in Algo. 1.

Algorithm 1 Our Watermark Generation

Require: Fixed hash H_f, bias δ, sharpness factor θ, thresholds τ_1, τ_2
Ensure: Watermarked text sequence $x_{0:T}$

Initialize: $x \leftarrow \emptyset$, $|s|_G \leftarrow 0$
Select low-entropy POS tags to form blue list B using H_f
Partition vocabulary into G, B, R using H_p and H_f
for $t = 0$ to $T-1$ **do**
  Calculate sharpness entropy $E_{\text{Sharpness}}(P_t, \theta)$
  if $E_{\text{Sharpness}} < \tau_1$ **then**
    $\delta^* \leftarrow 0$ {No bias for low sharpness}
  else
    Calculate z_{score} for previous tokens
    if $z_{\text{score}} < \tau_2$ **then**
      $\delta^* \leftarrow \delta$ {Apply fixed bias}
    else
      $\delta^* \leftarrow \delta / \log(z)$ {Adaptive bias reduction (soft)}
      $\delta^* \leftarrow 0$ {Adaptive bias reduction (hard)}
      $\delta^* \leftarrow \frac{1}{2} cos(\frac{2\pi\eta(t-1)}{T} + \frac{\pi}{2}) + \frac{\delta}{2}$ {Adaptive bias reduction (wavy)}
    end if
  end if
  Adjust logits: logits $\leftarrow$ logits$_{LLM} + \delta^* \cdot \mathbb{I}(x_t \in G)$
  Sample next token: $x_t \leftarrow$ Softmax(logits)
  Update: $x \leftarrow x \cup \{x_t\}$, $|s|_G \leftarrow |s|_G + \mathbb{I}(x_t \in G)$
end for

return $x_{0:T}$

A.1 Details of Low-Entropy POS-Guided Token Partitioning

How to calculate the entropy of Part of Speech? To guarantee the semantic fidelity, tokens with low entropy can not be added bias. Because next token with low entropy represents that it has the highest probability, the main idea of one sentence will be changed if we modify the logits. Therefore, we define a "Sharpness Entropy" to determine whether introduce bias:

$$E_s(P_t, \theta) = -\sum_{t=0}^{N} (1 + \theta P_t) * \log_2(P_t). \quad (9)$$

where logits will not be added bias if $E_s(P_t, \theta)$ are lower than a threshold τ_1.

How to partition "blue list"? Firstly, we calculate the entropy of the next token of every Part of Speech E_{N-PoS}, and choose Part of Speech with Low Entropy. Then, we use the fixed hash H_f to seed fixed random number generator. Finally, we use fixed random number generator to generate "blue list" B of size $\alpha|V|$:

$$\alpha|V| \Leftarrow min(E_{N-PoS}) \quad (10)$$

How to Partition "Red-Green List" We compute a prefixed hash H_p of previous token x_{t-1} and use it to seed prefixed random number generator. Then we use prefixed random number generator partition the vocabulary into a "green list" G of size $\gamma|V|$ and a "red list" R of size $(1-\gamma)|V|$:

$$(1-\alpha)|V| \Rightarrow \gamma|V| + (1-\alpha-\gamma)|V| \quad (11)$$

where α, γ and $(1-\alpha-\gamma)$ are the size of blue, green and red list, respectively.

A.2 Details of Z-Score-Driven Dynamic Bias

How to calculate the z-score? To relieve the decrease of text-quality, bias should be adaptively adjusted according to the z-score of previous token sequence. Because the larger the bias added, the poorer quality of the watermarked text, when the z-score of previous token sequence is enough to detect watermark. Therefore, we define a threshold of z-score to determine that the bias should big or small. The z-score is calculated as follows:

$$z_{t-1} = \frac{|s|_G - \gamma * (t-1)}{\sqrt{(t-1) * \gamma * (1-\gamma)}} \quad (12)$$

where n_g is the number of tokens in green list.

How to calculate the "Dynamic Bias"?

If the z-score of previous token sequence is not enough to detect watermark, the bias should be bigger than original δ. On the contrary, the bias should be smaller than original δ. In addition to the "soft bias" (Fig. 6(a)) in the main text, we also designed

"hard bias" (Fig. 6(b)) and "wavy bias" (Fig. 6(c)). We calculate the "wavy bias" as follows:

$$\delta^* = \begin{cases} \delta, & \text{if } z_{t-1} \leq z + \epsilon \\ \frac{1}{2}cos(\frac{2\pi\eta(t-1)}{T} + \frac{\pi}{2}) + \frac{1}{\delta}, & \text{otherwise} \end{cases} \tag{13}$$

We calculate the "hard bias" as follows:

$$\delta^* = \begin{cases} \delta, & \text{if } z_{t-1} \leq z + \epsilon \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

How to calculate watermarked logits? Dynamic bias are introduced into tokens when them in green list.

$$logits^w = \begin{cases} logits(\cdot|x_{0:t-1}) + \delta^*, & \text{if } x_t \in G \\ logits(\cdot|x_{0:t-1}), & \text{if } x_t \in R \cup B \end{cases} \tag{15}$$

How to sample token? Sample the next token x_t from the watermarked logits:

$$x_t = Samp(Softmax(logits^{scale}(\cdot|x_{0:t-1}))) \tag{16}$$

A.3 Watermark Detection

How to judge watermarked or not? The detection process mirrors the KGW method, leveraging statistical analysis of green list token frequency to determine watermark presence. The cumulative z-score (eq. 6) measures the deviation of green list token count from the expected value. A threshold τ classifies the text based on the z-score:

$$\text{result} = \begin{cases} \text{non-watermarked}, & z_{\text{score}} \leq \tau \\ \text{watermarked}, & z_{\text{score}} > \tau \end{cases} \tag{17}$$

This binary decision efficiently identifies watermarked text while maintaining low false positive rates. The algorithm of watermark detection is shown in Algo. 2.

B Theoretical Foundation

In cross-modal tasks such as image captioning, preserving low-entropy nouns and adjectives is critical for semantic integrity. Substituting "white dog" with "black cat" (noun substitution) disrupts core image semantics more severely than modifying "running quickly" to "walking slowly" (adverb substitution), justifying the exclusion of blue list tokens from logit perturbations to maintain content fidelity.

Algorithm 2 Our Watermark Detection

Input: Text sequence $x_{0:T}$, green ratio γ, blue ratio α

Parameters: Threshold τ_2

Output: Detection result (Watermarked / non-watermarked)

```
Initialize: |s|_G ← 0
for each token x_t in x_{0:T} do
    if x_t ∈ Green List then
        |s|_G ← |s|_G + 1
    end if
end for
Calculate z-score:
    z_score = (|s|_G − γT) / sqrt(Tγ(1−γ))
if z_score > τ_2 then
    return Watermarked
else
    return non-watermarked
end if
```

B.1 Theoretical Proof of Detectability

Definition 2. *We define the "Sharpness Entropy" of discrete probability vector P_t with modulus θ as:*

$$E_s(P_t, \theta) = -\sum_{t=0}^{|V|}(1 + \theta P_t)\log_2(P_t) \tag{18}$$

where $|V|$ is the size of vocabulary (or token) set.

Similar to the "Spike Entropy" of KGW (Kirchenbauer et al., 2023), "Sharpness Entropy" serves as an indicator for the degree of distribution dispersion. The "sharpness entropy" attains its minimum value of 0 when the entire mass of P_t is concentrated at a single point. It reaches its maximum value of $(|V| + \theta)\log_2(|V|)$ when the mass of P_t is uniformly distributed. For large θ:

$$(1 + \theta P_t)\log_2(P_t) \approx \begin{cases} \theta, & \text{if } P_t > \frac{1}{\theta} \\ \log_2(\theta), & \text{if } P_t < \frac{1}{\theta} \end{cases} \tag{19}$$

For this reason, the "Sharpness Entropy" can be interpreted as a measure of the number of entries in P_t that are greater than $\frac{1}{\theta}$.

The following theorem predicts the number of green and blue list tokens that will appear in a sequence with a watermark.

Theorem 4. *Consider watermarked text sequences of T tokens. Each sequence is produced by sequentially sampling a raw probability vector P_t from*

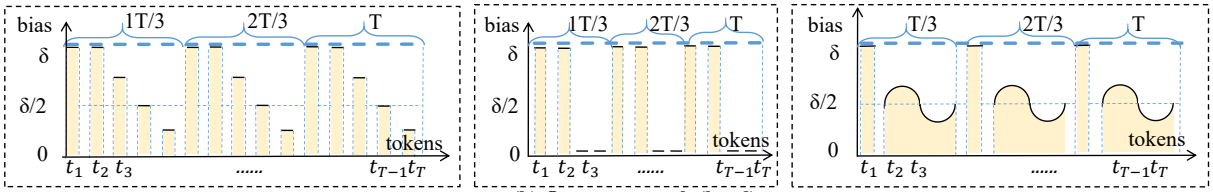


Figure 6: The comparisons of different bias.

the language model, sampling a random green list of size γN, and boosting the green list logits by δ^* before sampling each token. Let $|s|_G$ denote the number of green list tokens in sequence s.

If a randomly generated watermarked sequence has average spike entropy at least E_s^*, i.e.,

$$\frac{1}{T}\sum_{t=0}^{T} E_s\left(P_t, \frac{(\exp(\delta^*)-1)(1-\gamma)}{\log_2((1+(\exp(\delta^*)-1)\gamma)}\right) \geq E_s^*, \tag{20}$$

then the number of green list tokens in the sequence has expected value at least

$$\mathbb{E}|s|_G \geq \frac{\gamma\exp(\delta^*)T}{\log_2(1+(\exp(\delta^*)-1)\gamma)}E_s^*. \tag{21}$$

Furthermore, the number of green list tokens has variance at most

$$\begin{aligned} Var\ |s|_G \leq & T\frac{\gamma\exp(\delta^*)E_s^*}{\log_2(1+(\exp(\delta^*)-1)\gamma)} \\ & * \left(1-\frac{\gamma\exp(\delta^*)E_s^*}{\log_2(1+(\exp(\delta^*)-1)\gamma)}\right). \end{aligned} \tag{22}$$

If we have chosen $\gamma \geq .5$, then we can use the strictly looser but simpler bound

$$Var\ |s|_G \leq T\gamma(1-\gamma). \tag{23}$$

Remark 1. *It may seem like there are a lot of messy constants floating around in this bound. However, when we choose $\gamma = \frac{1}{2}$ and $\delta^* = \ln(2) \approx 0.7$, this bound simplifies to*

$$\mathbb{E}|s|_G \geq \frac{\ln 2}{\ln 1.5}TE_s^*, \tag{24}$$

$$Var|s|_G \leq \frac{\ln 2}{\ln 1.5}TE_s^*\left(1-\frac{\ln 2}{\ln 1.5}E_s^*\right) \tag{25}$$

where E_s^* is a bound on spike entropy with modulus $\frac{\ln 2}{\ln 2.25}$. If we study the "hard" red-green-blue list rules by choosing $\gamma = \frac{1}{2}$ and letting $\delta^* \to 0$, we have

$$\mathbb{E}|s|_G \geq TE_s^*, \quad Var|s|_G \leq TE_s^*(1-E_s^*) \tag{26}$$

where E_s^* is a bound on spike entropy with modulus 1.

B.2 Theoretical Proof of Semantic Fidelity

We formalize semantic fidelity loss using the expected PPL score discrepancy between original and watermarked sequences:$Q = \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\text{PPL}(y_t, \hat{y}_t)\right]$ where y_t and $\hat{y}_t$ denote the original and watermarked tokens at position t. The optimization problem balances quality loss Q against a minimum detection accuracy constraint using Lagrange multipliers: $\min_{\delta,\alpha} Q$ s.t. $\mathcal{A}(\rho) \geq \mathcal{A}_0$ The optimal bias δ^* decreases with the square root of sequence length and inversely with Sharpness Entropy, ensuring minimal distributional distortion while maintaining detectability.

B.3 Theoretical Proof of Robustness

Statistical Robustness against Attacks When the text is subjected to a deletion attack with a proportion of ρ, the effective detection length becomes $T' = T(1-\rho)$, and the detection statistic becomes:

$$z' = \frac{\hat{\gamma}T' - \gamma T}{\sqrt{\gamma(1-\gamma)T}} \tag{27}$$

By adjusting the proportion of the blue list α, it can be proved that when the attack intensity $\rho \leq 0.5$, the detection power is maintained:

$$\beta = \Phi\left(\frac{\Delta\gamma\sqrt{T}-\epsilon}{\sqrt{\gamma(1-\gamma)}}\right) \tag{28}$$

Information-theoretic Limit against Adversarial Attacks Under the character replacement attack, the surviving information of the watermark I satisfies:

$$I \geq \sum_{t=1}^{T}\left[\log_2\frac{\hat{p}(y_t)}{p(y_t)} - D_{\text{KL}}(\hat{p}||p)\right] \tag{29}$$

When the attack intensity is ρ, the lower bound of the surviving information is:

$$\begin{aligned} I \geq & T(1-\rho)(\gamma\log_2\frac{\gamma+\Delta\gamma}{\gamma} \\ & + (1-\gamma)\log_2\frac{1-\gamma-\Delta\gamma}{1-\gamma}) \end{aligned} \tag{30}$$

C Experiment Setting

C.1 Metrics

To systematically evaluate the impact of watermarking on semantic fidelity, we introduce three relative metrics: Relative Changes Perplexity (RC-PPL), Relative Changes BLEU (RC-BLEU), and Relative Changes Log Diversity (RC-LD). These metrics are defined as:

$$\text{RC-PPL} = \frac{\text{PPL}^{w} - \text{PPL}^{nw}}{\text{PPL}^{nw}} \times 100\% \quad (31)$$

$$\text{RC-LD} = \frac{\text{LD}^{w} - \text{LD}^{nw}}{\text{LD}^{nw}} \times 100\% \quad (32)$$

Here, PPL^{w} and LD^{w} denote the perplexity and log diversity of watermarked text, while PPL^{nw} and LD^{nw} represent the corresponding metrics for non-watermarked text. For BLEU scores, BLEU^{w} and BLEU^{nw} follow the same convention. Higher values of LD, RC-LD, Precision (P), Recall (R), F1-score (F1), and Accuracy (ACC) indicate better performance, whereas lower PPL and RC-PPL values reflect superior semantic fidelity.

C.2 Hyper-parameters Analysis

the Part-of-Speech (PoS) selection: (Fig. 7)

- Previous studies have shown that low-entropy tokens (e.g., numerals and nouns) are less sensitive to watermark perturbations while maintaining grammatical integrity (Hoang et al., 2024). Fig. 7a illustrates the entropy distribution across different PoS categories, confirming that nouns and numerals exhibit significantly lower entropy compared to verbs or punctuation. To construct the blue list, we calculated the document frequency of each PoS in the C4 dataset (Fig. 7b). Numerals and nouns, which account for 18.7% of all tokens, were selected as blue tokens to minimize semantic disruption. The percentage of occurrences for each POS tag in one passage is as shown in Fig. 7c.

Temperature Sensitivity: (Table 8)

- At $T = 0.7$, Ours(S) achieves RC-PPL=10.63% (LLaMA-7B), maintaining F1=0.96 under 70% deletion.
- Figure 5b reveals that increasing temperature improves semantic fidelity (PPL↓ 19.4%) but slightly reduces robustness (F1↓ 3.2%).

Token Length: (Table 7)

- For short sequences (100 tokens), Ours(H) achieves F1=0.92 under 70% deletion, outperforming SIR (F1=0.75).
- Figure 5a demonstrates that longer sequences (500 tokens) enhance robustness (F1↑ 8.7%) due to increased watermark signal density.

Attack Resilience Analysis: (Table 6)

- **Word Deletion**: Ours(H) maintains F1=0.995 at 50% deletion, outperforming Unigram (F1=0.980).
- **Word Substitution**: Ours(S) achieves F1=0.90 at 70% substitution, significantly better than KGW (F1=0.81).
- **Paraphrase**: Ours(H) retains F1=0.89 under strong paraphrasing (BART-based), while SIR fails (F1=0.70).

C.3 The Setting of Baselines

We test nine baselines: KGW(Kirchenbauer et al., 2023), SWEET(Lee et al., 2024), EWD(Lu et al., 2024), Unigram(Zhao et al., 2023), DiPMark(Wu et al., 2023), EXP(Aaronson and Kirchner., 2022), SynthID(Dathathri et al., 2024), SIR(Liu et al., 2023) and Unbiased watermark(Hu et al., 2023). The configuration of them is as following.

- "algorithm name": "KGW", "gamma": 0.5, "delta": 2.0, "hash_key": 15485863, "prefix_length": 1, "z_threshold": 4.0, "f_scheme": "time", "window_scheme": "left", "temperature_inner": 1.0
- "algorithm_name": "SWEET", "gamma": 0.5, "delta": 2.0, "hash_key": 15485863, "z_threshold": 4.0, "prefix_length": 1, "entropy_threshold": 0.9, "temperature_inner": 1.0
- "algorithm_name": "EWD", "gamma": 0.5, "delta": 2.0, "hash_key": 15485863, "prefix_length": 1, "z_threshold": 4.0, "temperature_inner": 1.0
- "algorithm_name": "Unigram", "gamma": 0.5, "delta": 2.0, "hash_key": 15485863, "z_threshold": 4.0, "temperature_inner": 1.0

- "algorithm_name": "DIP", "gamma": 0.5, "alpha": 0.45, "key": 42, "prefix_length": 5, "z_threshold": 1.513, "ignore_history_generation": 0, "ignore_history_detection": 0, "temperature_inner": 1.0
- "algorithm_name": "EXP", "prefix_length": 4, "hash_key": 15485863, "threshold": 0.0001, "sequence_length": 200, "top_k": 0, "temperature_inner": 1.0
- "algorithm_name": "SynthID", "ngram_len": 5, "keys": [654, 400, 836, 123, 340, 443, 597, 160, 57, 29, 590, 639, 13, 715, 468, 990, 966, 226, 324, 585, 118, 504, 421, 521, 129, 669, 732, 225, 90, 960], "sampling_table_size": 65536, "sampling_table_seed": 0, "watermark_mode": "non-distortionary", "num_leaves": 2, "context_history_size": 1024, "detector_type": "mean", "threshold": 0.52, "temperature_inner": 1.0
- "algorithm_name": "SIR", "delta": 1.0, "chunk_length": 10, "scale_dimension": 300, "z_threshold": 0.2, "transform_model_input_dim": 1024, "transform_model_name": "watermark/sir/model/transform_model_cbert.pth", "embedding_model_path": "./models/compositional-bert-large-uncased/", "mapping_name": "watermark/sir/mapping/300_mapping_152064.json", "temperature_inner": 1.0
- "algorithm_name": "Unbiased", "gamma": 0.5, "key": 42, "prefix_length": 5, "z_threshold": 1.513, "ignore_history": 1, "temperature_inner": 1.0

D Additional results

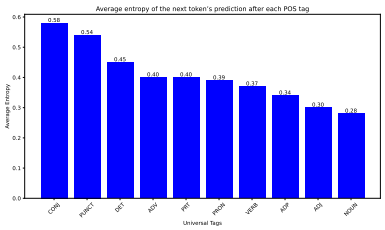
Table 8 demonstrates our framework's superiority across multiple dimensions:

- **Semantic Fidelity**: On the OPT-1.3B model, Ours(S) achieves RC-PPL=14.45% (compared to KGW's 23.95%) and RC-LD=-1.69% (vs. KGW's -9.44%).
- **Robustness**: Under 70% word deletion, Ours(H) maintains F1=0.97, outperforming KGW (F1=0.81) and SynthID (F1=0.67).
- **Detectability**: Both Ours(S) and Ours(H) achieve perfect detectability (ACC=1.00) due to preserved z-score calculations.

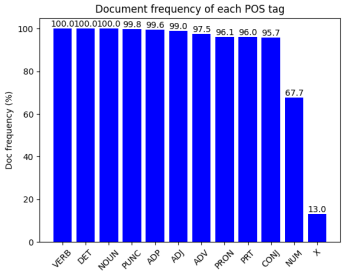
E Case Study: Watermarked Text Examples

Table 9 provides qualitative examples:

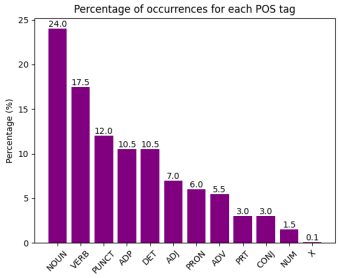
- **KGW**: "Bh art i has earned only Rs 58.19crore" (grammatical errors due to dynamic partitioning)
- **Ours**: "Bharati has earned only Rs 58.19 crore" (grammatically correct with static blue tokens)
- Detection statistics: Ours achieves z-score=4.1 (KGW=3.2) while maintaining lower PPL (10.1 vs. 12.1).



(a) Next-token prediction entropy distribution across POS categories.



(b) Document frequency of each POS tag.



(c) Percentage of occurrences for each POS tag.

Figure 7: Influence of POS.

Table 6: Robustness against different attacks with different ratio.

Attack	**Word Deletion**								**Word Substitution**								**Paraphrase**	
Ratio	**0.1**		**0.3**		**0.5**		**0.7**		**0.1**		**0.3**		**0.5**		**0.7**			
Metric	TPR	F1	TPR	F1	TPR	F1	TPR	F1	TPR	F1	TPR	F1	TPR	F1	TPR	F1	TPR	F1
KGW	1.000	1.000	0.995	0.995	0.970	0.951	0.965	0.806	1.000	1.000	0.990	0.995	0.965	0.941	0.905	0.889	0.890	0.858
Unigram	1.000	1.000	0.995	0.997	0.995	0.990	0.995	0.966	1.000	1.000	0.995	0.995	0.985	0.980	0.965	0.970	0.950	0.931
DiPMark	0.965	0.972	0.955	0.675	0.990	0.669	0.990	0.668	0.985	0.978	0.845	0.818	0.970	0.668	0.985	0.679	1.000	0.668
EXP	1.000	1.000	0.985	0.983	0.845	0.845	0.980	0.677	1.000	1.000	0.995	0.997	0.900	0.918	0.810	0.869	0.780	0.768
SynthID	0.990	0.992	0.865	0.874	0.810	0.675	0.995	0.664	0.990	0.990	0.885	0.876	0.920	0.733	0.940	0.686	0.860	0.703
SIR	0.930	0.951	0.805	0.854	0.660	0.767	0.995	0.664	0.935	0.957	0.850	0.881	0.750	0.792	0.875	0.734	0.775	0.836
Ours(S)	0.995	0.990	0.995	0.995	0.995	0.990	0.960	0.970	0.975	0.987	0.960	0.960	0.945	0.930	0.940	0.900	0.915	0.890
Ours(H)	0.995	0.990	0.995	0.990	0.975	0.980	0.970	0.965	0.970	0.970	0.970	0.940	0.830	0.850	0.875	0.805	0.910	0.900

Note: The length of token is 200. Ours(S) is our watermark with the soft way. Ours(H) is our watermark with the hard way.

Table 7: The results of different methods under different token length.

Token Length	Method	Semantic Fidelity				Robustness			Detectability	
		Perplexity		Diversity		Del(0.7)	Sub(0.7)	Para	Performers	
		PPL	RC-PPL	LD	RC-LD	F1	F1	F1	F1	ACC
100	KGW	9.79	+10.82%	10.66	+0.96%	0.77	0.85	0.77	0.94	0.94
	SWEET	9.26	+4.87%	9.94	-5.83%	0.71	0.75	0.76	0.95	0.95
	EWD	9.76	+10.53%	9.98	-5.48%	0.71	0.79	0.76	0.97	0.97
	Unigram	9.48	+7.38%	10.08	-4.51%	0.81	0.82	0.82	0.94	0.94
	DiPMark	9.43	+6.75%	10.61	+0.48%	0.66	0.66	0.66	0.83	0.82
	EXP	6.31	-28.58%	7.14	-32.43%	0.67	0.69	0.71	0.96	0.96
	SynthID	9.65	+9.33%	10.91	+3.34%	0.66	0.68	0.67	0.89	0.89
	SIR	9.79	+10.83%	9.36	-11.35%	0.67	0.67	0.67	0.82	0.82
	Unbiased	9.25	+4.80%	10.60	+0.35%	0.66	0.67	0.67	0.85	0.84
	Ours(S)	**6.57**	**+5.83%**	**6.88**	**-0.18%**	**0.72**	**0.80**	**0.78**	**0.99**	**0.99**
	Ours(H)	**6.60**	**+7.60%**	**6.88**	**-0.24%**	**0.77**	**0.80**	**0.81**	**0.97**	**0.97**
200	KGW	9.65	+33.86%	7.92	+2.24%	0.78	0.84	0.76	0.99	0.99
	SWEET	7.54	+4.53%	7.53	-2.90%	0.74	0.81	0.80	0.99	0.99
	EWD	7.97	+10.57%	7.41	-4.34%	0.77	0.82	0.84	1.00	1.00
	Unigram	7.72	+7.04%	7.62	-1.66%	0.83	0.84	0.83	0.97	0.97
	DiPMark	7.42	+2.92%	7.76	+0.08%	0.67	0.67	0.67	0.91	0.91
	EXP	6.31	-12.54%	7.14	-7.94%	0.67	0.68	0.68	0.96	0.96
	SynthID	7.47	+3.63%	7.79	+0.52%	0.66	0.67	0.67	0.96	0.96
	SIR	7.40	+2.63%	7.28	-6.10%	0.67	0.68	0.72	0.87	0.86
	Unbiased	7.51	+4.13%	7.71	-0.54%	0.67	0.67	0.67	0.92	0.92
	Ours(S)	**5.17**	**+10.78%**	**7.39**	**+0.25%**	**0.70**	**0.80**	**0.78**	**0.97**	**0.97**
	Ours(H)	**5.10**	**+9.13%**	**7.32**	**-0.65%**	**0.72**	**0.80**	**0.78**	**0.98**	**0.98**
300	KGW	8.60	+31.51%	7.49	+2.64%	0.82	0.92	0.79	1.00	1.00
	SWEET	6.85	+4.67%	7.08	-3.01%	0.76	0.82	0.78	0.99	0.99
	EWD	7.19	+9.91%	7.12	-2.45%	0.82	0.86	0.85	1.00	1.00
	Unigram	7.13	+8.95%	7.02	-3.83%	0.88	0.86	0.88	0.98	0.98
	DiPMark	6.71	+2.67%	7.33	+0.38%	0.66	0.68	0.67	0.96	0.96
	EXP	6.31	-3.58%	7.14	-2.26%	0.67	0.68	0.67	0.96	0.97
	SynthID	6.31	-3.58%	7.14	-2.26%	0.67	0.68	0.67	0.96	0.97
	SIR	6.78	+3.69%	7.05	-3.36%	0.67	0.67	0.75	0.91	0.91
	Unbiased	6.82	+4.36%	7.22	-1.04%	0.67	0.67	0.66	0.95	0.95
	Ours(S)	**8.82**	**+11.12%**	**8.87**	**-0.20%**	**0.70**	**0.75**	**0.73**	**0.94**	**0.94**
	Ours(H)	**8.73**	**+9.96%**	**8.94**	**+0.57%**	**0.70**	**0.75**	**0.73**	**0.95**	**0.95**
400	KGW	8.16	+30.69%	7.16	+1.96%	0.85	0.95	0.83	1.00	1.00
	SWEET	6.59	+5.56%	6.88	-1.97%	0.79	0.86	0.81	0.99	1.00
	EWD	6.72	+7.70%	6.81	-2.97%	0.81	0.90	0.86	1.00	1.00
	Unigram	6.73	+7.85%	6.86	-2.34%	0.89	0.88	0.89	0.99	0.99
	DiPMark	6.53	+4.61%	6.96	-0.81%	0.67	0.67	0.67	0.98	0.98
	EXP	6.31	+1.06%	7.14	+1.64%	0.67	0.68	0.69	0.97	0.97
	SynthID	6.31	+1.06%	7.14	+1.64%	0.67	0.68	0.69	0.97	0.97
	SIR	6.59	+5.54%	6.79	-3.22%	0.67	0.68	0.82	0.94	0.94
	Unbiased	6.62	+6.03%	7.05	+0.46%	0.67	0.67	0.66	0.97	0.97
	Ours(S)	**8.85**	**+11.47%**	**9.73**	**+9.47%**	**0.77**	**0.82**	**0.76**	**0.99**	**0.99**
	Ours(H)	**8.85**	**+11.47%**	**9.73**	**+9.47%**	**0.77**	**0.82**	**0.76**	**0.99**	**0.99**

note: 1. The number in the upper right corner represents the ranking. 2. ↑ denotes that the larger the value and ↓ means the opposite. 3. Temperature is 0.7 and model is DeepSeek-R1-Distill-Qwen-7B 4. Dataset is eli5.

Table 8: The results of different methods under temperature = 0.7.

Tasks	Datasets	Models	Methods	Semantic Fidelity				Robustness against Attacks						Detectability	
				Perplexity		Diversity		Del	Sub	Typo	Exp	Swap	LC	Performance	
				PPL↓	RC-PPL↓	LD↑	RC-LD↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	F1↑	ACC↑
Text Generation	**C4**	**OPT**	KGW	7.51	22.36%	6.80	-1.84%	0.97	0.96	0.96	1.00	0.99	1.00	1.00	1.00
			SWEET	7.50	22.11%	6.79	-2.09%	0.95	0.96	0.96	1.00	0.98	1.00	1.00	1.00
			EWD	7.46	21.46%	6.79	-2.04%	0.97	0.97	0.97	1.00	0.99	1.00	1.00	1.00
			Unigram	7.08	15.23%	6.17	-10.91%	0.99	0.99	0.97	1.00	1.00	0.99	1.00	1.00
			DiPMark	6.85	11.49%	7.11	2.61%	0.66	0.69	0.68	1.00	0.75	0.93	1.00	1.00
			EXP	19.91	224.28%	8.52	22.93%	0.82	0.90	0.89	1.00	0.98	0.99	1.00	1.00
			SynthID	15.36	150.15%	7.99	15.27%	0.85	0.88	0.86	1.00	0.97	1.00	1.00	1.00
			SIR	7.24	17.90%	6.70	-3.38%	0.81	0.81	0.81	0.99	0.94	0.98	0.99	0.99
			Unbiased	7.14	16.21%	7.12	2.74%	0.67	0.68	0.69	1.00	0.70	0.95	1.00	1.00
			Ours(H)	6.52	6.21%	6.93	0.03%	0.69	0.71	0.66	0.94	0.79	0.76	0.94	0.94
			Ours(S)	6.61	7.59%	6.85	-1.10%	0.80	0.84	0.67	0.99	0.91	0.91	0.99	0.99
Text Generation	**C4**	**GPT2**	KGW	8.04	25.21%	7.00	-0.02%	0.97	0.98	0.97	1.00	0.98	1.00	1.00	1.00
			SWEET	8.07	25.76%	7.03	0.44%	0.94	0.97	0.95	1.00	0.98	0.99	1.00	1.00
			EWD	8.14	26.83%	6.99	-0.16%	0.97	0.97	0.97	1.00	0.99	0.99	1.00	1.00
			Unigram	7.59	18.24%	6.90	-1.43%	1.00	0.99	0.99	1.00	1.00	0.99	1.00	1.00
			DIP	7.38	14.89%	7.24	3.40%	0.68	0.70	0.71	0.99	0.76	0.95	0.99	0.99
			EXP	23.87	271.74%	8.44	20.63%	0.84	0.92	0.92	1.00	0.98	0.99	1.00	1.00
			SynthID	18.24	184.13%	8.15	16.48%	0.81	0.88	0.84	1.00	0.94	0.99	1.00	1.00
			SIR	7.44	15.91%	6.79	-2.97%	0.73	0.75	0.71	0.97	0.89	0.94	0.97	0.97
			Unbiased	7.44	15.85%	7.29	4.12%	0.67	0.69	0.69	1.00	0.69	0.92	1.00	1.00
			Ours(H)	6.74	4.92%	6.98	-0.26%	0.71	0.71	0.67	0.93	0.79	0.81	0.93	0.93
			Ours(S)	6.81	6.10%	6.96	-0.53%	0.77	0.78	0.67	0.98	0.91	0.92	0.98	0.98
News Generation	**CNN-daily-Mail**	**Llama3**	KGW	4.81	12.19%	6.82	-0.33%	0.85	0.86	0.82	0.98	0.89	0.94	0.98	0.98
			SWEET	4.74	10.50%	6.88	0.62%	0.81	0.85	0.81	0.99	0.87	0.92	0.99	0.99
			EWD	4.91	14.40%	6.99	2.25%	0.84	0.87	0.85	0.99	0.89	0.94	0.99	0.99
			Unigram	5.02	16.91%	6.78	-0.88%	0.97	0.95	0.95	0.97	0.96	0.92	0.97	0.97
			DiPMark	4.47	4.08%	7.12	4.09%	0.67	0.67	0.67	0.93	0.68	0.77	0.93	0.94
			EXP	8.41	96.13%	7.19	5.12%	0.71	0.80	0.80	0.99	0.88	0.94	0.99	0.99
			SynthID	6.70	56.18%	7.42	8.46%	0.68	0.75	0.74	1.00	0.76	0.93	1.00	1.00
			SIR	4.85	13.14%	6.67	-2.45%	0.68	0.76	0.71	0.94	0.77	0.86	0.94	0.94
			Unbiased	4.45	3.82%	6.94	1.43%	0.67	0.68	0.67	0.93	0.68	0.76	0.92	0.92
			Ours(H)	4.59	6.92%	6.88	0.61%	0.75	0.70	0.67	0.93	0.81	0.83	0.93	0.93
			Ours(S)	4.67	8.76%	6.91	0.96%	0.79	0.71	0.67	0.96	0.83	0.85	0.96	0.96
Stroy Extension	**ROCStories**	**Qwen2.5**	KGW	8.13	26.22%	6.84	-0.80%	0.96	0.89	0.95	1.00	1.00	1.00	1.00	1.00
			SWEET	7.71	19.79%	6.77	-1.90%	0.94	0.92	0.94	0.99	0.98	0.99	0.99	1.00
			EWD	7.93	23.13%	6.83	-1.04%	0.94	0.93	0.95	1.00	0.99	1.00	1.00	1.00
			Unigram	7.89	22.45%	6.35	-8.04%	1.00	1.00	0.98	1.00	1.00	1.00	1.00	1.00
			DiPMark	7.26	12.70%	7.09	2.70%	0.67	0.67	0.68	1.00	0.74	0.97	1.00	1.00
			EXP	24.52	280.71%	8.93	29.35%	0.79	0.85	0.90	1.00	1.00	1.00	1.00	1.00
			SynthID	16.86	161.76%	8.02	16.17%	0.79	0.80	0.84	1.00	0.93	1.00	1.00	1.00
			SIR	7.67	19.09%	6.49	-5.87%	0.73	0.82	0.75	0.99	0.82	0.99	0.99	0.99
			Unbiased	7.28	12.99%	7.14	3.52%	0.68	0.67	0.69	1.00	0.70	0.97	1.00	1.00
			Ours(H)	7.06	9.70%	6.88	-0.29%	0.73	0.69	0.66	0.98	0.79	0.94	0.98	0.98
			Ours(S)	7.27	12.89%	6.97	1.04%	0.83	0.74	0.66	0.99	0.91	0.98	0.99	0.99
Long-form Question Aswer	**ELI5**	**DeepSeek-R1**	KGW	7.21	7.09%	7.17	-1.56%	0.90	0.91	0.90	1.00	0.92	0.99	1.00	1.00
			SWEET	6.91	2.70%	7.17	-1.56%	0.85	0.84	0.87	0.99	0.92	0.98	0.99	0.99
			EWD	7.12	5.86%	7.06	-3.07%	0.89	0.89	0.90	1.00	0.94	1.00	1.00	1.00
			Unigram	7.14	6.07%	7.08	-2.77%	0.94	0.89	0.93	0.98	0.95	0.97	0.98	0.98
			DiPMark	7.64	13.50%	7.28	0.01%	0.67	0.69	0.68	0.97	0.67	0.88	0.97	0.97
			EXP	9.39	39.53%	7.45	2.34%	0.70	0.78	0.77	0.99	0.82	0.98	0.99	1.00
			SynthID	9.32	38.43%	7.70	5.72%	0.68	0.73	0.75	1.00	0.78	0.98	1.00	1.00
			SIR	8.18	21.55%	7.23	-0.73%	0.67	0.69	0.67	0.93	0.67	0.87	0.93	0.94
			Unbiased	6.74	0.18%	7.19	-1.18%	0.67	0.67	0.68	0.95	0.68	0.85	0.95	0.95
			Ours(H)	7.06	4.84%	7.07	-2.93%	0.76	0.70	0.67	0.97	0.78	0.94	0.97	0.97
			Ours(S)	7.03	4.44%	7.09	-2.58%	0.80	0.68	0.67	0.96	0.80	0.95	0.96	0.97

note: 1. The number in the upper right corner represents the ranking. 2. ↑ denotes that the larger the value and ↓ means the opposite. 3. Temperature is 0.7.

Table 9: Examples of Prompts and Watermarked Text

Method	Prompt	Non-Watermarked Text	Watermarked Text	PPL	z
KGW	Belying expectations Prasar Bharti has earned only Rs 58.19 crore (Rs 581.9 million) as revenue during the	Be lying expectations , Pr asar Bh art i has earned only Rs 58 . 19 crore (Rs 5 81 . 9 million) as revenue during the first quarter of the current fiscal year , a drop of over 50 per cent from the same period last year . The revenue of the public broadcaster has been on a downward spiral since the government 's decision to privat ise it in 2012 . 	Be lying expectations , Pr asar Bh art i has earned only Rs 58 . 19 crore (Rs 5 81 . 9 million) as revenue during the first quarter of the current fiscal , a sharp fall from the Rs 1 , 9 21 crore (Rs 1 , 9 72 . 7 million) it earned during the same period last fiscal . The loss , however , was narrower than the Rs 1 	12.1	3.2
Method	**Prompt**	**Non-Watermarked Text**	**Watermarked Text**	**PPL**	**z**
EWD	Belying expectations Prasar Bharti has earned only Rs 58.19 crore (Rs 581.9 million) as revenue during the	Be lying expectations , Pr asar Bh art i has earned only Rs 58 . 19 crore (Rs 5 81 . 9 million) as revenue during the first quarter of the current fiscal year , a drop of over 50 per cent from the same period last year . The revenue of the public broadcaster has been on a downward spiral since the government 's decision to privat ise it in 2012 . 	Be lying expectations , Pr asar Bh art i has earned only Rs 58 . 19 crore (Rs 5 81 . 9 million) as revenue during the first quarter of the current fiscal . The revenue of the public broadcaster has dropped by Rs 1 . 5 billion compared to the same period last year . The revenue from advertisements fell by Rs 1 billion to Rs 5 . 5 billion . The revenue earned from broadcasting 	12.1	3.2
Method	**Prompt**	**Non-Watermarked Text**	**Watermarked Text**	**PPL**	**z**
Ours	Belying expectations Prasar Bharti has earned only Rs 58.19 crore (Rs 581.9 million) as revenue during the	Be lying expectations , Pr asar Bh art i has earned only Rs 58 . 19 crore (Rs 5 81 . 9 million) as revenue during the first quarter of the current fiscal year , a drop of over 50 per cent from the same period last year . The revenue of the public broadcaster has been on a downward spiral since the government 's decision to privat ise it in 2012 . 	Be lying expectations , Pr asar Bh art i has earned only Rs 58 . 19 crore (Rs 5 81 . 9 million) as revenue during the first week of the current financial year (July 1 , 2016 - July 31 , 2016) as against Rs 1 , 8 90 . 8 crore (Rs 2 , 8 90 million) earned by the company during the corresponding period of the previous financial year (	10.1	4.1