From Trade-off to Synergy: A Versatile Symbiotic Watermarking Framework for Large Language Models

Yidan Wang[1,2], Yubing Ren[1,2*], Yanan Cao[1,2*], Binxing Fang[3]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China

[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China

[3]Hainan Province Fang Binxing Academician Workstation, Hainan, China

{wangyidan, renyubing}@iie.ac.cn

Abstract

The rise of Large Language Models (LLMs) has heightened concerns about the misuse of AI-generated text, making watermarking a promising solution. Mainstream watermarking schemes for LLMs fall into two categories: logits-based and sampling-based. However, current schemes entail trade-offs among robustness, text quality, and security. To mitigate this, we integrate logits-based and sampling-based schemes, harnessing their respective strengths to achieve synergy. In this paper, we propose a versatile symbiotic watermarking framework with three strategies: serial, parallel, and hybrid. The hybrid framework adaptively embeds watermarks using token entropy and semantic entropy, optimizing the balance between detectability, robustness, text quality, and security. Furthermore, we validate our approach through comprehensive experiments on various datasets and models. Experimental results indicate that our method outperforms existing baselines and achieves state-of-the-art (SOTA) performance. We believe this framework provides novel insights into diverse watermarking paradigms. Our code is available at https://github.com/redwyd/SymMark.

1 Introduction

The exceptional capabilities of large language models (LLMs) (Touvron et al., 2023; Zhang et al., 2022) have revolutionized various fields, including creative content generation and automated writing, etc. The widespread accessibility of LLMs has significantly reduced the barriers to using AI-generated content, enabling broader adoption across diverse domains. While this democratization of technology brings substantial benefits, it also introduces critical challenges, including the potential misuse of LLMs for generating malicious content, violating intellectual property rights, and spreading disinformation (Liu et al., 2024b). To address these risks, watermarking has emerged as a promising solution for ensuring the traceability, authenticity, and accountability of LLM-generated content. By embedding invisible identifiers within generated text, watermarking provides a robust mechanism to trace content origins and mitigate misuse.

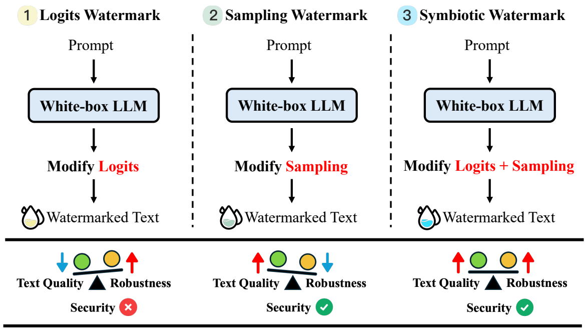

Figure 1: Paradigm comparison between our symbiotic watermark framework SymMark and existing logits-based watermark / sampling-based watermark.

However, existing watermarking methods face fundamental limitations that hinder their effectiveness in diverse and adversarial scenarios (Kirchenbauer et al., 2023; Kuditipudi et al., 2024). A key challenge lies in balancing **robustness** and **text quality**—increasing watermark strength often compromises the fluency and diversity of generated text while prioritizing quality can weaken robust to adversarial attacks (Wu et al., 2023; Zhao et al., 2024; Dathathri et al., 2024). Moreover, the security of watermarks remains a pressing issue. Current methods, such as the KGW family, are vulnerable to attacks like watermark stealing, where adversaries can potentially reverse-engineer watermark rules via frequency analysis, undermining their effectiveness (Jovanović et al., 2024; Pang et al., 2024; Wu and Chandrasekaran, 2024). Finally, as shown in Figure 1, the field lacks golden design principles, as both logits-based and sampling-based watermarkings face inherent trade-offs.

Can robustness, text quality, and security be har-

[*]Corresponding Authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 10306–10322
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

monized to work together, rather than being treated as conflicting objectives? Drawing inspiration from *symbiosis* in natural ecosystems, where different entities coexist and thrive through mutual benefits, we explore a novel perspective for watermarking. We introduce **SymMark**, a versatile symbiotic watermarking framework that transcends the traditional trade-offs in watermarking design. By transforming these trade-offs into synergy, SymMark combines the strengths of logits-based and sampling-based watermarking, providing an innovative solution that ensures robustness, text quality, and security, even under adversarial conditions.

Building on this symbiotic perspective, SymMark explores three strategies to integrate logits-based and sampling-based watermarking. Serial Symbiotic Watermarking **(Series)** embeds both watermarks in each token, ensuring high detectability. However, overly strong watermarks can degrade text quality. Parallel Symbiotic Watermarking **(Parallel)** alternates between the two methods at the token level, balancing robustness and text quality. Yet, it lacks flexibility, unable to adaptively select the optimal watermarking strategy for each token. To address these issues, we introduce Hybrid Symbiotic Watermarking **(Hybrid)**, our primary configuration. Hybrid applies a non-linear combination of both watermarking methods, adaptively choosing the most suitable strategy for each token. This may involve applying both watermarks, only one, or skipping watermarking altogether, depending on the token's context. By dynamically selecting the best strategy based on token and semantic entropy (Shannon, 1948; Farquhar et al., 2024), Hybrid enhances watermark security, resilience, and fluency. Additionally, we propose a unified algorithm to detect all three strategies effectively and efficiently.

Extensive experiments across multiple datasets and models consistently reveal that SymMark outperforms existing baselines. Specifically, the Serial excels in detectability and robustness, while the Parallel preserves high text quality without weakening watermark strength. Hybrid integrates the strengths of both approaches, making it the most comprehensive and effective strategy. Our main contributions are as follows:

- We systematically explore the integration of logits-based and sampling-based watermarking methods, pioneering a comprehensive approach to their synergy.
- We propose a versatile symbiotic watermarking framework, **SymMark**, which incorporates three distinct strategies: Series, Parallel, and Hybrid.
- Our exhaustive experiments demonstrate that the SymMark framework achieves state-of-the-art (SOTA) performance in terms of detectability, robustness, text quality, and security.

2 Related Work

The current mainstream LLM watermarking during the generation stage can be categorized into logits-based and sampling-based.

Logits-based Watermarking. The pioneering KGW method (Kirchenbauer et al., 2023) uses a hash key to divide the vocabulary into red and green lists, favoring green tokens in the output. To enhance watermark robustness, Unigram (Zhao et al., 2024) introduces a fixed red-green vocabulary partitioning scheme. Ren et al. (2024b) incorporate the vocabulary's prior distribution, and Ren et al. (2024a); He et al. (2024); Liu et al. (2024a); Liu and Bu (2024); Huo et al. (2024); Fu et al. (2024b); Chen et al. (2024) determine logits partitioning using semantic embeddings. Hu et al. (2024); Wu et al. (2024) explore unbiased watermarking to ensure identical expected distributions between watermarked and non-watermarked texts. To improve watermarked text quality, SWEET (Lee et al., 2024), EWD (Lu et al., 2024), and Wouters (2023) optimize watermarking from an entropy perspective. Furthermore, Guan et al. (2024); Fernandez et al. (2023); Wang et al. (2024); Yoo et al. (2024) investigate multi-bit watermarks to obtain higher capacity and convey more information.

Sampling-based Watermarking. In token-level sampling watermarking, Christ et al. (2024) employ a pseudo-random number to guide token generation, though it is unsuitable for real-world LLMs. Meanwhile, AAR (Aaronson, 2023) utilizes exponential minimum sampling to embed the watermark, while Fu et al. (2024a); Kuditipudi et al. (2024) build on this method to enhance text diversity further. Zhu et al. (2024) advocate contrastive decoding for sampling, and Dathathri et al. (2024) devise a tournament sampling scheme that preserves text quality while ensuring high detection accuracy. In sentence-level sampling watermarking, SemStamp (Hou et al., 2024a) divides the semantic space into watermarked and non-watermarked regions using locality-sensitive hashing. k-SemStamp (Hou et al., 2024b) further opti-

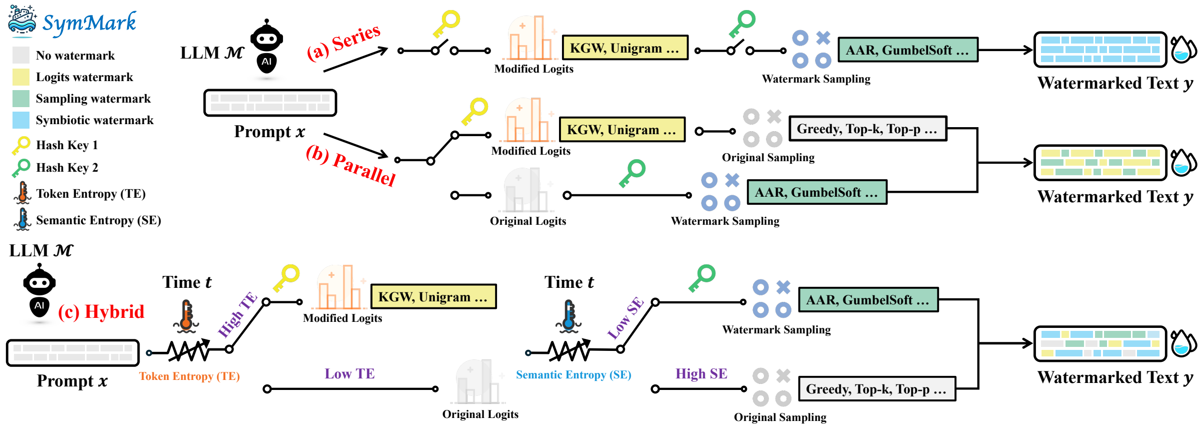

Figure 2: A Versatile Symbiotic Watermark Framework for LLMs.

mizes this process with a K-means clustering (MacQueen et al., 1967) algorithm.

3 Preliminary

3.1 LLM Generation

LLM $\mathcal{M}$ is a transformer-based (Vaswani, 2017) autoregressive neural network, characterized by its vocabulary $\mathcal{V}$ and parameters θ. The generation process of $\mathcal{M}$ involves two steps: **(1)** given prompt x and the previously generated tokens $y_{<t} = \{y_1, ..., y_{t-1}\}$, calculate t-th token's logits vector $l_t = \mathcal{M}(\cdot \mid x, y_{<t})$ of length $|\mathcal{V}|$, and then normalize it through softmax function to obtain a probability vector $p_t = \mathrm{softmax}(l_t)$; **(2)** Sample the t-th token based on p_t. Common sampling methods include greedy search, beam search, and multinomial sampling, among others.

3.2 LLM Watermarking

LLM watermarking is embedded into the token generation process by modifying one of two stages: (i) the logits generation stage, or (ii) the sampling stage. A typical watermarking in the logits stage is **KGW** (Kirchenbauer et al., 2023), which partitions the vocabulary into red and green lists with the γ ratio. This is achieved by hashing the previous k tokens with the watermark key ξ and applying a δ bias to the logits of each token in the green list, making the LLM more inclined to generate these tokens. During detection, hypothesis testing can determine if the text of length L contains a watermark by analyzing the number of green list tokens n_{green}. Specifically, if the proportion of green tokens significantly exceeds γ, with a high z-score $= (n_{\text{green}} - \gamma L)/\sqrt{L\gamma(1-\gamma)}$ above the threshold, the text is considered watermarked. Zhao et al. (2024) propose **Unigram**, a robust variant of KGW, that utilizes a fixed global split between red and green lists to generate watermark logits. However, Unigram is susceptible to statistical analysis, which could reveal the tokens classified as green. In contrast, the watermark in the sampling stage avoids altering the logits and embeds the watermark by modifying the sampling algorithm. **AAR** (Aaronson, 2023) proposes an exponential scheme to select tokens using $y_t = \arg\max_{i \in \mathcal{V}} (r_t^i)^{1/p_t^i}$, where $r_t \in [0,1]^{|\mathcal{V}|}$ is a random sequence, obtained by hashing the previous h tokens with a fixed watermark key ξ or by shifting the watermark key (Kuditipudi et al., 2024) to get multiple random sequences $r = \xi^{(1)}, ..., \xi^{(m)}$. During detection, if the hash scores r_t of the tokens in the observed sequence are high, the p-value will be low, indicating the presence of a watermark.

4 SymMark

This section first introduces three symbiotic watermark strategies—Series, Parallel, and Hybrid. Then outlines a unified symbiotic watermark detection algorithm.

4.1 Series Symbiotic Watermark

To fully embed the two watermarks and maximize the watermark signal, we designed the series symbiotic watermark, as illustrated in Figure 2 (a). When LLM generates t-th token, we first apply a logits-based watermarking $\mathcal{A}_w$ (e.g., KGW, Unigram, etc.) to modify the logits distribution l_t, followed by normalization via softmax function. During the sampling stage, we employ a sampling-based watermarking $\mathcal{S}_w$ (e.g., AAR, EXP, etc.) to generate the current token y_t:

Algorithm 1: Hybrid Symbiotic Watermark

```
Input: LLM $\mathcal{M}$, prompt $x$, ComputeEntropy $\mathcal{E}$
Params: Length $T$, TE Threshold $\alpha$, SE Threshold $\beta$
Output: Watermarked Text $y_{1:T}$
for $t = 1, 2..., T$ do
    $l_t \leftarrow \mathcal{M}(x, y_{<t})$
    $\hat{l}_t \leftarrow l_t$
    // Compute Two Entropy
    $H_{TE}, H_{SE} \leftarrow \mathcal{E}(l_t)$
    // Add Logits Watermark
    if $H_{TE} > \alpha$ then
        $\hat{l}_t \leftarrow \mathcal{A}_w(l_t)$
    end
    $\hat{p}_t \leftarrow \text{softmax}(\hat{l}_t)$
    // Add Sampling Watermark
    else if $H_{SE} < \beta$ then
        $y_t \sim S_w(\hat{p}_t)$
        continue
    end
    // Origin Sampling Method
    $y_t \sim S_o(\hat{p}_t)$
end
```

$$y_t = \mathcal{S}_w(\text{softmax}(\mathcal{A}_w(l_t))) \tag{1}$$

4.2 Parallel Symbiotic Watermark

To independently embed two watermark signals while minimizing their mutual interference, we propose a parallel symbiotic watermark, as shown in Figure 2 (b). This approach embeds either a logits-based or sampling-based watermark as the LLM generates the current token y_t. Specifically, at odd positions, the logits-based watermarking $\mathcal{A}_w$ modifies the logits distribution to embed the watermark, preserving the original sampling algorithm $\mathcal{S}_o$. At even positions, the logits distribution remains unchanged, embedding the watermark with the sampling-based watermarking $\mathcal{S}_w$. The formal representation is as follows, where $k \in \mathbb{N}$:

$$y_t = \begin{cases} \mathcal{S}_o(\text{softmax}(\mathcal{A}_w(l_t))), & t = 2k \\ \mathcal{S}_w(\text{softmax}(l_t)), & t = 2k+1 \end{cases} \tag{2}$$

4.3 Hybrid Symbiotic Watermark

To achieve a synergy between logits-based and sampling-based watermarks, we propose an adaptive hybrid symbiotic watermarking method, as illustrated in Figure 2 (c). This approach leverages two key entropy measures to dynamically decide the watermarking strategy: token entropy determines whether to apply logits-based watermarking, while semantic entropy governs the use of sampling-based watermarking.

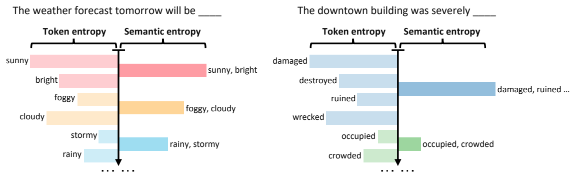

Figure 3: High Token Entropy with High Semantic Entropy (Left) and Low Semantic Entropy (Right).

Token Entropy Derived from Shannon entropy (Shannon, 1948), quantifies the uncertainty in the logits distribution of a token at the current time step t. Given the model's logits output, we apply softmax normalization to obtain the probability p_t^i for each token $i \in \mathcal{V}$, and compute token entropy as follows:

$$H_{TE} = -\sum_i p_t^i \log p_t^i, \quad i \in \mathcal{V} \tag{3}$$

Token entropy serves as the basis for applying logits watermarking because it reflects the model's confidence in generating a particular token. **Low token entropy** (high confidence) indicates the model strongly prefers a specific token, meaning that altering logits may significantly affect the fluency and naturalness of the generated text. Thus, applying logits watermarking could be intrusive. **High token entropy** (low confidence) indicates the model exhibits greater uncertainty, with multiple competing candidates in the logits distribution. Since the token choice is inherently unstable, modifying logits introduces minimal disruption to text quality while ensuring effective watermark embedding.

Semantic Entropy Semantic entropy measures the diversity of the top-k candidate tokens at time step t in terms of their semantic meaning. To compute semantic entropy, we extract the embeddings of the top-k tokens from the logits distribution and cluster them into n groups $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_n\}$ using K-means (MacQueen et al., 1967). The logits are then merged according to the cluster assignments, as shown in Equation 4, and the final semantic entropy is computed from the merged logits, as detailed in Equation 5.

$$q_t^j = \sum_{i=1}^{|\mathcal{C}_j|} p_t^i, \quad i \in \mathcal{C}_j \tag{4}$$

$$H_{SE} = -\sum_j q_t^j \log q_t^j, \quad j \in \{1, ..., n\} \tag{5}$$

Algorithm 2: Symbiotic Watermark Detection

```
Input: $\mathcal{M}, y_{1:T}, \mathcal{D}_l, \mathcal{D}_s, z_1, z_2$
Output: $I$: True (Watermarked) or False
$I_l \leftarrow$ False
$I_s \leftarrow$ False
// Logits Watermark Detection
if $\mathcal{D}_l(\mathcal{M}, y_{1:T}) > z_1$ then
    $I_l \leftarrow$ True
end
// Sampling Watermark Detection
if $\mathcal{D}_s(\mathcal{M}, y_{1:T}) > z_2$ then
    $I_s \leftarrow$ True
end
$I \leftarrow I_l \mid I_s$
```

Semantic entropy determines whether to apply sampling watermarking by assessing how semantically diverse the top-ranked candidates are. As illustrated in Figure 3, **low semantic entropy** (high semantic similarity) means that the top candidates have similar meanings, implying that replacing one with another will have a negligible impact on text interpretation. Thus, adding a sampling watermark is unlikely to alter the meaning of the generated content. While **high semantic entropy** (low semantic similarity) indicates the top candidates exhibit substantial semantic variation. In such cases, altering the sampling process could disrupt the intended meaning of the sentence, making sampling watermarking undesirable. Experimental analysis is provided in Appendix H.

Algorithm 1 details the overall process. Given a logits distribution generated by the LLM $\mathcal{M}$, we first compute token entropy H_{TE} and semantic entropy H_{SE}. If H_{TE} exceeds the predefined threshold α, logits watermarking is applied; otherwise, the logits remain unchanged. After normalization via softmax and sampling, we check H_{SE}: if it falls below the predefined threshold β, sampling watermarking is applied, ensuring that the final text preserves semantic integrity. This hybrid strategy dynamically selects the optimal watermarking method for each token, achieving robust and high-quality watermark embedding.

4.4 Symbiotic Watermark Detection

Algorithm 2 presents the symbiotic watermark detection process. Given the watermark model $\mathcal{M}$, the generated content $y_{1:T}$, the logits-based detection algorithm D_l, and the sampling-based detection algorithm D_s, the watermark is deemed present if any watermark signal is detected due to the method's low false positive rate. Theoretically, tokens can be grouped according to different symbiotic watermark frameworks for detection. Further analysis is provided in Appendix I.

5 Experimental Setup

Dataset and Prompt. To measure detectability, we follow Kirchenbauer et al. (2023); Zhao et al. (2024) and use subsets of the news-like C4 dataset (Raffel et al., 2020) and the long-form OpenGen dataset (Krishna et al., 2023) to insert watermarks. For each sample, the last 200 tokens are treated as natural text (i.e., human-written), while the remaining tokens from the start are used as prompts. We then generate $T = 200 \pm 30$ tokens (i.e., watermarked text) using LLMs conditioned on the prompts. To evaluate text quality, we followed the Waterbench (Tu et al., 2024) framework and tested four downstream tasks: Factual Knowledge, Long-form QA, Code Completion, and Text Summarization. Details are in Appendix C.

Models. We conducted experiments using three model series: the OPT series (OPT-6.7B, OPT-2.7B, OPT-1.3B) (Zhang et al., 2022), the LLaMA series (LLaMA3-8B-Instruct, LLaMA2-7B-chat-hf) (Dubey et al., 2024; Touvron et al., 2023), and the GPT series (GPT-J-6B) (Wang and Komatsuzaki, 2021). Notably, semantic clustering requires using a model with the same tokenizer as the original watermark model.

Baselines. We compared SymMark with dozens of existing methods, including logits-based watermark KGW (Kirchenbauer et al., 2023), Unigram (Zhao et al., 2024), SWEET (Lee et al., 2024), EWD (Lu et al., 2024), DIP (Wu et al., 2024), Unbiased (Hu et al., 2024) and sampling-based watermark AAR (Aaronson, 2023), EXP (Kuditipudi et al., 2024), ITS (Kuditipudi et al., 2024), GumbelSoft (Fu et al., 2024a), SynthID (Dathathri et al., 2024). Detailed introductions are in Appendix D.

Evaluation Metrics. Watermark detectability is evaluated using True Positive Rate (TPR), True Negative Rate (TNR), Best F1 Score, and AUC metrics. Watermark robustness is assessed through the AUROC curve, which illustrates the FPR (False Positive Rate) and TPR across varying thresholds.

Implementation Details. Our symbiotic watermark selects the representative logits-based Unigram watermark (Zhao et al., 2024), with the classic sampling-based AAR watermark (Aaronson, 2023). The hybrid symbiotic watermark employs

Watermark	C4 DATASET								OPENGEN DATASET							
	OPT-6.7B				GPT-J-6B				OPT-6.7B				GPT-J-6B			
	TPR	TNR	F1	AUC	TPR	TNR	F1	AUC	TPR	TNR	F1	AUC	TPR	TNR	F1	AUC
Logits Watermark																
KGW	0.990	1.000	0.994	0.999	0.995	0.995	0.995	0.999	1.000	1.000	1.000	1.000	0.995	0.990	0.992	0.997
DIP	0.985	0.995	0.989	0.999	0.990	1.000	0.994	0.995	0.995	0.995	0.995	0.998	0.940	0.995	0.966	0.985
EWD	0.995	0.995	0.995	0.997	0.995	1.000	0.997	0.999	1.000	1.000	1.000	1.000	0.995	0.995	0.995	0.998
SWEET	0.985	1.000	0.992	0.998	1.000	0.995	0.997	0.999	0.990	1.000	0.994	0.999	0.980	1.000	0.990	0.990
Unigram	0.995	1.000	0.997	0.998	0.995	1.000	0.997	0.999	1.000	1.000	1.000	1.000	0.990	1.000	0.994	0.999
Unbiased	0.980	0.990	0.984	0.995	0.975	1.000	0.987	0.998	1.000	0.980	0.990	0.999	0.975	1.000	0.987	0.991
Sampling Watermark																
AAR	0.995	1.000	0.997	0.999	0.995	1.000	0.997	0.995	1.000	1.000	1.000	1.000	0.995	1.000	0.997	0.999
EXP	0.975	0.925	0.951	0.960	0.975	0.945	0.960	0.970	0.980	0.925	0.953	0.960	0.990	0.965	0.977	0.977
ITS	0.965	0.950	0.957	0.968	0.980	0.985	0.982	0.987	0.925	0.890	0.909	0.928	0.985	0.970	0.978	0.979
GumbelSoft	0.975	1.000	0.987	0.983	0.990	1.000	0.994	0.995	1.000	1.000	1.000	1.000	0.985	1.000	0.992	0.994
SynthID	0.985	0.995	0.989	0.998	1.000	1.000	1.000	1.000	0.995	1.000	0.997	0.999	0.955	0.995	0.974	0.995
Symbiotic Watermark (Ours)																
Series	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**
Parallel	0.995	0.995	0.995	0.997	1.000	0.990	0.995	0.998	1.000	0.990	0.995	0.999	1.000	0.995	0.997	0.997
Hybrid	**0.995**	**1.000**	**0.997**	**0.999**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**0.995**	**1.000**	**0.997**	**0.999**

Table 1: Detectability of OPT-6.7B and GPT-J-6B under different watermarking algorithms on C4 and OpenGen.

the K-means (MacQueen et al., 1967) clustering algorithm with the following default hyperparameters: Top-k token numbers $k = 64$, clusters number $n = 10$, token entropy threshold $\alpha = 1.0$, and semantic entropy threshold $\beta = 0.5$. Detailed Hyperparameter Analysis is in Appendix G.

6 Experimental Analysis

To demonstrate SymMark's superiority, we evaluated it in four aspects: detectability, robustness, text quality, and security. The experimental results show that the Serial excels in detectability and robustness, Parallel better preserves text quality, and Hybrid achieves the best overall balance.

6.1 Detectability

Table 1 presents the overall watermark detection results for two datasets and four base models.

Series scheme achieves state-of-the-art (SOTA) detectability performance. Series scheme exhibits a perfect TPR of **1.000** across all datasets and models, signifying no false positives, which is crucial given the higher impact of false positives in watermarking contexts. This is due to the injection of double watermark signals into each token, reinforcing the watermark presence throughout the sequence. However, this enhanced detectability comes at the cost of text quality, as strong constraints are imposed on token selection at both the logits and sampling stages.

Parallel scheme demonstrates competitive detectability performance, with an average F1/AUC improvement of 1.60%/1.35% over sampling watermark. Despite each token being modified by only one of the two watermarking strategies (logits or sampling), sufficient watermark signal remains for detection. This result highlights that doubling watermarking is not always necessary for detection.

Hybrid scheme consistently outperforms baselines across various datasets and base model configurations, evidencing its remarkable generalization. Specifically, Compared to the sampling watermark, Hybrid's F1/AUC performance improves by 1.90%/1.52% on average. This scheme adaptively assigns watermarking strategies based on entropy characteristics, which enables optimal watermark placement, ensuring high detectability while preserving text quality.

6.2 Text Quality

To evaluate the impact of our watermarking framework on text quality, we focus on perplexity and downstream tasks. Table 2 and Figure 4 show that our hybrid scheme achieves minimal performance drop and the lowest perplexity than baselines.

Perplexity. To assess the fluency of watermarked text, we used LLaMA2-7B to compute the perplexity (PPL) of texts generated by models of varying sizes with different watermarking algorithms. As shown in Figure 4, the Parallel scheme results in a

Model	T1: Short Q, Short A *Factual Knowledge*				T2: Short Q, Long A *Long-form QA*				T3: Long Q, Short A *Reasoning & Coding*				T4: Long Q, Long A *Summarization*			
+ Watermark	TPR	TNR	GM	DROP	TPR	TNR	GM	DROP	TPR	TNR	GM	DROP	TPR	TNR	GM	DROP
LLAMA3-8B	-	-	57.50	-	-	-	24.05	-	-	-	48.43	-	-	-	27.18	-
+ KGW	0.815	0.700	56.00	↓ 2.61%	0.990	0.975	23.32	↓ 3.04%	0.740	0.845	36.40	↓ 24.8%	0.955	0.985	26.66	↓ 1.91%
+ Unigram	0.955	0.360	51.00	↓ 11.3%	0.965	0.990	23.24	↓ 3.37%	0.775	0.695	40.95	↓ 15.4%	0.915	0.890	26.89	↓ 1.07%
+ EWD	0.860	0.745	49.00	↓ 14.8%	1.000	1.000	23.52	↓ 2.20%	0.740	0.850	35.45	↓ 26.8%	0.965	0.990	26.68	↓ 1.84%
+ AAR	0.685	0.930	46.00	↓ 18.3%	0.995	1.000	21.95	↓ 8.73%	0.910	0.990	38.95	↓ 19.6%	1.000	0.995	25.14	↓ 7.51%
+ SynthID	0.780	0.530	51.00	↓ 11.3%	0.990	0.970	23.60	↓ 1.87%	0.790	0.695	39.10	↓ 19.3%	0.955	0.935	26.83	↓ 1.29%
+ **Series**	0.970	0.935	55.00	↓ 4.35%	0.950	1.000	21.82	↓ 9.27%	0.770	0.995	41.26	↓ 14.8%	0.930	1.000	26.22	↓ 3.53%
+ **Parallel**	0.965	0.450	52.00	↓ 9.57%	0.730	0.970	22.35	↓ 7.07%	0.765	1.000	42.63	↓ 12.0%	0.910	0.940	26.76	↓ 1.55%
+ **Hybrid**	1.000	0.960	57.00	↓ **0.87%**	0.965	1.000	23.61	↓ **1.83%**	0.925	0.990	42.65	↓ **11.9%**	0.965	0.995	26.92	↓ **0.96%**

Table 2: The performance of various watermarking algorithms across four different downstream tasks using True Positive Rate (TPR), True Negative Rate (TNR), Generation Metric (GM), and Generation Quality Drop (Drop).

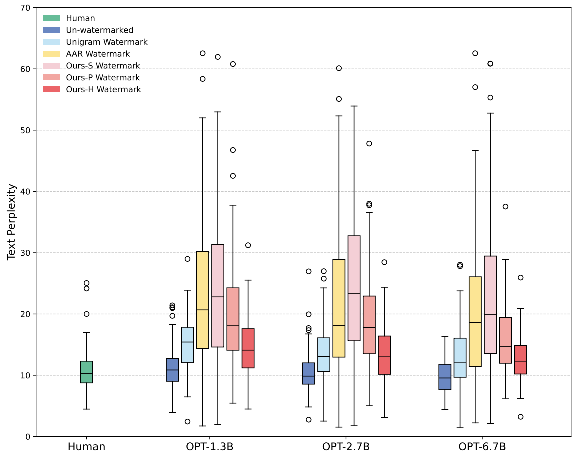

Figure 4: A comparison of PPL across three symbiotic watermarking schemes with different model sizes.

lower PPL compared to the Serial scheme, as double watermarking per token degrades text quality more than single watermarking. Unlike Parallel watermarking, which groups tokens by odd and even positions, hybrid watermarking introduces semantic entropy and adaptively applies stage-specific watermarks, effectively managing text quality and achieving the lowest PPL.

Downstream Task. Fidelity is the cornerstone of watermarking algorithms, to further validate the impact of watermarking on text quality, we followed Waterbench (Tu et al., 2024) settings and considered four downstream tasks (Details refer to Appendix C). The results in Table 2 indicate that the longer the generated answers (e.g., Task 2 and Task 4), the smaller the impact of the injected watermark on downstream tasks. Across all tasks, our hybrid scheme consistently achieves a high detection rate and superior task performance. Specifically, performance drops by only 0.87% on Task 1 and 0.96% on Task 4, demonstrating minimal distortion. Compared to baselines, SynthID imposes relatively minor text quality degradation but suffers from a lower detection rate, whereas other baselines exhibit either excessive text degradation or weaker detectability. In contrast, the Hybrid scheme strategically ensures strong detectability while preserving text fidelity, more suitable for real-world deployment.

6.3 Robustness to Real-world Attacks

Ensuring the robustness of watermarking schemes against various attacks is crucial for real-world applicability (Kirchenbauer et al., 2024). To provide comprehensive evidence of SymMark's robustness, we conduct experiments to test its resilience against four attacks: **Editing, Copy-Paste, Back-Translation, and Rephrasing**. Details are in Appendix F.

The ROC curves and AUC values for comparison in Figure 5 indicate **Hybrid's consistently robust watermark detection capabilities facing all attack scenarios**. The average AUC values of serial and hybrid symbiotic watermarks are 0.987 and 0.984, respectively, significantly outperforming Unigram, the previously most robust method, with an AUC of 0.951. The Parallel scheme shows a relatively lower AUC, suggesting that injecting only one watermark signal per token is more vulnerable to adversarial modifications.

Hybrid excels in robustness is due to: (1) Dual-signal Injection. Hybrid ensures that even if one watermarking signal is partially disrupted, the other remains intact, enabling reliable detection; (2) Entropy-driven Adaptation. Unlike fixed strategies, Hybrid is driven by entropy to adaptively select watermarking constraints, ensuring both imperceptibility and resilience; (3) Cross-attack Gen-

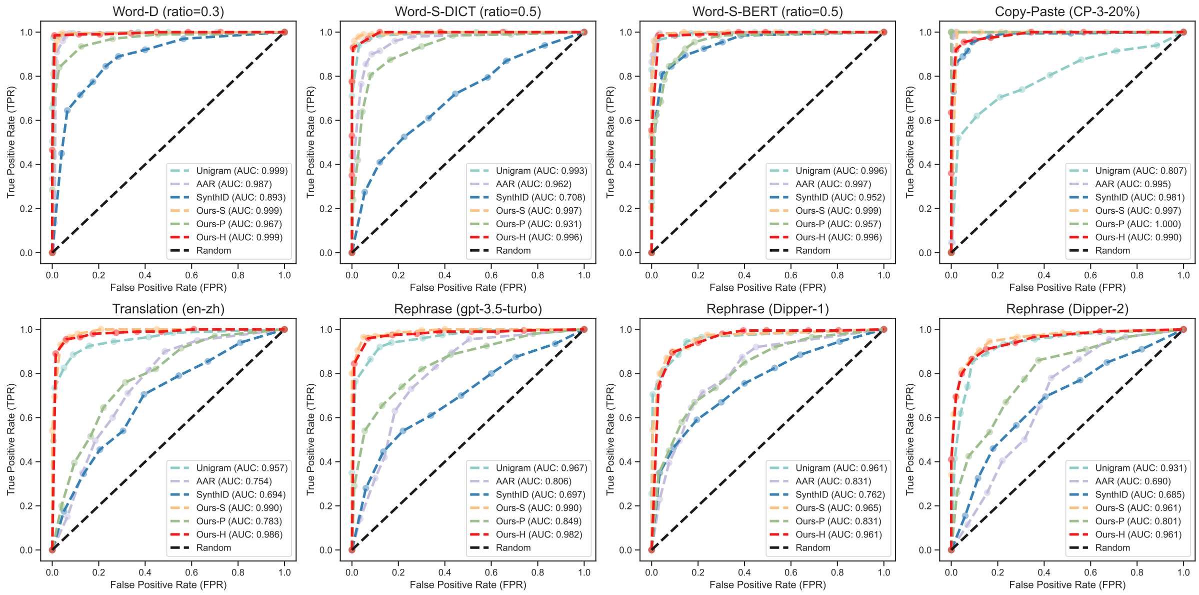

Figure 5: The AUROC curve of watermarked text generated by OPT-6.7B under various attacks on C4 dataset.

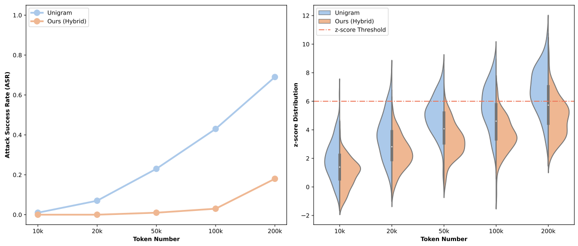

Figure 6: The ASR of watermark stealing for varying numbers of tokens (left) and the z-score distribution of spoofing watermark (right) on LLaMA2-7B-chat-hf.

eralization. While some methods perform well on specific attacks, Hybrid maintains high detection rates across diverse attack categories, making it practical for real-world deployment where adversarial conditions are unpredictable.

6.4 Security

Existing watermark stealing strategies, such as those targeting logits-based methods (e.g., the KGW family), are ineffective against sampling-based watermarks, which remain immune to such attacks. To explore the security of symbiotic watermarks, we apply the watermark stealing method and perform a spoofing attack (Jovanović et al., 2024; Pang et al., 2024) on the Unigram and our Hybrid. Detailed settings are in Appendix J.

Figure 6 presents stealing results. The left panel depicts the Attack Success Rate (ASR) of watermark stealing, while the right panel presents the z-score distribution of spoofed Unigram and our Hybrid across different token counts. As the number of tokens obtained by the attacker increases, so does the ASR and z-score. However, the ASR and z-score of Hybrid scheme is much lower than that of the naive Unigram. When generating 200,000 tokens, the ASR for the original Unigram reaches 69%, whereas the ASR for our symbiotic watermark scheme is only 18%.

The enhanced security of the Hybrid scheme stems from its non-linear combination of logits-based and sampling-based watermarking methods. Since the symbiotic watermarking rules are influenced not only by the logits but also by the inherent randomness in the sampling process, attackers are unable to reconstruct the watermarking rules purely through token frequency statistics or distribution modeling. This makes the Hybrid scheme significantly more resistant to watermark stealing attacks, offering enhanced security, particularly in adversarial environments where attackers are actively attempting to subvert the watermark.

7 Conclusion

This paper introduces a versatile symbiotic watermarking framework including three strategies: Serial, Parallel, and Hybrid. The Hybrid symbiotic watermark strategy leverages token and semantic entropy to balance detectability, robustness, text quality, and security. Experimental results across various datasets and models demonstrate the effectiveness of our method, shifting the focus from trade-offs to synergy. In the future, we will explore additional symbiotic watermarking paradigms, investigating perspectives beyond entropy to further advance watermarking techniques.

8 Limitations

This paper explores combining logits-based and sampling-based watermarks from an entropy perspective, while acknowledging that entropy is not the only evaluation metric. Future research could adopt other mathematical or information-theoretic tools to enhance symbiotic watermark design. Metrics like information gain and signal-to-noise ratio, alongside entropy, may offer deeper insights into watermark performance, robustness, and efficiency. These metrics can support the development of more adaptable watermarking schemes for diverse applications. Considering alternative metrics may lead to more flexible watermark designs suitable for varied scenarios. Despite limitations, we believe the symbiotic watermark concept offers a novel perspective and meaningful direction for advancing LLM watermarking in this fast-evolving field.

9 Ethical Statement

With the rapid development of large language models (LLMs) and their widespread applications, incorporating watermarks into LLM-generated content facilitates traceability, thereby significantly enhancing transparency and accountability. Building on previous research, this paper seeks to achieve a balance among the detectability, text quality, security, and robustness of watermarks. We aspire for the framework proposed in this paper to offer novel insights into watermarking methodologies and to be further utilized in safeguarding intellectual property, curbing misinformation, and mitigating AIGC misuse, including academic fraud, thereby fostering public trust in AI technologies.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (No.U2336202) and Hainan Province Fang Binxing Academician Workstation (No.YSGZZ2023003). The authors gratefully acknowledge this support, which made the completion of this work possible.

References

Scott Aaronson. 2023. Watermarking of large language models. In *Large Language Models and Transformers Workshop at Simons Institute for the Theory of Computing, 2023*.

Liang Chen, Yatao Bian, Yang Deng, Deng Cai, Shuaiyi Li, Peilin Zhao, and Kam-Fai Wong. 2024. WatME: Towards lossless watermarking through lexical redundancy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9166–9180, Bangkok, Thailand. Association for Computational Linguistics.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Miranda Christ, Sam Gunn, and Or Zamir. 2024. Undetectable watermarks for language models. In *The Thirty Seventh Annual Conference on Learning Theory*, pages 1125–1139. PMLR.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm. *Company Blog of Databricks*.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, et al. 2024. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Alexander Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. 2019. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1074–1084, Florence, Italy. Association for Computational Linguistics.

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.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three bricks to consolidate watermarks for large language models. *Preprint*, arXiv:2308.00113.

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. 2024a. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5791–5808, Bangkok, Thailand. Association for Computational Linguistics.

Yu Fu, Deyi Xiong, and Yue Dong. 2024b. Watermarking conditional text generation for ai detection: Unveiling challenges and a semantic-aware watermark remedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18003–18011.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. 2024. On the learnability of watermarks for language models. In *The Twelfth International Conference on Learning Representations*.

Batu Guan, Yao Wan, Zhangqian Bi, Zheng Wang, Hongyu Zhang, Pan Zhou, and Lichao Sun. 2024. CodeIP: A grammar-guided multi-bit watermark for large language models of code. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 9243–9258, Miami, Florida, USA. Association for Computational Linguistics.

Zhiwei He, Binglin Zhou, Hongkun Hao, Aiwei Liu, Xing Wang, Zhaopeng Tu, Zhuosheng Zhang, and Rui Wang. 2024. Can watermarks survive translation? on the cross-lingual consistency of text watermark for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4115–4129, Bangkok, Thailand. Association for Computational Linguistics.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024a. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4067–4082, Mexico City, Mexico. Association for Computational Linguistics.

Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. 2024b. k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1706–1715, Bangkok, Thailand. Association for Computational Linguistics.

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

Mingjia Huo, Sai Ashish Somayajula, Youwei Liang, Ruisi Zhang, Farinaz Koushanfar, and Pengtao Xie. 2024. Token-specific watermarking with enhanced detectability and semantic coherence for large language models. In *Forty-first International Conference on Machine Learning*.

Nikola Jovanović, Robin Staab, and Martin Vechev. 2024. Watermark stealing in large language models. In *ICLR 2024 Workshop on Reliable and Responsible Foundation Models*.

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.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2024. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Frederick Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2024. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*.

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.

Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. 2023. GPT detectors are biased against non-native english writers. In *ICLR 2023 Workshop on Trustworthy and Reliable Large-Scale Machine Learning Models*.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024a. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip Yu. 2024b. A survey of text watermarking in the era of large language models. *ACM Computing Surveys*, 57(2):1–36.

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. *Preprint*, arXiv:2401.13927.

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.

Yiyang Luo, Ke Lin, and Chao Gu. 2024. Lost in overlap: Exploring watermark collision in llms. *Preprint*, arXiv:2403.10020.

James MacQueen et al. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

R OpenAI et al. 2023. Gpt-4 technical report. *ArXiv*, 2303:08774.

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. No free lunch in LLM watermarking: Trade-offs in watermarking design choices. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Hao Peng, Xiaozhi Wang, Shengding Hu, Hailong Jin, Lei Hou, Juanzi Li, Zhiyuan Liu, and Qun Liu. 2022. COPEN: Probing conceptual knowledge in pre-trained language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5015–5035, Abu Dhabi, United Arab Emirates. 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. *Journal of machine learning research*, 21(140):1–67.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2024a. A robust semantics-based watermark for large language model against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 613–625, Mexico City, Mexico. Association for Computational Linguistics.

Yubing Ren, Ping Guo, Yanan Cao, and Wei Ma. 2024b. Subtle signatures, strong shields: Advancing robust and imperceptible watermarking in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5508–5519, Bangkok, Thailand. Association for Computational Linguistics.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. 2024. Can AI-generated text be reliably detected?

C. E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

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

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2024. WaterBench: Towards holistic evaluation of watermarks for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1517–1542, Bangkok, Thailand. Association for Computational Linguistics.

A Vaswani. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. `https://github.com/kingoflolz/mesh-transformer-jax`.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. 2024. Towards codable watermarking for injecting multi-bits information to LLMs. In *The Twelfth International Conference on Learning Representations*.

Bram Wouters. 2023. Optimizing watermarks for large language models. *arXiv preprint arXiv:2312.17295*.

Qilong Wu and Varun Chandrasekaran. 2024. Bypassing LLM watermarks with color-aware substitutions. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8549–8581, Bangkok, Thailand. Association for Computational Linguistics.

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

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023. Dipmark: A stealthy, efficient and resilient watermark for large language models. *arXiv preprint arXiv:2310.07710*.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2024. Advancing beyond identification: Multi-bit watermark for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4031–4055, Mexico City, Mexico. 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, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Zhaoxi Zhang, Xiaomei Zhang, Yanjun Zhang, Leo Yu Zhang, Chao Chen, Shengshan Hu, Asif Gill, and Shirui Pan. 2024. Large language model watermark stealing with mixed integer programming. *arXiv preprint arXiv:2405.19677*.

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

Chaoyi Zhu, Jeroen Galjaard, Pin-Yu Chen, and Lydia Chen. 2024. Duwak: Dual watermarks in large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 11416–11436, Bangkok, Thailand. Association for Computational Linguistics.

A Efficient Analysis

Method	KGW	AAR	EXP	Series	Parallel	Hybrid
Generation Time	8.475s	8.605s	8.260s	8.745s	12.675s	15.575s
Detection Time	0.035s	0.045s	65.74s	0.050s	0.060s	0.050s

Table 3: The computational efficiency analysis of different watermarking for each text of length 200 tokens.

All experiments were conducted on two NVIDIA A100 GPUs. Table 3 presents the average time required by several representative watermarking methods to generate and detect watermark texts of 200 tokens using OPT-6.7B. Our symbiotic watermarking strategy achieves nearly the same efficiency as existing methods in watermark detection. Although our hybrid watermarking method incurs additional computation time for token and semantic entropy during watermark text generation, this overhead remains acceptable in practical applications and contributes to enhanced robustness, security, and text quality. Furthermore, this overhead could be mitigated if entropy calculation were integrated into the Hugging Face[1] tool library in the future.

B Distinguishing Human-Written Text

Based on Liang et al. (2023), we evaluated our method using the TOEFL dataset, comprising non-native English writing samples, as shown in Figure 7. The experimental results show that our approach reliably identifies text with watermarks while non-native English writing samples are susceptible to misclassification by existing AIGT (AI-generated text) detection methods. These findings highlight the practicality and reliability of our watermarking method, which achieves a near-zero FPR.

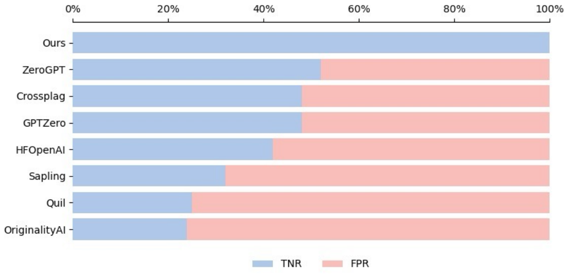

Figure 7: Comparing AIGT detection methods and ours in distinguishing human-written text on TOEFL dataset.

C Downstream Task Datasets

Referring to Waterbench (Tu et al., 2024), we utilize the following datasets:

- **Category 1 (Short Input, Short Answer)** includes the concept-probing Copen dataset (Peng et al., 2022), with 200 samples selected from the CIC and CSJ tasks. Given the short output length, the **F1 score** is chosen as the evaluation metric. The max_new_tokens parameter for model generation is set to **16**.

- **Category 2 (Short Input, Long Answer)** utilizes 200 samples from the ELI5 dataset (Fan et al., 2019), a long-form question-answering dataset originating from the Reddit forum "Explain Like I'm Five." **Rouge-L** is employed as the evaluation metric. The max_new_tokens parameter for model generation is set to **300**.

- **Category 3 (Long Input, Short Answer)** addresses the code completion task, utilizing 200

[1] https://huggingface.co/

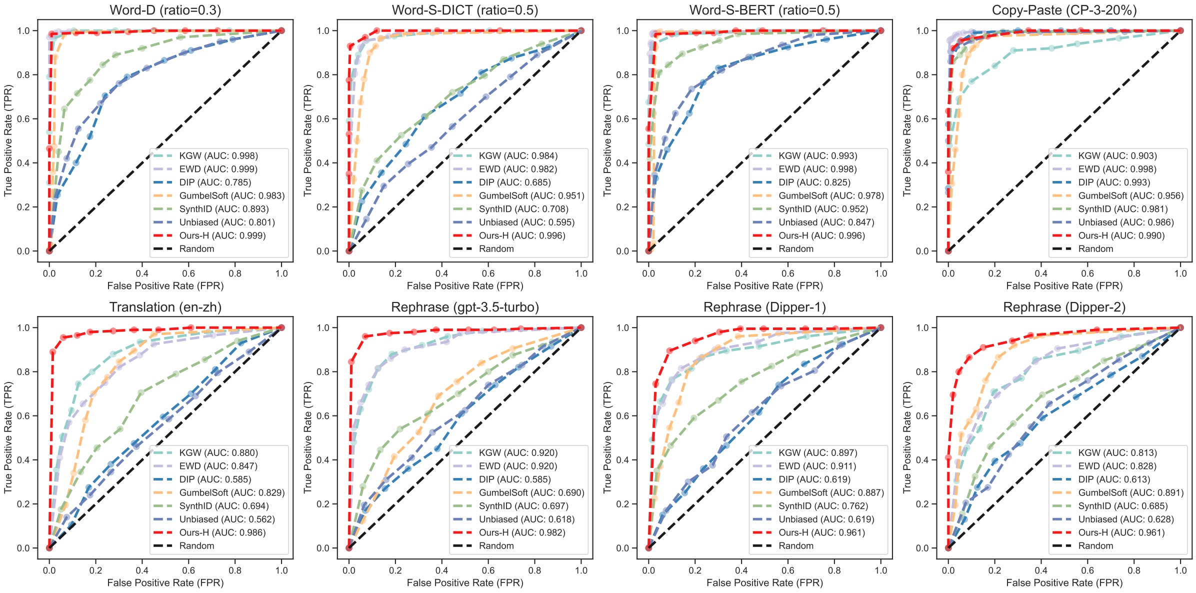

Figure 8: The AUROC curve of watermarked text generated by OPT-6.7B under various attacks on C4 dataset.

samples from the LCC dataset (Chen et al., 2021). This dataset is created by filtering single-file code samples from GitHub, with the **Edit Similarity** metric adopted for evaluation. The max_new_tokens parameter for model generation is set to **64**.

- **Category 4 (Long Input, Long Answer)** involves 200 samples from the widely-used MultiNews dataset (Fabbri et al., 2019), a multi-document news summarization dataset. **Rouge-L** serves as the evaluation metric. The max_new_tokens parameter for model generation is set to **512**.

D Baseline Settings

We use MarkLLM (Pan et al., 2024) toolkit to implement both the baseline and our proposed method, as detailed below:

- **KGW** proposed by Kirchenbauer et al. (2023), the details of the parameters are as follows: γ = 0.5, δ = 0.2, ξ = 15485863, prefix_length = 1, z_threshold = 4.0, window_scheme = "left".

- **Unigram** proposed by Zhao et al. (2024), the details of the parameters are as follows: γ = 0.5, δ = 2.0, ξ = 15485863, z_threshold = 4.0

- **DIP** proposed by Wu et al. (2024), the details of the parameters are as follows: γ = 0.5, α = 0.45, key = 42,prefix_length = 5, z_threshold=1.513

- **SWEET** proposed by Lee et al. (2024), the details of the parameters are as follows: γ = 0.5, δ = 2.0, ξ = 15485863, prefix_length = 1, z_threshold = 4.0, entropy_threshold = 0.9

- **EWD** proposed by Lu et al. (2024), the details of the parameters are as follows: γ = 0.5, δ = 2.0, ξ = 15485863, prefix_length = 1, z_threshold=4.0

- **Unbiased** proposed by Hu et al. (2024), the details of the parameters are as follows: γ = 0.5, key = 42, prefix_length = 5, z_threshold=1.513

- **AAR** proposed by Aaronson (2023), the details of the parameters are as follows: prefix_length = 4, ξ = 15485863, p_value = 1e-4, sequence_length = 200

- **EXP** proposed by Kuditipudi et al. (2024), the details of the parameters are as follows: pseudo_length = 420, sequence_length = 200, n_runs = 100, key = 42, p_threshold = 0.2

- **ITS** proposed by Kuditipudi et al. (2024), the details of the parameters are as follows: pseudo_length = 256, sequence_length = 200, n_runs = 500, key = 42, p_threshold = 0.1

- **GumbelSoft** proposed by Fu et al. (2024a), the details of the parameters are as follows: prefix_length = 2, eps = 1e-20, threshold = 1e-4, sequence_length = 200, temperature = 0.7

- **SynthID** proposed by Dathathri et al. (2024), the details of the parameters are as follows: n = 5,

Watermark	C4 Dataset								OpenGen Dataset							
	OPT-6.7B				GPT-J-6B				OPT-6.7B				GPT-J-6B			
	TPR	TNR	F1	AUC	TPR	TNR	F1	AUC	TPR	TNR	F1	AUC	TPR	TNR	F1	AUC
KGW + AAR Watermark																
Series	1.000	0.995	0.998	0.999	1.000	1.000	1.000	1.000	1.000	0.995	0.998	0.999	1.000	1.000	1.000	1.000
Parallel	1.000	0.970	0.985	0.990	1.000	0.980	0.990	0.992	0.995	0.955	0.975	0.976	0.985	0.980	0.983	0.985
Hybrid	0.995	1.000	0.997	0.999	1.000	1.000	1.000	1.000	0.995	1.000	0.998	0.999	0.995	0.995	0.995	0.997
Unbiased + AAR Watermark																
Series	0.985	1.000	0.993	0.999	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.995	1.000	0.997	0.997
Parallel	0.835	1.000	0.918	0.914	0.890	1.000	0.942	0.954	0.885	0.990	0.934	0.957	0.945	1.000	0.972	0.974
Hybrid	0.970	1.000	0.985	0.994	0.920	1.000	0.956	0.973	0.995	1.000	0.997	0.998	0.965	1.000	0.982	0.992
KGW + GumbelSoft Watermark																
Series	0.985	1.000	0.992	0.993	0.970	1.000	0.985	0.988	1.000	1.000	1.000	0.996	0.975	1.000	0.987	0.996
Parallel	0.935	1.000	0.967	0.992	0.955	0.995	0.974	0.993	0.980	0.990	0.985	0.995	0.900	1.000	0.947	0.997
Hybrid	0.955	1.000	0.977	0.998	0.985	1.000	0.992	0.994	0.980	0.995	0.987	0.999	0.950	0.990	0.969	0.993
Unigram + GumbelSoft Watermark																
Series	0.995	1.000	0.997	0.995	0.995	0.980	0.988	0.999	0.975	0.995	0.985	0.999	0.995	0.995	0.995	0.996
Parallel	0.870	1.000	0.930	0.993	0.985	0.955	0.970	0.978	0.920	0.985	0.951	0.981	0.940	0.965	0.952	0.993
Hybrid	0.955	1.000	0.977	0.994	0.960	0.975	0.967	0.999	0.980	1.000	0.990	0.999	0.990	0.990	0.990	0.995

Table 4: Evaluating the detectability of different symbiotic watermarking algorithms on C4 and OpenGen.

sampling_size = 65536, seed = 0, mode = "non-distortionary", num_leaves = 2, context_size = 1024, detector_type = "mean", threshold = 0.52

E Watermark Selection

In our symbiotic framework SymMark, we adopt the Unigram method (Zhao et al., 2024) for logits-based watermarking, as it surpasses the KGW algorithm (Kirchenbauer et al., 2023) in robustness and maintains relatively high text quality compared to other logits-based watermarking methods, including Unbiased, DIP, and SWEET. For sampling-based watermarking, we select the AAR (Aaronson, 2023) algorithm to improve both robustness and security. This choice is motivated by the extremely low detection efficiency of the EXP and ITS (Kuditipudi et al., 2023) watermarks, as shown in Table 3, along with the relatively poor detectability of both GumbelSoft (Fu et al., 2024a) and SynthID (Dathathri et al., 2024). The parameter settings remain identical to the baselines.

We explored additional watermark combinations, with detection results summarized in Table 4. Theoretically, both the KGW family (Unigram, SWEET, etc.) and the ARR family (EXP, GumbelSoft, etc.) can be integrated into our framework. As shown in Figure 9, the corresponding PPL results of KGW and AAR further validate that our hybrid symbiotic watermarking strategy effectively balances detectability and text quality.

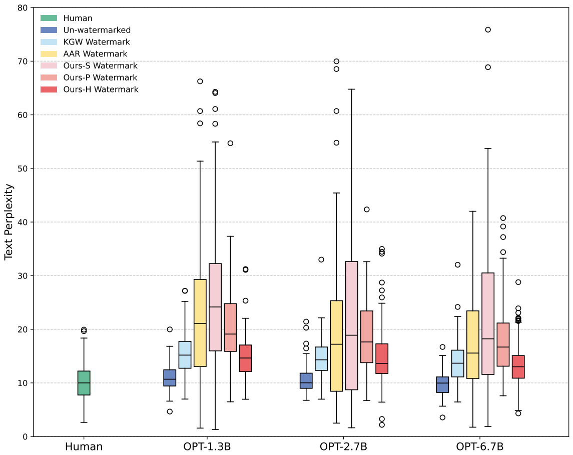

Figure 9: A comparison of PPL across three symbiotic watermarking schemes with different model sizes.

F Attack Settings

Besides the method presented in Figure 5, the AUROC curves for the attack robustness tests of the other baseline methods are illustrated in Figure 8. The specific parameter settings for various attack scenarios are as follows:

- **Word-D** Randomly delete 30% of the words in the watermark text.
- **Word-S-DICT** Replace 50% of the words with their synonyms based on the WordNet (Miller, 1995) dictionary.
- **Word-S-BERT** Replace 50% of the words

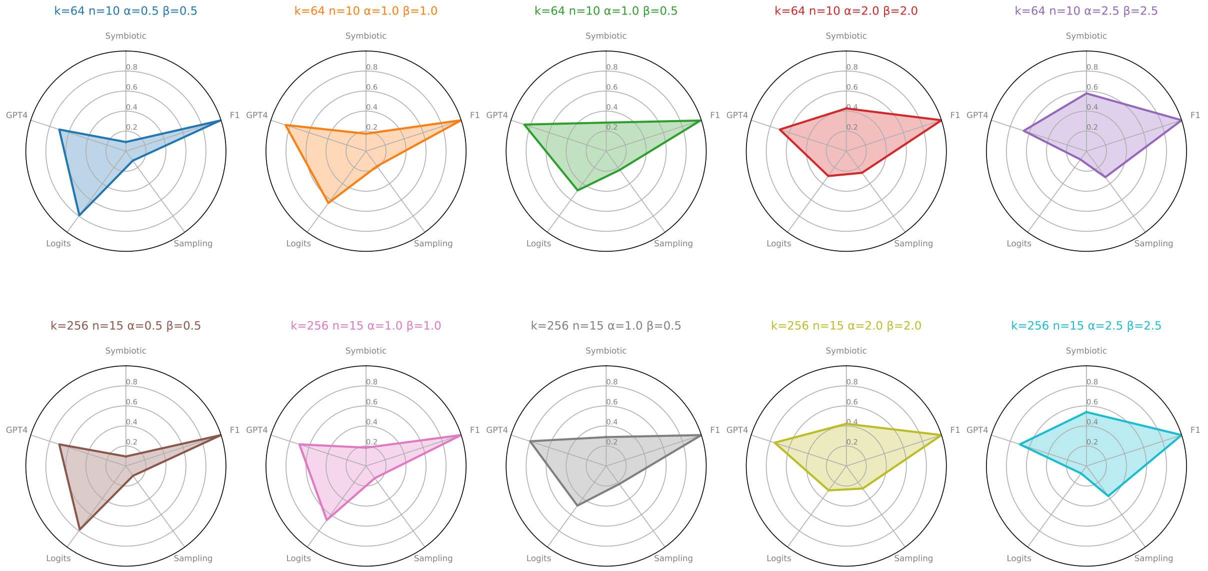

Figure 10: Hyperparameter Analysis of Top-k Selection, Number of Clusters n, TE threshold α and SE threshold β.

with contextually appropriate synonyms using BERT's (Devlin, 2018) embeddings.

- **Copy-Paste** Only 20% of the watermark text is retained, distributed across three locations in the document.
- **Translation** Translate the text from English to Chinese and then back to English using the fine-tuned T5 translation model [2].
- **Rephrase (GPT-3.5-turbo)** Call GPT-3.5-turbo API to paraphrase the text with low creativity (temperature = 0.2).
- **Rephrase (Dipper-1)** Use the DIPPER (Krishna et al., 2023) model for a restatement attack, focusing on lexical diversity without changing sentence structure. (lex_diversity = 60, order_diversity = 0, max_new_tokens = 200)
- **Rephrase (Dipper-2)** Use DIPPER again, with both lexical and order diversity, generating even more varied restatements. (lex_diversity=60, order_diversity=60, max_new_tokens=200)

G Hyperparameter Analysis

We randomly sampled 50 instances from the C4 dataset and embedded our hybrid symbiotic watermarks into the OPT-6.7B model. We analyzed the detection F1 scores and GPT-4's evaluations of text quality under varying token entropy and semantic entropy thresholds, with the results displayed in Figure 10. The prompt used for GPT-4 (OpenAI et al., 2023) to evaluate watermarked text quality in Figure 10 and Figure 11 is as follows:

GPT-4 Judge

> "You are given a prompt and a response, and you need to grade the response out of 100 based on: Accuracy (20 points) - correctness and relevance to the prompt; Detail (20 points) - comprehensiveness and depth; Grammar and Typing (30 points) - grammatical and typographical accuracy; Vocabulary (30 points) - appropriateness and richness. Deduct points for shortcomings in each category. Note that you only need to give an overall score, no explanation is required."

The impact of top-k and cluster number n. As shown in Figure 10, under different top-k and n settings, the variations in F1 and GPT-4 scores closely follow the changes in the entropy threshold. This indicates that top-k and the number of clusters have minimal impact on semantic entropy calculation. Therefore, for clustering efficiency, we set top-k to 64 and n to 10.

The impact of entropy thresholds α and β. In Figure 10, "Symbiotic" represents the ratio of embedding logits to sampling watermarked tokens, "Logits" denotes the ratio of embedding logits watermark tokens, and "Sampling" refers to the ratio of embedding sampling watermark tokens. When the token and semantic entropy thresholds are low,

[2] https://huggingface.co/utrobinmv/

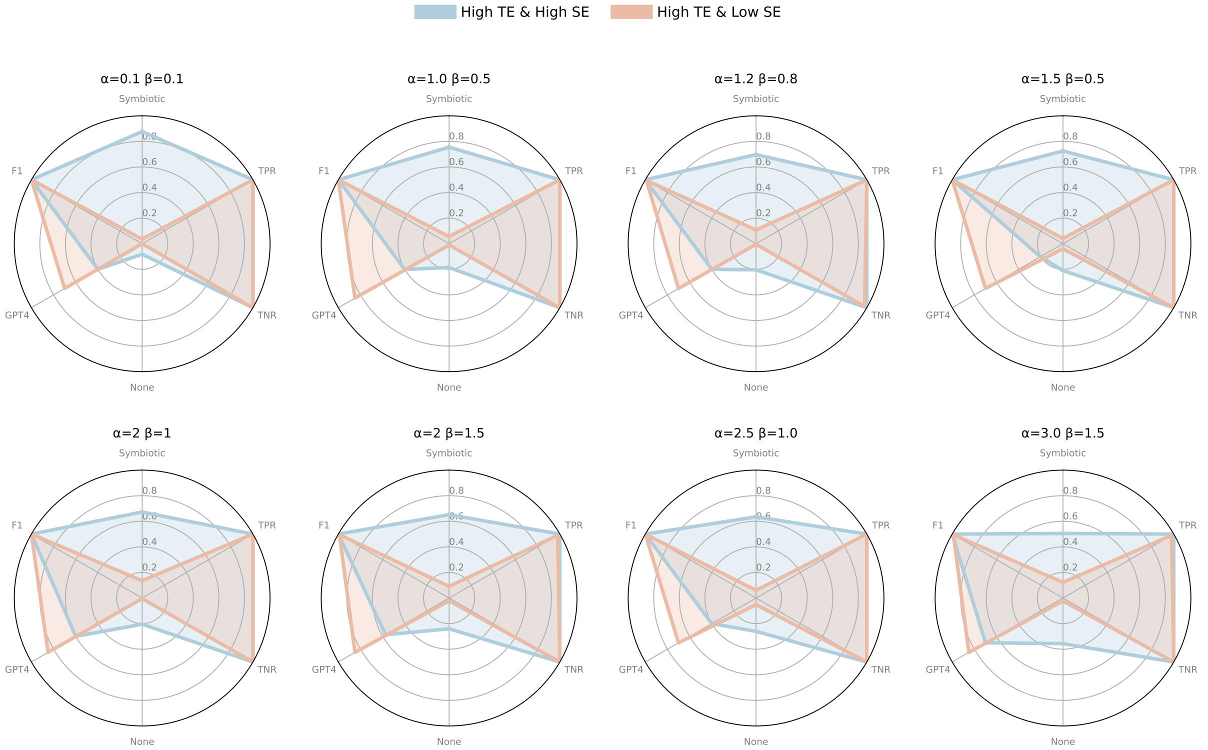

Figure 11: Comparison of two watermarking schemes: high versus low token and semantic entropy. "Symbiotic" refers to embedding logits and sampling watermarked tokens, while "None" refers to unwatermarked tokens.

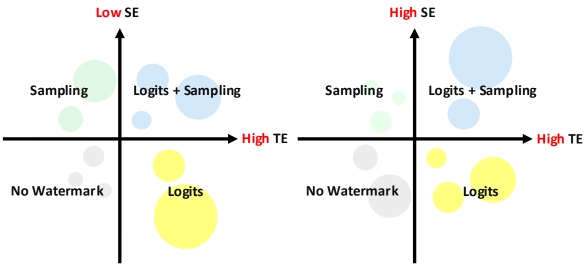

Figure 12: Scheme 1 (Left), Scheme 2 (Right)

the proportion of symbiotic watermarks remains low. As these thresholds increase, the proportion of symbiotic watermarks correspondingly rises. The two extreme cases of hybrid watermarks, corresponding to series and parallel configurations, constrain the impact of entropy thresholds on the detectability F1 score. However, an increased proportion of symbiotic watermarks more significantly affects text quality. Based on our experiments on the demo dataset, we set the token entropy threshold (α) to 1.0 and semantic entropy threshold (β) to 0.5 to achieve an optimal trade-off between detectability and text quality.

H The impact of Semantic Entropy

We compared two entropy combination schemes:

- **Scheme 1** (we adopted): Embeds symbiotic watermarks at high token entropy and low semantic entropy.
- **Scheme 2**: Embeds symbiotic watermarks at high token entropy and high semantic entropy.

The experimental results for various token and semantic entropy thresholds are shown in Figure 11 and 12. While both schemes demonstrate good detectability, Scheme 1 (GPT-4) significantly outperforms Scheme 2 in text quality assessment. This suggests that embedding watermarks on tokens with low semantic entropy has a lesser impact on text quality than embedding them on tokens with high semantic entropy. Even when watermarks are applied to tokens with low semantic entropy, the semantic integrity of the sampled tokens remains largely unchanged.

Furthermore, our experiments show that when token entropy is low, semantic entropy is also low, while when token entropy is high, semantic entropy can vary between high and low. Consequently, in many samples, numerous tokens are not embedded with the watermark in Scheme 2, negatively affecting watermark detection performance. In contrast, Scheme 1 successfully embeds sufficient watermark signals in nearly all cases, while preserving

Algorithm 3: Group Watermarked Token

```
Input: $\mathcal{M}, y_{1:T}, \alpha, \beta$, FLAG
Output: $Y_l, Y_s$
// Serial Watermark Group
if FLAG = "S" then
    $Y_l \leftarrow y_{1:T}$
    $Y_s \leftarrow y_{1:T}$
end
// Parallel Watermark Group
else if FLAG = "P" then
    if $i \bmod 2 == 0$ then
        $Y_l$.append$(y_i)$
    end
    else if $i \bmod 2 == 1$ then
        $Y_s$.append$(y_i)$
    end
end
// Hybrid Watermark Group
else if FLAG = "H" then
    for $i = 1, ..., T$ do
        $H_{TE}, H_{SE} \leftarrow$ ComputeEntropy$(y_{1:i})$
        // High Token Entropy
        if $H_{TE} > \alpha$ then
            $Y_l$.append$(y_i)$
        end
        // Low Semantic Entropy
        if $H_{SE} < \beta$ then
            $Y_s$.append$(y_i)$
        end
    end
end
```

the text quality. Therefore, we choose to embed two watermark signals when token entropy is high and semantic entropy is low.

I Group-based Detection

We also explored a group-based detection algorithm (Algorithm 3). Tokens are first grouped into logits-based and sampling-based categories. For serial watermarks, all tokens are grouped since each contains two watermarks. For parallel watermarks, tokens are divided by odd and even positions. For hybrid watermarks, tokens are grouped based on entropy values calculated from token and semantic entropy. Detection then uses the methods in Algorithm 4. However, this approach has several drawbacks: (1) a more complex process; (2) lower efficiency, especially for hybrid watermarks due to entropy calculation; (3) poor robustness, as the position of parallel watermarks may vary.

Therefore, this paper employs Algorithm 2 for detection, as it directly identifies watermark signals in all tokens of the generated text. This method has demonstrated outstanding practical performance, is easy to implement, and ensures high watermark detection efficiency, as shown in Table 3.

Algorithm 4: Group-based Detection

```
Input: $\mathcal{M}, Y_l, Y_s, \mathcal{D}_l, \mathcal{D}_s, z_1, z_2$
Output: $I$: True (Watermarked) or False
$I_l \leftarrow$ False
$I_s \leftarrow$ False
// Logits Watermark Detection
if $\mathcal{D}_l(\mathcal{M}, Y_l) > z_1$ then
    $I_l \leftarrow$ True
end
// Sampling Watermark Detection
if $\mathcal{D}_s(\mathcal{M}, Y_s) > z_2$ then
    $I_s \leftarrow$ True
end
// Combine Detection Results
$I \leftarrow I_l \mid I_s$
```

J Watermark Stealing Settings

Since mainstream watermark attack methods (Jovanović et al., 2024; Zhang et al., 2024; Sadasivan et al., 2024; Gu et al., 2024; Luo et al., 2024; Pang et al., 2024) primarily target the red-green word list approach rather than the sampling method, we follow Jovanović et al. (2024) to conduct a watermark-stealing attack, assuming the attacker has access to the distribution of unwatermarked tokens. In this attack, we query the watermarked LLM to generate a total of 200k tokens, estimate the watermark pattern, and subsequently launch spoofing attacks based on the estimated pattern.

Specifically, we use watermarked text generated from the C4 dataset to learn the watermark, then execute a watermark spoofing attack on Dolly-CW datasets (Conover et al., 2023) containing 100 samples. To ensure experimental fairness, the logits-based watermark in our hybrid symbiotic watermark employs the Unigram algorithm with identical hash keys and parameters $\gamma = 0.25$, $\delta = 0.4$. For the sampling-based watermark, we utilize the AAR (Aaronson, 2023) algorithm. We use LLaMA2-7B-chat-hf as both the watermark and attack model, with the watermark spoofing strength set to 5.0. All other parameter settings remain consistent with those in our main experiment.

During the watermark detection stage, we set the spoofing watermark z-score threshold to 6 and apply the original KGW watermark detection algorithm to analyze n spoofing samples. If the computed z-score exceeds 6, the attack is deemed successful; otherwise, it is considered unsuccessful. Consequently, the attack success rate (ASR) is determined as follows:

$$\text{ASR} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}[\text{z-score}_i > 6] \tag{6}$$