SAMARK: A SELF-ANCHORED WATERMARKING WITH PARAGRAPH-LEVEL PARAPHRASE ROBUSTNESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic-Level Watermarking (SWM) improves robustness against text modifications by treating sentences as the basic unit. However, *robustness to paragraph-level paraphrasing remains difficult because such attacks globally disrupt watermark signals by changing sentence order*. Therefore, we propose **SAMARK, a self-anchored watermarking framework that removes the dependency on sentence order by establishing a step-independent green region in semantic space**. To improve detectability, we introduce a multi-channel hyperbolic scoring mechanism that amplifies watermark signals while suppressing noise from weakly aligned candidates. We further propose a diversity-aware filtering strategy that combines hard filtering with soft regularization, extending beyond simple n-gram repetition filters to address semantic redundancy. Experiment show SAMARK achieves up to 90.2% TP@FP1% under typical paragraph-level paraphrasing attacks, outperforming the strongest baseline by more than 30% averagely, while maintaining generation quality competitive with unwatermarked text and breaking the robustness-quality trade-off that limits prior methods. See code at URL.

1 INTRODUCTION

The rapid advancement of generative AI (GenAI) has transformed content creation across many domains, from education to software development. Yet the growing ability of these models to produce realistic and persuasive content has also raised concerns about authenticity, attribution, and misuse. LLM watermarking, which embeds imperceptible yet algorithmically detectable signals into model outputs to identify AI-generated text, has emerged as a promising way to mitigate such risks.

During generation, watermarking subtly modifies model outputs while preserving meaning and fluency, embedding a hidden, consistent signal that serves as a verifiable marker of GenAI origin. The watermark can later be detected from the generated output, enabling verification without relying on inherent statistical artifacts of the language model and offering a more resilient way to distinguish machine-generated content from human-authored work.

Token-level Watermarking. Token-level watermarking (TWM) schemes for LLMs have been widely studied. Green-Red schemes (Kirchenbauer et al., 2023a; Zhao et al., 2023) assign tokens to keyed green and red subsets and bias sampling toward green tokens; the detector then tests whether the generated text contains an unusually high fraction of green tokens. This mechanism provides a simple statistical signal, but it changes the original sampling distribution of the LLM and can degrade text quality (Hu et al., 2023). Distortion-free methods, including Gumbel Watermarking (Aaronson, 2023) and PRC-based pseudorandom error-correcting schemes (Christ & Gunn, 2024), reduce this distributional distortion and provide theoretical guarantees on text quality. However, their guarantees mainly address generation quality rather than edit robustness. Local modifications such as word deletion, substitution, and paraphrasing can still remove or dilute token-level evidence, which limits the reliability of TWM in adversarial or post-edited settings (Hou et al., 2023).

Semantic-level Watermarking. To improve robustness to such attacks, semantic-level watermarking (SWM) approaches, such as SemStamp (Hou et al., 2023; 2024; Dabiriaghdam & Wang, 2025), treat a semantically complete sentence as the basic watermarking unit. These methods use rejection sampling to ensure that generated sentences fall within a valid semantic region of the embedding space, analogous to the green list in Green-Red watermarking. Recently, PMark (Huo et al., 2026) increased watermark density in SWM by applying multi-channel constraints during sampling, further improving robustness against sentence-level paraphrasing attacks. Despite these efforts, current SWM methods still rely on context hashing or fixed private keys to partition green and red regions during sampling (Hou et al., 2023; 2024; Huo et al., 2026). This reliance makes them vulnerable to paragraph-level paraphrasing attacks (PPA), which can globally disrupt watermark signals

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

by changing sentence order, as illustrated in Figure 1. Achieving robustness to PPA has long been regarded as an intractable problem in the community (Huo et al., 2026).

Is a PPA-robust SWM method possible? Given the status quo, we ask the following question:

Can we develop a high-quality semantic-level watermarking scheme that is inherently robust to paragraph-level paraphrasing attacks?

An affirmative answer would reshape the practical landscape of SWM. Specifically, our goal is to realize an SWM method that is robust and quality-preserving, thereby facilitating practical usage.

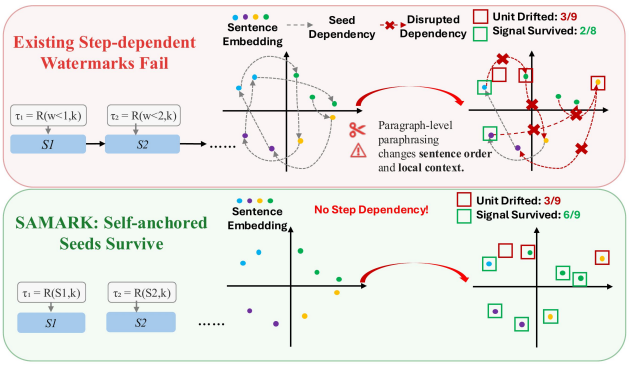


Figure 1: Comparison between step-dependent watermarks and our proposed SAMARK.

Self-anchored Paradigm. In this work, we propose SAMARK, a novel self-anchored semantic watermarking framework designed to withstand PPAs. We identify that the vulnerability of existing watermarking schemes and their variants (Kirchenbauer et al., 2023a; Dathathri et al., 2024; Hou et al., 2023; Huo et al., 2026) largely stems from their reliance on step-dependent random seeds. Motivated by the observation that a unit's intrinsic semantic content remains relatively invariant under paraphrasing, SAMARK derives its random seed solely from the semantics of the current unit itself and establishes a step-independent green region in semantic space. This self-anchored design enables the watermark to survive both textual perturbations and sentence reordering.

Multi-channel Hyperbolic Scoring and Quality Control. We further observe that concentrating watermark scores near semantic-region boundaries increases the likelihood of sign flips under paraphrasing, as illustrated in Figure 2. To address this issue, we introduce a multi-channel hyperbolic scoring mechanism that amplifies robust watermark evidence while suppressing noise from weakly aligned candidates. Finally, we employ diversity-aware quality control that combines hard semantic filtering with soft regularization to reduce redundancy and encourage lexical novelty. Empirical evaluations demonstrate that SAMARK maintains strong detectability under PPAs, where existing baselines largely fail. Meanwhile, it preserves generation quality competitive with that of unwatermarked text, substantially narrowing the conventional trade-off between watermark robustness and text quality. **Key contributions of this paper include:**

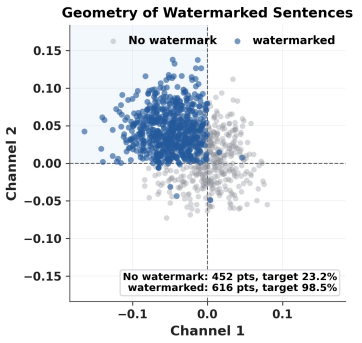


Figure 2: Illustration of watermarked semantic distribution.

❶ We identify step-dependent random seeds as a major source of vulnerability to PPAs and show that intrinsic unit semantics provide a more reliable invariant for robust watermarking.

❷ We reveal that watermark-score concentration near semantic-region boundaries undermines post-attack detectability by increasing the risk of sign flips under paraphrasing.

❸ We propose SAMARK, a self-anchored watermarking framework that derives seeds solely from each unit's semantics and establishes step-independent green regions, providing robustness to both textual perturbations and sentence reordering.

❹ We introduce multi-channel hyperbolic scoring and diversity-aware quality control to strengthen watermark evidence while preserving generation quality, and validate their effectiveness through comprehensive experiments.

2 Problem Definition and Preliminaries

To motivate our approach, we first formalize the general framework of existing watermarking methods and then analyze why they fail under paragraph-level paraphrasing attacks (PPA). This analysis reveals that step-dependent random seeds are the root cause of vulnerability, guiding us toward a self-anchored solution.

2.1 Problem Setup and General Framework

Problem Setup. Let $\mathcal{V}$ denote the vocabulary of an LLM π. The text generation process is denoted by $x \xrightarrow{\pi} y$, where a response y is sampled from the model given an input prompt x. We define $\mathcal{V}^*$ as the set of all semantically complete sentences, including the null sentence of length zero. Let $u \in \Omega$ denote the basic unit of watermark generation, where Ω represents the set of all possible units, such as a token $v \in \mathcal{V}$ in a token-level watermarking scheme or a sentence $s \in \mathcal{V}^*$ in a semantic-level one. At each generation step t, the probability distribution of producing the next unit $u_t \in \Omega$ given the preceding context $u_{<t} = (u_1, \ldots, u_{t-1})$ is denoted by $p_\pi(\cdot \mid u_{<t})$.

General Framework. Existing generative watermarking schemes typically consist of three key components: a random seed generator $\mathfrak{R}$, a sampling algorithm $\mathfrak{G}$, and a scoring function $\mathfrak{S}$. We introduce a private key k drawn from a key space $\mathcal{K}$. The watermarked output follows a conditional distribution $p_\pi^w(u_t \mid u_{<t}; k)$. The unified generation and detection process can be formalized as:

- **Fetching Random Seed ($\mathfrak{R}$).** At step t, a random seed r_t is generated. For context-hashing-based watermarks, $r_t = \mathfrak{R}(u_{t-w}, \ldots, u_{t-1}; k)$, where w denotes the context window size. For fixed key, the seed is step-dependent but context-independent: $r_t = \mathfrak{R}(t; k)$.
- **Output Sampling ($\mathfrak{G}$).** The seed r_t defines a green region partition $G_t = \mathcal{P}(\Omega; r_t)$, and the output u_t is preferentially sampled from G_t via $u_t = \mathfrak{G}(\pi, u_{<t}, G_t) \sim p_\pi^w(\cdot \mid u_{<t}; k)$.
- **Detection ($\mathfrak{S}$).** The detector quantifies the correlation between generated units and reconstructed seeds. The overall detection score is $\mathfrak{S}(X) = \sum_{t=1}^{T} \mathfrak{S}(u_t, r_t)$.

Robustness is typically pursued by maximizing $\mathfrak{S}(X)$ during generation while maintaining text quality through a carefully designed sampling algorithm $\mathfrak{G}$.

2.2 Brief Introduction to Existing Methods and Their Limitations

Token- and Semantic-level Green-Red Paradigm. The Green-Red paradigm was initially introduced for token-level watermarking (Kirchenbauer et al., 2023a). At generation step t, a seed $r_t = \mathfrak{R}(u_{t-w}, \ldots, u_{t-1}; k)$ is used to partition the vocabulary $\mathcal{V}$ into a green list G_t and a red list $R_t = \mathcal{V} \setminus G_t$, with $|G_t| = \gamma|\mathcal{V}|$. Given the model logits $\ell_v^{(t)}$ for each token $v \in \mathcal{V}$, TWM increases the probability of green tokens by adding a positive logit bias δ:

$$p_\pi^w(v \mid u_{<t}; k) = \begin{cases} \dfrac{\exp(\ell_v^{(t)} + \delta)}{\Lambda}, & v \in G_t \\ \dfrac{\exp(\ell_v^{(t)})}{\Lambda}, & v \in R_t \end{cases}, \text{where } \Lambda = \sum_{v' \in G_t} \exp(\ell_{v'}^{(t)} + \delta) + \sum_{v' \in R_t} \exp(\ell_{v'}^{(t)}).$$

Semantic-level watermarking follows the same partitioning principle, but the partition is defined over the unit space $\Omega = \mathcal{V}^*$ rather than the token vocabulary. Given a context-dependent seed $r_t = \mathfrak{R}(u_{t-w}, \ldots, u_{t-1}; k)$, the semantic space is partitioned into $G_t = \mathcal{P}(\Omega; r_t)$ and $R_t = \Omega \setminus G_t$, often through locality-sensitive hashing over sentence embeddings. SWM then performs rejection sampling: a candidate sentence u is accepted only if $u \in G_t$. Equivalently, the watermarked distribution is the model distribution truncated to the green region:

$$p_\pi^w(u \mid u_{<t}; k) = \begin{cases} \dfrac{p_\pi(u \mid u_{<t})}{Z_t}, & u \in G_t \\ 0, & u \in R_t \end{cases}, \text{where } Z_t = \sum_{u' \in G_t} p_\pi(u' \mid u_{<t}).$$

We identify two fundamental vulnerabilities in existing watermarking schemes that stem from their reliance on step-dependent random seeds.

Context-Dependent Seeds are Fragile. Most existing TWMs (Kirchenbauer et al., 2023a; Dathathri et al., 2024) and SWMs (Hou et al., 2023; 2024) rely on context-hashing mechanisms, where the random seed depends strictly on the preceding context: $r_t = \mathfrak{R}(u_{t-h}, \ldots, u_{t-1}; k)$. Paraphrasing attacks involving synonym substitutions and unit modifications perturb the context sequence, thereby altering the reconstructed seeds. The detection score changes to $\mathfrak{S}(X') = \sum_t \mathfrak{S}(u'_t, \mathfrak{R}(u'_{t-h}, \ldots, u'_{t-1}; k))$, shifting from the watermarked distribution $\mathcal{D}_w$ back toward the unwatermarked distribution $\mathcal{D}_n$ and causing detection failure.

Step-Dependent Seeds Cannot Survive Reordering. Some watermarking schemes (Huo et al., 2026) use step-dependent fixed keys $r_t = \mathfrak{R}(t; k)$ to avoid context dependency. However, they remain vulnerable to unit reordering. When paraphrasing attacks alter the sequence order, the detection score becomes a mismatched sum $\mathfrak{S}(X') = \sum_t \mathfrak{S}(u'_t, r_t)$, leading to a drop of detectable signals.

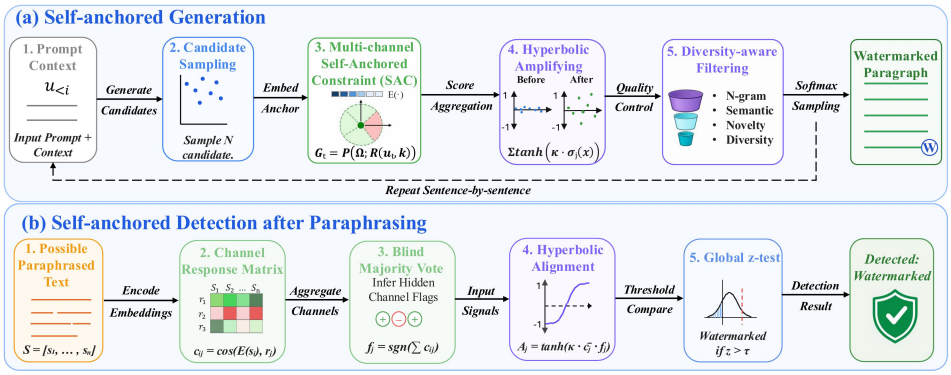


Figure 3: Overall framework of SAMARK.

Semantics as the Only Invariant. From the above analysis, we conclude that *the most reliable invariant property of watermarked text under PPA is the intrinsic semantic content of each unit*, denoted by $\phi(u_t)$. This semantic representation is preserved even when the surface form or step position changes. This insight motivates our self-anchored paradigm: by making the random seed depend solely on the unit's own semantics, i.e., $r_t = \mathfrak{R}(\phi(u_t); k)$, we can achieve robustness against both perturbation and reordering.

2.3 THE ROBUSTNESS-QUALITY TRADE-OFF

Pursuing Robustness Degrades Quality. While existing methods strive to enhance robustness, they often introduce quality degradation (Giboulot & Furon, 2024). Watermark generators tend to repeat previous n-grams and sentences because of the strong constraints imposed by limited green sets. Although n-gram repetition filters (Pan et al., 2024) have been used to avoid looped generation, their effectiveness is limited when detecting semantic-level overlap. This trade-off constrains the practical potential of robust watermarking schemes.

Smarter Filtering for Better Robustness-Quality Trade-off. We observe that simple surface-level filters fail to capture semantic redundancy. To address this, we propose a diversity-aware filtering mechanism that operates at both lexical and semantic levels, enabling the selection of candidates that are simultaneously high-scoring (robust) and diverse (high-quality). This pushes the frontier of the robustness-quality trade-off.

3 SAMARK: SELF-ANCHORED WATERMARK

Building on the insights from Section 2, we propose SAMARK, a self-anchored watermarking framework that achieves robustness against paragraph-level paraphrasing by making each sentence's watermark signal depend solely on its own semantics. Our method combines three key components: (1) a theoretically grounded self-anchored constraint that establishes a step-independent green region, (2) a multi-channel hyperbolic scoring mechanism that amplifies watermark signals for improved detectability, and (3) a diversity-aware filtering strategy that maintains text quality without sacrificing robustness. Figure 3 illustrates the overall framework of our proposed method. A complete pseudocode of our method can also be found in Appendix D. To provide a deeper understanding of SAMARK, we also conduct extensive mechanism analyses, which can be found in Appendix D.1.

3.1 RETHINKING SAMPLING STRATEGY WITH SELF-ANCHORED CONSTRAINTS

In this section, we formulate the constraints for a self-anchored watermark by replacing the step-dependent components in the general framework with step-independent counterparts. We define the Self-Anchored Condition (SAC) as:

$$\text{SAC:} \quad \begin{cases} r_t = \mathfrak{R}(u_t; k) \\ G_t = \mathcal{P}(\Omega; r_t) \\ u_t = \mathfrak{G}(\pi, u_{<t}, G_t) \sim p_\pi^w(\cdot \mid u_{<t}, k) \in G_t \end{cases} \tag{1}$$

Theorem 1 (Step-independent Feasible Region)**.** *There exists a step-independent feasible region $\mathcal{G} \subseteq \Omega$ such that any unit $u \in \mathcal{G}$ strictly satisfies Eq. 1 at any generation step t, while any $v \notin \mathcal{G}$ consistently violates it.*

Proof. By the definition of the Self-Anchored Condition in Eq. 1, the random seed $r_t = \mathfrak{R}(u_t; k)$ and the corresponding partition $G_t = \mathcal{P}(\Omega; r_t)$ depend exclusively on the current unit u_t and the fixed key k. They are strictly decoupled from the generation step t and the preceding context $u_{<t}$. Therefore, we can explicitly construct the step-independent feasible region $\mathcal{G}$ as:

$$\mathcal{G} = \{u \in \Omega \mid u \in \mathcal{P}(\Omega; \mathfrak{R}(u; k))\}.$$

For any unit $u \in \mathcal{G}$, assigning $u_t = u$ at an arbitrary step t deterministically yields the seed $r_t = \mathfrak{R}(u; k)$ and the partition $G_t = \mathcal{P}(\Omega; r_t)$. By the definition of the set $\mathcal{G}$, it holds that $u_t \in G_t$, which strictly satisfies the constraint in Eq. 1. Conversely, for any unit $v \notin \mathcal{G}$, it inherently follows that $v \notin \mathcal{P}(\Omega; \mathfrak{R}(v; k))$. Consequently, assigning $u_t = v$ invariably results in $u_t \notin G_t$, which consistently violates Eq. 1 regardless of the step t. This completes the proof. □

Theorem 1 demonstrates that the self-anchored conditions in Eq. 1 are mathematically equivalent to establishing a globally static green region within the entire space Ω per response.

3.2 SAMARK GENERATOR: MULTI-CHANNEL HYPERBOLIC SAMPLING

We adopt the multi-channel sampling paradigm from PMark (Huo et al., 2026) and introduce a hyperbolic-enhanced scoring mechanism for candidate sampling. Given c pivot vectors $V = \{v_1, \ldots, v_c\} \in \mathbb{R}^d$ living in the embedding space of text encoder $\mathcal{E}$ and a target flag pattern $R = \{r_1, \ldots, r_c\} \in \{-1, 1\}^c$, the step-independent green region is defined as $\{u \in \Omega \mid \text{sgn}(\langle \mathcal{E}(u), v_j \rangle) = r_j, \forall j \in \{1, \ldots, c\}\}$, with $\langle \cdot, \cdot \rangle$ denoting the inner product and $\text{sgn}(\cdot)$ denoting the signum function. To add feasibility, the target flag pattern R is chosen uniformly at random for each query on the generation side and is not shared with the detection side, reducing storage overhead. The generation process proceeds as follows:

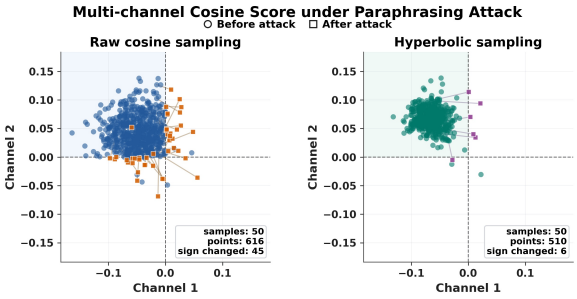


Figure 4: Multi-channel semantic deviation under paraphrasing attacks under different settings.

- **Candidate Generation:** Given context $u_{<t}$ and model π, sample N candidate sentences from the natural distribution to form the pool $W = \{x_1, \ldots, x_N\}$, where $x_i \sim p_\pi(\cdot \mid u_{<t})$.
- **Embedding and Channel Matching:** Extract semantic embeddings $\mathcal{E}(x)$ for all $x \in W$. Filter W to retain only candidates whose similarity signs strictly match the target pattern R across all c channels, yielding the matched subset:

$$V_{\text{match}} = \{x \mid \text{sgn}(\langle \mathcal{E}(x), v_j \rangle) = r_j, \forall j \in \{1, \ldots, c\}\}. \tag{2}$$

- **Hyperbolic Scoring:** For each $x \in V_{\text{match}}$, compute the signed cosine similarity per channel $\sigma_j(x) = r_j \cdot \cos(\mathcal{E}(x), v_j)$. We apply a hyperbolic tangent transform to amplify candidates with strong channel alignment while suppressing near-zero noise:

$$S_{\text{wm}}(x) = \sum_{j=1}^{c} \tanh(\kappa \cdot \sigma_j(x)), \tag{3}$$

  where $\kappa > 0$ is a sharpness parameter controlling the amplification strength (default $\kappa = 30$). The $\tanh$ function saturates at ± 1, ensuring that candidates with large $|\sigma_j(x)|$ contribute maximally while those with weak alignment are suppressed. The final score incorporates diversity regularization (detailed in Section 3.4):

$$S(x) = S_{\text{wm}}(x) + \lambda_{\text{div}} \cdot D(x) + \lambda_{\text{nov}} \cdot \mathcal{N}(x), \tag{4}$$

  where $D(x)$ and $\mathcal{N}(x)$ are the semantic diversity and vocabulary novelty terms.
- **Softmax Selection:** Rather than deterministic top-k selection (Giboulot & Furon, 2024), a softmax distribution over the matched candidates is introduced for controlled stochasticity:

$$q(x) = \frac{\exp(\epsilon \cdot S(x))}{\sum_{x' \in V_{\text{match}}} \exp(\epsilon \cdot S(x'))}, \tag{5}$$

  where $\epsilon > 0$ is the temperature. The watermarked sentence u_t is then sampled as $u_t \sim q(\cdot)$.

In Figure 4, we display the influence of hyperbolic scoring under the Doc-P I attack. Compared with naive cosine scoring in previous SWM methods, hyperbolic scoring prevents the concentration on the boundary of the green regions.

3.3 SAMARK DETECTOR: MAJORITY-VOTE BLIND DETECTION

We introduce a blind, sentence-level detection scheme that extracts watermark evidence across multiple channels without prior knowledge of the target flags R. The detection process mirrors the generation-side hyperbolic transform for consistency.

- **Sentence Segmentation:** Segment the input text into a sequence of sentences $S = [s_1, \ldots, s_n]$.
- **Embedding and Channel Response:** Extract embeddings $\mathcal{E}(s_i)$ and deterministically reconstruct the pivot vectors $V = \{v_1, \ldots, v_c\}$ using the shared key. Compute the channel response matrix $C \in \mathbb{R}^{n \times c}$: $\boldsymbol{C_{i,j} = \cos(\mathcal{E}(s_i), v_j)}$.
- **Blind Flag Inference:** Since the detector has no access to the true flags R, we infer them via majority voting over the directional responses. For each channel j, aggregate the responses across all sentences to estimate the flag as $\hat{\boldsymbol{r}}_{\boldsymbol{j}} = \text{sgn}\left(\sum_{i=1}^{n} \boldsymbol{C_{i,j}}\right)$. Under the watermarking hypothesis, the majority of sentences will have been selected to align with the true flag pattern, making this inference reliable.
- **Hyperbolic Score Alignment:** Align the raw channel responses using the inferred flags and apply the same hyperbolic transform used during generation: $\boldsymbol{A_{i,j} = \tanh(\kappa \cdot C_{i,j} \cdot \hat{r}_j)}$, where $A_{i,j}$ measures the evidence strength of the s_i in the j-th channel. Similarly, this transform amplifies strong watermark signals while suppressing noise from weakly aligned sentences.
- **Robust Detection via z-Test:** Treat the transformed alignment matrix A as a flattened distribution of $n \cdot c$ samples. The global z-statistic is computed as:

$$z = \frac{\bar{A}}{\hat{\sigma}_A / \sqrt{nc}}, \text{ where } \quad \bar{A} = \frac{1}{nc} \sum_{i,j} A_{i,j}, \quad \hat{\sigma}_A = \sqrt{\frac{1}{nc} \sum_{i,j} (A_{i,j} - \bar{A})^2}.$$

  The text is classified as watermarked if z exceeds a predefined confidence threshold τ.

3.4 DIVERSITY-AWARE GENERATION

While the watermark scoring mechanism in Eq. 3 ensures strong detectability, greedily selecting high-scoring candidates may lead to repetitive or semantically monotonous text (Giboulot & Furon, 2024; Pan et al., 2024). To maintain generation quality, we introduce a two-stage diversity control mechanism: a hard filtering stage that removes overly similar candidates, followed by soft regularization that encourage diverse selections. Such additional optimization ensures the diversity and quality of generated text, avoiding quality degradation resulting from the strong sampling constraints.

3.4.1 HARD FILTERING

Before scoring, we apply two sequential filters to the candidate pool W:

- **N-gram Overlap Filter:** For each candidate x, we compute the fraction of its n-grams (default $n = 4$) that already appear in the generated context $u_{<t}$. Let $\mathcal{G}_n(x)$ denote the set of n-grams in x and $\mathcal{G}_n(u_{<t})$ denote those in the context. Candidates with n-gram overlap ratio $\boldsymbol{\rho_{\text{ngram}}(x) = \frac{|\mathcal{G}_n(x) \cap \mathcal{G}_n(u_{<t})|}{|\mathcal{G}_n(x)|} \geq \theta_{\text{ngram}} \in (0, 1]}$ are filtered out, where θ_{ngram} is the overlap threshold.
- **Semantic Similarity Filter:** Let $\mathcal{H} = \{u_1, \ldots, u_{t-1}\}$ denote the previously generated sentences. To mitigate semantic repeating, we filter out a candidate x if $\boldsymbol{\rho_{\text{sem}}(x) = \max_{u \in \mathcal{H}} \cos(\mathcal{E}(x), \mathcal{E}(u)) \geq \theta_{\text{sem}}}$, where $\theta_{\text{sem}} \in (0, 1]$ is the similarity threshold.

3.4.2 SOFT REGULARIZATION

For unfiltered candidates, we augment the watermark score with two diversity-promoting terms:

- **Semantic Diversity Bonus:** We reward candidates that are semantically dissimilar to previously selected sentences $\boldsymbol{D(x) = 1 - \rho_{\text{sem}}(x) = 1 - \max_{u \in \mathcal{H}} \cos(\mathcal{E}(x), \mathcal{E}(u))}$. This term ranges in $[0, 1]$, with higher values indicating greater semantic novelty.
- **Vocabulary Novelty Bonus:** We encourage candidates that introduce new vocabulary. Let $\mathcal{V}(x)$ denote the set of unique words in x and $\mathcal{V}_{<t}$ denote the accumulated vocabulary from the context. The novelty score is $\boldsymbol{\mathcal{N}(x) = \frac{|\mathcal{V}(x) \setminus \mathcal{V}_{<t}|}{|\mathcal{V}(x)|}}$, representing the fraction of words in x that have not appeared in the generated text so far.

The final selection score in Eq. 4 balances watermark strength with diversity through the weights λ_{div} and λ_{nov}. Setting these weights to zero recovers the pure watermark-driven selection, while positive values encourage more varied and natural-sounding text. An illustration of the SAMARK sampling process is provided in Appendix D.2.

4 EXPERIMENTAL RESULTS

In this section, we conduct extensive experiments to answer the following research questions: (**RQ1**) How does SAMARK perform under various PPAs? (**RQ2**) Can SAMARK preserve generation quality in practice? (**RQ3**) What is the computational cost of SAMARK compared with other SWM baselines? (**RQ4**) How sensitive is SAMARK to its key components or parameters?

4.1 EXPERIMENT SETUP

Dataset and Baselines. Following previous work (Huo et al., 2026), we evaluate SAMARK on 500 samples from C4 (Raffel et al., 2020) and BOOKSUM (Kryściński et al., 2021), using Nemotron-Nano-9B-v2-Base () and Mistral-Small-3.1-24B-Base (Mistral AI, 2025) as backbone models. Our baselines follow the official MarkLLM implementation (Pan et al., 2024), including token-level methods KGW (Kirchenbauer et al., 2023a), MorphMark (Wang et al., 2025), EXP (Aaronson, 2023), and SynthID (Dathathri et al., 2024), as well as semantic-level baselines SemStamp (Hou et al., 2023), k-SemStamp (Hou et al., 2024), and PMark (Huo et al., 2026). For SAMARK, we set the number of channels to 2 and the sample budget to $N = 64$. We report two variants: **SAMARK**, which uses the full diversity-aware sampling and soft-regularization pipeline, and **SAMARK** †, which disables these optimization terms. More implementation details are provided in Appendices E.1 and E.2.

Metrics. We assess watermark effectiveness in terms of detectability through **TP@FP1%** and robustness under paragraph-level attacks, such as Paraphrase Attack **Doc-P** and Back-Translation Attack **Doc-T** by GPT-4.1-mini (OpenAI, 2022). We use two paraphrasing prompts: *Doc-P I*, "Please rewrite the following text:" and *Doc-P II*, "Please rewrite the following text, avoiding using the same words or phrases in the original text:". We regard *Doc-P II* as the most representative and challenging scenario in practice. Appendix E.3 provides more details about attack settings.

Text Quality. Although perplexity (PPL) is widely used in prior work, it is insufficient for our setting: low-quality outputs with repeated sentences can receive artificially low PPL because repetitive patterns are easy for language models to predict. We therefore include three diversity-oriented metrics: **Sentence Duplicate ratio (SD)**, which is lower when generation is less redundant; **Distinct-2 (D-2)**, which is higher when lexical variety is greater; and **4-gram Repeat ratio (4g)**, which is lower when text is less repetitive. To evaluate text quality fairly, we use GPT-4.1-mini to conduct pairwise blind review between each watermarking method and unwatermarked baseline for each sample, producing **Win (PS-W)**, **Lose (PS-L)**, and **Tie (PS-T)**. Computation details in Appendix E.4.

4.2 DETECTABILITY, ROBUSTNESS AND TEXT QUALITY (RQ1 & RQ2)

To answer **RQ1** and **RQ2**, we comprehensively compare SAMARK with five widely used token-level watermarks and three semantic-level watermarks under various attacks. Table 1 jointly presents the BOOKSUM and C4 results. Additional paragraph-level attacks and full results with all metrics are provided in Appendices F.1 and F.2, respectively.

SAMARK achieves dominant robustness under most attack scenarios. SAMARK attains the highest detection rates across attack settings on both models and datasets. On Mistral with BOOKSUM under Doc-P II, SAMARK achieves 88.1 TP@FP1%, compared with 57.5 for the next-best baseline, yielding a 30.6 improvement. Similar advantages persist on Nemotron and C4. As a result, most baselines fail to maintain detectable signals under extreme paraphrasing, whereas SAMARK preserves robustness.

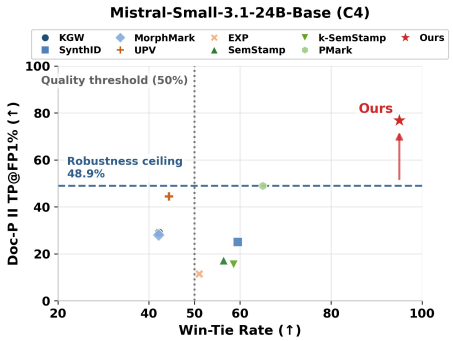


Figure 5: Robustness–quality trade-off on Mistral+C4. The x-axis reports the PS-W + PS-T (no worse than) rate.

The robustness advantage expands with stronger adversaries. SAMARK's performance gap over baselines widens as attack intensity increases. From Doc-P I to Doc-P II on Mistral-BOOKSUM, KGW and SynthID drops by 29.6 and 13.4 points in TP@FP1% respectively, while SAMARK decreases by only 2.4. The same pattern appears across models: on Nemotron-C4, UPV decreases by 21.8 points between attack levels, while SAMARK drops by only 5.4 points. This stability under escalating adversarial pressure indicates that self-anchored encoding preserves watermark signals even when surface-level features are substantially altered.

Table 1: Results on **BOOKSUM** and **C4**. We report TP@FP1%, pairwise LLM Judge W/L/T%, and AvgRank (Δ vs. unwatermarked). **Bold** = best, underlined = second-best. † denotes SAMARK without quality optimization.

Method	BOOKSUM								C4							
	TP@FP1%↑				*Quality*			*Rank*	*TP@FP1%*↑				*Quality*			*Rank*
	No Atk	Doc-P I	Doc-P II	Doc-T	PS-W↑	PS-L↓	PS-T	Avg↓ (Δ)	No Atk	Doc-P I	Doc-P II	Doc-T	PS-W↑	PS-L↓	PS-T	Avg↓ (Δ)
Mistral-Small-3.1-24B-Base-2503																
None	–	–	–	–	–	–	–	5.76 (+0.00)	–	–	–	–	–	–	–	5.69 (+0.00)
EXP	99.8	38.2	17.2	29.6	52.4	46.2	1.4	5.56 (-0.20)	99.4	24.6	11.5	36.4	50.8	49.0	0.2	5.38 (-0.32)
KGW	100.0	64.0	34.4	54.4	38.0	50.2	11.8	6.65 (+0.89)	100.0	77.7	29.0	64.2	41.6	57.8	0.6	6.56 (+0.87)
UPV	99.0	61.4	23.6	38.0	47.2	51.4	1.4	6.12 (+0.35)	99.7	67.1	44.5	71.5	44.0	55.6	0.4	6.17 (+0.48)
SynthID	99.8	23.8	10.4	12.6	54.2	38.6	7.2	5.11 (-0.65)	99.2	44.1	25.1	44.1	59.1	40.5	0.4	4.64 (-1.05)
MorphMark	99.4	56.0	34.2	45.8	47.6	51.4	1.0	6.08 (+0.32)	99.7	55.5	28.0	52.7	41.7	57.9	0.4	6.17 (+0.48)
SemStamp	96.0	49.8	39.8	58.6	53.2	39.2	7.6	5.04 (-0.72)	95.1	28.6	17.2	54.7	56.4	43.6	0.0	5.19 (-0.50)
k-SemStamp	98.6	61.7	43.1	65.9	51.4	47.2	1.4	4.94 (-0.82)	92.5	22.5	15.5	44.5	58.4	41.4	0.2	4.93 (-0.77)
PMark	98.0	51.5	57.5	77.0	57.2	37.6	5.2	4.61 (-1.15)	96.2	40.0	48.9	67.6	64.8	35.0	0.2	4.17 (-1.52)
SAMARK	98.0	**90.5**	**88.1**	**87.1**	47.9	45.2	6.9	5.14 (-0.62)	94.5	**78.6**	**76.8**	**84.2**	82.0	5.0	13.0	6.10 (+0.41)
SAMARK †	98.7	88.7	89.0	90.3	33.0	59.0	8.0	–	95.2	78.8	76.9	90.0	78.0	6.0	16.0	–
Nemotron-Nano-9B-v2-Base																
None	–	–	–	–	–	–	–	5.06 (+0.00)	–	–	–	–	–	–	–	5.23 (+0.00)
EXP	99.8	42.2	16.2	37.8	30.6	45.8	23.6	5.81 (+0.75)	100.0	39.1	16.9	58.6	36.7	58.4	5.0	6.21 (+0.98)
KGW	100.0	76.8	45.6	66.8	32.0	42.4	25.6	5.97 (+0.91)	100.0	55.9	27.3	56.2	39.6	56.0	4.4	5.98 (+0.75)
UPV	99.4	71.0	48.8	61.6	39.2	38.2	22.6	5.42 (+0.36)	99.7	75.0	53.2	79.5	42.2	53.2	4.6	5.64 (+0.41)
SynthID	99.8	31.4	15.6	21.2	47.8	33.4	18.8	4.38 (-0.67)	100.0	42.9	22.6	53.0	54.8	41.4	3.8	4.60 (-0.63)
MorphMark	100.0	74.0	46.2	64.0	36.4	40.4	23.2	5.57 (+0.51)	99.8	48.3	20.4	49.0	45.0	50.8	4.2	5.76 (+0.53)
SemStamp	97.0	46.9	34.5	64.3	39.2	41.6	19.2	5.24 (+0.18)	96.4	40.4	27.9	61.7	52.4	44.8	2.8	4.73 (-0.50)
k-SemStamp	97.6	51.7	37.1	63.7	35.0	45.0	20.0	5.58 (+0.52)	90.5	15.6	14.6	32.1	39.2	56.2	4.6	5.56 (+0.33)
PMark	96.6	24.2	28.3	46.3	26.0	49.8	24.2	6.22 (+1.16)	96.4	24.7	28.3	45.0	40.0	55.0	5.0	5.31 (+0.08)
SAMARK	98.8	**92.6**	**90.2**	**88.2**	34.2	46.0	19.8	5.75 (+0.69)	93.8	**78.3**	**72.9**	**82.3**	69.4	5.2	25.4	5.98 (+0.75)
SAMARK †	99.4	86.6	84.8	91.2	20.4	56.4	23.2	–	94.8	76.6	73.8	83.8	28.0	68.6	3.4	–

SAMARK maintains competitive text quality while achieving dominant robustness. Pairwise LLM-judge results show that SAMARK achieves high win rates on C4, suggesting that sentence-level filtering can introduce beneficial diversity. The trade-offs in Figure 5 show a consistent pattern across all backbone–dataset settings, where SAMARK surpasses the robustness ceiling of existing methods while maintaining competitive text quality. On Mistral+BOOKSUM, its AvgRank of 5.14 ($\Delta = -0.62$) outperforms the unwatermarked baseline (5.76) and remains comparable to SynthID (5.11), confirming that the robustness gains do not come at the expense of text quality.

SAMARK shows architecture-agnostic and distribution-agnostic consistency. SAMARK maintains TP@FP1% above 72% under all attack scenarios across both backbone models, while baselines show high variance. Under Doc-P II, PMark drops by 29.2 points across models, whereas SAMARK exhibits minimal fluctuation across datasets with different distributions. SAMARK consistently achieves at least 97.0% AUC across all 16 attack-model-dataset combinations, and no baseline matches this uniformity. These results suggest that the self-anchored encoding strategy captures fundamental linguistic invariants rather than model-specific or domain-specific artifacts. Results on additional backbones and datasets are provided in Appendix F.3.

Robustness under word-level attacks. Beyond paragraph-level paraphrasing, we further evaluate word-level perturbations that alter local lexical surface forms. Specifically, Word-S replaces a fraction of words with synonyms or context-compatible alternatives, while preserving most sentence-level semantics. As shown in Figure 6, existing semantic-level watermarks such as SemStamp, k-SemStamp, and PMark degrade substantially because their verification relies on stable sentence segmentation and sentence-internal structure. A similar trend appears under Word-D (Appendix F.4). In contrast, SAMARK remains more stable across perturbation types, because its signal is anchored to sentence-level semantics rather than exact token realization. This result is consistent with Table 1, and shows that SAMARK mitigates the structural fragility of prior semantic-level baselines without inheriting the strongest quality penalties of robust token-level methods. Additional robustness–quality trade-offs are provided in Appendix F.5.

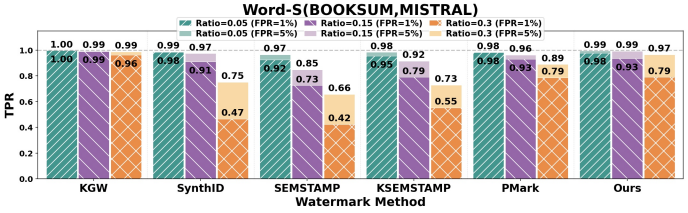


Figure 6: Word-S attack results.

4.3 EFFICIENCY PERFORMANCE (RQ3)

SAMARK achieves the lowest overall computation overhead among semantic-level methods. We compare the computation overhead of SAMARK and existing semantic-level baselines on BOOKSUM with Mistral-Small-3.1-24B-Base-2503, evaluating with the `vLLM` backend. As shown in Table 2, SAMARK achieves a generation latency of 0.22 s/token and 4.61 s/sentence, comparable to PMark and approximately 1.7× faster than SemStamp and k-SemStamp. It also consumes the fewest sampled tokens per output token and per sentence. During detection, SAMARK requires only a single embedding-model pass and lightweight algebraic operations, achieving 0.0014 s/token and over 130× lower latency than PMark.

Table 2: Computation overhead of different methods. L_t and L_s denote generation latency per token and per sentence; T_t and T_s denote sampled tokens per generated token and per sentence; L_d denotes detection latency per output token.

Method	Generation				Detection
	L_t (s)$_\downarrow$	L_s (s)$_\downarrow$	$T_t{}_\downarrow$	$T_s{}_\downarrow$	L_d (s)$_\downarrow$
SemStamp	0.395	7.58	64.50	1294.76	0.0063
k-SemStamp	0.361	6.65	65.05	1255.72	0.0043
PMark	0.205	4.63	49.73	1146.01	0.1861
SAMARK	0.218	4.61	47.44	1080.97	0.0014

4.4 HYPERPARAMETER SENSITIVITY ANALYSIS (RQ4)

To answer **RQ4**, we conduct a systematic sensitivity analysis on the key hyperparameters of SAMARK using Mistral on BOOKSUM. Here, we focus on the sample budget N that influences the efficiency of SWMs directly. An ablation of the hyperbolic transform and sensitivity analyses of the diversity-related parameters, component-wise ablations of diversity-aware filtering, and are provided in Appendix F.6 and F.7 respectively.

Larger sample budgets improve both robustness and quality. Table 3 shows that increasing the sample budget N from 64 to 128 and 200 consistently improves detection rates under all attack scenarios. At $N = 128$, SAMARK achieves 100.0% TP@FP1% under Doc-P I and 93.8% under Doc-P II, compared to 93.1% and 88.8% at $N = 64$. The quality metrics also benefit from larger budgets, as a richer candidate pool provides more opportunities to find sentences that simultaneously satisfy the watermark constraint and the diversity criteria. The 4-gram repetition rate drops from 4.96% at $N = 64$ to 1.53% at $N = 128$, and Distinct-2 improves from 0.82 to 0.88. We adopt $N = 64$ as the default to balance performance with computational cost, as discussed in Section 4.2. Additional score-distribution analyses and qualitative examples are provided in Appendix F.8.

Table 3: Effect of sample budget N on Mistral+BOOKSUM.

N	*TP@FP1%*$_\uparrow$			*Quality*		
	Doc-P I	**Doc-P II**	**Doc-T**	**SD%**$_\downarrow$	**D-2**$_\uparrow$	**4g%**$_\downarrow$
64	93.1	88.8	90.6	0.27	0.82	4.96
128	100.0	93.8	93.8	0.52	0.88	1.53
200	97.0	95.0	90.5	0.08	0.84	4.49

5 RELATED WORK

LLM watermarking includes token-level methods that control token sampling and semantic-level methods that embed signals in sentence representations (Kirchenbauer et al., 2023a; Aaronson, 2023; Hou et al., 2023; 2024; Dabiriaghdam & Wang, 2025). Although semantic-level schemes improve resilience to lexical rewriting, most rely on context-dependent hashing or partitioning; paragraph-level edits that reorder, split, or merge sentences can therefore disrupt their evidence. PMark strengthens semantic watermarking with multi-channel constraints but retains this dependence (Huo et al., 2026). In contrast, SAMARK self-anchors watermark evidence to each sentence and couples it with diversity-aware quality control. A complete review is provided in Appendix G.

6 CONCLUSION

Existing token-level and semantic-level watermarks rely on context-aware seeds or predefined keys during generation, making them vulnerable to paragraph-level paraphrasing that alters wording and sentence order. To address this limitation, we introduce SAMARK, a self-anchored watermarking framework whose green region at each step depends only on the semantics of the current sentence. By removing contextual-order dependence, SAMARK improves robustness to global paraphrasing. To the best of our knowledge, it is the first watermarking framework to achieve paragraph-level paraphrase robustness. Experiments show that SAMARK consistently outperforms existing SWM baselines in robustness while maintaining competitive generation quality, offering a more reliable paradigm for semantic-level watermarking.

REFERENCES

Scott Aaronson. Simons institute talk on watermarking of large language models, 2023. *URL https://simons.berkeley.edu/talks/scott-aaronson-ut-austin-openai-2023-08-17*, 2023.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

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

Ruibo Chen, Yihan Wu, Yanshuo Chen, Chenxi Liu, Junfeng Guo, and Heng Huang. A watermark for order-agnostic language models. In *International Conference on Learning Representations*, volume 2025, pp. 4372–4388, 2025.

Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Annual International Cryptology Conference*, pp. 325–347. Springer, 2024.

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

Aloni Cohen, Alexander Hoover, and Gabe Schoenbach. Watermarking language models for many adaptive users. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2583–2601. IEEE, 2025.

Amirhossein Dabiriaghdam and Lele Wang. Simmark: A robust sentence-level similarity-based watermarking algorithm for large language models. *arXiv preprint arXiv:2502.02787*, 2025.

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

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. Publicly-detectable watermarking for language models. *arXiv preprint arXiv:2310.18491*, 2023.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. In *2023 IEEE international workshop on information forensics and security (WIFS)*, pp. 1–6. IEEE, 2023.

Yu Fu, Deyi Xiong, and Yue Dong. 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, pp. 18003–18011, 2024.

Eva Giboulot and Teddy Furon. Watermax: breaking the llm watermark detectability-robustness-quality trade-off. *Advances in Neural Information Processing Systems*, 37:18848–18881, 2024.

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

Abe Bohan Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. *arXiv preprint arXiv:2402.11399*, 2024.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*, 2023.

Jiahao Huo, Shuliang Liu, Bin Wang, Junyan Zhang, Yibo Yan, Aiwei Liu, Xuming Hu, and Mingxun Zhou. PMark: Towards robust and distortion-free semantic-level watermarking with channel constraints. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=EhDgP69DJG.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pp. 604–613, 1998.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *International Conference on Machine Learning*, pp. 17061–17084. PMLR, 2023a.

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

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. *arXiv preprint arXiv:2105.08209*, 2021.

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

Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S Yu. An unforgeable publicly verifiable watermark for large language models. *arXiv preprint arXiv:2307.16230*, 2023.

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

Mistral AI. Mistral small 3.1: The best model in its weight class. Technical report, Mistral AI, 2025. URL https://mistral.ai/news/mistral-small-3-1.

OpenAI. ChatGPT, 2022. URL https://openai.com/blog/chatgpt.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, et al. Markllm: An open-source toolkit for llm watermarking. *arXiv preprint arXiv:2405.10051*, 2024.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for {AI-generated} text. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 201–220, 2025.

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

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language model against paraphrasing. *arXiv preprint arXiv:2311.08721*, 2023.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. *arXiv preprint arXiv:2307.15992*, 2023.

Zongqi Wang, Tianle Gu, Baoyuan Wu, and Yujiu Yang. Morphmark: Flexible adaptive watermarking for large language models. *arXiv preprint arXiv:2505.11541*, 2025.

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

Linyu Wu, Linhao Zhong, Wenjie Qu, Yuexin Li, Yue Liu, Shengfang Zhai, Chunhua Shen, and Jiaheng Zhang. Dmark: Order-agnostic watermarking for diffusion large language models. *arXiv preprint arXiv:2510.02902*, 2025.

Xiaojun Xu, Jinghan Jia, Yuanshun Yao, Yang Liu, and Hang Li. Robust multi-bit text watermark with llm-based paraphrasers. *arXiv preprint arXiv:2412.03123*, 2024.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 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)*, pp. 4031–4055, 2024.

Zhuohao Yu, Xingru Jiang, Weizheng Gu, Yidong Wang, Qingsong Wen, Shikun Zhang, and Wei Ye. SAEMark: Steering personalized multilingual LLM watermarks with sparse autoencoders. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=tXnyVPNOfa.

Junyan Zhang, Shuliang Liu, Aiwei Liu, Yubo Gao, Jungang Li, Xiaojie Gu, and Xuming Hu. Cohemark: A novel sentence-level watermark for enhanced text quality. *arXiv preprint arXiv:2504.17309*, 2025.

Yuehan Zhang, Peizhuo Lv, Yinpeng Liu, Yongqiang Ma, Wei Lu, Xiaofeng Wang, Xiaozhong Liu, and Jiawei Liu. Personamark: Personalized llm watermarking for model protection and user attribution. *arXiv preprint arXiv:2409.09739*, 2024.

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

Tong Zhou, Xuandong Zhao, Xiaolin Xu, and Shaolei Ren. Bileve: Securing text provenance in large language models against spoofing with bi-level signature. *Advances in Neural Information Processing Systems*, 37:56054–56075, 2024.

A AI USAGE STATEMENT

We used AI assistants for two purposes: (1) generating routine code and boilerplate functions, which were subsequently reviewed and debugged by humans, and (2) performing grammatical review and sentence-level editing of the manuscript. The research methodology, findings, and analysis were independently proposed and conducted.

B ETHICS CONSIDERATIONS

This work studies text watermarking for identifying AI-generated content. Our experiments use publicly available datasets and generated model outputs, and do not involve recruiting human subjects or collecting personal information. Potential risks include false-positive detections and misuse of watermark detectors for unsupported attribution decisions. We therefore report low-FPR detection metrics and emphasize that watermarking evidence should be interpreted as a statistical signal rather than definitive proof of authorship.

C LIMITATIONS AND FUTURE WORK

Despite its strong paragraph-level paraphrase robustness, SAMARK has several limitations. First, like other semantic-level watermarking methods, it requires sampling multiple sentence candidates before selecting one that satisfies the target constraint, since current autoregressive LLMs are decoded token by token rather than sentence by sentence. This extra sampling loop makes generation latency higher than that of token-level methods. Second, our diversity-aware filtering improves text quality in practice but does not yet provide a formal distortion-free guarantee, because its hard filters and soft regularizers still modify the effective sampling distribution. Finally, our work focuses on zero-bit watermarking, which embeds only a yes/no signal into generated text and is not directly applicable to multi-bit scenarios that require payload encoding.

Several directions could address these limitations. On the efficiency side, emerging architectures such as diffusion language models and large concept models operate natively over sentence- or concept-level units and may provide a more direct interface for semantic watermarking, removing the need for repeated candidate sampling; complementary techniques such as steering generation could further guide the model toward valid candidates in a single pass. On the distortion side, a deeper theoretical understanding of the geometry of semantic embedding spaces, beyond the empirically motivated hyperbolic transform used here, could inform sampling strategies that remain robust to paragraph-level paraphrasing while achieving a formal distortion-free guarantee. For the multi-bit extension, the self-anchored partition can be generalized from a binary green/red split to a multi-way partition of the semantic space, allowing each sentence to carry several payload bits while preserving robustness to paraphrasing. We leave these as future work.

D DETAILED METHODOLOGY

In Algorithm 7 we provide the pseudocode of our generation and detection algorithm for better comprehension.

D.1 ADDITIONAL MECHANISM ANALYSES

D.1.1 EMPIRICAL AND CROSS-FITTED CALIBRATION

Empirical calibration controls false positives without sacrificing detectability. To verify the nominal FP1% operating point, we compare two detectors under repeated held-out calibration. The original detector estimates channel signs from all sentences and scores those same sentences, whereas the cross-fitted detector splits the sentences into two halves and uses each half to estimate the signs for the other. As shown in Table 4, the held-out FPR remains between 0.89% and 1.05% across all settings, while the TPR stays close to the corresponding no-attack result. Cross-fitting produces nearly identical FPR and TPR. We therefore use the original detector with setting-specific empirical calibration and treat cross-fitting as a robustness check.

D.1.2 AUTOMATIC QUALITY AUDIT

Algorithm 1: SAMARK Generation

Input : LLM π; context $u_{<1}$; embedding model $\mathcal{E}$; private key k; sentences T; channels c; budget N; κ; ϵ; $\lambda_{\text{div}}, \lambda_{\text{nov}}$, $\theta_{\text{ngram}}, \theta_{\text{sem}}$

Output : Generated sentences $u_1, \ldots, u_T$

Derive pivot vectors $V = \{v_1, \ldots, v_c\}$ and flags $R = \{r_1, \ldots, r_c\}$ from k;
$\mathcal{H} \leftarrow \emptyset$; $\mathcal{V}_{<1} \leftarrow \text{vocab}(u_{<1})$;
for $t \leftarrow 1$ **to** T **do**
  Sample $W = \{x_1, \ldots, x_N\}$ with $x_i \sim p_\pi(\cdot \mid u_{<t})$;
  $W \leftarrow \{x \in W \mid \rho_{\text{ngram}}(x) < \theta_{\text{ngram}}\}$;
  $W \leftarrow \{x \in W \mid \rho_{\text{sem}}(x) < \theta_{\text{sem}}\}$;
  $V_{\text{match}} \leftarrow \{x \in W \mid \forall j : \text{sgn}(\cos(\mathcal{E}(x), v_j)) = r_j\}$;
  foreach $x \in V_{match}$ **do**
    $S_{\text{wm}}(x) \leftarrow \sum_{j=1}^{c} \tanh(\kappa \cdot r_j \cdot \cos(\mathcal{E}(x), v_j))$;
    $S(x) \leftarrow S_{\text{wm}}(x) + \lambda_{\text{div}} D(x) + \lambda_{\text{nov}} \mathcal{N}(x)$;
  end
  $q(x) \leftarrow \exp(\epsilon \cdot S(x)) / \sum_{x'} \exp(\epsilon \cdot S(x'))$;
  Sample $u_t \sim q(\cdot)$; $\mathcal{H} \leftarrow \mathcal{H} \cup \{u_t\}$;
end
return $u_1, \ldots, u_T$;

Algorithm 2: SAMARK Detection

Input : Text X; embedding model $\mathcal{E}$; private key k; channels c; sharpness κ; threshold τ

Output : `True` (watermarked) or `False`

Segment X into sentences $S = [s_1, \ldots, s_n]$;
Derive pivot vectors $V = \{v_1, \ldots, v_c\}$ from k;
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** c **do**
    $C_{i,j} \leftarrow \cos(\mathcal{E}(s_i), v_j)$;
  end
end
for $j \leftarrow 1$ **to** c **do**
  $\hat{r}_j \leftarrow \text{sgn}\left(\sum_{i=1}^{n} C_{i,j}\right)$;
end
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow 1$ **to** c **do**
    $A_{i,j} \leftarrow \tanh(\kappa \cdot C_{i,j} \cdot \hat{r}_j)$;
  end
end
$\bar{A} \leftarrow \frac{1}{nc} \sum_{i,j} A_{i,j}$;
$\hat{\sigma}_A \leftarrow \sqrt{\frac{1}{nc} \sum_{i,j} (A_{i,j} - \bar{A})^2}$;
$z \leftarrow \bar{A} \cdot \sqrt{nc} \,/\, \hat{\sigma}_A$;
return $(z > \tau)$;

Figure 7: SAMARK watermarking algorithms. **Left**: multi-channel hyperbolic generation with diversity-aware filtering. **Right**: blind majority-vote detection with hyperbolic score alignment.

Table 4: Empirical-calibration audit at the nominal FP1% operating point. CV denotes cross-validation. Results are averaged over 20 repetitions of five-fold CV, and each detector entry reports held-out FPR/TPR (%).

Dataset	Backbone	No-attack TP↑	Original detector	Cross-fitted detector
BOOKSUM	Mistral	98.0	1.02/97.83	1.03/97.91
BOOKSUM	Nemotron	98.8	1.02/98.72	1.03/98.80
C4	Mistral	92.8	1.00/92.14	0.89/92.13
C4	Nemotron	89.9	1.02/89.24	1.05/89.23

SAMARK maintains lexical diversity with limited repetition. To complement subjective LLM-based judgments with reproducible surface-level evidence, we conduct an automatic audit of lexical diversity and repetition. As shown in Table 5, SAMARK achieves Distinct-2 scores of 0.746–0.844, while duplicate-sentence rates remain below 0.7% across all settings. These automatic measurements complement the LLM-judge evaluation by directly auditing diversity and repetition.

Table 5: Automatic quality audit of SAMARK outputs. 4g and SD denote repeated 4-grams and duplicate sentences, respectively.

Dataset	Backbone	D-2↑	4g%↓	SD%↓
BOOKSUM	Mistral	0.817	5.18	0.19
BOOKSUM	Nemotron	0.746	10.62	0.62
C4	Mistral	0.823	6.58	0.54
C4	Nemotron	0.844	5.25	0.65

D.1.3 MARGIN AND CHANNEL-SIGN STABILITY

SAMARK preserves channel signs under semantic perturbations. To examine whether paraphrasing changes the semantic-region assignment that carries the watermark, we align pre- and post-attack sentences and measure channel-wise sign flips. As shown in Table 6, only 1.15%–4.81% of aligned sentence–channel pairs cross the region boundary after document-level attacks, even though their mean semantic cosine ranges from 0.716 to 0.891. This low sign-flip rate supports our central

Table 7: Detection boundaries of SAMARK at FP1%. The upper panel varies the watermarked-sentence proportion ρ in 12-sentence interleaved texts; the lower panel varies text length for fully watermarked texts. All entries are TP@FP1%$_{\uparrow}$.

Mixed-source texts (12 sentences)

Dataset	Backbone	$\rho = 25\%$	$\rho = 50\%$	$\rho = 75\%$	$\rho = 100\%$
BOOKSUM	Mistral	1.7	5.4	36.6	98.0
BOOKSUM	Nemotron	1.7	6.2	39.0	99.3
C4	Mistral	2.0	6.6	25.3	94.1
C4	Nemotron	1.7	5.7	23.1	92.6

Fully watermarked texts

Dataset	Backbone	2 sent.	4 sent.	6 sent.	8 sent.	12 sent.
BOOKSUM	Mistral	51.5	90.8	93.9	97.5	98.0
BOOKSUM	Nemotron	60.5	94.0	98.5	99.5	99.3
C4	Mistral	33.4	70.8	83.1	87.3	94.1
C4	Nemotron	33.2	69.1	78.1	84.7	92.6

motivation for keeping watermark evidence away from the decision boundary: a larger semantic margin makes the self-anchored signal less likely to change sign under paraphrasing.

Table 6: Channel-sign stability under document-level attacks. Each entry reports sign-flip rate (%)/mean semantic alignment cosine; lower flip rates and higher cosine values indicate stronger stability.

Dataset	Backbone	Doc-P I	Doc-P II	Doc-T
BOOKSUM	Mistral	1.55/0.862	2.25/0.846	3.22/0.859
BOOKSUM	Nemotron	1.15/0.852	1.63/0.847	2.39/0.878
C4	Mistral	2.49/0.846	3.28/0.822	2.34/0.891
C4	Nemotron	3.77/0.731	4.81/0.716	4.39/0.800

D.1.4 MIXED-SOURCE AND SHORT-TEXT BOUNDARIES

Reliable detection requires sufficient watermark coverage and text length. To characterize the minimum amount of watermark evidence required for reliable detection, we vary both the fraction of watermarked sentences and the total text length. As shown in Table 7, detection is weak when only 25%–50% of a 12-sentence text is watermarked, but rises sharply as coverage approaches 100%. Short texts exhibit a similar boundary: fully watermarked two-sentence inputs achieve only 33.2%–60.5% TP@FP1%, whereas performance reaches 69.1%–94.0% with four sentences and 92.6%–99.3% with 12 sentences. These results clarify that SAMARK is intended for paragraph-level text with sufficient watermarked evidence rather than isolated sentences or lightly watermarked mixtures.

D.2 ILLUSTRATION OF SAMARK SAMPLING PROCESS

In Figure 8, each point denotes one sampled next-sentence candidate s_i from a pool of size N. Let e_i be its sentence embedding, e_{prev} be the previous selected sentence embedding, and $\{v_k\}_{k=1}^{b}$ be channel pivots with channel signs $w_k \in \{-1, +1\}$. We define channel cosine values

$$c_{ik} = \cos(e_i, v_k), \quad \tilde{c}_{ik} = w_k\, c_{ik}, \tag{6}$$

and the quality score

$$q_i = \cos(e_i, e_{\text{prev}}). \tag{7}$$

The robustness score before and after hyperbolic transformation is

$$r_i^{\text{raw}} = \sum_{k=1}^{b} \tilde{c}_{ik}, \qquad r_i^{\tanh} = \sum_{k=1}^{b} \tanh(\kappa\, \tilde{c}_{ik}), \tag{8}$$

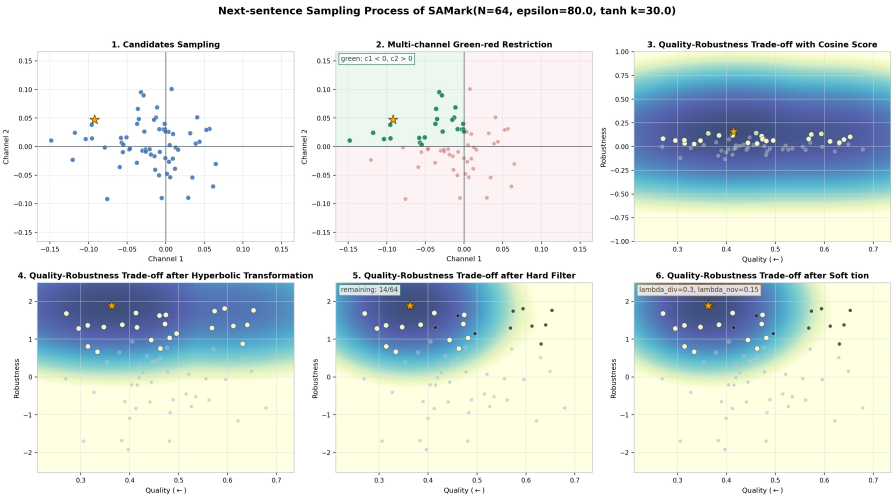


Figure 8: Illustration of the next-sentence sampling process of SAMark.

where κ is the tanh scale.

Panel 1 plots (c_{i1}, c_{i2}) for all candidates. Panel 2 applies the multi-channel green-red restriction and keeps candidates with matched channel signs (mask m_i^{green}). Panels 3–4 visualize the quality-robustness trade-off (q_i, r_i) with sampling probabilities

$$p_i \propto \exp(\epsilon r_i), \tag{9}$$

computed on active candidates under m_i^{green}, using r_i^{raw} in Panel 3 and $r_i^{\tanh}$ in Panel 4.

Panel 5 adds a hard filter mask m_i^{hard} (e.g., overlap and semantic-similarity constraints), and samples by

$$p_i^{\text{hard}} \propto \exp\big(\epsilon r_i^{\tanh}\big)\, \mathbf{1}[m_i^{\text{hard}}]. \tag{10}$$

Panel 6 adds soft regularization with diversity and novelty:

$$\begin{aligned} r_i^{\text{reg}} &= r_i^{\tanh} + \lambda_{\text{div}} d_i + \lambda_{\text{nov}} n_i, \\ p_i^{\text{reg}} &\propto \exp\big(\epsilon r_i^{\text{reg}}\big)\, \mathbf{1}[m_i^{\text{hard}}], \end{aligned} \tag{11}$$

where $d_i = 1 - \text{sim}_i$ and n_i is a lexical novelty bonus. The final selected sentence is the argmax under p_i^{reg}.

E Experiment Setup Details

E.1 Baselines

MarkLLM. We use the official MarkLLM implementation (Pan et al., 2024)[1] to reproduce the results of various token-level baselines, including KGW (Kirchenbauer et al., 2023a), UPV (Liu et al., 2023), MorphMark (Wang et al., 2025), EXP (Aaronson, 2023), and SynthID (Dathathri et al., 2024). During generation, we set `max_new_tokens` to 256, which is comparable to the generation length used by semantic-level methods. The temperature and `top_p` are fixed at 0.7 and 0.95, respectively, across all experiments to ensure a fair comparison. In addition, for baselines that depend on external networks (e.g., UPV (Liu et al., 2023)), we use the official weights provided by MarkLLM.

k-SemStamp. We adapt the official k-SemStamp implementation from MarkLLM to align it with other semantic-level methods. Specifically, we set `max_new_sentences` to 12 (instead of a token count), and apply this setting consistently across all semantic-level baselines. We set `max_trials` to the default value of 100 for SemStamp and k-SemStamp. For the embedding model, we use the fine-tuned `all-mpnet-base-v2` (Reimers & Gurevych, 2019)[2] provided by (Hou et al., 2024). We also use the k-means centroid weights[3] released by (Hou et al., 2024), trained on the C4 (Raffel

[1] https://github.com/THU-BPM/MarkLLM
[2] https://huggingface.co/AbeHou/SemStamp-c4-sbert
[3] https://github.com/abehou/SemStamp

et al., 2020) and BookSum (Kryściński et al., 2021) datasets. All other hyperparameters are kept at their default settings in the k-SemStamp implementation of MarkLLM.

SemStamp. For SemStamp (Hou et al., 2023), we use the authors' official implementation, setting temperature $= 0.7$, `top_p` $= 0.95$, and `max_new_sentences` $= 12$ to match the other semantic-level methods. We also use the fine-tuned embedding model for a fair comparison.

PMark. For PMark (Huo et al., 2026), we use the authors' official implementation of the online PMark variant, setting temperature $= 0.7$, `top_p` $= 0.95$, and `max_new_sentences` $= 12$ to match the other semantic-level methods, with `all-mpnet-base-v2` as the embedding model. The channel number is set to 4, and the sample budget is $N = 64$.

SAMARK. During generation, we use the original `all-mpnet-base-v2` embedding model (Reimers & Gurevych, 2019)[4] without fine-tuning. Pivot vectors are generated via QR decomposition of a Gaussian random matrix to ensure orthogonality. We set temperature $= 0.7$, `top_p` $= 0.95$, and `max_new_sentences` $= 12$, consistent with other semantic-level methods. The sample budget is $N = 64$, and the number of channels is $b = 2$.

E.2 PROMPT TEMPLATES

We summarize the prompt templates used in the evaluation pipeline.

Doc-T. The back-translation attack uses two translation prompts:

- EN→ZH: "Translate the following text from English to Chinese. Only output the translation, nothing else:"
- ZH→EN: "Translate the following text from Chinese to English. Only output the translation, nothing else:"

LLM pairwise judge. For blind ranking, we use pairwise comparison with dataset-specific system prompts:

- `booksum`: "Compare Summary A/B by accuracy, completeness, coherence, writing quality, and absence of obvious errors."
- `c4`: "Compare Continuation A/B by topical relevance, coherence, writing quality, informativeness, and absence of obvious errors."

Both settings require JSON-only output, as shown in Figure 9.

Pairwise Judge Output Template

```
{"winner": "A" or "B" or "tie", "reason": "<one sentence>"}
```

Figure 9: Output template for the LLM pairwise judge.

For example, the pairwise user query template of `booksum` is shown in Figure 10.

To ensure blind evaluation, watermarked and unwatermarked texts are assigned to A/B in randomized order for each sample, and final win/lose/tie is mapped back to the watermarked side.

LLM ranking prompts. For multi-method blind ranking, we evaluate multiple candidates (A/B/C/...) jointly with dataset-specific system prompts:

- `booksum`: "Rank all summaries by accuracy, completeness, coherence, writing quality, and absence of obvious errors."
- `c4`: "Rank all continuations by topical relevance, coherence, writing quality, informativeness, and absence of obvious errors."

The output is constrained to JSON, as shown in Figure 11, where the first element is best. Ties are explicitly allowed using entries such as `"A=B"`.

For instance, the ranking query template of `booksum` is shown in Figure 12.

For each sample, method outputs are first randomly mapped to blind labels A/B/C/..., then mapped back after parsing.

[4] https://huggingface.co/sentence-transformers/all-mpnet-base-v2

BOOKSUM Pairwise Query Template

```
## Book Chapter Opening
{prompt}

## Summary A
{text_a}

## Summary B
{text_b}

Which summary is better? Return ONLY JSON:
{"winner": "A" or "B" or "tie", "reason": "..."}
```

Figure 10: Pairwise user query template for `booksum`.

Multi-method Ranking Output Template

```
{"ranking": ["X", "Y", ...], "reason": "<one sentence>"}
```

Figure 11: Output template for multi-method blind ranking.

E.3 Attack Details

Doc-P. We conduct document-level paraphrase attacks using GPT-4.1-mini (Achiam et al., 2023). Two prompt templates are used to instruct paraphrasing at different adversarial strengths. Attack Prompt I: "Please rewrite the following text:"; Attack Prompt II: "Please rewrite the following text, avoiding using the same words or phrases in the original text:". All attacks are applied at the paragraph level across methods and settings to simulate real-world paragraph-level paraphrasing scenarios.

Doc-T. We implement the back-translation attack following MarkLLM (Pan et al., 2024). Specifically, we use GPT-4.1-mini to translate watermarked English text into Chinese and then back into English in an attempt to remove watermark evidence. For the two translation steps, the prompts are "Translate the following text from English to Chinese. Only output the translation, nothing else:" and "Translate the following text from Chinese to English. Only output the translation, nothing else:", respectively.

Word-D and Word-S. We implement word deletion (Word-D) and synonym substitution (Word-S) attacks following the official MarkLLM implementation. We set the attack ratios to 5%, 15%, and 30% to evaluate the overall robustness of our method. We apply word-level attacks to the whole paragraph rather than sentence by sentence, unlike PMark (Huo et al., 2026). Additional experimental results are reported in Appendix F.4.

Multi-turn Doc-P. We apply Doc-P II for five successive turns using GPT-4.1-mini, with each turn paraphrasing the output from the preceding turn rather than the original watermarked text. The prompt is: "Please rewrite the following text, avoiding using the same words or phrases in the original text. Preserve every fact, entity, relation, chronology, and the overall meaning. Output only the rewritten text."

Merge. We prompt GPT-4.1-mini to merge adjacent sentences while preserving the original content. The prompt is: "Rewrite the passage below by merging adjacent sentences where it is natural to do so. Preserve every fact, entity, relation, and the overall meaning. Do not add commentary, headings, or explanations. Output only the rewritten passage."

Split. We prompt GPT-4.1-mini to split long sentences into shorter ones while preserving their meaning and factual content. The prompt is: "Rewrite the passage below by splitting some long sentences into shorter, natural sentences. Preserve every fact, entity, relation, and the overall meaning. Do not add commentary, headings, or explanations. Output only the rewritten passage."

Summarize. We prompt GPT-4.1-mini to summarize each watermarked text to approximately 70% of its original word count. The prompt is: "Rewrite the passage below as a controlled summary of

BOOKSUM Ranking Query Template

```
## Book Chapter Opening
{prompt}

## Summary A
{text_a}

## Summary B
{text_b}
...

Rank ALL summaries from best to worst.
Ties are allowed.
Return ONLY JSON:
{"ranking": ["X", "Y", ...], "reason": "..."}
For ties: ["X=Y", "Z"] means X and Y tied for 1st, Z is 3rd.
```

Figure 12: Ranking user query template for `booksum`.

approximately 70 percent of its original word count. Preserve the central facts, entities, relations, and chronology; do not invent information. Do not add commentary, headings, or explanations. Output only the summary."

Doc-P II (Kimi). We use Kimi-K2.5 as an independent attacker under the Doc-P II setting. The prompt is: "Please rewrite the following text, avoiding using the same words or phrases in the original text. Preserve every fact, entity, relation, chronology, and the overall meaning. Output only the rewritten text."

Adaptive Removal. Starting from watermarked text, a score-oracle attacker generates four rewrites with GPT-4.1-mini and retains the candidate with the lowest watermark score. The prompt is: "Rewrite the passage to reduce the SAMARK watermark detector score while preserving every fact, entity, relation, chronology, and the natural English. Use a genuine semantic paraphrase, not deletion or nonsense. Return exactly one rewrite."

Adaptive Forgery. Starting from unwatermarked text, the same score-oracle attacker generates four rewrites and retains the candidate with the highest watermark score. The prompt is: "Rewrite this ordinary unwatermarked passage into a fluent, fact-preserving version that maximizes the chance of passing the SAMARK watermark detector. Do not mention watermarking or add false facts. Return exactly one rewrite."

E.4 QUALITY METRICS

For each generated sample x, we compute three automatic diversity metrics in addition to the pairwise LLM judge. Let $\mathcal{S}(x)$ denote the sequence of normalized sentences in x after sentence tokenization, trimming, and lowercasing, and let $\mathcal{G}_n(x)$ denote the sequence of word n-grams obtained after lowercasing and whitespace tokenization. Sentence Duplicate percentage (SD) is the fraction of generated sentences that duplicate an earlier sentence in the same output:

$$\mathrm{SD}(x) = 100 \times \frac{|\mathcal{S}(x)| - |\operatorname{uniq}(\mathcal{S}(x))|}{|\mathcal{S}(x)|}. \tag{12}$$

Distinct-2 (D-2) measures bigram diversity:

$$\text{D-2}(x) = \frac{|\operatorname{uniq}(\mathcal{G}_2(x))|}{|\mathcal{G}_2(x)|}. \tag{13}$$

The 4-gram Repeat percentage (4g) measures repetitions beyond the first occurrence of each 4-gram:

$$4\mathrm{g}(x) = 100 \times \frac{|\mathcal{G}_4(x)| - |\operatorname{uniq}(\mathcal{G}_4(x))|}{|\mathcal{G}_4(x)|}. \tag{14}$$

If the denominator is zero, the corresponding metric is set to 0. We report the average of each metric over the evaluation set. Lower SD and 4g indicate less repetition, while higher D-2 indicates greater lexical diversity.

Table 8: Additional paragraph-level attack results for SAMARK on Mistral-Small-3.1-24B-Base-2503 with BOOKSUM. For Adaptive Forgery, the TP columns report false-acceptance rates at the corresponding thresholds, and lower TP and AUC values indicate stronger resistance.

Attack family	Attack setting	TP@FP1%	TP@FP5%	AUC
Multi-turn Doc-P	Turn 1↑	86.7	97.0	99.32
	Turn 2↑	84.9	97.8	99.25
	Turn 3↑	83.9	95.6	99.01
	Turn 4↑	83.1	96.8	99.00
	Turn 5↑	83.5	95.4	98.95
Structural	Merge↑	90.4	97.0	99.33
	Split↑	83.2	96.8	98.98
	Summarize↑	84.3	96.6	99.30
Independent model	Doc-P II (Kimi)↑	73.8	91.9	98.11
Adaptive	Adaptive Removal↑	81.0	92.0	98.50
	Adaptive Forgery↓	3.5	16.5	65.62

Table 9: Full results on **BOOKSUM** dataset. For detection, we report TP@FP=1%/TP@FP=5%/AUC (%). **Doc-P** denotes document-level paraphrase attack (GPT) with Attack Prompt I and II; **Doc-T** denotes document-level translation attack (GPT). For text quality, we report Sentence Duplicate% (SD, lower is better), Distinct-2 (D-2, higher is better), and pairwise LLM Judge Win/Lose/Tie% against unwatermarked text. **Bold** = best, underlined = second-best.

Method	*Detection*				*Text Quality*				
	No Atk	Doc-P I	Doc-P II	Doc-T(GPT)	SD%↓	D-2↑	PS-W%	PS-L%	PS-T%
Mistral-Small-3.1-24B-Base-2503									
SemStamp (Hou et al., 2023)	96.0/98.0/99.5	49.8/64.3/91.1	39.8/54.6/85.7	58.6/72.7/92.1	8.9	0.761	53.2	39.2	7.6
k-SemStamp (Hou et al., 2024)	98.6/99.6/99.9	61.7/78.8/94.1	43.1/62.1/88.3	65.9/81.6/95.3	9.5	0.756	51.4	47.2	1.4
KGW (Kirchenbauer et al., 2023a)	100.0/100.0/100.0	64.0/84.0/96.6	34.4/59.6/89.7	54.4/76.0/95.0	0.4	0.948	38.0	50.2	11.8
SynthID (Dathathri et al., 2024)	99.8/99.8/100.0	23.8/47.2/80.8	10.4/24.2/64.5	12.6/35.4/75.1	0.2	0.910	54.2	38.6	7.2
MorphMark (Wang et al., 2025)	99.4/99.8/100.0	56.0/78.6/94.7	34.2/55.8/86.9	45.8/74.0/92.7	0.2	0.942	47.6	51.4	1.0
UPV (Liu et al., 2023)	99.0/99.4/98.7	61.4/77.0/85.7	23.6/39.6/65.9	38.0/57.2/75.3	0.7	0.928	47.2	51.4	1.4
EXP (Aaronson, 2023)	99.8/99.8/100.0	38.2/57.4/86.8	17.2/25.0/69.9	29.6/45.2/82.8	10.2	0.770	52.4	46.2	1.4
PMark (Huo et al., 2026)	98.0/98.4/99.4	51.5/66.7/89.9	57.5/71.5/92.0	77.0/82.8/94.3	4.0	0.788	57.2	37.6	5.2
SAMARK (Ours)	98.0/99.6/99.8	**90.5/97.9/99.5**	**88.1/97.4/99.3**	**87.1/97.1/99.1**	0.1	0.816	47.9	45.2	6.9
NVIDIA-Nemotron-Nano-9B-v2-Base									
SemStamp (Hou et al., 2023)	97.0/98.2/99.7	46.9/63.9/90.2	34.5/50.5/86.3	64.3/76.6/93.9	15.9	0.687	39.2	41.6	19.2
k-SemStamp (Hou et al., 2024)	97.6/98.8/99.2	51.7/74.3/93.4	37.1/58.9/88.6	63.7/79.0/93.2	17.9	0.696	35.0	45.0	20.0
KGW (Kirchenbauer et al., 2023a)	100.0/100.0/100.0	76.8/90.0/97.8	45.6/72.2/93.1	66.8/85.2/97.0	0.1	0.941	32.0	42.4	25.6
SynthID (Dathathri et al., 2024)	99.8/100.0/100.0	31.4/52.6/82.8	15.6/29.8/68.4	21.2/46.0/81.6	0.1	0.897	47.8	33.4	18.8
MorphMark (Wang et al., 2025)	100.0/100.0/100.0	74.0/88.4/97.7	46.2/70.8/92.7	64.0/85.6/96.5	0.2	0.937	36.4	40.4	23.2
UPV (Liu et al., 2023)	99.4/99.6/98.8	71.0/83.4/87.7	48.8/65.8/78.9	61.6/77.6/85.4	0.4	0.916	39.2	38.2	22.6
EXP (Aaronson, 2023)	99.8/99.8/100.0	42.2/57.6/86.9	16.2/25.4/71.7	37.8/55.0/83.5	16.7	0.686	30.6	45.8	23.6
PMark (Huo et al., 2026)	96.6/98.4/99.5	24.2/43.5/82.0	28.3/48.1/84.0	46.3/60.1/87.9	15.1	0.732	26.0	49.8	24.2
SAMARK (Ours)	98.8/99.8/99.9	**92.6/97.6/99.4**	**90.2/96.4/99.1**	**88.2/96.2/99.0**	0.6	0.746	34.2	46.0	19.8

F ADDITIONAL EXPERIMENTS

F.1 ADDITIONAL PARAGRAPH-LEVEL ATTACKS

As shown in Table 8, SAMARK remains robust under multi-turn, structural, independent-model, and adaptive-removal attacks, while maintaining a low false-acceptance rate under adaptive forgery.

F.2 FULL RESULTS

Tables 9 and 10 present the complete experimental results with all metrics.

F.3 ADDITIONAL BACKBONE AND DATASET RESULTS

We further evaluate five representative methods on an additional backbone and dataset suite, including Gemma3-12B-PT and reddit-TIFU. Every cell in Tables 11 and 12 is TP@FP1% over 500 matched samples. The extension runs contain the three available detection conditions (No Atk, Doc-P I, and Doc-P II).

Table 10: Full results on **C4** dataset. For detection, we report TP@FP=1%/TP@FP=5%/AUC (%). **Doc-P** denotes document-level paraphrase attack (GPT) with Attack Prompt I and II; **Doc-T** denotes document-level translation attack (GPT). For text quality, we report Sentence Duplicate% (SD, lower is better), Distinct-2 (D-2, higher is better), and pairwise LLM Judge Win/Lose/Tie% against unwatermarked text. **Bold** = best, underlined = second-best.

Method	*Detection*				*Text Quality*				
	No Atk	Doc-P I	Doc-P II	Doc-T(GPT)	SD%↓	D-2↑	PS-W%	PS-L%	PS-T%
Mistral-Small-3.1-24B-Base-2503									
SemStamp (Hou et al., 2023)	95.1/99.0/99.7	28.6/58.1/87.4	17.2/31.0/73.2	54.7/77.8/92.1	8.4	0.788	56.4	43.6	0.0
k-SemStamp (Hou et al., 2024)	92.5/99.5/99.0	22.5/59.0/87.5	15.5/40.2/79.8	44.5/74.5/91.1	5.3	0.821	58.4	41.4	0.2
KGW (Kirchenbauer et al., 2023a)	100.0/100.0/100.0	77.7/**90.7/97.6**	29.0/47.2/88.3	64.2/84.5/96.3	0.0	0.953	41.6	57.8	0.6
SynthID (Dathathri et al., 2024)	99.2/99.2/99.7	44.1/58.3/84.4	25.1/37.2/69.9	44.1/59.4/83.6	0.1	0.921	59.1	40.5	0.4
MorphMark (Wang et al., 2025)	99.7/100.0/100.0	55.5/77.4/95.4	28.0/51.4/89.2	52.7/76.8/95.8	0.0	0.947	41.7	57.9	0.4
UPV (Liu et al., 2023)	99.7/100.0/99.7	67.1/85.9/84.3	44.5/74.6/75.2	71.5/92.8/85.8	0.1	0.931	44.0	55.6	0.4
EXP (Aaronson, 2023)	99.4/99.4/99.4	24.6/48.0/82.6	11.5/22.7/70.5	36.4/60.4/88.0	6.6	0.810	50.8	49.0	0.2
PMark (Huo et al., 2026)	96.2/96.4/98.9	40.0/54.0/84.3	48.9/60.4/86.4	67.6/73.8/92.8	3.9	0.811	64.8	35.0	0.2
SAMARK (Ours)	94.5/97.6/99.5	**78.6**/89.3/97.5	**76.8/87.5/97.0**	**84.2/93.0/98.3**	0.5	0.814	82.0	5.0	13.0
NVIDIA-Nemotron-Nano-9B-v2-Base									
SemStamp (Hou et al., 2023)	96.4/98.0/99.0	40.4/54.8/87.4	27.9/44.2/78.4	61.7/71.6/90.6	14.3	0.700	52.4	44.8	2.8
k-SemStamp (Hou et al., 2024)	90.5/99.2/98.2	15.6/49.2/84.4	14.6/40.3/81.1	32.1/65.7/88.4	13.9	0.721	39.2	56.2	4.6
KGW (Kirchenbauer et al., 2023a)	100.0/100.0/100.0	55.9/80.3/96.0	27.3/49.9/88.6	56.2/80.6/96.2	0.2	0.957	39.6	56.0	4.4
SynthID (Dathathri et al., 2024)	100.0/100.0/100.0	42.9/59.9/84.1	22.6/35.0/69.0	53.0/69.3/87.6	0.0	0.904	54.8	41.4	3.8
MorphMark (Wang et al., 2025)	99.8/100.0/100.0	48.3/71.4/93.8	20.4/41.5/85.8	49.0/78.1/95.3	0.1	0.951	45.0	50.8	4.2
UPV (Liu et al., 2023)	99.7/100.0/99.8	75.0/**91.7**/87.9	53.2/79.2/77.7	79.5/**94.6**/88.0	0.2	0.923	42.2	53.2	4.6
EXP (Aaronson, 2023)	100.0/100.0/99.8	39.1/50.3/83.4	16.9/25.1/69.7	58.6/70.6/90.2	12.4	0.748	36.7	58.4	5.0
PMark (Huo et al., 2026)	96.4/97.8/98.8	24.7/39.7/78.9	28.3/40.0/79.3	45.0/57.4/86.1	9.6	0.728	40.0	55.0	5.0
SAMARK (Ours)	93.8/97.0/99.2	**78.3**/90.6/**97.2**	**72.9/89.2/97.1**	**82.3**/93.8/**97.8**	0.4	0.826	69.4	5.2	25.4

Table 11: Extension results for Gemma3-12B-PT. We report TP@FP1% under no attack and two Doc-P attacks. **Bold** = best and underlined = second-best within each dataset–attack column.

Method	BOOKSUM			C4			Reddit-TIFU-500		
	TP@FP1%			*TP@FP1%*			*TP@FP1%*		
	No Atk	Doc-P I	Doc-P II	No Atk	Doc-P I	Doc-P II	No Atk	Doc-P I	Doc-P II
KGW	**100.0**	60.8	25.2	99.8	38.4	11.6	**100.0**	0.6	15.2
SynthID	**100.0**	31.6	5.8	**100.0**	45.2	20.6	99.6	5.6	14.4
SemStamp	12.8	33.8	27.0	93.8	44.2	30.6	99.0	58.2	43.2
PMark	97.6	43.6	**79.6**	96.8	46.0	53.6	97.0	30.4	37.6
SAMARK	99.6	**89.2**	63.6	93.8	**75.6**	**73.2**	99.2	**87.6**	**84.0**

Table 12: Extension results on Reddit-TIFU-500. We report TP@FP1% under no attack and two document-level paraphrase attacks. **Bold** = best and underlined = second-best within each backbone–attack column.

Method	Mistral-Small-3.1-24B-Base-2503			NVIDIA-Nemotron-Nano-9B-v2-Base		
	TP@FP1%			*TP@FP1%*		
	No Atk	Doc-P I	Doc-P II	No Atk	Doc-P I	Doc-P II
KGW	**100.0**	54.2	29.2	**100.0**	60.0	32.2
SynthID	**100.0**	5.8	12.8	98.8	7.2	11.6
SemStamp	98.2	53.2	46.0	97.8	50.6	41.0
PMark	96.8	30.0	35.4	95.6	37.4	44.6
SAMARK	99.8	**82.6**	**80.8**	99.0	**82.8**	**81.0**

F.4 Word-Deletion Attack Results

In Figure 13, we illustrate the results of various watermark methods under Word-D attacks.

F.5 Additional Robustness–Quality Trade-offs

Figure 14 complements the Mistral+C4 view in Figure 5 with the other three backbone–dataset settings.

F.6 Diversity-aware Filtering Ablation

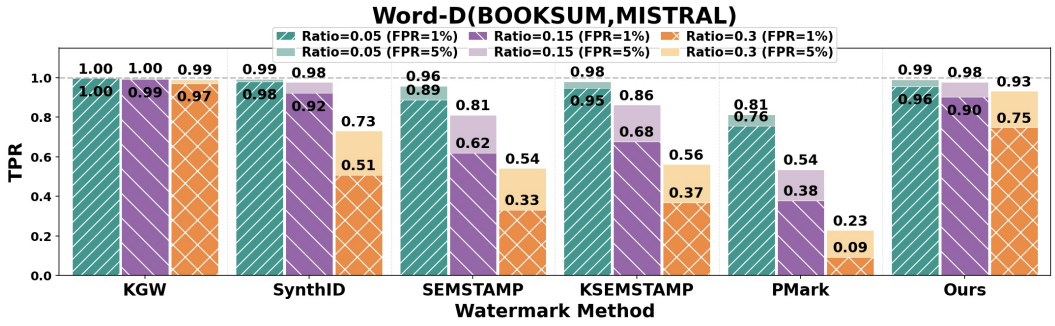
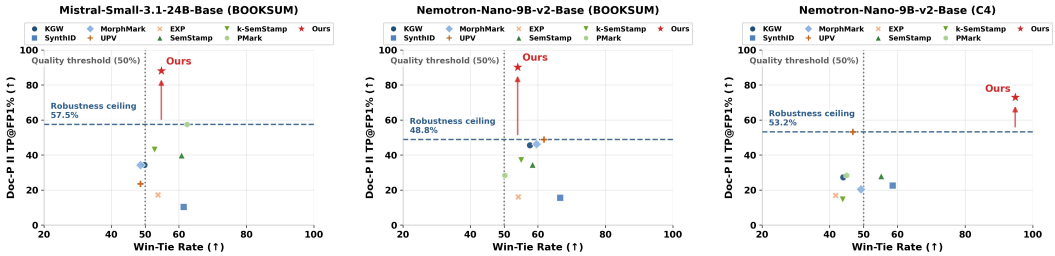


Figure 13: Word-D attack results.



(a) Mistral + BOOKSUM (b) Nemotron + BOOKSUM (c) Nemotron + C4

Figure 14: Additional robustness–quality trade-offs. The x-axis reports the PS-W + PS-T (no worse than) rate.

Table 13: Ablation study on diversity-aware filtering. Each row disables one component respectively.

Config	*Quality*			*LLM Judge*		
	SD%↓	D-2↑	4g%↓	PS-W	PS-L	PS-T
Default	0.27	0.825	4.96	47.9	45.2	6.9
w/o θ_{ngram}	1.35	0.792	8.31	48.8	47.5	3.8
w/o θ_{sem}	1.25	0.797	7.55	47.5	52.5	0.0
w/o λ_{div}	0.10	0.776	7.26	45.0	53.8	1.3
w/o λ_{nov}	0.71	0.762	8.70	46.3	47.5	6.3
All Off	7.65	0.570	29.11	33.0	59.0	8.0

Each diversity component contributes to text quality. Table 13 presents the ablation results where each diversity component is individually disabled. Removing the n-gram overlap filter raises the 4-gram repetition rate from 4.96% to 8.31% and the sentence duplicate rate from 0.27% to 1.35%. Disabling the semantic similarity filter produces a similar degradation, with the 4-gram repetition rate increasing to 7.55% and the sentence duplicate rate rising to 1.25%. Turning off the diversity weight λ_{div} reduces Distinct-2 from 0.825 to 0.776 and lowers the LLM judge win rate from 47.9% to 45.0%, while disabling the novelty weight λ_{nov} causes the largest drop in Distinct-2 to 0.762 and the highest 4-gram repetition rate among single-component ablations at 8.70%. When all four components are disabled simultaneously, the quality degradation compounds dramatically: the sentence duplicate rate surges to 7.65%, the 4-gram repetition rate reaches 29.11%, and the LLM judge win rate drops to 33.0%. These results confirm that the hard filtering and soft regularization mechanisms are complementary, each addressing a distinct aspect of text diversity.

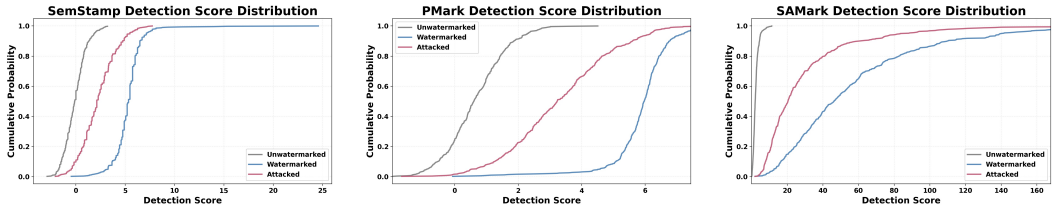


Figure 15: Detection score distributions for unwatermarked, watermarked, and attacked samples on BOOKSUM with Mistral-Small-3.1-24B-Base-2503.

F.7 Diversity Parameter Sensitivity

Table 14: Sensitivity to diversity parameters on Mistral+BOOKSUM. Default values are marked with $*$.

λ_{div}	SD%↓	D-2↑	4g%↓	W%	λ_{nov}	SD%↓	D-2↑	4g%↓	W%
0.0	0.10	0.78	7.26	45.0	0.0	0.71	0.76	8.70	46.3
0.1	0.45	0.78	7.67	47.5	0.1	0.61	0.78	7.10	48.8
0.35*	0.27	0.82	4.96	47.9	0.2*	0.27	0.82	4.96	47.9
0.7	0.83	0.83	5.16	57.5	0.3	0.31	0.84	4.37	51.3
1.0	0.62	0.83	5.40	60.0	0.5	0.10	0.85	3.27	56.3
θ_{sem}	SD%↓	D-2↑	4g%↓	W%	θ_{ngram}	SD%↓	D-2↑	4g%↓	W%
0.6	0.71	0.84	4.73	46.3	0.2	0.32	0.83	3.58	51.3
0.7	0.10	0.84	3.35	48.8	0.3	0.31	0.81	4.81	45.0
0.8*	0.27	0.82	4.96	47.9	0.4*	0.27	0.82	4.96	47.9
0.9	0.52	0.80	7.51	48.8	0.6	0.52	0.81	6.80	40.0
1.0	1.25	0.80	7.55	47.5	1.0	1.35	0.79	8.31	48.8

Soft regularization weights exhibit a clear quality improvement trend. As shown in Table 14, increasing λ_{div} from 0 to 1.0 steadily improves Distinct-2 from 0.78 to 0.83 and raises the LLM judge win rate from 45.0% to 60.0%, confirming that the semantic diversity bonus effectively encourages varied sentence selection. Similarly, increasing λ_{nov} from 0 to 0.5 reduces the 4-gram repetition rate from 8.70% to 3.27% and improves Distinct-2 from 0.76 to 0.85, demonstrating that the vocabulary novelty bonus successfully promotes lexical variety. Both parameters show monotonic quality gains without degrading detection performance, as the watermark scoring mechanism in Eq. 3 remains the dominant component in the final selection score.

Hard filtering thresholds control the quality floor. The semantic similarity threshold θ_{sem} and the n-gram overlap threshold θ_{ngram} act as hard constraints that remove overly similar candidates before scoring. Relaxing θ_{sem} from 0.7 to 1.0 increases the 4-gram repetition rate from 3.35% to 7.55% and raises the sentence duplicate rate from 0.10% to 1.25%, as more semantically redundant candidates enter the scoring pool. A similar pattern holds for θ_{ngram}: increasing it from 0.2 to 1.0 raises the 4-gram repetition rate from 3.58% to 8.31%. These results confirm that hard filtering provides a reliable quality floor by eliminating degenerate candidates early in the pipeline.

Mistral-Small-3.1-24B-Base-2503, BOOKSUM

Prompt: Dr. W. H. R. Rivers, a psychiatrist at a mental hospital in Craiglockhart, Scotland, silently reads a letter written by Siegfried Sassoon in July 1917.

Watermarked Text: The letter, addressed to the commander in chief of the British army, challenges the conduct of the war. Sassoon, a war poet, is at the hospital to recover from a self-inflicted wound. Rivers, a psychologist, is charged with curing Sassoon and returning …… **(Score: *9.72*, is_watermarked: True)**

NVIDIA-Nemotron-Nano-9B-v2-Base, C4

Prompt: Mark Twain sought solace and inspiration in the hills of Weggis, Switzerland.

Watermarked Text: He found a sense of calm and rejuvenation amidst the tranquil surroundings, which helped him recharge creatively. The serene environment of Weggis provided him with the space and solitude he needed to write and reflect. This retreat was not just a physical escape but also a mental and emotional …… **(Score: *4.56*, is_watermarked: True)**

Figure 16: Case study of text generated by SAMark.

F.8 Case Study

Score distribution analysis. Figure 15 shows the same trend at the distribution level. For SemStamp and PMark, the attacked score distribution shifts toward the unwatermarked distribution, which increases overlap and weakens threshold-based detection. For SAMark, attacked samples still remain clearly separated from unwatermarked samples, although their scores are lower than the original watermarked scores. This separation explains the higher robustness observed in the main results, where SAMark preserves a usable detection margin under aggressive paraphrasing. We also include certain specific cases in Figure 16 for reference.

G Additional Related Work

In recent years, LLM watermarking techniques have advanced rapidly, with KGW (Kirchenbauer et al., 2023a) being a seminal contribution that embeds statistically detectable signals into text by introducing logit biases to pseudorandomly selected tokens. Following the taxonomy adopted in prior work, existing approaches can be broadly categorized into three types: (1) zero-bit watermarking, (2) multi-bit watermarking, and (3) semantic-level watermarking. Our work falls within the third category, while drawing inspiration from the other two.

G.1 Zero-bit Watermarking

The primary objective of zero-bit watermarking is to determine whether a given text is machine-generated, with KGW (Kirchenbauer et al., 2023a) being the most representative. Subsequent work optimizes the KGW framework in both *detectability* and *text quality*.

Detectability. EWD (Lu et al., 2024) enhances sensitivity in low-entropy contexts (e.g., code) by prioritizing high-entropy tokens via entropy-based z-score weighting. WinMax (Kirchenbauer et al., 2023b) combats signal dilution in mixed texts using a sliding-window approach to dynamically compute multi-window z-scores and select maxima, boosting robustness. ITS and EXP (Kuditipudi et al., 2023) further enhance detectability by embedding a secret key sequence into tokens and computing alignment costs during detection to mitigate adversarial token manipulations.

Text Quality. Existing methods mitigate quality degradation by either embedding watermarks in high-entropy, low-semantic-impact positions using entropy-aware red-green list distributions (Wouters, 2023), optimizing list distributions via semantic alignment to disperse similar words across lists (Chen et al., 2024), or increasing context-relevant word selection probability (Fu et al., 2024). Beyond autoregressive LLMs, recent studies have also investigated order-agnostic watermarking for non-autoregressive paradigms (Chen et al., 2025; Wu et al., 2025).

G.2 Multi-bit Watermarking

Many applications require watermarks to convey richer information, such as user identifiers and timestamps, which zero-bit methods cannot support. Hence, another line of work focuses on embedding a multi-bit message (i.e., a binary string containing several bits). Existing solutions fall into two main categories: the first splits the vocabulary into distinct groups (like color categories) to encode data bits (Fernandez et al., 2023), and the second divides the text into chunks with each chunk holding part of the payload (Wang et al., 2023). However, when either method partitions too finely (e.g., too many groups or tiny chunks), the watermark becomes weak and hard to detect. To address this, Yoo et al. (Yoo et al., 2024) merge both approaches, avoiding extreme splitting while maintaining enough data capacity. More recently, Qu et al. (Qu et al., 2025) empirically observe that existing schemes suffer from either decoding efficiency or detection accuracy when encoding larger bit payloads, and propose a segmented construction to balance the two; Xu et al. (Xu et al., 2024) further refine payload robustness through majority-bit-aware aggregation. Another direction focuses on designing robust mechanisms that can trace content in settings with many adaptive users attempting to evade detection (Cohen et al., 2025), establishing rigorous statistical guarantees for ultra-low false positive rates while extending the mechanism to multi-bit message embedding.

G.3 Semantic-level Watermarking

The aforementioned methods operate primarily at the token level, making watermark signals vulnerable to adversarial rewriting such as paraphrasing. Sentence-level semantic watermarking (SWM) addresses this by embedding signals into sentence semantics, thereby offering robustness against lexical rewrites.

A representative work is SemStamp (Hou et al., 2023), which extends the idea of prefix-hashing from KGW into the sentence semantic space. Specifically, it employs locality-sensitive hashing (Indyk & Motwani, 1998) to compute a hash from the preceding sentence, which defines the "green" region for embedding the subsequent sentence. During detection, SemStamp counts the number of sentences whose embeddings fall within the corresponding green regions, conditioned on their preceding sentences, and leverages this statistic to perform a watermark-detection test. A follow-up, k-SemStamp (Hou et al., 2024), mitigates the quality degradation caused by SemStamp's rigid restriction to a single semantic partition by allowing sampling from the k nearest-neighbor partitions, thereby improving fluency and semantic preservation. Similarly, SemaMark (Ren et al., 2023) also relies on prefix hashing but adopts a Normalized Embedding Ring strategy. Beyond hash-based approaches, CoheMark (Zhang et al., 2025) and SimMark (Dabiriaghdam & Wang, 2025) explore more flexible watermarking criteria using fuzzy c-means clustering or sentence similarity, while PersonaMark (Zhang et al., 2024) hashes sentence structures for personalized embedding. PMark (Huo et al., 2026) addresses the distorted sentence distribution observed in prior methods and proposes a median-estimation-based approach to achieve distortion-free semantic-level watermarking with multi-channel constraints. Other recent efforts explore sparse autoencoders (Yu et al., 2026) and LLM-based paraphrasers (Xu et al., 2024) for sentence-level multi-bit embedding.

However, existing SWM methods inherit KGW's *prefix-based* design: the watermark signal in each sentence is pseudorandomly conditioned on its preceding context. Such dependence on sentence order makes them highly vulnerable to paragraph-level paraphrasing attacks that involve sentence reordering, splitting, or merging. Meanwhile, quality control in SWM remains underexplored, as simple n-gram filters fail to address semantic-level redundancy across sentences.

Our work falls within the domain of semantic-level watermarking but is inspired by token-level designs that couple watermarks with a global secret key sequence (Kuditipudi et al., 2023), a concept also explored in prior works including undetectable watermarks (Christ et al., 2024), bi-level signatures (Zhou et al., 2024), and publicly-detectable watermarks (Fairoze et al., 2023). These approaches either operate at the token level or fail to maintain robustness under structural or paragraph-level perturbations. Meanwhile, although both our method and existing multi-bit watermarking approaches embed multi-bit-like signals into text, our goal is not to transmit a specific payload across the entire document; instead, we propose a *self-anchored* semantic-level watermarking framework that severs the reliance on contextual order and incorporates diversity-aware filtering for quality preservation, pushing the Pareto frontier of the robustness-quality trade-off in text watermarking.