

CAD-Mark: Context-Aware Distributional Watermarking for Robust LLM-Generated Text Detection

Zhaoheng Huang¹, Dacheng Wen², Yutao Zhu^{1*},
Zhicheng Dou^{1*}, Ji-Rong Wen¹, and Fangzhao Wu^{3*}

¹Gaoling School of Artificial Intelligence, Renmin University of China

²School of Computing and Data Science, The University of Hong Kong

³Microsoft Research Asia

{huangzh, ytzhu, dou}@ruc.edu.cn, fangzwu@microsoft.com

Abstract

The increasing prevalence of texts generated by large language models (LLMs) drives a critical demand for reliable LLM text detection, where text watermarking has emerged as a promising proactive solution. Among existing methods, sentence-level watermarking is more robust to semantic-preserving adversarial attacks, but it still faces two key limitations: generation often requires iterative rejection sampling, and detection treats all sentence-level signals as equally reliable. To address these limitations, we propose CAD-Mark, a context-aware distributional sentence-level watermarking framework. During generation, CAD-Mark partitions a batch of candidate sentences with a context-derived adaptive boundary and samples the output sentence from the subset corresponding to the secret bits, embedding multiple bits per sentence without iterative rejection sampling. During detection, it estimates sentence-level reliability from re-sampled candidates and aggregates bit-level signals with a diversity-weighted z -test. Experiments on two benchmarks demonstrate that CAD-Mark consistently outperforms strong baselines in TPR@1%FPR while preserving generation quality with comparable cost.

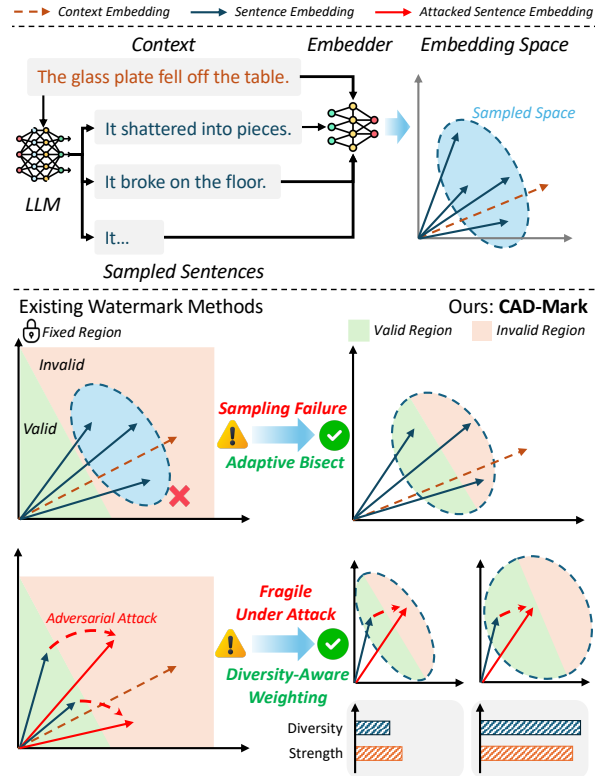


Figure 1: Existing methods use fixed boundaries, risking sampling failures and vulnerability to attacks. CAD-Mark resolves this through adaptive candidate bisection and diversity-aware weighting for enhanced robustness.

1 Introduction

The rapid advancement of large language models (LLMs) (Zhao et al., 2023) has made LLM-generated text increasingly prevalent, raising concerns about misinformation, copyright misuse, and academic dishonesty (Zhang et al., 2023; Karamolegkou et al., 2023). As a result, reliably detecting LLM-generated text has become an important problem. LLM text watermarking (Liu et al., 2025) provides a proactive solution by embedding traceable signals during generation, allowing generated content to be identified at the point of creation.

LLM watermarking methods have evolved from token-level to sentence-level paradigms. Early token-level methods typically use a secret key to partition the vocabulary into “green” and “red” token sets, and bias decoding toward “green” tokens (Kirchenbauer et al., 2023). The detector then identifies watermarked text by checking whether “green” tokens appear more often than expected by chance. However, this token-level intervention may degrade generation quality and is often vulnerable to semantic-preserving adversarial attacks such as paraphrasing, which can replace many tokens while keeping the original semantics.

*Corresponding authors.

This limitation has motivated a shift from token-level to sentence-level watermarking (SWM). Conceptually, SWM extends the green-red token partitioning idea from the vocabulary to the sentence semantic space. Instead of biasing the model toward “green” tokens, SWM samples multiple candidate sentences and selects one whose semantic representation falls into a target valid region in the semantic space, analogous to the “green” token set in token-level watermarking. Although SWM improves robustness against adversarial attacks, existing methods still rely on **fixed or heuristic semantic boundaries** to decide whether a candidate sentence is valid (Hou et al., 2024a,b; Huo et al., 2025; Aghdam and Wang, 2025). As shown in Figure 1, these constrained designs introduce limitations in both generation and detection.

First, on the generation side, fixed or heuristic semantic boundaries can cause inefficient iterative rejection sampling. Even a batch of candidates may contain no sentence in the target valid region, forcing repeated resampling and sometimes leading to failure within a fixed retry budget. This results in high and unstable generation cost. **Second, on the detection side**, fixed partition boundaries provide only a validity decision, without estimating how reliable the watermark signal is under the current context. Since different contexts induce candidate distributions with different semantic diversity, some watermark signals are robust while others are ambiguous and vulnerable to attacks such as paraphrasing and back-translation. Uniform aggregation then allows these vulnerable signals to reduce the final detection score.

To address these limitations, we propose CAD-Mark, a context-aware distributional sentence-level watermarking framework. During generation, CAD-Mark samples a batch of candidate sentences and uses the preceding context embedding to define an adaptive semantic boundary. Candidates are partitioned according to their relative similarity to the context, and the subset matching the secret bit is selected. By extending this context-based boundary into multiple orthogonal directions, CAD-Mark embeds multiple bits per sentence without iterative rejection sampling. During detection, CAD-Mark re-samples candidates under each context to reconstruct the local reference distribution and estimate sentence-level semantic diversity. It then aggregates bit verification results with a diversity-weighted z -test, emphasizing reliable sentence positions while down-weighting ambiguous signals.

We conduct experiments on two widely used benchmarks with three generators, comparing CAD-Mark against five token-level and four sentence-level baselines. Results show that CAD-Mark achieves superior robustness against adversarial attacks and consistently outperforms strong baselines in TPR@1%FPR, while preserving text quality and maintaining comparable token generation cost. The code and data are released at <https://github.com/WaldenRUC/CAD-Mark>.

Our contributions can be summarized as follows:

- We propose a context-aware sentence-level watermarking framework that embeds multiple bits per sentence by adaptively bisecting candidate distributions along context-derived orthogonal pivots, avoiding iterative rejection sampling.
- We design a diversity-weighted z -test that assigns larger weights to reliable sentence positions and down-weights ambiguous signals based on candidate distributional diversity, improving detection robustness against adversarial attacks.
- Experiments on two benchmarks and three generators show that CAD-Mark improves detection performance over baselines, while preserving text quality with comparable generation token cost.

2 Related Work

2.1 Token-Level LLM Text Watermark

Token-level watermarking embeds signals by modifying the logit distribution during generation. KGW (Kirchenbauer et al., 2023) partitions the vocabulary into “green” and “red” lists via hash values and boosts the logits of green-list tokens to create detectable statistical patterns. Subsequent works improve quality through high-entropy token selection (Lee et al., 2024; Lu et al., 2024), adaptive watermark strength (Wang et al., 2025; Liu and Bu, 2024), or semantic-aware logit adjustment (Liu et al., 2024b; Ren et al., 2024). However, token-level methods usually require access to LLM-predicted logits, making them less applicable to black-box LLMs. Furthermore, altering token logits may degrade the generation quality of LLMs, and these token-level signals remain vulnerable to semantic-preserving adversarial attacks such as paraphrasing.

2.2 Sentence-Level LLM Text Watermark

Sentence-level watermarking treats each complete sentence as the basic watermarking unit, and embeds signals in the semantic space. Sem-

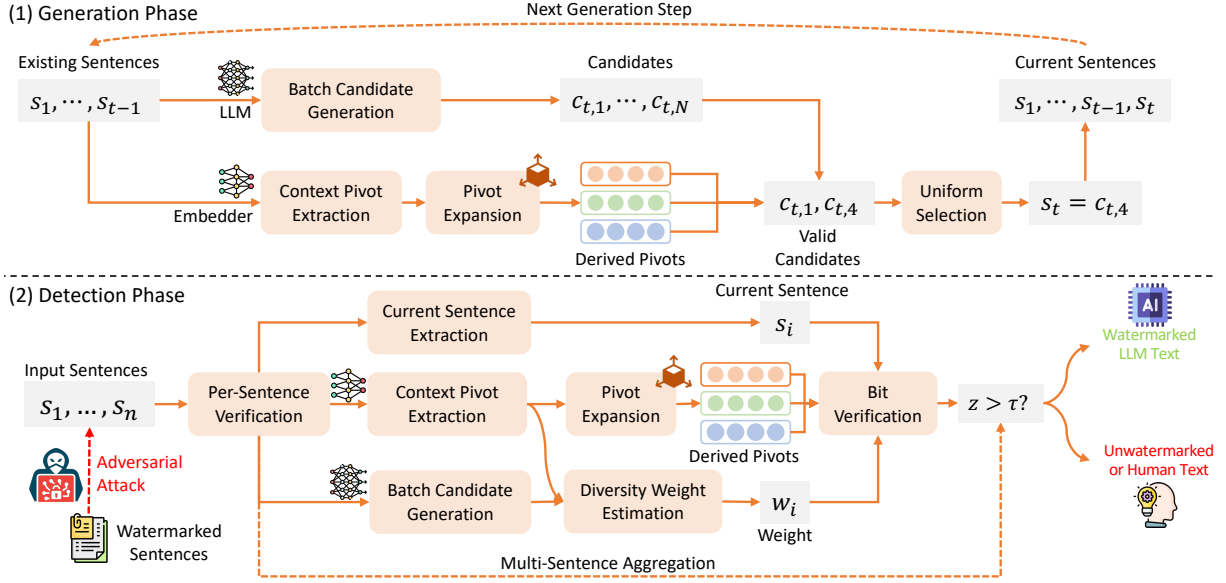


Figure 2: Overview of CAD-Mark. **Generation:** N candidate sentences from the LLM are sequentially bisected along k orthogonal pivots derived from the context $\mathbf{h}_t$, embedding k bits per sentence. **Detection:** Each sentence is verified against k bits. A diversity weight w_t measured from resampled candidates reweights the contribution of each sentence in a weighted z -test for robust detection.

Stamp (Hou et al., 2024a) and k-SemStamp (Hou et al., 2024b) use rejection sampling to map generated sentences into semantic regions defined by random signatures or pre-computed cluster centroids. However, these fixed semantic partitions can cause repeated resampling or sampling failures when no candidate falls into the target region, leading to high generation cost. Subsequent work (Huo et al., 2025) mitigates space distortion by utilizing orthogonal basis vectors, but it is not conditioned on the preceding context, ignoring the semantic coherence between the generated sentence and its context. Alternatively, SimMark (Aghdam and Wang, 2025) attempts to enforce similarity between the generated sentence and prior context within a fixed range, yet this scalar constraint still relies on a rigid boundary and does not account for the varying reliability of sentence-level signals during detection.

In contrast, CAD-Mark adaptively partitions the current candidate distribution using context-derived semantic boundaries and reweights sentence-level signals according to candidate diversity, improving robustness without iterative rejection sampling.

3 Methodology: CAD-Mark

3.1 Problem Formulation

Given a text $T = [s_1, s_2, \dots, s_n]$ consisting of n sentences, a sentence-level watermarking scheme

embeds $k \geq 1$ secret bits per sentence, yielding a total of nk binary observations $\{g_i\}_{i=1}^{nk}$, where $g_i = 1$ indicates that the i -th bit is successfully verified and considered *valid*, i.e., the sentence falls into the expected region of the semantic space according to the shared key and the context semantic.

Watermark detection is cast as a one-sided hypothesis test:

$$\begin{aligned} H_0 &: g_i \stackrel{\text{i.i.d.}}{\sim} \text{Bernoulli}(p), \quad i = 1, \dots, nk, \\ H_1 &: \mathbb{E}[g_i] > p. \end{aligned}$$

Under H_0 (unwatermarked text), each verification bit matches the secret bit only by chance. The null match probability p is determined by how the semantic partition divides the reference distribution. In our framework, the adaptive median partition is designed to balance the candidate distribution, so we use $p = \frac{1}{2}$ as the reference null probability.

Under H_1 (watermarked text), the generation process deliberately selects sentences that match the secret bits, yielding an agreement rate significantly above p .

3.2 Watermarked Text Generation

Like prior sentence-level methods (Hou et al., 2024b; Aghdam and Wang, 2025), we generate watermarked text sentence by sentence starting from a prompt sentence s_1 . However, instead of repeatedly sampling and rejecting individual sentences

until one meets a fixed constraint, at each step t we draw a single batch of N candidate sentences $\mathcal{C}_t = \{c_{t,1}, \dots, c_{t,N}\}$ from the LLM conditioned on $s_1, \dots, s_{t-1}$ and select from them directly. Each candidate $c_{t,j}$ is embedded by a general encoder $\mathcal{E}$ and normalized as $\mathbf{e}_{t,j} = \mathcal{E}(c_{t,j}) / \|\mathcal{E}(c_{t,j})\|$. We use the embedding of the w most recent sentences as a *context pivot* $\mathbf{h}_t$:

$$\mathbf{h}_t = \frac{\mathcal{E}([s_{t-w}; \dots; s_{t-1}])}{\|\mathcal{E}([s_{t-w}; \dots; s_{t-1}])\|}.$$

To strengthen the watermark signal, we embed k bits per sentence along k mutually orthogonal directions. The first pivot is defined as $\mathbf{p}_1 = \mathbf{h}_t$. For the remaining pivots, we use the secret key $\mathcal{K}$ as the seed of a deterministic pseudo-random generator to produce $k-1$ random vectors in the embedding space. Applying Gram–Schmidt orthogonalization to these vectors together with $\mathbf{p}_1$ yields $\mathbf{p}_2, \dots, \mathbf{p}_k$, ensuring $\mathbf{p}_i \perp \mathbf{p}_j$ for all $i \neq j$. Because the same key is shared by the generator and detector, these pivots can be reconstructed during detection.

We perform a sequential bisection to filter the candidates. Let $\mathcal{C}_t^{(0)} = \mathcal{C}_t$ be the initial candidate batch. For each bit $i \in \{1, \dots, k\}$, we project the surviving candidates from the previous step $\mathcal{C}_t^{(i-1)}$ onto the i -th pivot $\mathbf{p}_i$:

$$\sigma_{t,j}^{(i)} = \mathbf{e}_{t,j}^\top \mathbf{p}_i, \quad \forall c_{t,j} \in \mathcal{C}_t^{(i-1)}.$$

Let $\hat{\mu}_t^{(i)}$ be the median of these projections $\{\sigma_{t,j}^{(i)}\}$. The subset of candidates that satisfy the i -th secret bit $b_t^{(i)}$ is given by:

$$\mathcal{C}_t^{(i)} = \begin{cases} \{c_{t,j} \in \mathcal{C}_t^{(i-1)} \mid \sigma_{t,j}^{(i)} \geq \hat{\mu}_t^{(i)}\} & \text{if } b_t^{(i)} = 1, \\ \{c_{t,j} \in \mathcal{C}_t^{(i-1)} \mid \sigma_{t,j}^{(i)} < \hat{\mu}_t^{(i)}\} & \text{if } b_t^{(i)} = 0. \end{cases}$$

The final watermarked sentence s_t is then drawn uniformly from the final surviving set $\mathcal{C}_t^{(k)}$.

SimMark (Aghdam and Wang, 2025) is the most related prior sentence-level method, as it also uses the prior context to constrain sentence generation. However, it requires the sentence-context similarity to fall within a fixed interval, which depends on distributional assumptions and may lead to rejection failures. In contrast, our adaptive bisection computes the boundary from the current candidate batch, making the partition well-defined and retaining valid candidates when $N \geq 2^k$. Moreover, each median split balances the reference candidate distribution, so we use $p = \frac{1}{2}$ as the reference null probability in detection.

Algorithm 1 Watermarked Text Generation

Require: Prompt s_1 , LLM $\mathcal{M}$, encoder $\mathcal{E}$, secret bits $\{\mathbf{b}_t \in \{0, 1\}^k\}$, key $\mathcal{K}$, window w , batch size N
Ensure: Watermarked text $T = [s_1, s_2, \dots, s_n]$

- 1: **for** $t = 2, 3, \dots, n$ **do**
- 2: $\mathcal{C}_t \leftarrow \{c_{t,1}, \dots, c_{t,N}\}$ sampled from $\mathcal{M}(\cdot \mid s_1, \dots, s_{t-1})$
- 3: $\mathbf{e}_{t,j} \leftarrow \text{Normalize}(\mathcal{E}(c_{t,j}))$ for all $j \in \{1, \dots, N\}$
- 4: $\mathbf{h}_t \leftarrow \text{Normalize}(\mathcal{E}([s_{t-w}; \dots; s_{t-1}]))$
- 5: $(\mathbf{p}_1, \dots, \mathbf{p}_k) \leftarrow \text{ORTHOPIVOTS}(\mathbf{h}_t, k, \mathcal{K})$
- 6: $\mathcal{A} \leftarrow \mathcal{C}_t$ {Initialize surviving candidate set}
- 7: **for** $i = 1, \dots, k$ **do**
- 8: $\hat{\mu}_t^{(i)} \leftarrow \text{MEDIAN}(\{\mathbf{e}_{t,j}^\top \mathbf{p}_i \mid c_{t,j} \in \mathcal{A}\})$
- 9: $\mathcal{A} \leftarrow \{c_{t,j} \in \mathcal{A} \mid (\mathbf{e}_{t,j}^\top \mathbf{p}_i \geq \hat{\mu}_t^{(i)}) = b_t^{(i)}\}$
- 10: **end for**
- 11: $s_t \leftarrow \text{UNIFORMSAMPLE}(\mathcal{A})$; append s_t to T
- 12: **end for**
- 13: **return** T

Algorithm 2 Watermarked Text Detection

Require: Text T , LLM $\mathcal{M}$, encoder $\mathcal{E}$, secret bits $\{\mathbf{b}_t\}$, key $\mathcal{K}$, window w , batch N , threshold τ
Ensure: T is watermarked or not

- 1: **for** $t = 2, \dots, n$ **do**
- 2: $\mathcal{C}_t \leftarrow \text{Sample } N \text{ candidates}; \mathcal{A} \leftarrow \mathcal{C}_t$
- 3: $\mathbf{e}_{t,j} \leftarrow \text{Normalize}(\mathcal{E}(c_{t,j}))$ for all $j \in \{1, \dots, N\}$
- 4: $\mathbf{h}_t \leftarrow \text{Normalize}(\mathcal{E}([s_{t-w}; \dots; s_{t-1}]))$
- 5: $(\mathbf{p}_1, \dots, \mathbf{p}_k) \leftarrow \text{ORTHOPIVOTS}(\mathbf{h}_t, k, \mathcal{K})$
- 6: $w_t \leftarrow \text{Std}(\{\mathbf{e}_{t,j}^\top \mathbf{h}_t \mid c_{t,j} \in \mathcal{C}_t\})$ {Diversity weight}
- 7: **for** $i = 1, \dots, k$ **do**
- 8: $\hat{\mu}_t^{(i)} \leftarrow \text{MEDIAN}(\{\mathbf{e}_{t,j}^\top \mathbf{p}_i \mid c_{t,j} \in \mathcal{A}\})$
- 9: $\hat{g}_t^{(i)} \leftarrow \text{SOFTSCORE}(\mathbf{e}_{s_t}^\top \mathbf{p}_i - \hat{\mu}_t^{(i)}, b_t^{(i)})$
- 10: $\mathcal{A} \leftarrow \{c_{t,j} \in \mathcal{A} \mid (\mathbf{e}_{t,j}^\top \mathbf{p}_i \geq \hat{\mu}_t^{(i)}) = b_t^{(i)}\}$ {Recursive update}
- 11: **end for**
- 12: **end for**
- 13: $z \leftarrow \sum_{t,i} w_t (\hat{g}_t^{(i)} - 1/2) / \sqrt{\frac{1}{4} \sum_{t,i} w_t^2}$
- 14: **return** $z > \tau$

3.3 Watermarked Text Detection

Given an input text $T = [s_1, s_2, \dots, s_n]$, the detector is assumed to have access to the same generator $\mathcal{M}$, encoder $\mathcal{E}$, secret key $\mathcal{K}$, and deterministic seed with the generator. The detection proceeds in two stages: per-sentence verification and paragraph-level aggregation.

Per-sentence verification. For each sentence s_t , the detection phase reconstructs the generation context by (i) drawing N candidate sentences $\mathcal{C}_t$ from $\mathcal{M}$ conditioned on $s_1, \dots, s_{t-1}$, and (ii) rebuilding the k orthogonal pivots $\{\mathbf{p}_1, \dots, \mathbf{p}_k\}$ using the same context embedding and the shared seed determined by $\mathcal{K}$ as in generation. To align with the generation process, the detection phase performs the same sequential bisection on $\mathcal{C}_t$. For each bit $i \in \{1, \dots, k\}$, the detection phase identifies the median $\hat{\mu}_t^{(i)}$ from the projections of the current sur-

living candidate subset onto $\mathbf{p}_i$. The verification bit $g_t^{(i)}$ is then determined by checking whether the input sentence s_t falls on the side specified by the secret bit $b_t^{(i)}$:

$$g_t^{(i)} = \mathbb{1} \left[\left(\mathbf{e}_{s_t}^\top \mathbf{p}_i \geq \hat{\mu}_t^{(i)} \right) = b_t^{(i)} \right],$$

where $\mathbf{e}_{s_t} = \mathcal{E}(s_t) / \|\mathcal{E}(s_t)\|$. A value of $g_t^{(i)} = 1$ indicates that the sentence agrees with the secret bit on pivot i . We follow prior work on soft counting (Hou et al., 2024a; Aghdam and Wang, 2025; Huo et al., 2025) to mitigate the sensitivity of hard binary decisions near the partition boundary. Instead of the strict indicator $g_t^{(i)}$, we replace it with a soft score. Let $\delta_t^{(i)} = \mathbf{e}_{s_t}^\top \mathbf{p}_i - \hat{\mu}_t^{(i)}$, we define:

$$\tilde{g}_t^{(i)} = \begin{cases} 1, & (2b_t^{(i)} - 1) \delta_t^{(i)} \geq -\epsilon, \\ \exp(-\kappa |\delta_t^{(i)}|), & \text{otherwise,} \end{cases}$$

where ϵ is a small tolerance for numerical stability and κ controls the decay rate. A sentence on the correct side of the boundary receives full credit, while one on the wrong side receives an exponentially decaying score rather than 0, reducing variance in the paragraph-level statistic.

Diversity-aware weighting. Not all sentence positions carry equally reliable watermark signals. When the re-sampled candidates under a context are tightly clustered, the bisection is less discriminative and the resulting bit verification is more vulnerable to semantic perturbations. We therefore use the dispersion of candidate projections as a proxy for sentence-level signal reliability:

$$w_t = \text{Std} \left(\{ \mathbf{e}_{t,j}^\top \mathbf{h}_t \}_{j=1}^N \right).$$

A larger standard deviation indicates a more diverse candidate distribution and a more distinguishable watermark signal. The same weight is shared across all k bits of sentence t , since they come from the same context-conditioned candidate distribution. We empirically compare different proxy choices in Section 4.3.

Weighted z -test. Collecting all nk bit-level observations and their weights, the statistical detection z -score is:

$$z = \frac{\sum_{t,i} w_t \tilde{g}_t^{(i)} - \frac{1}{2} \sum_{t,i} w_t}{\frac{1}{2} \sqrt{\sum_{t,i} w_t^2}}.$$

Given a threshold τ , the text is classified as watermarked if $z > \tau$, and as human-written or unwatermarked otherwise. (Pseudocode is in Algorithm 2).

Theoretical analysis. We establish three properties of the weighted z -test (proofs in Appendix D):

- (i) **Scale invariance:** The z -score depends only on relative weight ratios, not on absolute magnitudes.
- (ii) **Asymptotic validity:** Under H_0 , the z -score converges to $\mathcal{N}(0, 1)$ as $n \rightarrow \infty$, provided no single weight dominates the total weight mass. This condition is mild in our setting because diversity weights are bounded and candidate distributions are non-degenerate in practice.
- (iii) **Optimality:** By the Cauchy–Schwarz inequality, the expected z -score is maximized when $w_t \propto |p_t - \frac{1}{2}|$, where p_t is the agreement probability at sentence t ; assigning large weights to weak signals strictly reduces detection power.

4 Experiments

4.1 Experimental Setup

Datasets Following Huo et al. (2025); Hou et al. (2024a); Wang et al. (2025), we use C4 (Raffel et al., 2019) and BookSum (Kryscinski et al., 2022) as benchmarks for evaluation. Each dataset contains 500 samples, and we follow prior work to use OPT-1.3B (Zhang et al., 2022) and Mistral-7B-v0.1 (Jiang et al., 2023) as backbone models to ensure a fair comparison with baselines. We also evaluate our approach on Qwen3-4B (Team, 2025).

Baselines We compare our proposed approach with various baselines, implemented by MarkLLM (Pan et al., 2024): **Token-level methods** include KGW (Kirchenbauer et al., 2023), UPV (Liu et al., 2024a), MorphMark (Wang et al., 2025), SIR (Liu et al., 2024b), SynthID (Dathathri et al., 2024). **Sentence-level methods** include SemStamp (Hou et al., 2024a), k-SemStamp (Hou et al., 2024b), SimMark (Aghdam and Wang, 2025), and PMark (Huo et al., 2025).

Metrics Following prior work (Huo et al., 2025), we evaluate watermarks in three dimensions:

Detectability. We use AUROC and $\text{TPR}@ \{1\%, 5\% \} \text{FPR}$ to measure watermark detection performance.

Quality. We evaluate text fluency via perplexity (PPL) using the associated LLM generator.

Robustness. We conduct various adversarial attacks using MarkLLM (Pan et al., 2024). Paraphrasing employs Dipper (Krishna et al., 2023) and GPT-4.1-nano (OpenAI, 2023). Back-translation (**Doc-T**) uses Qwen2.5-7B-Inst (Yang et al., 2024) via Spanish. Word-level attacks include deletion

Dataset	Method	No Attack	Word-D	Word-S	Doc-T	Doc-Dipper	Doc-GPT	Avg. Attack
<i>OPT-1.3B</i>								
C4	KGW	96.8 / 97.8 / 99.4	83.8 / 90.8 / 93.9	84.8 / 90.2 / 92.9	68.0 / 80.4 / 90.7	30.4 / 51.8 / 88.6	41.4 / 60.0 / 91.0	61.7 / 74.6 / 91.4
	UPV	81.0 / 89.6 / 95.4	77.8 / 86.6 / 94.2	76.0 / 85.4 / 94.5	58.4 / 68.6 / 86.7	17.6 / 41.4 / 80.0	38.4 / 64.0 / 87.4	53.6 / 69.2 / 88.6
	MorphMark	97.0 / 97.8 / 99.0	83.0 / 93.6 / 96.7	89.2 / 91.8 / 96.4	64.6 / 80.0 / 94.8	26.6 / 50.4 / 87.9	31.6 / 53.6 / 88.2	59.0 / 73.9 / 92.8
	SIR	96.6 / 97.2 / 98.6	87.6 / 95.2 / 98.6	91.2 / 93.4 / 96.9	78.8 / 86.4 / 96.6	24.0 / 42.2 / 77.4	35.6 / 55.0 / 75.4	63.4 / 74.4 / 89.0
	SynthID	99.8 / 100 / 100	93.2 / 99.0 / 99.6	94.4 / 96.8 / 99.4	71.8 / 84.2 / 94.6	24.8 / 40.0 / 73.8	32.2 / 51.6 / 81.0	63.3 / 74.3 / 89.7
	SemStamp	89.2 / 96.0 / 98.7	83.4 / 90.8 / 96.9	80.4 / 86.2 / 96.4	71.6 / 86.6 / 95.3	25.8 / 44.6 / 81.1	55.0 / 74.4 / 92.1	63.2 / 76.5 / 92.4
	k-SemStamp	85.8 / 94.2 / 98.1	82.6 / 85.8 / 94.9	81.4 / 87.2 / 94.4	65.8 / 74.2 / 93.2	28.8 / 46.6 / 82.3	58.8 / 73.2 / 90.0	63.5 / 73.4 / 91.0
	SimMark	85.0 / 90.2 / 94.4	84.8 / 86.2 / 92.5	84.2 / 88.2 / 93.4	71.6 / 77.4 / 91.5	37.8 / 52.4 / 83.6	59.2 / 74.6 / 93.1	67.5 / 75.8 / 90.8
	PMark	96.6 / 98.2 / 99.9	86.4 / 93.4 / 98.1	88.4 / 94.8 / 98.3	82.8 / 93.8 / 98.3	64.6 / 83.2 / 96.1	79.6 / 86.0 / 96.8	80.4 / 90.2 / 97.5
	CAD-Mark (Ours)	99.2 / 99.6 / 99.9	89.8 / 97.0 / 99.1	92.2 / 97.2 / 99.6	87.0 / 95.4 / 99.1	70.0 / 84.6 / 96.8	91.2 / 96.0 / 99.1	86.0 / 94.0 / 98.7
BookSum	KGW	95.8 / 98.2 / 99.4	86.8 / 91.2 / 95.9	87.2 / 93.0 / 96.0	69.8 / 86.8 / 96.8	39.0 / 73.2 / 93.0	37.4 / 70.8 / 92.0	64.0 / 83.0 / 94.7
	UPV	81.4 / 87.4 / 96.0	76.2 / 84.4 / 94.9	77.8 / 84.6 / 95.0	42.6 / 62.0 / 87.2	24.0 / 44.4 / 82.1	54.6 / 69.8 / 89.6	55.0 / 69.0 / 89.8
	MorphMark	95.8 / 98.2 / 99.2	91.0 / 95.0 / 98.7	90.8 / 94.8 / 98.5	64.6 / 83.4 / 95.2	39.0 / 71.6 / 92.8	30.8 / 63.0 / 90.7	63.2 / 81.6 / 95.2
	SIR	98.4 / 99.2 / 99.7	93.2 / 96.8 / 98.6	91.6 / 98.0 / 99.4	60.6 / 84.2 / 94.1	23.0 / 36.8 / 62.7	22.6 / 37.8 / 77.2	58.2 / 70.7 / 86.4
	SynthID	99.6 / 99.6 / 99.7	93.0 / 96.0 / 98.1	95.8 / 96.8 / 98.8	61.6 / 74.8 / 92.9	29.0 / 33.6 / 73.1	35.4 / 42.8 / 76.0	63.0 / 68.8 / 87.8
	SemStamp	98.4 / 99.0 / 99.8	78.2 / 89.6 / 94.8	84.0 / 94.4 / 98.6	68.4 / 84.4 / 96.2	34.6 / 56.4 / 89.3	66.4 / 81.8 / 96.2	66.3 / 81.3 / 95.0
	k-SemStamp	97.2 / 99.8 / 99.9	71.4 / 85.4 / 94.0	80.6 / 92.2 / 98.1	57.2 / 82.4 / 96.4	20.4 / 49.4 / 87.6	60.4 / 86.4 / 97.1	58.0 / 79.2 / 94.6
	SimMark	87.4 / 89.8 / 96.2	72.0 / 85.0 / 92.5	72.6 / 85.8 / 95.2	73.2 / 85.8 / 95.4	42.0 / 64.0 / 89.2	72.2 / 84.4 / 94.8	66.4 / 81.0 / 93.4
	PMark	98.8 / 99.0 / 99.3	86.0 / 92.8 / 96.2	92.0 / 95.2 / 96.3	81.8 / 92.6 / 97.7	64.8 / 83.6 / 95.9	76.6 / 85.4 / 95.8	80.2 / 89.9 / 96.4
	CAD-Mark (Ours)	99.4 / 99.6 / 99.9	90.8 / 95.6 / 98.7	94.8 / 98.6 / 99.6	92.6 / 97.8 / 99.3	84.0 / 91.2 / 97.4	92.0 / 96.4 / 99.3	90.8 / 95.9 / 98.9
<i>Mistral-7B</i>								
C4	KGW	100 / 100 / 100	91.6 / 94.6 / 97.0	94.0 / 96.0 / 97.9	71.8 / 92.4 / 98.4	32.8 / 60.8 / 91.9	73.6 / 88.2 / 93.9	72.8 / 86.4 / 95.8
	UPV	99.4 / 100 / 100	91.8 / 95.0 / 96.8	92.6 / 95.8 / 98.1	70.0 / 95.6 / 98.6	19.2 / 56.8 / 91.5	58.2 / 87.4 / 93.9	66.4 / 86.1 / 95.8
	MorphMark	99.8 / 100 / 100	93.0 / 97.0 / 99.9	92.2 / 96.8 / 98.3	64.2 / 85.0 / 97.2	26.4 / 55.8 / 90.4	52.0 / 76.2 / 95.7	65.6 / 82.2 / 96.3
	SIR	99.8 / 100 / 100	93.8 / 97.4 / 99.6	93.2 / 96.8 / 97.8	86.2 / 92.6 / 98.6	67.0 / 84.4 / 88.9	83.4 / 88.6 / 96.3	84.7 / 92.0 / 96.2
	SynthID	99.8 / 100 / 100	92.2 / 96.0 / 99.5	93.6 / 97.8 / 99.9	67.6 / 80.0 / 93.0	22.8 / 43.4 / 75.7	30.6 / 48.2 / 78.0	61.4 / 73.1 / 89.2
	SemStamp	93.0 / 97.8 / 99.5	83.4 / 92.2 / 95.5	87.8 / 91.2 / 96.2	51.6 / 70.0 / 91.6	23.4 / 44.8 / 79.5	43.6 / 62.4 / 88.4	58.0 / 72.1 / 90.2
	k-SemStamp	87.4 / 97.0 / 98.9	85.4 / 90.6 / 94.3	82.4 / 85.6 / 93.9	57.6 / 74.2 / 91.8	10.4 / 36.6 / 80.4	25.0 / 57.4 / 89.3	52.2 / 68.9 / 89.9
	SimMark	88.4 / 94.0 / 96.0	81.2 / 90.0 / 93.2	79.0 / 85.0 / 90.8	54.2 / 68.6 / 85.6	15.6 / 35.2 / 81.4	22.8 / 47.4 / 85.3	50.6 / 65.2 / 87.3
	PMark	99.0 / 99.4 / 99.8	89.2 / 92.8 / 96.7	90.6 / 94.8 / 97.9	85.8 / 93.0 / 98.6	60.0 / 77.2 / 94.7	82.4 / 90.2 / 96.5	81.6 / 89.6 / 96.9
	CAD-Mark (Ours)	100 / 100 / 100	93.8 / 96.4 / 99.4	95.6 / 98.2 / 99.7	88.4 / 94.6 / 99.0	79.2 / 91.8 / 98.2	89.8 / 96.0 / 99.2	89.4 / 95.4 / 99.1
BookSum	KGW	100 / 100 / 100	95.6 / 97.8 / 98.9	95.8 / 98.8 / 99.9	77.0 / 92.6 / 98.3	33.4 / 63.8 / 91.0	67.4 / 89.4 / 97.4	73.8 / 88.5 / 97.1
	UPV	99.4 / 100 / 100	95.8 / 98.0 / 99.2	96.6 / 99.4 / 99.9	86.2 / 95.0 / 98.9	35.6 / 61.6 / 93.1	71.0 / 89.2 / 97.8	77.0 / 88.6 / 97.8
	MorphMark	100 / 100 / 100	96.8 / 98.8 / 99.4	96.4 / 99.4 / 99.9	64.8 / 86.0 / 96.5	20.8 / 54.6 / 88.6	50.6 / 79.6 / 94.7	65.9 / 83.7 / 95.8
	SIR	99.8 / 100 / 100	97.0 / 98.6 / 99.2	95.8 / 99.2 / 99.8	89.8 / 95.2 / 98.5	77.8 / 88.6 / 96.7	79.4 / 91.6 / 98.4	88.0 / 94.6 / 98.5
	SynthID	99.4 / 99.6 / 99.8	96.6 / 98.4 / 99.7	95.2 / 99.2 / 99.8	55.2 / 73.6 / 93.5	11.2 / 29.4 / 74.7	17.2 / 36.4 / 79.2	55.1 / 67.4 / 89.4
	SemStamp	97.4 / 99.2 / 99.9	61.6 / 81.8 / 95.1	73.8 / 88.2 / 97.6	59.8 / 77.4 / 95.6	28.6 / 55.0 / 89.0	54.4 / 78.8 / 95.4	55.6 / 76.2 / 94.5
	k-SemStamp	96.2 / 99.4 / 99.7	52.8 / 76.6 / 93.1	66.2 / 88.6 / 97.4	47.2 / 77.0 / 95.1	22.8 / 51.4 / 88.6	53.0 / 84.4 / 96.7	48.4 / 75.6 / 94.2
	SimMark	78.4 / 86.0 / 95.4	73.2 / 78.8 / 88.2	50.4 / 72.8 / 92.2	51.6 / 73.2 / 92.7	40.0 / 65.8 / 90.8	52.0 / 72.2 / 92.1	53.4 / 72.6 / 91.2
	PMark	99.4 / 99.6 / 99.9	90.2 / 95.2 / 98.7	92.6 / 95.6 / 97.6	84.8 / 93.4 / 97.8	78.6 / 89.0 / 96.4	85.8 / 90.0 / 98.6	86.4 / 92.6 / 97.8
	CAD-Mark (Ours)	100 / 100 / 100	93.2 / 97.6 / 99.5	96.2 / 98.4 / 99.8	93.4 / 97.6 / 99.5	84.8 / 92.8 / 98.5	94.4 / 98.8 / 99.7	92.4 / 97.0 / 99.4

Table 1: Detection results against various attacks. All numbers in each cell are in percentages and correspond to **TPR@1%** / **TPR@5%** / **AUC**, respectively. The best results are in **bold** for each metric. Results for Qwen3-4B are provided in Appendix B.

(**Word-D**) and synonym substitution (**Word-S**) at the rate of 0.15. More implementation details can be found in Appendix A.

4.2 Overall Detection Performance

Table 1 summarizes the detection performance across two datasets and two generators under five types of adversarial attacks, and results on Qwen3-4B are provided in Appendix B. We find that:

(1) **Token-level methods** achieve strong performance without attack, with SynthID reaching near-perfect scores on most settings. However, their performance degrades sharply under adversarial attacks, and the document-level attacks, including Doc-T, Doc-Dipper, and Doc-GPT, cause the most severe drops. These attacks rewrite larger portions of the text while preserving the overall semantics, confirming that token-level signals are abundant but are fragile to attacks that alter the token sequence.

(2) **Existing sentence-level methods** show improved robustness under word-level attacks but remain limited under document-level attacks due to their rigid constraints. CAD-Mark consistently achieves the best average robustness across all settings. On OPT-1.3B, CAD-Mark improves the average attacked TPR@1% over the best baseline by +5.6 points on C4 and +10.6 points on BookSum. The gains are especially pronounced under the strongest document-level attacks. Meanwhile, CAD-Mark maintains near-best no-attack performance (99.2–100% TPR@1%), demonstrating that robustness is not achieved at the cost of detection sensitivity.

4.3 Effect of Diversity-Aware Weighting

We compare four weighting strategies based on the standard deviation weight: Uniform ($w_t = 1$), Std (w_t), Var (w_t^2), and Std³ (w_t^3). Figure 3 reports

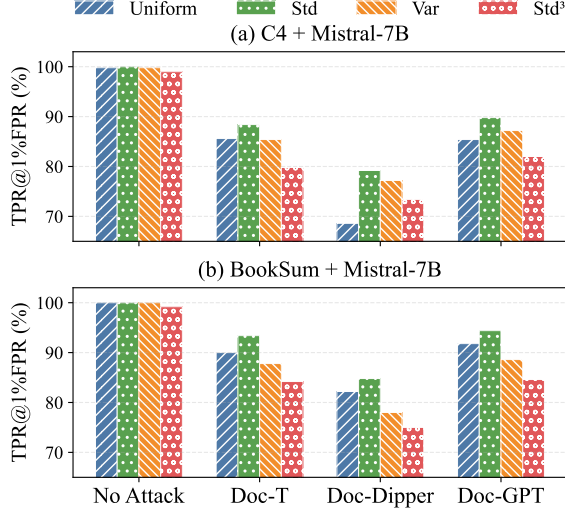


Figure 3: TPR@1%FPR under different weighting strategies. **Std weighting** achieves the best balance, and stronger powers over-amplify weight differences and hurt robustness.

TPR@1%FPR on Mistral-7B across C4 and BookSum under document-level attacks. We find that Std weighting consistently outperforms Uniform, with gains up to +10.6 points on Doc-Dipper in C4, showing that candidate diversity is an effective proxy for signal reliability. In contrast, stronger powers such as Var and Std³ degrade performance, suggesting that over-amplified weight differences become less aligned with the true signal strength. Under no attack, all strategies perform comparably ($\geq 99\%$), indicating that diversity-aware weighting mainly improves robustness without sacrificing no-attack setting detectability.

4.4 Text Quality Analysis

Figure 4 visualizes the trade-off between text quality (PPL) and detection robustness (TPR@1%FPR under Doc-Dipper attack) on the C4 dataset with Mistral-7B, where bubble size reflects PPL range across generated samples. More results are provided in Appendix C.

We observe that token-level methods suffer from noticeably higher PPL and larger quality variance due to logit manipulation. Among sentence-level baselines, PMark achieves competitive PPL but falls 19.2 points behind CAD-Mark in robust detection. CAD-Mark simultaneously achieves the lowest PPL and the highest TPR@1%FPR, demonstrating that our candidate batch selection strictly preserves the natural generated text distribution while the context-derived orthogonal pivots and diversity-

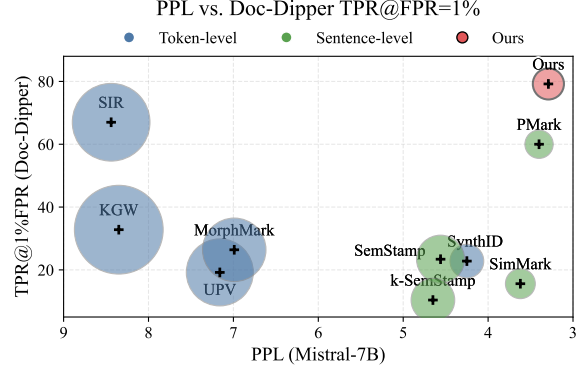


Figure 4: Trade-off between text quality (PPL) and robustness (TPR@1%FPR under Doc-Dipper) on Mistral-7B + C4. Bubble size reflects PPL range. **The upper-right corner is optimal** (low PPL and high TPR).

$k \backslash N$	16	32	64	128
1	29.4 / 52.8	40.6 / 65.6	37.2 / 68.0	36.0 / 60.8
2	39.6 / 58.8	60.2 / 84.4	60.4 / 85.6	65.0 / 86.8
3	67.0 / 84.6	73.6 / 89.2	84.0 / 92.0	80.6 / 91.2
4	71.0 / 83.2	76.6 / 91.2	85.0 / 91.6	79.8 / 92.6

Table 2: Dipper/GPT-attacked TPR@1%FPR (%) under varying k and N on the BookSum dataset with OPT-1.3B.

aware weighting effectively strengthen the watermark signal against adversarial attacks.

4.5 Hyperparameter Analysis

Table 2 studies the effect of the bits per sentence k and candidate pool size N on OPT-1.3B + BookSum under Doc-Dipper and GPT attacks.

We observe two primary findings. **First**, increasing k generally improves robustness by providing more bit-level observations for the weighted z -test, making detection less vulnerable to adversarial attacks like paraphrasing perturbations. **Second**, a larger N stabilizes median boundary estimation, especially when multiple bisections are applied with larger k . However, the gains diminish beyond $N = 64$. Since each bisection reduces the surviving candidate pool, k and N are coupled. We therefore set $\{k, N\} = \{3, 64\}$ by default, which provides strong robustness without requiring an excessively large candidate pool.

Effect of Context Window Size. The context window w controls how much prior context contributes to the context pivot $\mathbf{h}_t$. Figure 5 reports the effect of the context window size w on OPT-1.3B + BookSum with $k = 3$ and $N = 64$. The

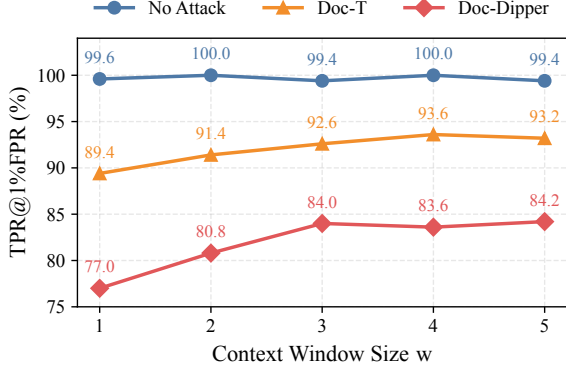


Figure 5: Effect of the context window size w on detection robustness (TPR@1%FPR) under different attacks.

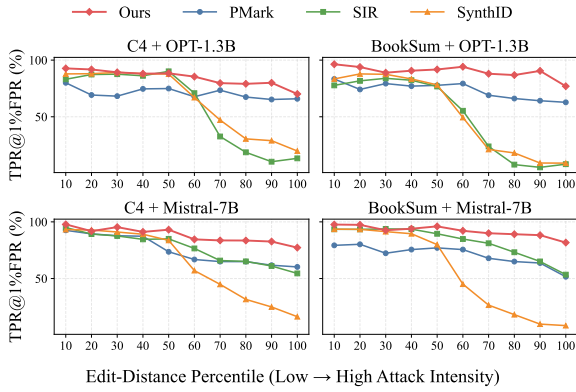


Figure 6: Detection robustness across varying attack intensities. CAD-Mark maintains superior stability and detectability as edit distance increases, outperforming both token- and sentence-level baselines.

no-attack performance remains consistently high across all values of w , showing that w has limited influence on original watermarked text detectability. Under attacks, increasing w from 1 to 3 clearly improves robustness by providing a more stable context pivot. When $w \geq 3$, performance becomes relatively stable, and the gains from longer contexts are marginal. Therefore, we set $w = 3$ as the default to balance robustness and context length.

4.6 Robustness under Attack Intensity

To analyze how detection robustness changes with attack strength, we sort attacked samples by their normalized character-level Levenshtein distance from the original watermarked text and divide them into ten equal-frequency bins. Each bin therefore contains the same number of samples with increasing perturbation intensity.

Figure 6 reports TPR@1%FPR within each bin under a per-method threshold calibrated on its own

Method	C4	BookSum
SemStamp	1,527.63	1,420.98
k-SemStamp	239.85	265.40
SimMark	197.25	179.42
PMark	336.42	309.01
CAD-Mark	321.84	313.58

Table 3: Average number of generated tokens required to produce one accepted watermarked sentence.

human samples. Across all four settings, CAD-Mark attains the highest TPR in almost every bin and shows the smoothest degradation as edit distance grows. In contrast, the token-level baselines SIR and SynthID are competitive only on lightly perturbed text and drop sharply as edit distance increases, since their token-level signals can be easily invalidated by adversarial attacks. This contrast highlights the reliability of sentence-level watermarking under realistic, intensity-varying attacks.

4.7 Generation Token Cost

Table 3 reports the average number of generated tokens required to produce one accepted watermarked sentence, directly measuring the generation-side cost of sentence-level watermarking. CAD-Mark uses less than one-fourth of the tokens required by SemStamp on both datasets, because it avoids fixed semantic boundaries that can trigger repeated rejection sampling when no candidate falls into the target region. Instead, CAD-Mark adaptively partitions the current candidate batch and selects a valid sentence directly. Meanwhile, CAD-Mark has comparable token consumption to PMark but improves average TPR@1%FPR under attack by up to 10.6 points, indicating that its robustness gains come from more effective watermark design rather than increased generation cost.

5 Conclusion

In this paper, we introduce CAD-Mark, a context-aware sentence-level watermarking framework for robust LLM-generated text detection. During generation, CAD-Mark embeds multiple bits per sentence via adaptive context-derived partitioning, avoiding iterative rejection sampling. During detection, it uses diversity-aware weighted aggregation to emphasize reliable signals and down-weight ambiguous ones in a weighted z -test. Experiments on two benchmarks and three generators show that CAD-Mark achieves strong robustness against adversarial attacks while preserving generation quality

with comparable generation token cost.

Limitations

We acknowledge some limitations of our study: (1) Reference generator access. CAD-Mark assumes access to the same or a compatible reference generator during detection, because it re-samples candidate continuations to reconstruct the local reference distribution. This setting is suitable for LLM providers auditing their own watermarked models, but may not apply when the generator is unknown or inaccessible. (2) Candidate sampling cost. Although CAD-Mark avoids iterative rejection sampling and the associated risk of sampling failures, it still relies on a candidate pool, which is common in sentence-level watermarking methods. Future work may investigate faster candidate generation or selection strategies to obtain valid watermarked sentences with fewer samples.

Acknowledgment

This work is supported by the National Natural Science Foundation of China No. 62402497 and No.62272467. The work is partially done at the Beijing Key Laboratory of Research on Large Models and Intelligent Governance.

References

- AmirHossein Dabiri Aghdam and Lele Wang. 2025. [Simmark: A robust sentence-level similarity-based watermarking algorithm for large language models](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 30785–30806. Association for Computational Linguistics.
- Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Posen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, A. Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, and 5 others. 2024. [Scalable watermarking for identifying large language model outputs](#). *Nat.*, 634(8035):818–823.
- Frank E Harrell and CE Davis. 1982. A new distribution-free quantile estimator. *Biometrika*, 69(3):635–640.
- Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024a. [Semstamp: A semantic watermark with paraphrastic robustness for text generation](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024*, pages 4067–4082. Association for Computational Linguistics.
- Abe Bohan Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. 2024b. [k-semstamp: A clustering-based semantic watermark for detection of machine-generated text](#). In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, volume ACL 2024 of *Findings of ACL*, pages 1706–1715. Association for Computational Linguistics.
- Jiahao Huo, Shuliang Liu, Bin Wang, Junyan Zhang, Yibo Yan, Aiwei Liu, Xuming Hu, and Mingxun Zhou. 2025. [Pmark: Towards robust and distortion-free semantic-level watermarking with channel constraints](#). *CoRR*, abs/2509.21057.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *CoRR*, abs/2310.06825.
- Antonia Karamolegkou, Jiaang Li, Li Zhou, and Anders S  gaard. 2023. [Copyright violations and large language models](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 7403–7412. Association for Computational Linguistics.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. [A watermark for large language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.
- Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. [Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Wojciech Kryscinski, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. 2022. [BOOKSUM: A collection of datasets for long-form narrative summarization](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6536–6558, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

- Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2024. [Who wrote this code? watermarking for code generation](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 4890–4911. Association for Computational Linguistics.
- Aiwei Liu, Leyi Pan, Xuming Hu, Shuang Li, Lijie Wen, Irwin King, and Philip S. Yu. 2024a. [An unforgeable publicly verifiable watermark for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024b. [A semantic invariant robust watermark for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip S. Yu. 2025. [A survey of text watermarking in the era of large language models](#). *ACM Comput. Surv.*, 57(2):47:1–47:36.
- Yepeng Liu and Yuheng Bu. 2024. [Adaptive text watermark for large language models](#). In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, volume 235 of *Proceedings of Machine Learning Research*, pages 30718–30737. PMLR / OpenReview.net.
- Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. 2024. [An entropy-based text watermarking detection method](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 11724–11735. Association for Computational Linguistics.
- OpenAI. 2023. [GPT-4 technical report](#). *CoRR*, abs/2303.08774.
- Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuan-dong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2024. [Markllm: An open-source toolkit for LLM watermarking](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: EMNLP 2024 - System Demonstrations, Miami, Florida, USA, November 12-16, 2024*, pages 61–71. Association for Computational Linguistics.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *CoRR*, abs/1910.10683.
- Nils Reimers and Iryna Gurevych. 2019. [Sentence-bert: Sentence embeddings using siamese bert-networks](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 3980–3990. Association for Computational Linguistics.
- Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. 2024. [A robust semantics-based watermark for large language model against paraphrasing](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 613–625, Mexico City, Mexico. Association for Computational Linguistics.
- Qwen Team. 2025. [Qwen3 technical report](#). *CoRR*, abs/2505.09388.
- Zongqi Wang, Tianle Gu, Baoyuan Wu, and Yujiu Yang. 2025. [Morphmark: Flexible adaptive watermarking for large language models](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 4842–4860. Association for Computational Linguistics.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona T. Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. [OPT: open pre-trained transformer language models](#). *CoRR*, abs/2205.01068.
- Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2023. [Siren’s song in the AI ocean: A survey on hallucination in large language models](#). *CoRR*, abs/2309.01219.
- Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2023. [A survey of large language models](#). *CoRR*, abs/2303.18223.

A Implementation Details

We conduct experiments using the Mark-LLM toolkit (Pan et al., 2024). We use the all-mpnet-base-v2 encoder (Reimers and

Gurevych, 2019) for sentence embeddings. For soft scoring, we follow PMark (Huo et al., 2025) and SimMark (Aghdam and Wang, 2025), setting $\kappa = 250$ and $\epsilon = 0.001$. For a fair comparison, we use the Harrell–Davis median estimator (Harrell and Davis, 1982) for robust median estimation and set the candidate pool size to $N = 64$ for both PMark and CAD-Mark. For watermarked text generation, we set the maximum number of new tokens to 256 and the maximum number of generated sentences to 12 for all methods. We sample with temperature 0.7 and top- p 0.95. For SimMark, we follow the original paper (Aghdam and Wang, 2025) and use the recommended validity interval $[0.68, 0.76]$.

B Additional Experimental Results

Table 4 reports additional results on Qwen3-4B. The results follow the same trend as the main experiments: CAD-Mark preserves strong no-attack detectability while achieving the best average robustness under adversarial attacks on both datasets. In particular, its advantage is more evident under document-level attacks, indicating that the proposed context-aware bisection and diversity-aware weighting generalize well to another LLM generator backbone.

C Text Quality Analysis

Figure 7, 8, and 9 provide additional trade-off results between text quality and detection robustness across more datasets and generators. The overall trend is consistent with Figure 4: token-level methods tend to have higher PPL due to logit manipulation, while sentence-level methods better preserve fluency. Among them, CAD-Mark achieves the strongest robustness while keeping PPL comparable to or lower than existing sentence-level baselines. This further confirms that the robustness gains of CAD-Mark do not come at the cost of generation quality.

D Theoretical Analysis

In this appendix, we provide detailed proofs for the theoretical properties of the diversity-aware weighted z -test presented in Section 3.3. We first introduce the notation and then prove scale invariance, asymptotic validity, and optimal weighting. We also discuss why orthogonal pivots provide approximately independent bit-level observations.

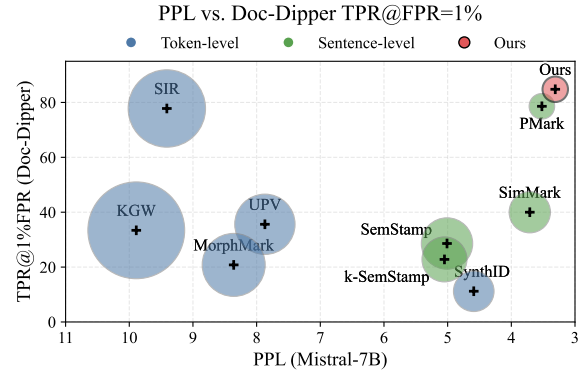


Figure 7: Trade-off between PPL and robustness (TPR@1%FPR under Doc-Dipper) on Mistral-7B + BookSum. Bubble size reflects PPL range. **The upper-right corner is optimal** (low PPL and high TPR).

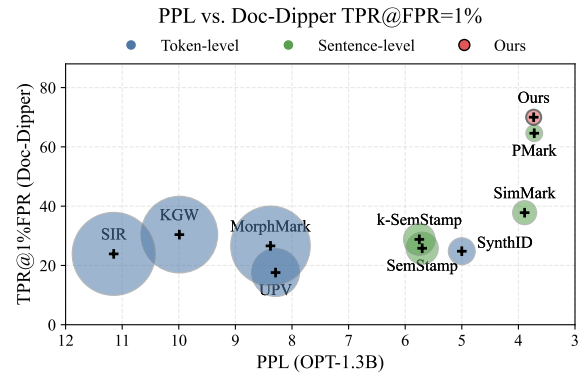


Figure 8: Trade-off between PPL and robustness (TPR@1%FPR under Doc-Dipper) on OPT-1.3B + C4. Bubble size reflects PPL range. **The upper-right corner is optimal** (low PPL and high TPR).

D.1 Notation and Setup

Consider a watermarked text $T = [s_1, s_2, \dots, s_n]$ with n sentences. At each sentence position t , the detector produces k bit-level observations $g_t^{(1)}, g_t^{(2)}, \dots, g_t^{(k)}$, where $g_t^{(i)} = 1$ indicates that sentence s_t agrees with the corresponding secret bit along pivot $\mathbf{p}_i$. Each observation is assigned a sentence-level weight $w_t > 0$, shared across all k bits of sentence t . Thus, there are nk bit-level observations in total.

For brevity, let $\sum_{t,i}$ denote $\sum_{t=1}^n \sum_{i=1}^k$. The weighted z -score is:

$$z = \frac{\sum_{t,i} w_t \left(g_t^{(i)} - \frac{1}{2} \right)}{\frac{1}{2} \sqrt{\sum_{t,i} w_t^2}}. \quad (1)$$

Under the null hypothesis H_0 , the text is not watermarked, so each bit agrees with the secret bit only by chance. We therefore assume

Dataset	Method	No Attack	Word-D	Word-S	Doc-T	Doc-Dipper	Doc-GPT	Avg. Attack
<i>Qwen3-4B</i>								
C4	KGW	97.4 / 98.4 / 99.4	93.2 / 97.4 / 99.1	95.6 / 97.0 / 99.3	81.4 / 90.2 / 96.1	46.0 / 69.6 / 92.9	51.6 / 69.4 / 91.9	73.6 / 84.7 / 95.9
	UPV	92.6 / 96.2 / 99.0	86.6 / 91.4 / 97.2	86.6 / 90.6 / 97.2	79.6 / 92.6 / 98.3	42.6 / 71.0 / 92.0	48.6 / 74.4 / 91.4	68.8 / 84.0 / 95.2
	MorphMark	95.6 / 97.6 / 99.4	91.6 / 95.6 / 99.0	93.0 / 96.6 / 98.9	77.6 / 88.0 / 97.0	37.8 / 61.4 / 90.0	50.4 / 70.6 / 91.7	70.1 / 82.4 / 95.3
	SIR	99.0 / 99.0 / 99.2	95.4 / 98.0 / 98.5	96.8 / 97.4 / 98.4	86.4 / 91.6 / 96.1	62.6 / 75.8 / 94.5	57.0 / 68.4 / 90.5	79.6 / 86.2 / 95.6
	SynthID	99.8 / 99.8 / 99.9	97.2 / 98.0 / 99.2	97.0 / 97.6 / 99.4	75.4 / 84.0 / 95.1	28.8 / 46.0 / 75.8	40.2 / 55.8 / 80.2	67.7 / 76.3 / 89.9
	SemStamp	94.8 / 97.2 / 99.5	80.0 / 88.2 / 94.6	77.8 / 86.0 / 96.5	65.0 / 78.2 / 94.5	55.2 / 73.2 / 91.6	53.0 / 71.6 / 91.5	66.2 / 79.4 / 93.7
	k-SemStamp	86.4 / 97.4 / 99.1	81.8 / 89.0 / 93.5	61.8 / 79.2 / 94.9	59.0 / 77.0 / 92.9	46.8 / 68.6 / 88.9	55.8 / 77.0 / 93.5	61.0 / 78.2 / 92.7
	SimMark	61.6 / 81.4 / 91.6	78.0 / 84.2 / 93.2	62.4 / 83.8 / 95.0	54.6 / 74.8 / 89.6	45.8 / 65.6 / 86.6	40.8 / 61.2 / 88.6	56.3 / 73.9 / 90.6
	PMark	95.2 / 96.2 / 99.4	85.6 / 91.6 / 97.4	90.8 / 95.0 / 98.4	80.8 / 88.2 / 97.5	64.2 / 78.8 / 95.5	80.2 / 85.6 / 95.8	80.3 / 87.8 / 96.9
	CAD-Mark (Ours)	98.2 / 99.6 / 99.9	95.0 / 97.6 / 99.4	96.2 / 97.8 / 99.4	87.4 / 92.8 / 98.1	70.8 / 82.0 / 96.1	84.0 / 89.8 / 97.6	86.7 / 92.0 / 98.1
BookSum	KGW	98.0 / 98.4 / 99.6	94.0 / 95.8 / 98.4	92.4 / 96.8 / 99.1	77.0 / 83.6 / 96.1	48.0 / 70.2 / 94.5	65.8 / 83.0 / 96.6	75.4 / 85.9 / 96.9
	UPV	96.6 / 98.0 / 99.6	93.4 / 96.8 / 98.9	93.0 / 96.2 / 98.8	82.4 / 90.8 / 97.5	61.2 / 81.6 / 95.2	63.6 / 85.4 / 96.1	78.7 / 90.2 / 97.3
	MorphMark	96.6 / 98.0 / 99.4	94.0 / 95.8 / 99.2	95.0 / 96.2 / 99.2	81.6 / 90.6 / 97.7	46.4 / 70.8 / 93.2	59.2 / 78.6 / 95.7	75.2 / 86.4 / 97.0
	SIR	97.6 / 100 / 99.9	89.8 / 96.6 / 99.3	88.2 / 96.8 / 99.3	61.8 / 85.8 / 96.1	37.4 / 63.8 / 90.5	48.0 / 80.8 / 95.1	65.0 / 84.8 / 96.1
	SynthID	99.8 / 100 / 100	93.4 / 96.6 / 99.1	95.0 / 98.6 / 99.5	55.8 / 79.2 / 94.7	6.2 / 23.8 / 73.2	15.0 / 37.4 / 79.9	53.1 / 67.1 / 89.3
	SemStamp	95.4 / 97.8 / 99.5	81.8 / 86.0 / 94.2	73.4 / 88.2 / 97.8	60.4 / 79.8 / 95.2	31.6 / 55.4 / 87.6	58.2 / 78.0 / 94.8	61.1 / 77.5 / 93.9
	k-SemStamp	94.2 / 98.6 / 99.8	76.2 / 80.6 / 93.2	66.8 / 89.2 / 97.9	59.8 / 74.8 / 95.0	21.6 / 48.6 / 88.1	56.4 / 82.4 / 96.1	56.2 / 75.1 / 94.1
	SimMark	82.0 / 86.8 / 93.1	67.8 / 77.6 / 85.3	68.6 / 82.8 / 89.2	51.8 / 71.6 / 86.4	29.6 / 51.4 / 81.6	43.0 / 60.4 / 86.0	52.2 / 68.8 / 85.7
	PMark	95.0 / 97.2 / 99.0	91.0 / 94.8 / 97.0	89.8 / 94.0 / 97.3	83.4 / 90.6 / 96.4	65.4 / 83.2 / 95.1	79.2 / 87.6 / 96.0	81.8 / 90.0 / 96.4
	CAD-Mark (Ours)	98.8 / 99.4 / 99.9	95.0 / 97.4 / 98.8	95.4 / 98.0 / 99.3	86.4 / 92.2 / 98.3	69.8 / 84.8 / 95.9	85.2 / 90.8 / 97.0	86.4 / 92.6 / 97.9

Table 4: Detection results against various attacks. All numbers in each cell are in percentages and correspond to **TPR@1%FPR / TPR@5%FPR / AUC**, respectively. The best results are in bold for each metric.

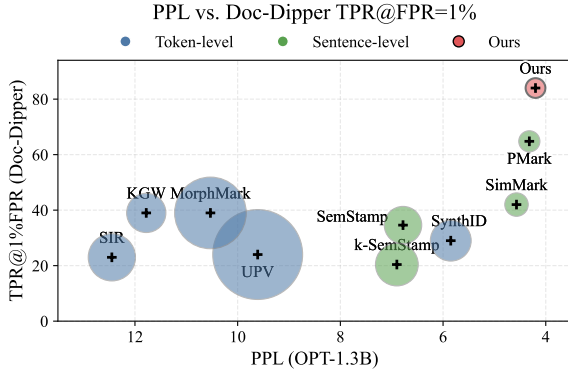


Figure 9: Trade-off between PPL and robustness (TPR@1%FPR under Doc-Dipper) on OPT-1.3B + BookSum. Bubble size reflects PPL range. **The upper-right corner is optimal** (low PPL and high TPR).

$$g_t^{(i)} \sim \text{Bernoulli}\left(\frac{1}{2}\right),$$

with $\mathbb{E}[g_t^{(i)}] = \frac{1}{2}$ and $\text{Var}(g_t^{(i)}) = \frac{1}{4}$. Under the alternative hypothesis H_1 , the text is watermarked. We denote the agreement probability at sentence position t by

$$p_t = \mathbb{E}[g_t^{(i)}] = \frac{1}{2} + \delta_t,$$

where $\delta_t \geq 0$ represents the signal strength at position t .

D.2 Scale Invariance

Proposition 1. *For any constant $c > 0$, replacing w_t with $c \cdot w_t$ yields the same z -score: $z(c \cdot \mathbf{w}) = z(\mathbf{w})$.*

Proof. Let $w'_t = c \cdot w_t$ for all t . Substituting into Eq. (1):

Numerator:

$$\begin{aligned} & \sum_{t,i} w'_t g_t^{(i)} - \frac{1}{2} \sum_{t,i} w'_t \\ &= c \sum_{t,i} w_t g_t^{(i)} - \frac{c}{2} \sum_{t,i} w_t \\ &= c \left(\sum_{t,i} w_t g_t^{(i)} - \frac{1}{2} \sum_{t,i} w_t \right). \end{aligned}$$

Denominator:

$$\frac{1}{2} \sqrt{\sum_{t,i} (w'_t)^2} = \frac{1}{2} \sqrt{c^2 \sum_{t,i} w_t^2} = \frac{c}{2} \sqrt{\sum_{t,i} w_t^2}.$$

Ratio:

$$\begin{aligned} z' &= \frac{c \left(\sum_{t,i} w_t g_t^{(i)} - \frac{1}{2} \sum_{t,i} w_t \right)}{\frac{c}{2} \sqrt{\sum_{t,i} w_t^2}} \\ &= \frac{\sum_{t,i} w_t g_t^{(i)} - \frac{1}{2} \sum_{t,i} w_t}{\frac{1}{2} \sqrt{\sum_{t,i} w_t^2}} = z. \end{aligned}$$

Conclusion. The z -score depends only on the relative ratios w_{t_1}/w_{t_2} among sentences within the same text. Global rescaling has no effect. This ensures that the detection outcome is consistent across texts whose candidate distributions have different absolute scales of diversity, and that the weighting scheme does not require cross-sample calibration. $\square$

D.3 Asymptotic Validity

Proposition 2. Under H_0 , the weighted z -score converges in distribution to $\mathcal{N}(0, 1)$ as $n \rightarrow \infty$, provided the weights satisfy the Feller condition:

$$\frac{\max_{1 \leq t \leq n} w_t^2}{\sum_{t=1}^n w_t^2} \xrightarrow{n \rightarrow \infty} 0. \quad (2)$$

Proof. Under H_0 , define the centered random variables:

$$X_{t,i} = w_t \left(g_t^{(i)} - \frac{1}{2} \right).$$

Since the bit-level observations are independent under H_0 , the variables $X_{t,i}$ are also independent and satisfy:

$$\begin{aligned} \mathbb{E}[X_{t,i}] &= 0, \\ \text{Var}(X_{t,i}) &= w_t^2 \cdot \text{Var}(g_t^{(i)}) = \frac{1}{4} w_t^2. \end{aligned}$$

The z -score can be written as:

$$z = \frac{\sum_{t=1}^n \sum_{i=1}^k X_{t,i}}{\sqrt{\sum_{t=1}^n \sum_{i=1}^k \text{Var}(X_{t,i})}} = \frac{S_{nk}}{\sqrt{V_{nk}}},$$

where $S_{nk} = \sum_{t=1}^n \sum_{i=1}^k X_{t,i}$ and $V_{nk} = \sum_{t=1}^n \sum_{i=1}^k \text{Var}(X_{t,i}) = \frac{k}{4} \sum_{t=1}^n w_t^2$.

By the Lindeberg–Feller CLT, it suffices to show that for every $\epsilon > 0$:

$$\frac{1}{V_{nk}} \sum_{t=1}^n \sum_{i=1}^k \mathbb{E} \left[X_{t,i}^2 \mathbb{1} \left\{ |X_{t,i}| > \epsilon \sqrt{V_{nk}} \right\} \right] \xrightarrow{n \rightarrow \infty} 0. \quad (3)$$

Since $g_t^{(i)} \in \{0, 1\}$, we have:

$$|X_{t,i}| = \frac{1}{2} w_t.$$

Therefore, the indicator in Eq. (3) can be non-zero only when:

$$\frac{1}{2} w_t > \epsilon \sqrt{V_{nk}} = \epsilon \cdot \frac{\sqrt{k}}{2} \sqrt{\sum_{\ell=1}^n w_\ell^2},$$

i.e.,

$$w_t > \epsilon \sqrt{k \sum_{\ell=1}^n w_\ell^2}.$$

However, the Feller condition implies:

$$\frac{\max_t w_t}{\sqrt{\sum_{\ell=1}^n w_\ell^2}} \rightarrow 0.$$

Thus, for sufficiently large n , no w_t can satisfy the above inequality for any fixed $\epsilon > 0$. Hence the indicator term is eventually zero for all t and i , and the Lindeberg condition holds. Therefore,

$$z = \frac{S_{nk}}{\sqrt{V_{nk}}} \xrightarrow{d} \mathcal{N}(0, 1) \quad \text{under } H_0.$$

In CAD-Mark, the weights are bounded because they are computed as the standard deviation of cosine similarities:

$$w_t = \text{Std} \left(\{ \mathbf{e}_{t,j}^\top \mathbf{h}_t \}_{j=1}^N \right), \quad \mathbf{e}_{t,j}^\top \mathbf{h}_t \in [-1, 1].$$

Therefore, $0 \leq w_t \leq 1$. The Feller condition holds as long as the candidate distributions are non-degenerate and no single sentence position dominates the total weight mass, which is mild in practice. $\square$

D.4 Optimality

Proposition 3. Under H_1 , $\mathbb{E}[z]$ is maximized when $w_t \propto \delta_t$, where $\delta_t = p_t - \frac{1}{2}$ is the signal strength at sentence t .

Proof. Under H_1 , the expected value of each observation is $\mathbb{E}[g_t^{(i)}] = \frac{1}{2} + \delta_t$. We compute the expected numerator and denominator of z separately.

Step 1: Expected numerator.

$$\begin{aligned} & \mathbb{E} \left[\sum_{t,i} w_t g_t^{(i)} - \frac{1}{2} \sum_{t,i} w_t \right] \\ &= \sum_{t,i} w_t \left(\mathbb{E}[g_t^{(i)}] - \frac{1}{2} \right) \\ &= k \sum_{t=1}^n w_t \delta_t. \end{aligned}$$

Step 2: Denominator (deterministic). The denominator is:

$$\frac{1}{2} \sqrt{\sum_{t=1}^n k w_t^2} = \frac{\sqrt{k}}{2} \sqrt{\sum_{t=1}^n w_t^2}.$$

Step 3: Expected z -score. Combining Steps 1 and 2:

$$\mathbb{E}[z] = \frac{k \sum_{t=1}^n w_t \delta_t}{\frac{\sqrt{k}}{2} \sqrt{\sum_{t=1}^n w_t^2}} = 2\sqrt{k} \cdot \frac{\sum_{t=1}^n w_t \delta_t}{\sqrt{\sum_{t=1}^n w_t^2}}. \quad (4)$$

Step 4: Apply Cauchy–Schwarz. Define vectors $\mathbf{w} = (w_1, \dots, w_n)$ and $\boldsymbol{\delta} = (\delta_1, \dots, \delta_n)$. By the Cauchy–Schwarz inequality:

$$\sum_{t=1}^n w_t \delta_t \leq \sqrt{\sum_{t=1}^n w_t^2} \cdot \sqrt{\sum_{t=1}^n \delta_t^2}$$

with equality if and only if $w_t = c \cdot \delta_t$ for some constant $c > 0$.

Substituting into Eq. (4):

$$\mathbb{E}[z] \leq 2\sqrt{k} \cdot \sqrt{\sum_{t=1}^n \delta_t^2} \quad (5)$$

with equality when $w_t = c \cdot \delta_t$.

Step 5: Consequence of misaligned weights.

Consider a sentence with weak signal ($\delta_t \approx 0$) assigned a large weight $w_t \gg 0$. This inflates $\sum_t w_t^2$ in the denominator without a proportional increase in $\sum_t w_t \delta_t$ in the numerator, strictly decreasing $\mathbb{E}[z]$ relative to the optimal assignment.

To see this concretely, consider two sentences with $\delta_1 > 0$ and $\delta_2 = 0$. Under uniform weighting $w_1 = w_2 = 1$:

$$\mathbb{E}[z]_{\text{uniform}} = 2\sqrt{k} \cdot \frac{\delta_1}{\sqrt{2}}.$$

Under optimal weighting $w_1 = \delta_1, w_2 = 0$:

$$\mathbb{E}[z]_{\text{optimal}} = 2\sqrt{k} \cdot \frac{\delta_1^2}{\delta_1} = 2\sqrt{k} \cdot \delta_1.$$

Thus $\mathbb{E}[z]_{\text{optimal}}/\mathbb{E}[z]_{\text{uniform}} = \sqrt{2} > 1$.

Conclusion. The expected detection z -score is maximized when sentence weights are proportional to their signal strengths. This result motivates our use of candidate diversity as a practical proxy for the unknown signal strength δ_t . We instantiate this proxy with $w_t = \text{Std}(\{\mathbf{e}_{t,j}^\top \mathbf{h}_t\}_{j=1}^N)$ and validate it empirically in Section 4.3. $\square$

D.5 Approximate Independence of Orthogonal Bits

Proposition 4. *If the pivots $\mathbf{p}_1, \dots, \mathbf{p}_k$ are mutually orthogonal and the covariance structure of candidate embeddings is approximately balanced across these pivot directions, then the projected similarities along different pivots are approximately uncorrelated.*

Proof. Let $\mathbf{e}_j \in \mathbb{R}^d$ be the normalized embedding of candidate sentence c_j . The projected similarity along pivot $\mathbf{p}_i$ is

$$\sigma_j^{(i)} = \mathbf{e}_j^\top \mathbf{p}_i.$$

For two distinct pivots $\mathbf{p}_i$ and $\mathbf{p}_\ell$ with $i \neq \ell$, the covariance between their projected similarities is

$$\begin{aligned} \text{Cov}(\sigma_j^{(i)}, \sigma_j^{(\ell)}) &= \text{Cov}(\mathbf{e}_j^\top \mathbf{p}_i, \mathbf{e}_j^\top \mathbf{p}_\ell) \\ &= \mathbf{p}_i^\top \boldsymbol{\Sigma} \mathbf{p}_\ell, \end{aligned}$$

where $\boldsymbol{\Sigma} = \text{Cov}(\mathbf{e}_j)$ is the covariance matrix of candidate embeddings.

When the candidate embedding distribution is approximately isotropic in the subspace spanned by the pivots, we have $\boldsymbol{\Sigma} \approx \lambda \mathbf{I}$ within this subspace. Therefore,

$$\mathbf{p}_i^\top \boldsymbol{\Sigma} \mathbf{p}_\ell \approx \lambda \mathbf{p}_i^\top \mathbf{p}_\ell = 0,$$

because $\mathbf{p}_i \perp \mathbf{p}_\ell$ by construction. Hence, the projected similarities along different orthogonal pivots are approximately uncorrelated.

The bit-level observation $g_t^{(i)}$ is obtained by comparing the projected similarity of sentence s_t with the median boundary along pivot $\mathbf{p}_i$:

$$g_t^{(i)} = \mathbb{1} \left[\left(\mathbf{e}_{s_t}^\top \mathbf{p}_i \geq \hat{\mu}_t^{(i)} \right) = b_t^{(i)} \right].$$

If the projected similarities are approximately jointly Gaussian, then approximate uncorrelatedness further implies approximate independence. Therefore, the bit-level observations produced by different orthogonal pivots can be treated as approximately independent.

Conclusion. Orthogonal pivots reduce redundancy among the k bit-level signals within each sentence. Under approximate isotropy and Gaussianity of the candidate embedding distribution, these signals provide approximately independent observations, increasing the effective number of observations from n to about nk . $\square$

E Use of AI assistants

We use ChatGPT to assist with writing clarity and presentation.¹

¹<https://chatgpt.com/>