

HSPMark: Desynchronization-Resilient Multi-Bit LLM Watermarking via Semantic Anchoring

Jing Wang¹, Pengyu Xiong^{1*}, Kejun Qian¹, Yi Yang², Jing Fan³

¹Huazhong University of Science and Technology

²Naval University of Engineering

³Dongfeng Motor Corporation Research and

Development Institute, Dongfeng Motor Group Company Limited

{cswjing, xpyu, qkjun}@hust.edu.cn, yangyi.ip@whu.edu.cn, fanjing@dfmc.com.cn

Abstract

Reliable multi-bit watermarking is essential for tracing the provenance of large language model (LLM) outputs. However, existing high-capacity schemes typically synchronize pseudo-random functions through discrete token n -gram matching, which makes them vulnerable to desynchronization under semantics-preserving paraphrasing attacks. To address this issue, we propose HSPMark, a hierarchical semantic projection marking mechanism for context-aware multi-bit watermarking, which shifts watermark synchronization from brittle token histories to continuous semantic anchors. HSPMark extracts stable contextual features using a pre-trained sentence-level encoder and a lightweight Robust Semantic MLP, then projects these features into context-conditioned pseudo-orthogonal bias matrices for multi-bit payload injection. To balance payload capacity and generation quality, HSPMark introduces an entropy-adaptive embedding mechanism that modulates logit bias according to predictive uncertainty. Extensive evaluations demonstrate that HSPMark improves robustness against paraphrase-induced desynchronization and local perturbations while maintaining a competitive trade-off among bit accuracy, payload capacity, and text quality compared with existing high-capacity baselines.

1 Introduction

Large Language Models (LLMs) increasingly require robust provenance tracking via multi-bit watermarking, which embeds identifiable metadata (e.g., user IDs) directly into generated text (Zellers et al., 2019; Kirchenbauer et al., 2023; Wang et al., 2024). However, current multi-bit frameworks can suffer substantial degradation under semantics-preserving evasion attacks, such as deep paraphrasing (Krishna et al., 2023; Xu et al., 2025). These tools structurally reorganize text while preserving

meaning, exposing a critical flaw in state-of-the-art schemes (Yoo et al., 2024; Qu et al., 2025; Park et al., 2026): their reliance on discrete, exact token matching. Paraphrasing alters the n -gram histories used to seed a pseudo-random function (PRF), irrecoverably corrupting the watermark state. This fatal desynchronization causes bit extraction accuracy to plummet to random guessing.

While recent zero-bit methods (Liu et al., 2024; Ren et al., 2024) use continuous semantic embeddings to resist structural edits, adapting this to multi-bit payloads is highly challenging. Multi-bit schemes require a large, separable injection space, and applying multi-dimensional logit biases can substantially increase text perplexity (PPL). To bridge this gap, we propose Hierarchical Semantic Projection Marking (HSPMark), a context-aware multi-bit watermarking framework designed to improve robustness against structural rewriting. HSPMark abandons brittle token histories, instead mapping preceding context into a continuous space via a Robust Semantic Multi-Layer Perceptron (MLP). To embed complex payloads without mutual interference, these robust features are projected into dynamically generated Pseudo-Orthogonal Bias Matrices. This design anchors the watermark state to sentence-level semantic context, making state reconstruction less sensitive to surface-level syntactic variation.

Specifically, to support high-capacity embedding without compromising generation quality, HSPMark incorporates an entropy-adaptive scaling mechanism. By reducing payload injection under high-confidence decoding steps and concentrating it in high-entropy regions, our framework preserves linguistic fluency while maintaining reliable bit-stream synchronization.

The main contributions of this paper are summarized as follows:

- We analyze paraphrase-induced desynchro-

*Corresponding author.

nization in n -gram based multi-bit watermarking, showing that exact token-history matching is a major source of payload recovery failure under semantic-preserving rewrites.

- We propose HSPMark, a context-aware multi-bit watermarking framework that replaces brittle token-history synchronization with hierarchical semantic projection, successfully integrating semantic synchronization with multi-bit payload recovery.
- We evaluate HSPMark under deep paraphrasing, local token perturbations, different payload sizes, and downstream generation tasks. Experimental results demonstrate that HSPMark improves robustness against paraphrase attacks while maintaining a favorable balance between payload recovery and text quality.

2 Related Work

2.1 LLM Watermarking for Provenance

Watermarking has become one of the main technical routes for tracing text generated by LLMs. Unlike post-hoc detectors, which classify a completed passage after generation, watermarking modifies the decoding distribution so that the generated text carries a verifiable statistical signal. [Kirchenbauer et al. \(2023\)](#) introduced a seminal logit-based watermarking scheme in which recent token history seeds a pseudo-random function that partitions the vocabulary, and decoding is biased toward one partition to create a detectable statistical signal.

Later studies refine this paradigm from different perspectives, including reliability analysis ([Kirchenbauer et al., 2024](#)), robustness under bounded edits ([Zhao et al., 2024](#)), distribution-aware watermark construction ([Fernandez et al., 2023](#)), scalable deployment ([Dathathri et al., 2024](#)), unbiased watermarking ([Hu et al., 2024](#); [Feng et al., 2025](#)), distortion-free sampling ([Kuditipudi et al., 2024](#)), and cryptographic or publicly verifiable formulations ([Christ et al., 2024](#); [Fairoze et al., 2025](#)). These works establish logit-based watermarking as a practical provenance mechanism, but most of them are designed for zero-bit verification: the detector decides whether a text is watermarked, rather than recovering an explicit payload.

2.2 Multi-bit Watermarking and Payload Recovery

Multi-bit watermarking extends watermarking from binary verification to traceable payload recovery. Instead of only proving that a passage was generated by a protected model, a multi-bit watermark can encode user identifiers, model versions, timestamps, or other generation-specific metadata. Recent work has explored this direction by mapping message bits to vocabulary partitions, token ranks, or structured decoding rules. Codable watermarking ([Wang et al., 2024](#)) studies how multi-bit information can be injected into LLM outputs. MPAC ([Yoo et al., 2024](#)) improves payload capacity through message-aware token partitioning. StealthInk ([Jiang et al., 2025](#)) aims to preserve stealthiness while embedding multiple bits. ECC ([Qu et al., 2025](#)) strengthens multi-bit recovery with structured block mapping and coding mechanisms, and WaterMod ([Park et al., 2026](#)) uses modular token-rank partitioning to balance probability mass across payload channels.

Although existing multi-bit watermarking schemes vary in payload assignment, they fundamentally depend on exact n -gram matching to synchronize PRF. This reliance on surface-level continuity renders them highly fragile to paraphrasing. Even minor structural edits—such as token insertions or substitutions—can irrecoverably disrupt the PRF seed, causing the detector to misalign bit channels and fail to recover the payload.

2.3 Robustness, Paraphrasing, and Desynchronization

Robustness is a central challenge for LLM watermarking because realistic users can rewrite watermarked text without changing its main meaning. Neural paraphraser such as Dipper ([Krishna et al., 2023](#)) can substantially alter lexical choices and sentence structure while preserving semantics. Local perturbations such as token deletion, insertion, and copy-paste editing are also enough to break the continuity assumed by token-history PRF. Existing robust watermarking studies improve detection through stronger tests, redundancy, distribution-preserving sampling, or edit-tolerant design ([Zhao et al., 2024](#); [Kuditipudi et al., 2024](#); [Kirchenbauer et al., 2024](#)). Toolkits and benchmarks such as MarkLLM ([Pan et al., 2024](#)) further highlight that robustness should be evaluated under both direct

perturbations and paraphrase-based attacks.

For multi-bit watermarking, robustness is limited less by signal strength than by synchronization. Token-history PRF requires the detector to reproduce the same n -gram state as the generator, but paraphrasing or local edits can easily change this state and shift the payload channels. Error-correction codes or stronger logit biases may absorb isolated token-level noise, yet they do not remove the dependence on exact token identity. This limitation motivates HSPMark to move beyond intact local token windows and reconstruct watermark states from semantic context, allowing the generator and detector to remain better aligned after meaning-preserving rewrites.

A natural way to reduce desynchronization is to anchor the watermark state in semantics instead of exact tokens. Semantic watermarking methods such as SemStamp (Hou et al., 2024) and SIR (Liu et al., 2024) show that continuous semantic information can make watermark detection less sensitive to paraphrasing. This idea is closely related to our motivation, but existing semantic robust watermarks are mainly designed for zero-bit detection or aggregate verification. Multi-bit watermarking requires a stronger property: the semantic state must not only remain stable after rewriting, but also support multiple distinguishable payload channels.

To address this problem, HSPMark anchors the watermark state in a continuous semantic space rather than brittle token histories. By extracting perturbation-resistant semantic features from sentence-level context, the framework ensures reliable state reconstruction under meaning-preserving rewrites. It then combines a context-conditioned pseudo-orthogonal projection to prevent inter-bit interference with an entropy-adaptive injection strategy, enabling robust multi-bit extraction without compromising generation quality.

3 Preliminaries

Multi-bit watermarking. Logit-based watermarking typically synchronizes the generator and the detector through a PRF seeded by the recent token history. Given a window of h previous tokens, the watermark state at step t can be written as

$$s_t = \text{PRF}(x_{t-h:t-1}). \quad (1)$$

The state s_t is then used to construct token partitions or decoding rules, and the model logits are biased so that a payload message $\mathbf{b} \in \{0, 1\}^D$ can

be recovered from the generated text. In this setting, reliable extraction requires the detector to reconstruct the same watermark state as the generator.

Desynchronization under rewriting. The main weakness of token-history-based synchronization is its dependence on exact surface forms. After paraphrasing or local editing, the rewritten sequence x' often no longer satisfies

$$x'_{t-h:t-1} = x_{t-h:t-1}. \quad (2)$$

As a result, the detector derives a different state s'_t , which changes the token partitions or payload channels used for extraction. This mismatch is especially harmful for multi-bit watermarking, where bit recovery depends not only on detecting a statistical signal but also on keeping multiple payload channels aligned. This motivates replacing brittle discrete synchronization with a more stable semantic anchoring mechanism.

4 Methodology

4.1 Problem Formulation and Overview

We propose HSPMark, a semantic-anchored multi-bit watermarking framework designed for autoregressive LLMs. Given a user prompt P , the objective is to imperceptibly embed a multi-bit binary payload $\mathbf{b} \in \{0, 1\}^D$ (where D is the payload capacity) into the generated text distribution. For mathematical symmetry during embedding and extraction, we map the binary payload into a bipolar form $\mathbf{m} = 2\mathbf{b} - 1 \in \{-1, +1\}^D$. Existing multi-bit watermarks fundamentally rely on exact, discrete n -gram token matching to synchronize their PRF. To address the inherent desynchronization vulnerability under paraphrasing attacks, HSPMark abandons discrete token matching, anchoring the watermark state in a continuous, perturbation-resistant semantic space. The framework integrates a sentence-level pseudo-orthogonal semantic projection with a token-level entropy-adaptive embedding mechanism.

4.2 Semantic Context Anchoring

To overcome the fragility of exact token matching, HSPMark introduces an auxiliary feature extraction network that maps historical textual context into a stable, dense vector. The watermark state is updated at the sentence level. For the i -th sentence being generated, the method constructs a context anchor C_i by extracting the preceding w com-

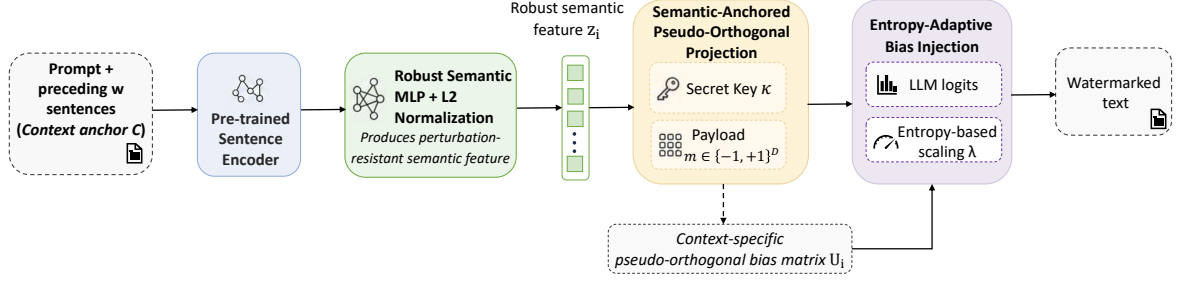


Figure 1: Pipeline of HSPMark embedding. Robust semantic features generate pseudo-orthogonal bias matrices, embedding multi-bit payloads into LLM logits via entropy-adaptive injection.

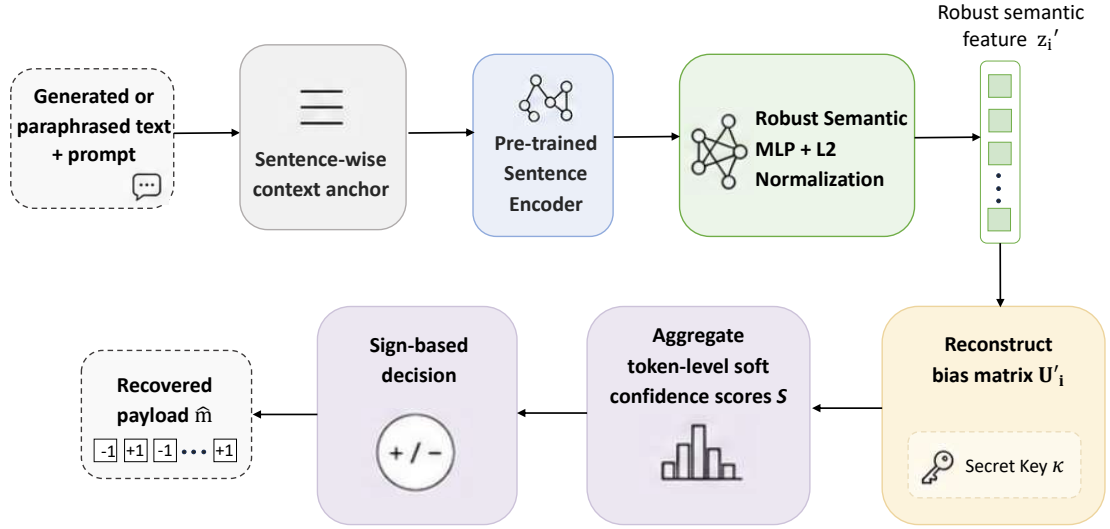


Figure 2: Pipeline of HSPMark extraction. The detector reconstructs the bias space from semantic anchors, recovering the payload via aggregated token-level confidence scores.

pleted sentences from the concatenated sequence $P \circ (s_1, \dots, s_{i-1})$.

The anchor C_i is first processed by a pre-trained base sentence encoder $E(\cdot)$. To filter out surface-level syntactic noise and extract invariant semantic features, the base representation is fed into a specialized Robust Semantic MLP, denoted as $g_\psi(\cdot)$. Finally, to eliminate scaling impacts on subsequent dot products, the output undergoes strict L_2 normalization to yield the robust semantic feature $\mathbf{z}_i \in \mathbb{R}^d$:

$$\mathbf{z}_i = \frac{g_\psi(E(C_i))}{\|g_\psi(E(C_i))\|_2} \quad (3)$$

During offline training, $E(\cdot)$ remains frozen to preserve its general semantic knowledge. The MLP g_ψ is trained independently as a lightweight calibration layer (see Appendix D for architecture and

training details). By minimizing the Cosine Embedding Loss between structurally diverse but semantically equivalent text pairs, g_ψ learns to align their representations, ensuring the context anchors remain stable against paraphrasing.

4.3 Semantic-Conditioned Pseudo-Orthogonal Bias Space

To encode multi-bit payloads without mutual interference, HSPMark projects the sentence-level semantic feature $\mathbf{z}_i$ into a dynamically generated pseudo-orthogonal space.

Using a secret key κ , HSPMark deterministically initializes a pseudo-random projection tensor $W \in \mathbb{R}^{D \times d \times r}$ and a vocabulary bucket mapping $\pi : \mathcal{V} \rightarrow \{1, \dots, r\}$. For the i -th sentence, the semantic feature $\mathbf{z}_i$ is projected into a bucket-level coding space and then expanded to the vocabulary

level:

$$\mathbf{U}_i[j, v] = \sum_{\ell=1}^d \mathbf{z}_i[\ell] W[j, \ell, \pi(v)].$$

Here, r is the number of projection buckets, and $\pi(v)$ assigns each vocabulary token to one bucket. The term pseudo-orthogonal does not refer to exact matrix orthogonalization. Instead, it denotes the low empirical correlation between different payload channels induced by keyed random projection, high-dimensional bucket mapping, and subsequent normalization. Each row of $\mathbf{U}_i$ therefore serves as a decorrelated coding channel for one payload bit. The raw watermark preference score for a candidate token $v \in \mathcal{V}$ is aggregated across all bit dimensions:

$$r_i(v) = \frac{1}{D} \sum_{j=1}^D m_j \mathbf{U}_i[j, v] \quad (4)$$

This score is then standardized across the vocabulary (using mean μ_i and standard deviation σ_i) to produce the normalized target bias $\hat{r}_i(v) = (r_i(v) - \mu_i)/(\sigma_i + \epsilon)$, where ϵ is a small constant (e.g., 10^{-5}) to prevent division by zero. Appendix C provides a first-order theoretical analysis showing that approximate row orthogonality in $\mathbf{U}_i$ suppresses the covariance terms between different payload channels, thereby reducing inter-channel crosstalk as the payload dimension D increases.

4.4 Entropy-Adaptive Watermark Embedding

While the target bias $\hat{r}_i(v)$ is fixed for the i -th sentence, imposing static logit biases can severely degrade the quality of LLM-generated text. Therefore, HSPMark modulates the injection strength at the token level using an Entropy-Adaptive Scaling mechanism.

During the autoregressive generation of the t -th token (which belongs to sentence i), let H_t denote the Shannon entropy of the model’s original predictive probability distribution. An adaptive scaling factor λ_t is computed:

$$\lambda_t = \text{clip} \left(\frac{H_t}{2.0}, 0.2, 1.2 \right) \quad (5)$$

Here, the clipping constants are empirically set to prevent extreme scaling values that could either degrade text fluency or wash out the watermark signal. By concentrating bias injection in high-entropy

Algorithm 1 Watermark Embedding

Input: Prompt P , Payload $\mathbf{b} \in \{0, 1\}^D$, Model $\mathcal{M}$, Encoder E , Projector g_ψ , Key κ , Window w , Strength α , Max length L

Output: Watermarked text sequence Y

- 1: Convert $\mathbf{b}$ to bipolar payload $\mathbf{m} = 2\mathbf{b} - 1$
- 2: Generate tensor $\mathbf{W}$ and mapping π from key κ
- 3: Initialize text $Y \leftarrow \emptyset$, sentence index $i \leftarrow 1$, and context sequence $S_{ctx} \leftarrow P$
- 4: Construct C_1 from S_{ctx} using preceding w sentences
- 5: Compute $\mathbf{z}_1 \leftarrow g_\psi(E(C_1))/\|g_\psi(E(C_1))\|_2$
- 6: Construct $\mathbf{U}_1$ and compute normalized bias $\hat{r}_1(v)$
- 7: **for** decoding step $t = 1$ to L **do**
- 8: Obtain original logits ℓ_t from $\mathcal{M}$ and calculate entropy H_t
- 9: Compute token-level entropy scaling factor λ_t
- 10: **for** each candidate token $v \in \mathcal{V}$ **do**
- 11: $\tilde{\ell}_t(v) \leftarrow \ell_t(v) + \alpha \cdot \lambda_t \cdot \hat{r}_i(v)$
- 12: **end for**
- 13: Sample next token $y_t \sim \text{softmax}(\tilde{\ell}_t)$
- 14: append y_t to Y
- 15: **if** y_t is an end-of-sentence token **then**
- 16: $i \leftarrow i + 1$
- 17: Update S_{ctx} with the newly generated sentence
- 18: Construct new anchor C_i from S_{ctx}
- 19: Compute $\mathbf{z}_i$, $\mathbf{U}_i$, and normalized bias $\hat{r}_i(v)$
- 20: **end if**
- 21: **end for**
- 22: **return** Y

(uncertain) regions and curtailing it in low-entropy (high-confidence) regions, the model preserves linguistic fluency. The original logits $\ell_t(v)$ are modified by adding the scaled, sentence-level target bias:

$$\tilde{\ell}_t(v) = \ell_t(v) + \alpha \cdot \lambda_t \cdot \hat{r}_i(v) \quad (6)$$

where α is the global watermark strength. The generation procedure is detailed in Algorithm 1.

4.5 Context-Aware Multi-bit Extraction

During extraction, HSPMark recovers the payload from a suspected generated text Y' without requiring the generation-time logits. To improve synchronization robustness, the original prompt P is utilized (see Appendix H for prompt-free extraction), and the text Y' is segmented into N sentences: $Y' = (s_1, s_2, \dots, s_N)$.

For each target sentence s_i , the detector reconstructs the context anchor C_i by extracting the preceding w sentences from the concatenated sequence $P \circ (s_1, \dots, s_{i-1})$. It then computes the semantic feature $\mathbf{z}'_i$ and reconstructs sentence-level bias matrix $\mathbf{U}'_i$ using the same secret key. A continuous soft confidence score vector $\mathbf{S} \in \mathbb{R}^D$ is accumulated by averaging the corresponding code

vectors:

$$\mathbf{S}[j] = \frac{1}{N_{\text{tok}}} \sum_{i=1}^N \sum_{v \in T(s_i)} \mathbf{U}'_i[j, v] \quad (7)$$

where $T(s_i)$ denotes the token sequence of sentence s_i , and $N_{\text{tok}} = \sum_{i=1}^N |T(s_i)|$ is the total number of evaluated tokens. Since the rows of $\mathbf{U}'_i$ are approximately centered and pseudo-orthogonal, the aggregated score mainly preserves the signal aligned with each target payload bit, while contributions from other bit channels tend to cancel out, as discussed in Appendix C. The final payload is extracted using a sign-based decision: $\hat{m}_j = \text{sgn}(\mathbf{S}[j])$, which is then mapped back to binary format as $\hat{b}_j = (\hat{m}_j + 1)/2$. The extraction process is outlined in Algorithm 2 of Appendix B.

5 Experiments

5.1 Experimental Setup

We compare HSPMark with state-of-the-art multi-bit text watermarking schemes, including StealthInk, MPAC, ECC, and the recently proposed WaterMod, evaluating their performance across capacity, reliability, and text quality. For text generation, unless noted otherwise, we employ Llama 2-7B (Touvron et al., 2023) as the backbone model and randomly sample 300 prompts from the C4 RealNewsLike (Raffel et al., 2020) dataset. Generation is executed with a temperature of 1.0 using a top-k sampling strategy ($k = 50$). For HSPMark, we set the global watermark strength to $\alpha = 2$ and fix the context window size to $w = 2$ by default, except in the hyperparameter sensitivity analysis. For the baseline methods, we follow the default parameter settings used in their original papers. Implementation details are provided in Appendix A.

5.2 Resilience Against Deep Paraphrasing and Perturbations

Most existing multi-bit watermarking schemes exhibit performance degradation when facing structural text alterations. We first assess the extraction robustness against deep paraphrasing attacks using the Dipper model (Krishna et al., 2023), which provides fine-grained control over lexical and order diversity. In these robustness evaluations, all evaluated methods are configured to generate sequences of 200 tokens. While lexical diversity controls the extent of synonym substitution, order diversity dictates the reordering and restructuring of syntactic frames.

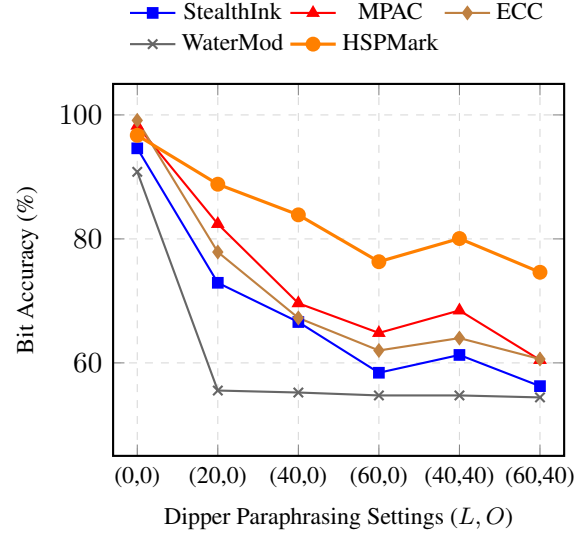


Figure 3: Bit accuracy under varying Dipper paraphrasing intensities (L, O) for an 8-bit payload.

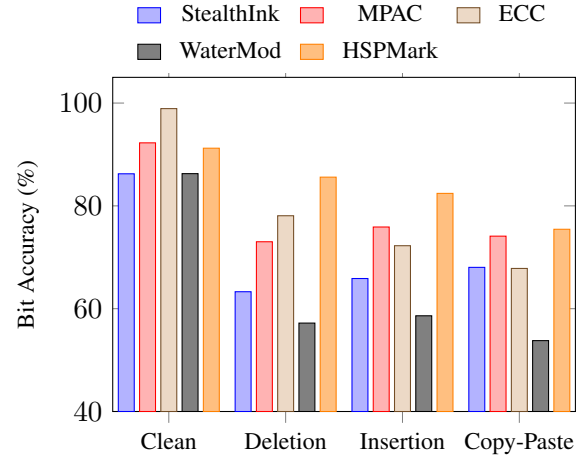


Figure 4: Bit accuracy under various discrete text perturbations for a 24-bit payload embedded within 200 tokens.

As illustrated in Figure 3 and Table 1, in a clean environment where the watermarked text remains untampered with, all evaluated methods effectively extract the embedded messages. However, under structural rewriting attacks, methods relying on discrete token matching exhibit a desynchronization phenomenon, leading to a decline in extraction accuracy. In contrast, HSPMark demonstrates a more stable synchronization capability. By anchoring the watermark in a continuous semantic space rather than discrete token sequences, our method maintains a stable extraction accuracy even when global sentence structures are rewritten.

Beyond structural rewriting, as illustrated in Figure 4, we evaluate the models’ resistance to

Table 1: Bit Accuracy against Dipper paraphrasing attacks across varying lexical and order diversity settings (L, O).

Payload	Method	Bit Accuracy (%)					
		Clean (0,0)	Low (20,0)	Med (40,0)	High (60,0)	Mixed (40,40)	Heavy (60,40)
8-bit	StealthInk	94.61	72.92	66.59	58.41	61.28	56.25
	MPAC	98.26	82.41	69.62	64.83	68.46	60.47
	WaterMod	90.82	55.54	55.22	54.75	54.75	54.43
	ECC	99.12	77.88	67.25	62.00	64.00	60.62
	HSPMark	96.71	88.82	83.88	76.32	80.06	74.62
16-bit	StealthInk	88.55	66.43	60.04	57.65	58.32	55.37
	MPAC	95.81	71.66	66.19	59.80	62.36	58.45
	WaterMod	88.18	50.59	50.08	50.00	49.92	48.99
	ECC	98.75	66.75	62.25	56.06	59.88	54.56
	HSPMark	94.47	77.67	71.48	67.06	70.38	65.10

discrete local perturbations, including token deletion, token insertion, and copy-paste attacks. For these evaluations, we apply a 24-bit payload setting. The perturbation intensity is specified as 0.3 for both deletion and insertion attacks (meaning 30% of the tokens are randomly manipulated), while the intensity for the copy-paste attack is set to 0.5. Traditional n-gram-based methods experience performance drops when local continuity is breached. The semantic representation mechanism of HSPMark better absorbs these local modifications, avoiding error accumulation caused by continuity interruptions and achieving more resilient payload recovery across various perturbation types.

5.3 Payload Capacity, Text Quality, and Extraction Efficiency

Increasing the payload capacity typically exacerbates the inherent trade-off between extraction reliability and text quality. Table 2 presents the performance of all evaluated methods across varying payload sizes (24, 36, and 48 bits) using a fixed generation length of 200 tokens. StealthInk prioritizes text stealthiness, achieving low PPL across all payload scales; however, its unbiased embedding design impacts extraction reliability under high payload constraints (dropping to 75.60% at 48 bits). Conversely, the ECC scheme achieves the highest extraction reliability among the compared methods by pairing rigorous error correction codes with static logit manipulation. However, this rigid logit biasing alters the original probability distribution, elevating text perplexity. To resolve this polarization, HSPMark implements an entropy-adaptive scaling mechanism. This dynamic approach achieves a balanced trade-off, maintaining a stable bit accu-

Table 2: Trade-off among payload size, bit accuracy, text quality measured by PPL, and extraction time for 200-token generations.

Payload	Method	Bit Accuracy (%) $\uparrow$	PPL $\downarrow$	Extraction Time (s) $\downarrow$
No Watermark		-	3.32	-
24-bit	StealthInk	86.39	3.66	0.12
	MPAC	91.92	4.41	0.10
	WaterMod	86.21	6.58	0.59
	ECC	96.79	6.98	0.28
	HSPMark	91.17	5.88	0.06
36-bit	StealthInk	81.40	3.49	0.12
	MPAC	86.11	4.40	0.10
	WaterMod	83.19	6.80	0.72
	ECC	91.17	6.94	0.87
	HSPMark	87.79	5.86	0.06
48-bit	StealthInk	75.60	3.57	0.13
	MPAC	81.24	4.26	0.10
	WaterMod	82.68	6.60	0.73
	ECC	85.59	6.89	1.34
	HSPMark	84.30	5.75	0.06

racy of 84.30% at 48 bits while keeping the text perplexity within an acceptable range.

In addition to payload capacity, computational efficiency represents another aspect of our framework. As reported in Table 2, HSPMark maintains an extraction time of 0.06 seconds across all payload scales. This is faster than the baseline methods, demonstrating its practical viability for latency-sensitive applications.

5.4 Downstream Utility Evaluation

To further evaluate whether watermark embedding affects task-oriented generation, we conduct experiments on two downstream NLP tasks: text summarization with BART-large (Lewis et al., 2020) on CNN/DailyMail (See et al., 2017), and machine translation with mBART (Liu et al., 2020) on the WMT’16 En-Ro (Bojar et al., 2016). Since these

Table 3: Impact of multi-bit watermarking on downstream NLP task performance.

Method	Text Summarization		Machine Translation	
	BERTScore $\uparrow$	ROUGE-1 $\uparrow$	BLEU $\uparrow$	BERTScore $\uparrow$
No Watermark	0.2723	0.3215	29.9856	0.6949
StealthInk	0.2727	0.3366	29.1380	0.6952
MPAC	0.2423	0.2967	27.1492	0.6789
WaterMod	0.2266	0.2879	17.4489	0.5745
ECC	0.1046	0.2216	4.3164	0.2019
HSPMark	0.2268	0.2817	25.2103	0.6605

tasks usually produce shorter outputs than open-ended generation, we use an 8-bit payload in this evaluation.

The results in Table 3 show that watermarking can affect task-oriented generation, with the impact varying across methods. Strong static-bias methods, such as ECC, cause clear degradation, particularly in machine translation. HSPMark performs better than these aggressive baselines, suggesting that entropy-adaptive scaling helps avoid excessive perturbation of high-confidence token choices. At the same time, HSPMark still lags behind the no-watermark setting and the more conservative StealthInk baseline on several metrics. These results indicate that HSPMark improves the robustness-utility trade-off, rather than completely preserving downstream task quality.

5.5 Ablation Studies

Entropy-adaptive scaling. We examine whether Shannon entropy is an appropriate uncertainty measure for adaptive watermark injection. Under a 24-bit payload, we replace entropy with either the top-1 token probability or the probability margin between the two most likely tokens, while keeping all remaining configurations unchanged.

As shown in Table 4, Shannon entropy achieves the best result across the evaluated recovery and PPL metrics. Unlike top-1 probability and probability margin, which retain information from only one or two tokens, Shannon entropy characterizes uncertainty over the complete next-token distribution and therefore provides a more informative signal for allocating watermark strength.

Context-window size. The context-window parameter w determines the number of preceding complete sentences used to construct each semantic anchor. To study its effect, we evaluate HSPMark with a 24-bit payload using $w \in \{1, 2, 3, 4\}$ under different paraphrasing intensities.

As illustrated in Figure 5, the context-window

Table 4: Ablation of uncertainty measures for adaptive scaling. ‘‘Avg Para’’ denotes the average over the evaluated non-clean Dipper settings, and ‘‘Heavy’’ denotes Dipper (60, 40).

Measure	Clean $\uparrow$	Avg Para $\uparrow$	Heavy $\uparrow$	PPL $\downarrow$
Entropy	92.68	73.49	66.58	5.83
Top-1 probability	91.43	71.70	65.59	6.27
Probability margin	91.96	71.91	65.74	6.35

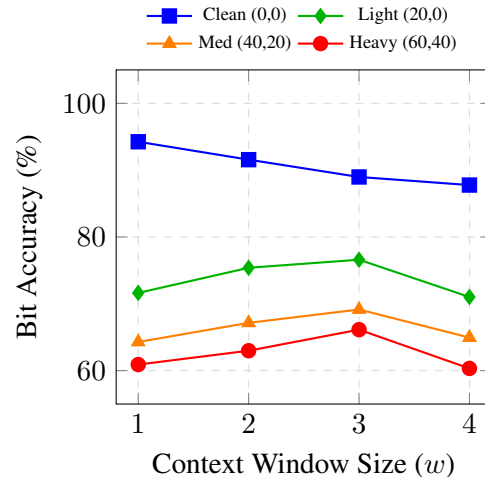


Figure 5: Impact of context window size w on watermark robustness under varying paraphrasing intensities.

size introduces a trade-off between contextual coverage and semantic noise. Specifically, a small window ($w=1$) demonstrates superior performance on clean text; however, it lacks sufficient semantic context to maintain stability under paraphrasing conditions. Expanding the window to $w=2$ or $w=3$ enhances robustness by offering more reliable semantic anchors. Nevertheless, a window size of $w=4$ introduces more distant and less relevant context, thereby introducing noise and diminishing accuracy. Overall, a window size of $w=2$ strikes the optimal balance between clean accuracy and paraphrasing robustness, warranting its adoption as the default experimental setting. We also provide additional ablations on the pseudo-orthogonal bias construction and the semantic feature extraction module in Appendices E and F, respectively.

6 Conclusion

This paper presented HSPMark, a semantic-anchored multi-bit watermarking framework for robust provenance tracking of LLM-generated text. HSPMark reconstructs watermark states from sentence-level semantic anchors, thereby mit-

igating desynchronization caused by paraphrasing and local perturbations. By combining pseudo-orthogonal bias spaces with entropy-adaptive injection, HSPMark balances payload recovery and generation quality. Experimental results demonstrate that HSPMark achieves a more stable robustness–utility trade-off than existing multi-bit watermarking baselines, highlighting the potential of semantic anchoring for reliable high-capacity watermarking.

Limitations

While HSPMark demonstrates strong robustness and utility across our evaluated settings, several directions remain open for further exploration.

On the one hand, the proposed semantic anchoring mechanism requires sufficient context to construct stable sentence-level anchors, so its effectiveness may decrease for very short generations or limited-context scenarios. Although HSPMark is robust to lexical and local structural variations, aggressive discourse-level rewrites, such as sentence reordering or merging, may affect anchor reconstruction. Modeling document-level discourse structure is left for future work.

On the other hand, the entropy-adaptive strategy intentionally reduces bias injection in low-entropy decoding regions to preserve fluency. This design improves generation quality, but it may limit payload recovery when generation is highly constrained, such as in code generation, template-style responses, or factual outputs with narrow token choices. Future work may explore task-aware adaptation strategies that adjust the embedding strength according to the generation scenario.

Ethics Statement

This work aims to protect the intellectual property of deep neural network models through watermark-based ownership verification. The proposed method does not collect personal data or attempt to identify individuals. The watermark is designed to have minimal impact on the normal performance of the model and is intended only for protection and legal ownership accountability. Potential misuse, such as false ownership claims or unauthorized watermark manipulation, should be prevented through proper verification protocols and responsible deployment.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62572208, 62402513), the Innovative Scientific and Technological Talents Program of Hubei Province, China (Project No. 2025DJA050), and the Natural Science Foundation of Hubei Province (Joint Fund Project), China (Grant No. 2025AFD748).

References

- Steven Bird and Edward Loper. 2004. [NLTK: The natural language toolkit](#). In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 214–217. Association for Computational Linguistics.
- Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, and 2 others. 2016. [Findings of the 2016 conference on machine translation](#). In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198. Association for Computational Linguistics.
- Miranda Christ, Sam Gunn, and Or Zamir. 2024. [Undetectable watermarks for language models](#). In *The Thirty Seventh Annual Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1125–1139. PMLR.
- Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, A. Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, and 5 others. 2024. [Scalable watermarking for identifying large language model outputs](#). *Nature*, 634(8035):818–823.
- Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahlouljifar, Mohammad Mahmoody, and Mingyuan Wang. 2025. [Publicly-detectable watermarking for language models](#). *IACR Communications in Cryptology*, 1(4):31.
- Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. 2025. [BiMark: Unbiased multi-layer watermarking for large language models](#). In *Proceedings of the 42nd International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 17049–17067. PMLR.
- Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. [Three bricks to consolidate watermarks for large language models](#).

- In *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE.
- Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024. [Semstamp: A semantic watermark with paraphrastic robustness for text generation](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4067–4082. Association for Computational Linguistics.
- Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2024. [Unbiased watermark for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*.
- Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. 2025. [StealthInk: A multi-bit and stealthy watermark for large language models](#). In *Forty-second International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. [A watermark for large language models](#). In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17061–17084. PMLR.
- John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2024. [On the reliability of watermarks for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*.
- 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*, volume 36, pages 27469–27500. Curran Associates, Inc.
- Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2024. [Robust distortion-free watermarks for language models](#). *Transactions on Machine Learning Research, TMLR 2024*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. [BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension](#). In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 7871–7880. Association for Computational Linguistics.
- Chin-Yew Lin. 2004. [Rouge: A package for automatic evaluation of summaries](#). In *Text summarization branches out*, pages 74–81. Association for Computational Linguistics.
- Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. 2024. [A semantic invariant robust watermark for large language models](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*.
- Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. [Multilingual denoising pre-training for neural machine translation](#). *Transactions of the Association for Computational Linguistics*, 8:726–742.
- Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, Irwin King, and Philip S. Yu. 2024. [MarkLLM: An open-source toolkit for LLM watermarking](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 61–71. Association for Computational Linguistics.
- Shinwoo Park, Hyejin Park, Hyeeseon Ahn, and Yo-Sub Han. 2026. [Watermod: Modular token-rank partitioning for probability-balanced llm watermarking](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 32683–32691.
- Matt Post. 2018. [A call for clarity in reporting BLEU scores](#). In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics.
- Wenjie Qu, Wengruai Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. 2025. [Provably robust multi-bit watermarking for AI-generated text](#). In *34th USENIX Security Symposium (USENIX Security 25)*, pages 201–220. USENIX Association.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. [Exploring the limits of transfer learning with a unified text-to-text transformer](#). *Journal of machine learning research*, 21(140):1–67.
- 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)*, pages 3982–3992. 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. Association for Computational Linguistics.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. [Get to the point: Summarization with pointer-generator networks](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083. Association for Computational Linguistics.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.

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

Xiaojun Xu, Jinghan Jia, Yuanshun Yao, Yang Liu, and Hang Li. 2025. [Robust multi-bit text watermark with LLM-based paraphraser](#). In *Proceedings of the 42nd International Conference on Machine Learning, ICML 2025*, volume 267 of *Proceedings of Machine Learning Research*, pages 69279–69292. PMLR.

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, Jixia Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2024. [Advancing beyond identification: Multi-bit watermark for large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4031–4055. Association for Computational Linguistics.

Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. 2019. [Defending against neural fake news](#). In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019*, volume 32. Curran Associates, Inc.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. [Bertscore: Evaluating text generation with BERT](#). In *International Conference on Learning Representations, ICLR 2020*.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. 2024. [Provable robust watermarking for ai-generated text](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024*.

Appendices

A Implementation Details

We evaluate HSPMark alongside state-of-the-art baselines across open-ended generation, downstream NLP tasks, and paraphrasing robustness tests. All experiments are conducted on an NVIDIA GeForce RTX 4090 GPU. To optimize memory efficiency during autoregressive decoding, the generative models are loaded in bfloat16 precision with dynamic device mapping and expandable memory segments enabled. The open-ended generation benchmarks are conducted using the Llama 2-7B model (Touvron et al., 2023). For downstream task evaluations, we employ BART-large (Lewis et al., 2020) for text summarization and mBART (Liu et al., 2020) for machine translation, matching the specifications described in the main text. The context anchoring mechanism extracts features using the all-MiniLM-L6-v2 (Reimers and Gurevych, 2019) sentence transformer, while adversarial paraphrasing attacks are driven by the Dipper model (Krishna et al., 2023). All utilized datasets – including the C4 RealNewsLike (Raffel et al., 2020), CNN/DailyMail (See et al., 2017), and WMT’16 En-Ro (Bojar et al., 2016) – along with the corresponding model checkpoints, are standard open-source assets that are used in accordance with their respective licenses and terms of use.

Our open-ended benchmark evaluates performance on 300 randomly sampled prompt prefixes, while downstream NLP task evaluations are conducted on 200 sampled instances. Evaluation subsets are sampled independently for each experiment and shared across all methods or variants compared within that experiment. Text processing and sentence boundary detection are handled via the NLTK tokenization module (Bird and Loper, 2004). The downstream text quality and semantic alignment metrics are implemented using the Hugging Face Evaluate library, specifically utilizing the rouge module (Lin, 2004) for summaries, sacrebleu (Post, 2018) for translation quality, and BERTScore (Zhang et al., 2020) for continuous semantic similarity verification. To ensure statistical stability and mitigate random variance, all performance metrics reported in our tables represent the mean results in the corresponding evaluation sets.

B Extraction Algorithm

We provide the pseudocode for HSPMark’s watermark extraction procedure in Algorithm 2. The extractor reconstructs context-conditioned bias matrices from sentence-level semantic anchors, accumulates token-level evidence across the suspected text, and recovers the payload through a lightweight sign-based decoding rule. This makes the recovery process explicit and complements the embedding procedure described in the main text.

Algorithm 2 Watermark Extraction

Input: Prompt P , Suspected text Y' , Encoder E , Projector g_ψ , Key κ , Payload length D , Window w

Output: Extracted payload $\hat{\mathbf{b}} \in \{0, 1\}^D$

- 1: Generate tensor $\mathbf{W}$ and mapping π from key κ
- 2: Segment Y' into N sentences $(s_1, s_2, \dots, s_N)$
- 3: Initialize score accumulator $\mathbf{S} \leftarrow \mathbf{0} \in \mathbb{R}^D$ and token count $N_{tok} \leftarrow 0$
- 4: **for** $i = 1$ to N **do**
- 5: Construct anchor C_i from $P \circ (s_1, \dots, s_{i-1})$ using preceding w sentences
- 6: Compute robust semantic feature $\mathbf{z}'_i \leftarrow g_\psi(E(C_i))/\|g_\psi(E(C_i))\|_2$
- 7: Reconstruct bias matrix $\mathbf{U}'_i \in \mathbb{R}^{D \times |\mathcal{V}|}$ using $\mathbf{z}'_i$, $\mathbf{W}$, and π
- 8: **for** each token v in $T(s_i)$ **do**
- 9: **for** $j = 1$ to D **do**
- 10: $\mathbf{S}[j] \leftarrow \mathbf{S}[j] + \mathbf{U}'_i[j, v]$
- 11: **end for**
- 12: $N_{tok} \leftarrow N_{tok} + 1$
- 13: **end for**
- 14: **end for**
- 15: Average confidence scores: $\mathbf{S} \leftarrow \mathbf{S} / \max(N_{tok}, 1)$
- 16: **for** $j = 1$ to D **do**
- 17: $\hat{m}_j \leftarrow 1$ if $\mathbf{S}[j] \geq 0$, else $\hat{m}_j \leftarrow -1$
- 18: $\hat{b}_j \leftarrow (\hat{m}_j + 1)/2$
- 19: **end for**
- 20: **return** $\hat{\mathbf{b}}$

C Theoretical Analysis of Channel Decoupling

We provide a concise analysis of why the pseudo-orthogonal bias matrix helps separate different payload channels. For simplicity, we analyze the clean extraction setting for a single sentence, where $\mathbf{z}'_i = \mathbf{z}_i$ and $\mathbf{U}'_i = \mathbf{U}_i$, and omit the sentence index i . Let

$$u_j(v) = \mathbf{U}_i[j, v]$$

denote the vocabulary-level bias vector for the j -th payload bit. The bipolar payload is $m \in \{-1, +1\}^D$, and the raw watermark score is

$$r(v) = \frac{1}{D} \sum_{j=1}^D m_j u_j(v). \quad (8)$$

The actual bias used during generation is the normalized score $\hat{r}(v)$, obtained by centering and rescaling $r(v)$.

Let $p_0(v)$ be the original next-token distribution before watermarking. After adding a small effective bias $\eta \hat{r}(v)$, where $\eta = \alpha \lambda_t$, the softmax distribution satisfies the first-order approximation

$$p_\eta(v) = p_0(v)(1 + \eta(\hat{r}(v) - \mathbb{E}_{p_0}[\hat{r}])) + O(\eta^2). \quad (9)$$

Therefore, the expected extraction score for bit k changes by

$$\mathbb{E}_{p_\eta}[u_k] = \mathbb{E}_{p_0}[u_k] + \eta \text{Cov}_{p_0}(u_k, \hat{r}) + O(\eta^2). \quad (10)$$

The first term is independent of the message and is close to zero after centering. Since $\hat{r}$ is only a centered and positively rescaled version of r , the message-dependent term has the same sign structure as $\text{Cov}_{p_0}(u_k, r)$. Define

$$A_{kj} = \text{Cov}_{p_0}(u_k, u_j).$$

Then

$$\text{Cov}_{p_0}(u_k, r) = \frac{1}{D} \left(m_k A_{kk} + \sum_{j \neq k} m_j A_{kj} \right). \quad (11)$$

Here, $m_k A_{kk}$ is the useful signal for the k -th bit, while the summation over $j \neq k$ is the crosstalk from other payload bits.

The pseudo-orthogonal construction is used to make the off-diagonal terms small. If the rows are normalized so that $A_{kk} \approx 1$, and

$$|A_{kj}| \leq \rho, \quad j \neq k,$$

then the total interference is bounded by

$$\left| \sum_{j \neq k} m_j A_{kj} \right| \leq (D-1)\rho. \quad (12)$$

Thus, when the off-diagonal correlation is sufficiently small, for example $(D-1)\rho < 1$, the expected score keeps the sign of the target bit m_k , up to the approximation error.

Returning to the multi-sentence extraction setting, HSPMark averages the channel scores over all evaluated tokens:

$$\mathbf{S}[k] = \frac{1}{N_{\text{tok}}} \sum_i \sum_{v \in T(s_i)} \mathbf{U}'_i[k, v]. \quad (13)$$

The useful signal has a consistent sign for the target bit, while off-channel terms are weakly correlated and tend to cancel during aggregation. This is why the sign rule

$$\hat{m}_k = \text{sgn}(\mathbf{S}[k])$$

can recover the payload bit reliably when the channels are well separated.

This also explains the failure mode of an unconstrained random projection. Although each off-diagonal term may be small on average, the extraction score contains $D - 1$ such terms. As D increases, their accumulated effect can become comparable to the target signal. By reducing these off-diagonal correlations, the pseudo-orthogonal bias space provides a more stable channel structure for high-capacity multi-bit watermarking.

D Training Details of the Robust Semantic MLP

The Robust Semantic MLP g_ψ is trained as a lightweight projection head on top of a frozen sentence encoder. We use all-MiniLM-L6-v2 as the base encoder $E(\cdot)$. During training, the parameters of $E(\cdot)$ are fixed, and only the MLP parameters are updated. This design keeps the general semantic representation from the pretrained encoder, while allowing the projected space to be slightly adapted to the synchronization requirement of HSPMark.

Given a text input x , the base encoder first produces a sentence embedding

$$e = E(x). \quad (14)$$

The MLP then maps this embedding into a normalized semantic feature space:

$$\tilde{z} = W_2 \sigma(\text{LayerNorm}(W_1 e + b_1)) + b_2, \quad (15)$$

$$z = \frac{\tilde{z}}{\|\tilde{z}\|_2}. \quad (16)$$

Here, $\sigma(\cdot)$ denotes the ReLU activation. The L2 normalization is applied so that the following semantic projection depends mainly on the direction of the representation rather than its scale.

For offline training, we use sentence-pair similarity data. Each training example is written as (x_a, x_b, s) , where $s \in [0, 5]$ is the annotated semantic similarity score. We normalize the score as

$$y = \frac{s}{2.5} - 1, \quad (17)$$

so that the supervision signal lies in $[-1, 1]$. The two sentences are encoded by the frozen encoder and then passed through the MLP:

$$z_a = g_\psi(E(x_a)), \quad z_b = g_\psi(E(x_b)). \quad (18)$$

The MLP is optimized with a cosine-based objective,

$$\mathcal{L}_{\text{MLP}} = \ell_{\cos}(z_a, z_b, y), \quad (19)$$

which encourages semantically similar sentence pairs to have closer projected representations, assigning lower cosine agreement to less related pairs. Once trained, g_ψ is fixed and shared by both the watermark embedding and extraction.

In our implementation, the MLP uses a hidden dimension of 256 and an output dimension of 128. We train it for 10 epochs with AdamW, a batch size of 64, and a learning rate of 1×10^{-3} . Since the sentence encoder is not fine-tuned, the MLP should be understood as a task-adaptive calibration layer rather than the main source of semantic understanding. The semantic information mainly comes from the pretrained encoder, while the MLP slightly adjusts the representation space to make the context anchors more stable under paraphrastic variation.

E Ablation on Pseudo-Orthogonal Bias Construction

To empirically validate the theoretical analysis presented in Appendix C, we conduct an ablation study focusing specifically on the high-capacity decoupling ability of the pseudo-orthogonal bias matrix.

We construct a baseline variant, HSPMark-Random, which replaces the structured pseudo-orthogonal projection described in Section 4.3 with an unstructured Gaussian projection. Specifically, this variant uses a standard random Gaussian matrix to map the robust semantic features directly into the vocabulary space, while all other components remain unchanged. We compare this against the full HSPMark framework, which adheres to the pseudo-orthogonal projection. We evaluate bit extraction accuracy across three payload constraints: low (8-bit), medium (24-bit), and high (48-bit), on untampered generated texts of 200 tokens.

As shown in Table 5, under a low payload setting (8-bit), both variants exhibit comparable performance, indicating that channel interference is not the primary bottleneck when the bit dimension is small. However, as the payload capacity increases to 48 bits, the extraction accuracy of HSPMark-Random deteriorates significantly, dropping to a

Table 5: Ablation study on pseudo-orthogonal bias construction across different payload sizes.

Payload	Variant	Bit Accuracy (%) $\uparrow$
8-bit	HSPMark-Random	95.83
	HSPMark (Full)	96.71
24-bit	HSPMark-Random	77.42
	HSPMark (Full)	91.17
48-bit	HSPMark-Random	54.68
	HSPMark (Full)	84.30

near-random guessing level of 54.68%. This empirical collapse is consistent with our theoretical analysis: without the structured pseudo-orthogonal construction, the aggregated crosstalk from interfering channels completely eclipses the target signal.

In contrast, the full HSPMark framework consistently maintains an accuracy of 84.30% at 48 bits. This result highlights the importance of pseudo-orthogonal projection in preserving stable channel separation for high-capacity multi-bit embedding.

F Ablation on Semantic Feature Extraction

To examine the effect of the semantic feature extraction module, we conduct an ablation study under the 24-bit payload setting with 200 generated tokens. We compare three variants of HSPMark. Full denotes the complete model with both the pretrained sentence encoder and the trained Robust Semantic MLP. Vanilla removes the MLP and directly uses the pretrained sentence encoder representation. Discrete removes semantic anchoring and falls back to token-history based synchronization.

Table 6: Ablation study on semantic feature extraction under Dipper paraphrasing attacks.

Method	Dipper Setting			
	Clean (0,0)	Low (20,0)	Med. (40,0)	High (60,0)
Full	90.08	75.00	72.42	71.03
Vanilla	89.78	71.95	67.26	66.99
Discrete	93.41	51.01	53.74	51.87

As shown in Table 6, the Discrete variant achieves the highest accuracy on clean text. This is expected because its token-history synchronization is still intact before rewriting. However, once paraphrasing is applied, its accuracy drops substantially, indicating that exact token matching is sensitive to surface-level changes.

Table 7: Semantic-anchor similarity under Dipper paraphrasing. Higher anchor similarity indicates more stable semantic-state reconstruction.

Dipper (L, O)	Vanilla $\uparrow$	Robust $\uparrow$
(20, 0)	0.7280	0.7558
(40, 0)	0.6770	0.7116
(60, 0)	0.6534	0.6834
(40, 40)	0.6534	0.6750
(60, 40)	0.6156	0.6569

The two semantic variants are more stable under Dipper attacks. Vanilla already improves the robustness over Discrete, which shows that the pretrained sentence encoder provides the main semantic stability. Compared with Vanilla, Full achieves a consistent but moderate improvement across all paraphrasing settings. This suggests that the Robust Semantic MLP further calibrates the pretrained embedding space for watermark synchronization, while the main robustness gain comes from replacing discrete token synchronization with semantic anchoring.

G Semantic Anchor Stability

To directly evaluate whether semantic anchors remain stable after paraphrasing, we compare the cosine similarity between the original and reconstructed anchors using the vanilla sentence encoder and the Robust Semantic MLP representation.

As shown in Table 7, the robust representation consistently yields higher anchor cosine similarity. Under the strongest Dipper setting (60, 40), it improves the similarity from 0.6156 to 0.6569. The decline in anchor similarity as attack intensity increases aligns with the accuracy degradation observed in Table 1, further supporting the connection between semantic-state stability and reliable payload recovery.

H Extraction Without the Original Prompt

When the original prompt is unavailable, the detector discards the first w complete sentences and reconstructs subsequent anchors using only the preceding sentences in the observed text.

As shown in Table 8, prompt-free extraction remains effective, although its accuracy decreases because the earliest anchors cannot be reconstructed and fewer token-level observations are available for score aggregation.

Table 8: Bit accuracy (%) with and without access to the original prompt.

Payload	With prompt $\uparrow$	Without prompt $\uparrow$
24-bit	91.17	85.40
36-bit	87.79	80.45
48-bit	84.30	72.41

Table 9: Bit accuracy (%) for partial-text extraction with a 16-bit payload.

Token length	Retained text ratio $\uparrow$	Bit Accuracy (%) $\uparrow$
500	25%	87.05
500	50%	92.89
500	75%	94.72
500	100%	94.77
1000	25%	93.34
1000	50%	93.84
1000	75%	94.42
1000	100%	95.14

I Extraction from Partial Text

We evaluate whether HSPMark can recover a payload from a contiguous fragment of a longer generation. When the original preceding context is unavailable, the first w complete sentences of the retained fragment are treated as a warm-up region. Payload scores are aggregated only over subsequent sentences whose anchors can be reconstructed from the locally observed context.

We generate sequences of 500 and 1,000 tokens with a 16-bit payload and retain contiguous fragments containing 25%, 50%, 75%, or 100% of the generated text.

As shown in Table 9, HSPMark maintains high bit accuracy when only part of a longer generation is available. For 500-token generations, bit accuracy increases from 87.05% at a 25% retention ratio to 94.77% when the complete text is retained. For 1,000-token generations, retaining only 25% of the text already achieves 93.34% bit accuracy. These results indicate that longer generations provide sufficient token-level evidence for reliable payload recovery even when only a partial fragment is observed.

J Cross-Backbone Evaluation

To assess whether the observations extend beyond Llama 2-7B, we repeat the 24-bit evaluation using Qwen2.5-7B (Yang et al., 2024) under the same generation and extraction protocol.

Table 10: Performance on Qwen2.5-7B with a 24-bit payload.

Method	Bit Accuracy (%) $\uparrow$	PPL $\downarrow$	Extraction Time (s) $\downarrow$
StealthInk	89.28	4.15	1.17
MPAC	92.73	5.37	1.22
WaterMod	86.80	7.46	0.48
ECC	98.63	8.09	0.46
HSPMark	93.77	6.37	0.05

As shown in Table 10, the Qwen2.5-7B results are consistent with those on Llama 2-7B: HSPMark provides competitive payload recovery while retaining substantially faster extraction than the evaluated baselines.