ALLEVIATING SEMANTIC NARROWING IN LLM WATERMARKING WITH HAMMING-NEIGHBORHOOD ACCEPTANCE

Anonymous authors
Paper under double-blind review

ABSTRACT

Semantic watermarking improves robustness against watermark removal attacks by embedding detectable signals into sentence-level representations. However, existing watermarking methods typically impose watermark-specific semantic preferences on generated sentences without explicitly accounting for the highly non-uniform and context-dependent semantic preference of LLM generation. When these two preferences are poorly aligned, many natural continuations become incompatible with the watermark, causing *semantic narrowing*: reduced semantic freedom, increased resampling cost, and potential degradation on tasks with strict semantic requirements. To alleviate this problem, we propose *HammingMark*, which uses the semantic hash of the preceding sentence as a dynamic center and accepts candidates whose hashes fall within its Hamming neighborhood. Defining watermark validity over a Hamming neighborhood in compact hash space retains a larger fraction of naturally likely semantic continuations. The coarse many-to-one hash mapping further allows diverse semantic realizations to remain watermark-valid. Experiments on C4 and BookSum show that HammingMark achieves strong robustness, high detectability, and near-unwatermarked generation quality, requiring only 2.2 sampled candidates per accepted sentence—a 72.8% reduction compared with the most sampling-efficient existing method. On more complex tasks with strict semantic constraints, HammingMark achieves the highest detection rates with the highest or tied-highest ROUGE-L scores, demonstrating its effectiveness in balancing watermark detectability and generation quality under constrained generation settings.

1 INTRODUCTION

Large language models (LLMs) have demonstrated remarkable capabilities in generating fluent and coherent text, which simultaneously raises growing concerns about content provenance, misuse, and copyright protection (Liang et al., 2026; Xu et al., 2025; Casper et al., 2026; Yang et al., 2025). Text watermarking provides a promising solution by embedding statistically detectable signals into model-generated content without altering its intended functionality (Kirchenbauer et al., 2023; Dathathri et al., 2024; Liu et al., 2026). However, conventional token-level watermarking methods encode such signals by manipulating token distributions and are therefore vulnerable to paraphrasing, synonym substitution, and other semantic-preserving attacks (Lu et al., 2024; Hu et al., 2024; Sun et al., 2026). Semantic-level watermarking has recently emerged as a more robust alternative against semantic-preserving attacks by shifting the sentence, rather than the token distributions (Liu et al., 2024; Hou et al., 2024a; Ren et al., 2024), hence is more attractive to reliable provenance verification of LLM-generated text.

Despite their improved robustness, existing semantic watermarking methods inevitably experience *semantic narrowing*: they typically sample multiple candidate sentences until finding one that falls into a predefined semantic subspace. For example, bucket-based methods require the generated sentence to fall into a designated semantic region (Hou et al., 2024a;b), while similarity-based methods constrain its embedding to a predefined similarity interval (Dabiriaghdam & Wang, 2025; Zhang et al., 2025). Multi-dimensional semantic partitioning methods similarly favor sentences located in watermark-specific regions of the embedding space (Huo et al., 2026). Although implemented

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

differently, these methods impose a watermark-specific semantic preference that may not align with the model's natural generation preference. Poor alignment can reject otherwise plausible, high-probability continuations, leading to inefficient, repeated sampling and potential semantic distortion. This problem is particularly severe in constrained tasks, where the set of acceptable continuations is already limited.

Our core observation is that the natural next-sentence generation is typically not uniformly distributed over the semantic space, but puts much weights on a context-dependent set of plausible semantic continuations (Meister et al., 2023; Farquhar et al., 2024; Aichberger et al., 2025). Previous semantic watermarking methods typically predefine a randomly positioned semantic subspace as the watermark-valid region, without considering whether this subspace aligns with the context-dependent semantic space favored by natural next-sentence generation.

Motivated by this principle, we propose *HammingMark*, which uses the semantics of the preceding sentence to dynamically determine which next-sentence candidates are compatible with the watermark. Rather than steering generation toward a fixed, context-independent semantic region, HammingMark adapts the watermark constraint to the local context already established by the generated text, making it more likely to preserve continuations that the model naturally prefers. Within this context-aligned design, watermark selectivity operates on coarse semantic hash codes, whose many-to-one mapping weakens the coupling between watermark validity and specific semantic content. The constraint thus remains selective without prescribing a single semantic direction, allowing multiple semantically distinct continuations to remain valid and preserving flexibility for discourse changes such as elaboration, contrast, and topic shifts.

Our contributions are threefold: 1) **A unified analysis of semantic narrowing.** We provide the first systematic analysis of *semantic narrowing*, a common limitation of existing semantic watermarking methods. 2) **A local-context-aligned semantic acceptance constraint for watermark embedding.** We propose *HammingMark*, which centers the Hamming acceptance neighborhood on the semantic code of the preceding sentence, allowing the watermark constraint to better align with naturally plausible continuations while preserving semantic flexibility. 3) **High-fidelity, efficient, and robust semantic watermarking.** Experiments on open-ended and constrained generation tasks show that HammingMark maintains strong detectability and robustness while substantially reducing semantic restriction and resampling overhead, requiring only 2.2 sampled candidates per accepted sentence—a 72.8% reduction compared with the most sampling-efficient existing method.

2 Related Work

SemStamp (Hou et al., 2024a) partitions the semantic space using locality-sensitive hashing, while K-SemStamp (Hou et al., 2024b) uses clustering-based regions. Both restrict generation to selected valid regions without accounting for the nonuniform, context-dependent distribution of natural continuations. These regions may therefore exclude much of the probability mass assigned to plausible continuations, causing semantic narrowing through misalignment with natural generation. SimMark (Dabiriaghdam & Wang, 2025) considers inter-sentence relations but requires consecutive sentences to satisfy a predefined similarity interval. However, natural discourse does not maintain a uniform similarity range: contrast, elaboration, and topic shifts may require continuations outside this interval, which the constraint consequently rejects. This interval-based criterion can also be sensitive to paraphrasing, as changes to both sentences may shift their measured similarity across the acceptance boundary even when their meanings are largely preserved.

PMark (Huo et al., 2026) introduces a proxy-function-based framework with multi-channel constraints. Its online variant offers strong theoretical guarantees through a distortion-free construction, but its substantial verification cost and demanding requirements for generator access, reproducible inference, and sentence-position alignment severely limit its practical applicability. Specifically, verification requires repeated sentence-level generation to reconstruct channel-wise medians, whose estimates may shift when prompts, context, or inference settings differ. Moreover, the strict sentence–key alignment requirement can render detection infeasible when sentence truncation or missing prompts disrupt this correspondence. A detailed discussion is provided in Appendix A. Its offline variant replaces dynamically estimated channel-wise medians with a fixed zero prior, eliminating generator queries but losing the exact distortion-free guarantee. Rather than rejecting candidates against an acceptance threshold, it selects from a sampled pool the candidate with the

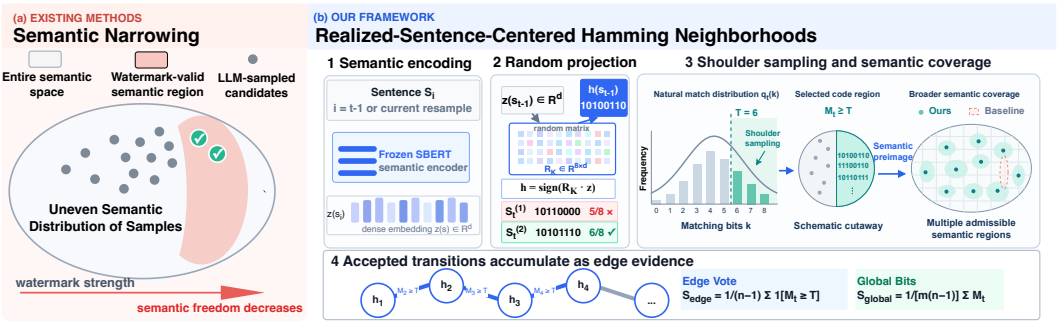


Figure 1: **Overview of HammingMark.** (a) Existing semantic watermarks may cause semantic narrowing when predefined watermark regions poorly overlap with natural generation. (b) HammingMark uses transition-centered Hamming neighborhoods for better alignment with natural generation, while coarse many-to-one hashing preserves diverse semantic transitions.

highest agreement with prescribed channel-wise random signs. The random projection matrix and target signs jointly impose fixed preferences in semantic space, concentrating selection probability on candidates favored by these preferences. Consequently, naturally plausible but lower-scoring continuations are systematically disadvantaged, inducing semantic narrowing even without explicit rejection sampling.

HammingMark defines watermark validity through a Hamming neighborhood centered on the preceding sentence's hash code, aligning the constraint with continuations naturally favored by the local context. Meanwhile, the coarse many-to-one mapping weakens the coupling between watermark validity and specific semantic content, allowing semantically distinct continuations to satisfy the same code-level constraint. Together, these mechanisms reduce semantic narrowing while preserving flexibility for contrast, elaboration, and topic shifts. A detailed analysis of semantic narrowing across methods is provided in Appendix B and C.

3 Method

3.1 Semantic Narrowing under Watermark Constraints

We characterize semantic narrowing in terms of the natural probability mass excluded by a watermark constraint.

At generation step t, let c_t denote the preceding context, and let $S_t \sim P_t(\cdot \mid c_t)$ denote a semantic continuation drawn from the context-dependent natural generation distribution. Here, $P_t(\cdot \mid c_t)$ captures the distribution of semantic continuations that the underlying language model would naturally produce given c_t. Let W_t denote the set of semantic continuations that satisfy the watermark constraint at step t.

We define the *natural acceptance mass* of the watermark constraint as

$$\alpha_t = \Pr_{S_t \sim P_t(\cdot|c_t)} [S_t \in W_t] . \tag{1}$$

The quantity α_t measures how much probability mass of the natural generation distribution remains compatible with the watermark constraint. Correspondingly, the excluded natural probability mass is $1 - \alpha_t$, which quantifies the degree of semantic narrowing induced by the watermark. A smaller α_t indicates poorer alignment between the watermark constraint and the natural distribution, causing more naturally likely continuations to be excluded.

3.2 HammingMark: Adaptive Hamming-Neighborhood Acceptance

We construct a context-dependent semantic acceptance rule based on transitions between compact hash codes. For each sentence S_t, we obtain a semantic embedding $e_t = F(S_t) \in \mathbb{R}^d$ and project it

using a fixed random-hyperplane matrix $R \in \mathbb{R}^{m \times d}$. The resulting semantic hash is

$$h_t = H(e_t) = \mathbf{1}[Re_t \geq 0], \qquad h_t \in \{0,1\}^m. \tag{2}$$

Random-hyperplane hashing preserves angular locality in probability: semantically similar embeddings are more likely to produce matching hash bits, providing a discrete but locality-sensitive representation of sentence semantics. A detailed analysis of this locality-preserving property is provided in Appendix D.

Given the preceding code h_{t-1}, we define the number of matching bits between two consecutive semantic hashes as $\mathrm{Match}(h_{t-1}, h_t) = m - d_H(h_{t-1}, h_t)$, where $d_H(\cdot,\cdot)$ denotes Hamming distance. HammingMark accepts the current sentence when

$$\mathrm{Match}(h_{t-1}, h_t) \geq T \quad \Longleftrightarrow \quad d_H(h_{t-1}, h_t) \leq r, \qquad r = m - T. \tag{3}$$

Accordingly, the valid next codes form the Hamming neighborhood

$$\mathcal{B}_r(h_{t-1}) = \{h \in \{0,1\}^m : d_H(h, h_{t-1}) \leq r\}. \tag{4}$$

Under the general formulation in Section 3.1, the natural acceptance mass of HammingMark is

$$\alpha_t^{\mathrm{HM}} = \Pr_{S_t \sim P_t(\cdot|c_t)} \left[H(F(S_t)) \in \mathcal{B}_r(h_{t-1})\right]. \tag{5}$$

During watermark generation, a candidate sentence S_t is accepted if $\mathrm{Match}(h_{t-1}, h_t) \geq T$ and is resampled otherwise. Thus, α_t^{HM} directly measures the natural probability mass retained by the HammingMark acceptance constraint.

The preceding hash h_{t-1} has two roles. First, it determines the center of the current Hamming acceptance constraint, making the watermark rule dependent on the realized preceding semantics rather than on an independently specified target. Second, it serves as one endpoint of the inter-sentence relation statistic used by the detector, so that the same pairwise semantic relation governs both watermark insertion and verification. HammingMark still performs semantic selection by rejecting candidates that violate equation 3; its advantage lies in how the acceptance constraint is placed and represented.

3.3 Why HammingMark Alleviates Semantic Narrowing

Under the formulation in Section 3.1, HammingMark alleviates semantic narrowing through two complementary mechanisms. First, the realized-sentence-centered Hamming constraint exploits the semantic continuity of natural sentence transitions through context-aligned shoulder sampling. Second, the coarse many-to-one hash representation allows semantically different continuations to remain valid under the same acceptance constraint.

Context-aligned shoulder sampling. As described in Section 3.2, HammingMark centers its acceptance constraint on the semantic hash of the preceding sentence. The key observation is that natural adjacent sentences typically exhibit semantic continuity, and random-hyperplane hashing preserves this locality in probability.

For a naturally sampled continuation S_t, define the natural match distribution and the corresponding HammingMark acceptance mass as

$$q_t(k) = \Pr_{S_t \sim P_t(\cdot|c_t)} \left[\mathrm{Match}\big(h_{t-1}, H(F(S_t))\big) = k\right], \qquad \alpha_t^{\mathrm{HM}} = \sum_{k=T}^{m} q_t(k). \tag{6}$$

To characterize this distribution, let e_{t-1} and e_t be normalized sentence embeddings with cosine similarity $\rho = e_{t-1}^\top e_t$. Under random-hyperplane hashing,

$$\mathrm{Match}(h_{t-1}, h_t) \mid \rho \sim \mathrm{Binomial}\left(m, 1 - \frac{\arccos(\rho)}{\pi}\right). \tag{7}$$

Therefore, $\Pr[\mathrm{Match}(h_{t-1}, h_t) \geq T \mid \rho]$ increases monotonically with semantic similarity. Since natural adjacent sentences already tend to exhibit semantic continuity, their match distribution is

biased toward moderate or high values. HammingMark thus performs a mild *shoulder sampling* over this existing distribution rather than directing generation toward an unrelated semantic target. Importantly, this shoulder truncation operates in the discrete code space rather than imposing a hard cutoff in the underlying semantic space.

For example, with $m = 8$ and $T = 6$, candidates producing four, five, or six matching bits are already relatively close in hash similarity. Selecting candidates with six or more matching bits therefore introduces a local preference along an existing natural transition tendency, rather than imposing a qualitatively different semantic direction.

Semantic diversity under coarse hashing. High natural acceptance alone does not guarantee that watermark-valid continuations are semantically diverse. HammingMark additionally exploits the many-to-one nature of compact semantic hashing. The mapping $H : \mathcal{Z} \to \{0,1\}^m$ compresses a high-dimensional semantic representation into a short binary code, so multiple distinct semantic representations may map to the same hash code.

For two normalized semantic representations z and z' with angular distance θ, random-hyperplane hashing gives

$$d_H(H(z), H(z')) \sim \text{Binomial}\left(m, \frac{\theta}{\pi}\right), \tag{8}$$

and therefore

$$\Pr[d_H(H(z), H(z')) \leq r] = \sum_{j=0}^{r} \binom{m}{j} \left(\frac{\theta}{\pi}\right)^j \left(1 - \frac{\theta}{\pi}\right)^{m-j}. \tag{9}$$

Semantic proximity increases the probability of nearby hash codes, but the converse need not hold for a short code. Since the compact hash provides only a sparse and coarse representation of the underlying semantic space, a higher bit match does not necessarily imply greater semantic similarity than a lower bit match. Importantly, equation 9 shows that even when two semantic representations are less similar, there can still be a nonzero probability that their hash codes fall within the same Hamming neighborhood. Therefore, although HammingMark performs shoulder sampling in the hash space, it does not explicitly filter out continuations solely because they have lower semantic similarity. Since the mapping is many-to-one, semantically different continuations may share the same code or map to different codes that are simultaneously accepted by the Hamming constraint. Watermark validity therefore does not correspond to a single semantic realization or direction. This flexibility is particularly useful for contrast, elaboration, topic shifts, and the introduction of new information, where a continuation may differ substantially from the preceding sentence while still satisfying the watermark rule. Appendix E provides a formal analysis of this many-to-one property.

Together, these two mechanisms address complementary sources of semantic restriction. Context-aligned shoulder sampling makes watermark selection follow an existing tendency of natural semantic transitions rather than imposing an unrelated semantic target, while coarse many-to-one hashing preserves semantic diversity among watermark-valid continuations.

3.4 Sequence-Level Detection

Given a text containing n sentences $S_1, \ldots, S_n$, the detector recovers their semantic hashes and adjacent matching scores as

$$h_t = H(F(S_t)), \qquad M_t = \text{Match}(h_{t-1}, h_t), \quad t = 2, \ldots, n. \tag{10}$$

We use two complementary sequence-level statistics.

Edge Vote. Each adjacent sentence pair is treated as one transition unit. A transition casts a valid vote when $M_t \geq T$, and the final score is

$$S_{\text{edge}} = \frac{1}{n-1} \sum_{t=2}^{n} \mathbf{1}[M_t \geq T]. \tag{11}$$

Global Bits. Instead of binarizing each transition, Global Bits aggregates all matching bits,

$$S_{\text{global}} = \frac{1}{m(n-1)} \sum_{t=2}^{n} M_t. \tag{12}$$

It therefore retains partial evidence from transitions that no longer exceed T after semantic-preserving perturbations.

For either detector $d \in \{\text{edge}, \text{global}\}$, a text is classified as watermarked when $S_d > \tau_d$, where τ_d is calibrated from unwatermarked text at the target false-positive rate. Detection requires only the fixed semantic encoder and projection matrix, without access to the generation model.

4 EXPERIMENTS

We organize our evaluation around the following research questions. **RQ1 (Detectability):** Can HammingMark be reliably distinguished from unwatermarked text across datasets and language models? **RQ2 (Robustness):** Does the watermark remain detectable after local edits, sentence-level paraphrasing, and document-level rewriting? **RQ3 (Text quality):** How well does HammingMark preserve the quality of generated text? **RQ4 (Sampling efficiency and semantic overlap):** How well does the watermark-valid region overlap with naturally preferred continuations, and how does this affect rejection-sampling efficiency? **RQ5 (Complex tasks):** Does HammingMark remain effective on conditional, long-form generation tasks? **RQ6 (Semantic sampling visualization):** How do different semantic watermarking methods constrain naturally sampled candidate continuations in the semantic space?

4.1 RQ1–RQ3: DETECTABILITY, ROBUSTNESS, AND QUALITY

HammingMark maintains near-perfect clean-text detection across all model–dataset settings, with $\text{TPR@1}\%$ of 0.97–1.00 and AUROC of 0.99–1.00. Edge Vote is particularly effective on clean text because it directly measures whether each transition satisfies the embedding rule.

Under paraphrasing and rewriting, Global Bits is generally more robust, achieving $\text{TPR@1}\%$ of 0.82–0.97 across all settings. Unlike Edge Vote, which discards a transition once it falls below the matching threshold, Global Bits retains partial evidence from modified transitions by aggregating all matching bits. Therefore, Edge Vote is preferable for ordinary clean-text verification, whereas Global Bits is better suited to attacked text.

HammingMark also preserves generation quality, with PPL differing from unwatermarked generation by at most 0.30 across all settings. This advantage follows from the relational Hamming-ball constraint: instead of forcing each sentence into a fixed semantic region, HammingMark retains a large set of admissible semantic realizations. The model can therefore continue exploring plausible continuations and select content appropriate to the context while embedding a detectable transition-level signal.

4.2 RQ4: SAMPLING EFFICIENCY AND NATURAL ACCEPTANCE

As shown in Table 2, HammingMark requires only 2.2 sampled candidates and 41.58 generated tokens for each accepted sentence. Compared with SimMark, the most efficient baseline, HammingMark reduces the number of sampled candidates by approximately 72.8% and token consumption by 77.7%.

To examine the source of this sampling efficiency, we estimate the natural acceptance mass defined in Section 3.1. For each of 100 prompts, we independently sample 300 candidate continuations from the unwatermarked model before applying any watermark constraint. These candidates are therefore samples from the context-dependent natural generation distribution $P_t(\cdot \mid c_t)$. We then apply each watermarking rule to the sampled candidates and measure the fraction that would be accepted without resampling.

Table 1: **Clean detectability, robustness, and generation quality on C4 and BookSum.** Each detection entry reports TPR@1%/TPR@5%/AUROC.

Method	Clean	Parrot	Pegasus	DIPPER	API	PPL ↓
Mistral-7B (Jiang et al., 2023) on C4 (Raffel et al., 2020)						
No watermark	–	–	–	–	–	4.51
KGW (Kirchenbauer et al., 2023)	**1.00/1.00/1.00**	0.59/0.72/0.90	0.51/0.63/0.89	0.43/0.49/0.84	0.31/0.39/0.80	8.00
SynthID (Dathathri et al., 2024)	**1.00/1.00/1.00**	0.32/0.45/0.82	0.23/0.42/0.80	0.35/0.40/0.80	0.24/0.38/0.79	**4.52**
MorphMark (Wang et al., 2025)	0.99/1.00/1.00	0.54/0.60/0.88	0.47/0.55/0.84	0.51/0.55/0.86	0.53/0.64/0.89	6.62
SIR (Liu et al., 2024)	0.98/1.00/1.00	0.51/0.63/0.91	0.56/0.60/0.90	0.44/0.57/0.84	0.38/0.46/0.82	8.35
K-SemStamp (Hou et al., 2024b)	0.83/0.88/0.98	0.43/0.58/0.86	0.46/0.65/0.89	0.66/0.87/0.95	0.31/0.40/0.80	10.78
SemStamp (Hou et al., 2024a)	0.82/0.87/0.97	0.56/0.77/0.91	0.79/0.90/0.97	0.76/0.94/0.98	0.32/0.48/0.84	9.36
SimMark (Dabiriaghdam & Wang, 2025)	0.71/0.82/0.95	0.28/0.43/0.82	0.35/0.51/0.84	0.23/0.41/0.80	0.25/0.36/0.78	8.34
PMark (Huo et al., 2026)	0.99/0.99/1.00	0.94/0.97/0.99	0.89/0.91/0.98	0.75/0.85/0.97	0.91/0.93/0.98	5.13
HammingMark (Edge Vote)	0.99/1.00/1.00	0.92/0.96/0.99	0.85/0.98/0.99	0.89/0.97/0.99	0.90/0.93/0.98	4.61
HammingMark (Global Bits)	0.98/1.00/1.00	**0.94/0.99/1.00**	**0.91/1.00/0.99**	**0.93/0.98/0.99**	**0.94/0.97/0.99**	4.61
Mistral-7B on BookSum (Kryściński et al., 2022)						
No watermark	–	–	–	–	–	6.20
KGW	**1.00/1.00/1.00**	0.34/0.43/0.86	0.40/0.45/0.82	0.35/0.41/0.81	0.16/0.23/0.64	10.14
SynthID	**1.00/1.00/1.00**	0.15/0.21/0.69	0.10/0.16/0.60	0.11/0.19/0.62	0.13/0.18/0.60	6.57
MorphMark	**1.00/1.00/1.00**	0.45/0.52/0.83	0.31/0.37/0.79	0.42/0.55/0.81	0.22/0.27/0.65	9.33
SIR	**1.00/1.00/1.00**	0.54/0.61/0.85	0.47/0.63/0.87	0.52/0.61/0.83	0.14/0.25/0.65	12.49
K-SemStamp	0.79/0.91/0.98	0.51/0.64/0.88	0.56/0.69/0.90	0.70/0.86/0.94	0.16/0.27/0.69	15.72
SemStamp	0.88/0.93/0.98	0.67/0.71/0.90	0.71/0.78/0.90	0.82/0.91/0.97	0.20/0.35/0.71	14.98
SimMark	0.77/0.86/0.96	0.29/0.52/0.81	0.44/0.58/0.84	0.21/0.43/0.80	0.14/0.21/0.62	14.25
PMark	0.97/0.99/0.99	**0.96/0.98/0.99**	0.79/0.91/0.98	0.78/0.93/0.98	0.71/0.84/0.94	8.26
HammingMark (Edge Vote)	**1.00/1.00/1.00**	0.91/0.94/0.99	0.86/0.95/0.99	0.91/0.94/0.98	0.88/0.92/0.98	**6.40**
HammingMark (Global Bits)	**1.00/1.00/1.00**	0.93/0.95/**0.99**	**0.90/0.99/0.99**	**0.91/0.94/0.99**	**0.92/0.95/0.99**	**6.40**
Qwen2.5-7B (Yang et al., 2024) on C4						
No watermark	–	–	–	–	–	5.52
KGW	**0.99/1.00/1.00**	0.08/0.13/0.66	0.08/0.11/0.67	0.06/0.12/0.61	0.02/0.08/0.49	7.29
SynthID	**0.99/1.00/1.00**	0.06/0.10/0.50	0.02/0.04/0.47	0.03/0.06/0.48	0.02/0.06/0.48	**5.30**
MorphMark	0.98/1.00/1.00	0.04/0.20/0.65	0.06/0.17/0.64	0.02/0.21/0.64	0.03/0.11/0.53	5.63
SIR	0.97/1.00/1.00	0.55/0.59/0.89	0.62/0.68/0.92	0.41/0.58/0.85	0.34/0.40/0.79	7.08
K-SemStamp	0.78/0.93/0.95	0.42/0.65/0.86	0.49/0.73/0.89	0.66/0.90/0.95	0.17/0.29/0.77	11.67
SemStamp	0.85/0.88/0.97	0.59/0.82/0.91	0.61/0.83/0.95	0.74/0.96/0.97	0.24/0.37/0.82	9.46
SimMark	0.72/0.81/0.94	0.27/0.50/0.82	0.38/0.52/0.86	0.29/0.41/0.78	0.31/0.46/0.83	8.11
PMark	0.99/1.00/0.99	0.96/0.98/0.99	0.85/0.91/0.96	0.91/0.94/0.97	0.83/0.89/0.98	6.32
HammingMark (Edge Vote)	0.98/0.99/0.99	0.93/0.94/0.98	**0.89/0.93/0.98**	0.92/0.94/0.98	0.94/0.98/0.99	5.38
HammingMark (Global Bits)	0.98/0.99/0.99	**0.97/0.98/0.99**	**0.89/0.93/0.98**	**0.94/0.95/0.99**	**0.95/0.98/0.99**	5.38
Qwen2.5-7B on BookSum						
No watermark	–	–	–	–	–	4.45
KGW	**0.99/1.00/1.00**	0.05/0.13/0.65	0.05/0.19/0.62	0.02/0.12/0.62	0.03/0.12/0.53	8.46
SynthID	**0.99/1.00/1.00**	0.03/0.08/0.54	0.05/0.12/0.56	0.00/0.09/0.55	0.01/0.08/0.51	**4.63**
MorphMark	0.97/0.99/1.00	0.02/0.05/0.57	0.02/0.07/0.55	0.02/0.06/0.56	0.01/0.09/0.51	7.31
SIR	0.95/0.98/0.99	0.54/0.63/0.84	0.60/0.67/0.91	0.46/0.66/0.85	0.10/0.23/0.72	9.76
K-SemStamp	0.82/0.91/0.96	0.44/0.70/0.88	0.53/0.61/0.84	0.68/0.75/0.90	0.14/0.28/0.75	10.30
SemStamp	0.87/0.90/0.97	0.57/0.84/0.92	0.45/0.71/0.89	0.70/0.78/0.91	0.17/0.33/0.76	11.49
SimMark	0.71/0.85/0.93	0.31/0.43/0.80	0.36/0.54/0.83	0.24/0.40/0.81	0.13/0.21/0.72	10.28
PMark	0.95/0.99/0.99	**0.92/0.98/0.99**	0.74/0.92/0.96	0.78/0.89/0.97	0.78/0.85/0.96	6.39
HammingMark (Edge Vote)	**0.99/1.00/1.00**	0.87/0.93/0.98	0.83/0.89/0.97	0.82/0.92/0.97	0.89/0.94/0.98	4.75
HammingMark (Global Bits)	0.97/0.99/1.00	0.93/0.95/0.99	**0.84/0.91/0.97**	**0.82/0.94/0.98**	**0.91/0.94/0.98**	4.75

Table 2: Sampling efficiency and estimated natural acceptance on C4. Lower sampling cost and higher natural acceptance are better. Random Code uses the same code-space coverage as HammingMark ($37/256$) but selects valid codes uniformly at random.

Method	SimMark	SemStamp	K-SemStamp	PMark	Random Code	HammingMark
Samples / sent. ↓	8.1	99.3	13.3	64.0	14.7	**2.2**
Tokens / sent. ↓	186.7	1694.4	246.9	1185.8	273.1	**41.58**
Natural acceptance $\widehat{\bar{\alpha}}$ ↑	0.34	0.04	0.13	/	0.15	**0.58**

For a given context c_t, we estimate the natural acceptance mass by

$$\hat{\alpha}_t = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}\left[S_t^{(i)} \in W_t\right], \qquad S_t^{(i)} \sim P_t(\cdot \mid c_t). \tag{13}$$

Table 3: Performance on complex generation tasks on Mistral-7B. Detection results are reported as TPR@1%, TPR@5%, and AUROC.

Method	ELI5 (Fan et al., 2019)				Multi-News (Fabbri et al., 2019)			
	@1 ↑	@5 ↑	AUC ↑	Rouge-L ↑	@1 ↑	@5 ↑	AUC ↑	Rouge-L ↑
No watermark	–	–	–	0.34	–	–	–	0.36
KGW	0.78	0.89	0.97	0.29	0.21	0.49	0.86	0.31
SynthID	0.43	0.67	0.92	0.31	0.08	0.17	0.64	**0.36**
MorphMark	0.49	0.64	0.90	0.29	0.26	0.41	0.88	0.35
SIR	0.63	0.75	0.93	0.28	0.21	0.36	0.85	0.34
SemStamp	0.22	0.36	0.82	0.31	0.03	0.19	0.64	0.35
K-SemStamp	0.13	0.24	0.71	0.32	0.03	0.16	0.64	0.33
SimMark	0.26	0.41	0.84	0.31	0.06	0.11	0.62	0.34
PMark	0.85	0.91	0.98	0.29	0.86	0.92	0.99	0.32
HammingMark	**0.95**	**0.95**	**0.99**	**0.33**	**0.91**	**1.00**	**0.99**	**0.36**

where $N = 300$ is the number of independently sampled natural candidates. We report $\widehat{\widehat{\alpha}}$, obtained by averaging $\hat{\alpha}_t$ across the 100 evaluation prompts. Thus, the values reported in Table 2 are direct Monte Carlo estimates of the natural probability mass retained by each watermark constraint. A larger natural acceptance mass means that a greater fraction of continuations produced by the unwatermarked model already satisfies the watermark rule. This quantity is directly related to rejection-sampling cost: when natural candidates are accepted more frequently, fewer candidates need to be sampled before obtaining a watermark-valid continuation. HammingMark achieves an estimated natural acceptance of 0.58, meaning that approximately 58% of naturally sampled continuations satisfy its watermark constraint without additional semantic redirection. This is consistent with its substantially lower sampling cost. We further evaluate the loss of semantic diversity induced by each method in Appendix F and G, complementing the natural acceptance analysis with a direct assessment of diversity preservation.

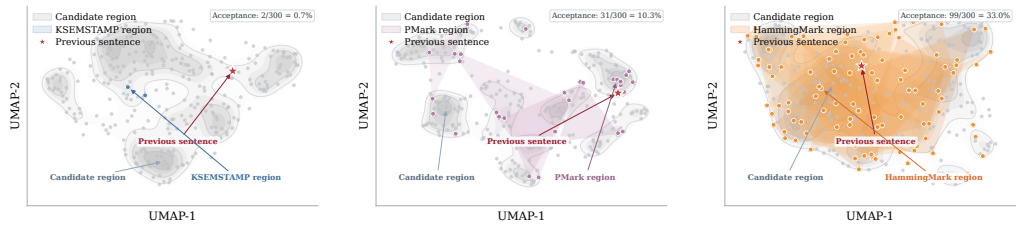
4.3 RQ5: Performance on Complex Generation Tasks

As shown in Table 3, HammingMark achieves TPR@1% of 0.95 on ELI5 and 0.91 on MultiNews while preserving the highest or tied-highest ROUGE-L. These tasks strongly constrain generation semantics, concentrating the natural distribution $P_t(\cdot \mid c_t)$ on task-consistent continuations and leaving little room for semantic redirection. Poorly aligned watermark constraints can therefore reject many high-probability candidates, yielding a small natural acceptance mass α_t and frequent resampling. HammingMark mitigates this issue through mild shoulder sampling around the realized semantic transition. Semantically coherent continuations are more likely to have high hash similarity, while the many-to-one hash mapping allows multiple semantic realizations to satisfy the same Hamming constraint. This preserves more natural probability mass, helping maintain task fidelity while accumulating a detectable watermark signal.

4.4 RQ6: Visualizing Natural Acceptance under Semantic Constraints

To provide an intuitive candidate-level view of natural acceptance, we independently sample 300 continuations from the unwatermarked model for a representative challenging prompt and project their sentence embeddings into two dimensions with UMAP. For each method, candidates satisfying its watermark constraint are highlighted within the same natural sample set. The highlighted fraction thus gives a prompt-level Monte Carlo estimate of the natural acceptance mass.

As shown in Figure 2, K-SemStamp accepts only 2 of the 300 natural candidates, while PMark accepts 31 and HammingMark accepts 99. Thus, for this prompt, HammingMark retains a substantially larger fraction of the continuations that the unwatermarked model naturally produces. The highlighted HammingMark candidates are also distributed across multiple parts of the projected sample cloud, illustrating that watermark-valid candidates are not confined to a single local semantic realization. This visualization complements the aggregate natural-acceptance results in Table 2. Importantly, the UMAP geometry is used only for qualitative illustration: semantic narrowing is quantified by the fraction of natural probability mass retained by the watermark constraint, rather than by the geometric area occupied by highlighted points in the two-dimensional projection.



(a) KSEMSTAMP resampling dimensionality reduction (b) PMark resampling dimensionality reduction (c) HammingMark resampling dimensionality reduction

Figure 2: **Visualization of natural candidates accepted by different watermark constraints.** UMAP projection of 300 independently sampled continuations from the unwatermarked model for a representative challenging prompt. All methods are compared under matched watermark effectiveness. For offline PMark, we highlight candidates matching all key-specified channel signs under the zero-median prior. Highlighted points denote candidates accepted by the watermark constraint.

4.5 Ablation Study

Effect of constraint placement. We examine whether HammingMark's higher natural acceptance is due simply to the number of valid hash codes or to how these codes are placed.

With $m = 8$ and $T = 6$, HammingMark accepts $\sum_{j=0}^{2} \binom{8}{j} = 37$ of the 256 possible hash codes. As a control, we construct a Random Code baseline that also accepts 37 codes, but selects them uniformly at random and independently of the preceding sentence. All other components and evaluation settings remain unchanged.

As shown in Table 2, Random Code achieves a natural acceptance of 0.15, whereas HammingMark reaches 0.58, despite identical code-space coverage. For a random set $\mathcal{A}$ of 37 codes, each code is included with probability $37/256$. Therefore, for any natural hash distribution $p_t(h)$,

$$\mathbb{E}_{\mathcal{A}}\left[\alpha_t(\mathcal{A})\right] = \sum_h p_t(h) \Pr_{\mathcal{A}}[h \in \mathcal{A}] = \frac{37}{256} \approx 0.1445. \tag{14}$$

The observed value of 0.15 is close to this random-set expectation.

In contrast, HammingMark places the same number of valid codes around the preceding sentence hash. Combined with the locality-preserving property of random-hyperplane hashing, this transition-centered placement makes the constraint better aligned with natural semantic transitions. The large gap between 0.15 and 0.58 therefore shows that natural acceptance depends strongly on constraint placement rather than code-space coverage alone.

5 Limitations

HammingMark depends on the quality of the semantic encoder and sentence segmentation; unstable embeddings or ambiguous sentence boundaries may alter hash codes and reduce detection reliability. Although HammingMark substantially improves sampling efficiency, it remains a rejection-sampling method and may incur additional cost under highly restrictive contexts or strict matching thresholds.

6 Conclusion

This work identifies *semantic narrowing* as the misalignment between watermark constraints and naturally preferred generation. We propose *HammingMark*, a realized-sentence-centered semantic watermarking method that reduces this mismatch through context-aligned Hamming acceptance while preserving semantic flexibility. Across open-ended and constrained generation tasks, HammingMark achieves strong detectability and robustness with near-unwatermarked quality and low sampling cost.

AI USE STATEMENT

In this work, generative AI tools were used to assist with language polishing, manuscript drafting and revision, formatting, and literature search. All AI-assisted outputs, including retrieved information and suggested text, were critically reviewed, verified, and revised by the authors. The authors take full responsibility for the accuracy, integrity, and final content of the paper. We take responsibility for the final content of this work, including any text or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work does not involve human-subject studies or the collection of personally identifiable or sensitive information. Our experiments are conducted using publicly available or appropriately licensed data and standard evaluation protocols. We do not intend the proposed methods to facilitate harmful, discriminatory, privacy-invasive, or otherwise unethical applications. Nevertheless, as with many machine learning methods, the proposed approach may inherit biases or limitations from the data and models on which it relies, and we encourage careful evaluation before deployment in safety- or socially sensitive settings. We have made our best effort to ensure research integrity, appropriate attribution, and compliance with relevant ethical and legal requirements. The authors declare no conflicts of interest that would affect the presentation or interpretation of the results.

REPRODUCIBILITY STATEMENT

We have made efforts to ensure the reproducibility of our results. The appendix provides the complete experimental setup and all details necessary to reproduce our experiments, including implementation details, evaluation configurations, hyperparameter settings, data processing procedures, and other relevant experimental specifications. We also provide additional results and analyses where appropriate to facilitate verification of the reported findings. Together, these materials are intended to enable independent reproduction of the main results presented in this work. The source code will be made publicly available upon acceptance of the paper.

REFERENCES

Lukas Aichberger, Kajetan Schweighofer, Mykyta Ielanskyi, and Sepp Hochreiter. Improving uncertainty estimation through semantically diverse language generation. In *International Conference on Learning Representations*, volume 2025, pp. 74479–74503, 2025.

Stephen Casper, Kyle O'Brien, Shayne Longpre, Elizabeth Seger, Kevin Klyman, Rishi Bommasani, Aniruddha Nrusimha, Ilia Shumailov, Sören Mindermann, Steven Basart, et al. Open technical problems in open-weight ai model risk management. *arXiv preprint arXiv:2608.07514*, 2026.

Amirhossein Dabiriaghdam and Lele Wang. 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*, pp. 30773–30794, 2025.

Prithiviraj Damodaran. Parrot: Paraphrase generation for nlu., 2021.

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

Alexander Richard Fabbri, Irene Li, Tianwei She, Suyi Li, and Dragomir Radev. Multi-news: A large-scale multi-document summarization dataset and abstractive hierarchical model. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 1074–1084, 2019.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. Eli5: Long form question answering. In *Proceedings of the 57th annual meeting of the association for computational linguistics*, pp. 3558–3567, 2019.

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

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

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

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 45408–45436, 2024.

Jiahao Huo, Shuliang Liu, Bin Wang, Junyan Zhang, Yibo Yan, Aiwei Liu, Xuming Hu, and Mingxun Zhou. Pmark: Towards robust and distortion-free semantic-level watermarking with channel constraints. In *International Conference on Learning Representations*, volume 2026, pp. 27100–27126, 2026.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL `https://arxiv.org/abs/2310.06825`.

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

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of ai-generated text, but retrieval is an effective defense. *Advances in neural information processing systems*, 36:27469–27500, 2023.

Wojciech Kryściński, Nazneen Rajani, Divyansh Agarwal, Caiming Xiong, and Dragomir Radev. Booksum: A collection of datasets for long-form narrative summarization. In *Findings of the association for computational linguistics: EMNLP 2022*, pp. 6536–6558, 2022.

Yuqing Liang, Jiancheng Xiao, Wensheng Gan, and Philip S Yu. Watermarking techniques for large language models: A survey. *Artificial Intelligence Review*, 59(2):74, 2026.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *International Conference on Learning Representations*, volume 2024, pp. 6499–6519, 2024.

Shuliang Liu, Xingyu Li, Hongyi Liu, Dong Fang, Yibo Yan, Bingchen Duan, Qi Zheng, Lingfeng Su, and Xuming Hu. Distilling the thought, watermarking the answer: A principle semantic guided watermark for large reasoning models. *arXiv preprint arXiv:2601.05144*, 2026.

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

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023.

Leyi Pan, Aiwei Liu, Zhiwei He, Zitian Gao, Xuandong Zhao, Yijian Lu, Binglin Zhou, Shuliang Liu, Xuming Hu, Lijie Wen, et al. Markllm: An open-source toolkit for llm watermarking. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 61–71, 2024.

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

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

Zewen Sun, Qian Jiang, Siyuan Sheng, and Liyao Xiang. Watermoe: Expert-routing-based watermarking for high fidelity and efficiency. *arXiv preprint arXiv:2607.13099*, 2026.

Zongqi Wang, Tianle Gu, Baoyuan Wu, and Yujiu Yang. 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)*, pp. 4842–4860, 2025.

Anyi Xu, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, Chenchen Ling, et al. Deepseek-v4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*, 2026.

Zhenhua Xu, Xubin Yue, Zhebo Wang, Haobo Zhang, Qichen Liu, Xixiang Zhao, Jingxuan Zhang, Wenjun Zeng, Wengpeng Xing, Dezhang Kong, et al. Copyright protection for large language models: A survey of methods, challenges, and trends. *arXiv preprint arXiv:2508.11548*, 2025.

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, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

Zhiguang Yang, Gejian Zhao, and Hanzhou Wu. Watermarking for large language models: A survey. *Mathematics*, 13(9):1420, 2025.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter Liu. Pegasus: Pre-training with extracted gap-sentences for abstractive summarization. In *International conference on machine learning*, pp. 11328–11339. PMLR, 2020.

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

A Practical Limitations of Online PMark

PMark (Huo et al., 2026) makes an important theoretical contribution to semantic watermarking by introducing a proxy-function framework with a distortion-free online construction. At each generation step, it samples candidate sentences and recursively partitions them into equally sized subsets using channel-wise empirical medians. Under the prescribed random channel decisions and balanced partitioning, each sampled candidate has the same marginal selection probability. Averaging over these decisions and candidate sampling therefore recovers the original next-sentence distribution. This provides a principled foundation for semantic watermarking. However, this generation-side guarantee does not by itself ensure inexpensive, reproducible, or context-independent verification.

Generator-dependent verification and computational cost. Online PMark requires the detector to resample N candidate continuations for each sentence under its preceding context and estimate channel-wise medians from these samples. For D documents containing L sentences each, this entails approximately DLN sentence generations, in addition to semantic encoding and evidence

aggregation. Verification therefore requires access to the original generator or a sufficiently faithful generation service, making large-scale screening costly and limiting deployment by independent verifiers. Hosting the generator also introduces substantial memory and inference requirements; relying on a remote service instead introduces availability, latency, and version-control dependencies.

Sensitivity to changes in the generation distribution. The published detector uses fresh samples rather than requiring exact replay of the original candidate pool. Consequently, a change in the sampling seed alone does not invalidate the distortion-free generation guarantee. Nevertheless, reliable median reconstruction depends on reproducing the relevant conditional generation distribution. Model updates or fine-tuning, changes in prompt formatting or decoding settings, and numerical differences that affect sampling can alter this distribution and its proxy medians. Even under unchanged settings, finite-sample estimation introduces discrepancies between generation and detection. PMark uses soft counting to mitigate such discrepancies, but this mechanism does not guarantee invariance to systematic changes in the generator or its inference configuration.

Key reuse and repeated-query limitations. PMark explicitly acknowledges in its Appendix D that its predefined, position-indexed random seeds leave n-shot undetectability unresolved, and restricts its guarantee to single-shot distortion-freeness (Huo et al., 2026). Specifically, averaging over random watermark decisions recovers the natural next-sentence distribution, but this does not establish that repeated responses under a reused key follow the joint distribution of independent unwatermarked samples. When the same prompt is queried repeatedly with the same key sequence, the corresponding channel preferences are repeatedly imposed rather than independently randomized across responses. Consequently, the single-shot guarantee does not ensure preservation of natural output variability in repeated-query settings. This is an acknowledged limitation of the construction: PMark leaves dynamic semantic-level seed generation for addressing this issue to future work.

Dependence on prompts and sentence–key alignment. For sentence s_t, the online detector reconstructs medians from the conditional distribution

$$P_M(\cdot \mid \pi, s_1, \ldots, s_{t-1}),$$

while evaluating watermark evidence with the corresponding position-indexed channel keys and prescribed proxy directions. This requires both the appropriate preceding context and the correct sentence index. When a circulated text contains only the generated response, the original prompt is unavailable, preventing faithful reconstruction of the conditioning context. Even when prompt text is included, an unmarked prompt–response boundary leaves the starting key index ambiguous. Prefix truncation or sentence insertion and deletion can further change the available context and shift subsequent sentence–key correspondence. These operations can occur through ordinary quotation or editing, without substantial semantic rewriting. The published detection procedure provides no explicit mechanism for recovering missing context or resynchronizing unknown sentence offsets. Searching over possible offsets would add computational cost and require appropriate false-positive calibration, while still not restoring missing conditioning information.

Overall, online PMark offers a valuable theoretical construction, but its practical verification remains tightly coupled to the generator, conditioning context, and sentence indexing used during generation. These dependencies substantially limit its suitability for scalable detection of independently circulated text.

B Semantic Narrowing and Watermark Effectiveness

Natural acceptance mass. Fix a generation context c_t and let z denote a normalized sentence embedding with natural distribution $p_t(z) = p(z \mid c_t)$. For a rejection-sampling watermark, let $a_t(z) \in \{0, 1\}$ be its acceptance rule under a fixed key and context. Define

$$\alpha_t = \int p_t(z) a_t(z)\, dz, \qquad N_t = 1 - \alpha_t. \tag{15}$$

Integrals are understood with respect to the embedding distribution, including discrete distributions. The excluded natural probability mass N_t is an operational measure of semantic narrowing; it is not

the proportional decrease in semantic diversity, entropy, or generation quality. For $\alpha_t > 0$, ideal rejection sampling with unlimited budget induces

$$p_t^{\mathrm{WM}}(z) = \frac{p_t(z)a_t(z)}{\alpha_t}. \tag{16}$$

We initially omit robustness margins and fallback, and introduce a finite budget below. Each method may use a different encoder and thus a different embedding distribution. Geometric reference calculations therefore cannot replace comparisons on common candidate texts from shared contexts.

B.1 Semantic Narrowing in Partition- and Similarity-Based Methods

SemStamp and K-SemStamp. SemStamp (Hou et al., 2024a) partitions embeddings using random hyperplane hashing, whereas K-SemStamp (Hou et al., 2024b) uses clustering. Both select valid partitions pseudorandomly. Ignoring margins, let $C_1, \ldots, C_K$ be the partitions and G_t the valid index set. Then

$$p_{t,j} = \int_{C_j} p_t(z)\, dz, \qquad \alpha_t = \sum_{j \in G_t} p_{t,j}, \qquad N_t = 1 - \sum_{j \in G_t} p_{t,j}. \tag{17}$$

Their default valid-partition ratio is $\gamma = 0.25$; SemStamp uses three hash bits (eight codes), and K-SemStamp uses eight clusters. If partitions have equal natural mass, then $\alpha_t = \gamma = 0.25$ and $N_t = 0.75$. Without equal masses, an ideal uniform random choice of γK valid partitions still gives $\mathbb{E}_G[\alpha_t] = \gamma$. Neither statement implies that a particular fixed valid set covers this mass. In particular, a uniform angle relative to the preceding sentence does not establish equal partition masses.

Natural continuations can concentrate in a few partitions. A valid set that misses these partitions may have very low acceptance mass, whereas one that includes them may have high mass. Increasing the number of valid partitions increases expected coverage under uniform selection, but need not yield appropriate partial coverage in every context. Nonuniformity does not necessarily reduce acceptance; the concern is context-dependent misalignment and uneven coverage.

Cosine-SimMark. For the cosine-similarity variant of SimMark (Dabiriaghdam & Wang, 2025), write $\theta = \arccos(z_{t-1}^\top z)$ and let $f_t(\theta)$ be its context-dependent density. Omitting fallback, an accepted cosine interval $[a, b] \subseteq [-1, 1]$ gives

$$\alpha_t^{\mathrm{Sim}} = \int_{\arccos b}^{\arccos a} f_t(\theta)\, d\theta. \tag{18}$$

Under the reference model $\theta \sim \mathrm{Uniform}(0, \pi)$,

$$\alpha_{\mathrm{unif}}^{\mathrm{Sim}} = \frac{\arccos a - \arccos b}{\pi} \approx 3.68\%, \qquad N_{\mathrm{unif}}^{\mathrm{Sim}} \approx 96.32\%, \tag{19}$$

using the reported main-experiment interval $[0.68, 0.76]$. A uniform angle is neither uniform cosine similarity nor a uniform distribution on a high-dimensional sphere. These values are reference calculations, not measured semantic losses. A fixed interval may miss the dominant mass in some contexts; actual acceptance can be above or below the reference value. The formula concerns angles in the normalized representation used for cosine scoring; it does not directly describe Euclidean-SimMark or angles before a PCA transformation.

B.2 Semantic Narrowing under Hamming-Neighborhood Acceptance

Fixed-key acceptance. For a fixed projection matrix R, HammingMark accepts according to

$$a_t^{\mathrm{HM}}(z) = \mathbb{I}[\mathrm{Match}(H_R(z_{t-1}), H_R(z)) \geq T], \qquad \alpha_t^{\mathrm{HM}} = \int p_t(z) a_t^{\mathrm{HM}}(z)\, dz, \tag{20}$$

with $N_t^{\mathrm{HM}} = 1 - \alpha_t^{\mathrm{HM}}$. For $m = 8, T = 6$, the accepted fraction of binary codes is

$$\gamma_{\mathrm{code}} = \frac{\binom{8}{6} + \binom{8}{7} + \binom{8}{8}}{2^8} = \frac{37}{256} \approx 14.45\%. \tag{21}$$

This counts codes, whereas α_t^{HM} weights them by natural generation probability. Thus $37/256$ is neither the actual acceptance mass nor, in general, the detection null hit probability.

Table 4: Idealized reference calculations, with margins omitted. The rows use different assumptions and averaging operations; they explain mechanisms rather than rank performance on one real generation distribution.

Method	Reference assumption	Accepted	Excluded
SemStamp	Equal partition masses	25%	75%
K-SemStamp	Equal partition masses	25%	75%
Cosine-SimMark	Uniform angle; $[a, b] = [0.68, 0.76]$	3.68%	96.32%
Random Code	Uniform selection of 37 codes; set average	14.45%	85.55%
HammingMark	Uniform angle; projection average	33.33%	66.67%

Random-projection reference. Hold the context, preceding embedding, and natural candidate distribution fixed independently of the projection draw. For independent isotropic random hyperplanes and a fixed pair at angle θ,

$$M \mid \theta \sim \mathrm{Binomial}(m, 1 - \theta/\pi), \qquad A_{\mathrm{HM}}(\theta) = \Pr_R(M \geq T \mid \theta). \tag{22}$$

Consequently,

$$\mathbb{E}_R[\alpha_t^{\mathrm{HM}}] = \int_0^\pi f_t(\theta) A_{\mathrm{HM}}(\theta)\, d\theta. \tag{23}$$

For a uniform angle and $1 \leq T \leq m$, substituting $u = 1 - \theta/\pi$ yields

$$\mathbb{E}_R[\alpha_{\mathrm{unif}}^{\mathrm{HM}}] = \sum_{k=T}^{m} \binom{m}{k} \int_0^1 u^k (1-u)^{m-k}\, du \tag{24}$$

$$= \sum_{k=T}^{m} \frac{1}{m+1} = \frac{m - T + 1}{m+1}. \tag{25}$$

Hence $m = 8, T = 6$ gives mean acceptance $1/3$ and mean excluded mass $2/3$. These are averages over random projections, not exact rates for every fixed matrix. They also cannot be multiplied as independent probabilities along a sequence generated with one shared key; generated contexts may themselves depend on that key.

Random Code control and reference comparison. In the same eight-bit code space, independently select a uniform subset G of exactly 37 valid codes. For any fixed natural code distribution,

$$\mathbb{E}_G[\alpha_t^{\mathrm{Random}}] = \sum_{h \in \{0,1\}^8} \Pr_t(H_R(z) = h) \Pr_G(h \in G) = \frac{37}{256}. \tag{26}$$

Compared with the HammingMark uniform-angle reference, this demonstrates that equal numbers of valid codes can carry different natural candidate mass because of how those codes are arranged.

Nonuniform natural continuations. equation 23 returns the analysis to the context-dependent angular distribution. Centering acceptance on the preceding hash is intended to align valid candidates with locally plausible continuations. A short hash neither uniquely specifies a continuous semantic realization nor imposes a deterministic cosine-similarity cutoff. Although many-to-one hashing also occurs in SemStamp, HammingMark further organizes valid codes as a neighborhood of the preceding code. These mechanisms aim to reduce misalignment, without guaranteeing higher acceptance or moderate coverage in every context. Claims of reduced narrowing should therefore be supported by measured acceptance on common context-specific candidate pools, together with independent semantic evaluations of what those accepted candidates preserve.

B.3 Effectiveness Beyond Enlarging the Acceptance Set

Relaxing baseline constraints. A natural question is whether existing methods can match HammingMark's sampling efficiency simply by enlarging their acceptance sets. To examine this possibility, we increase the valid-partition ratio of SemStamp and K-SemStamp and widen the acceptance interval of SimMark, targeting comparable numbers of sampled candidates per output sentence. A controlled comparison holds the generator, dataset, generation length, and sampling budget fixed, and separately calibrates each modified detector at 1% FPR using data disjoint from the test set.

Table 5: Relaxed-constraint comparison.

Method	Setting	Candidates/sentence	TPR@1% FPR
SemStamp	$\gamma = 0.5$	2.9	6%
K-SemStamp	$\gamma = 0.5$	3.1	9%
SimMark	$[a, b] = [0.40, 0.90]$	1.6	7%
HammingMark	$m = 8,\ T = 6$	2.2	100%

Acceptance mass and selection signal. Enlarging an acceptance set increases its natural probability mass, but does not necessarily preserve its ability to distinguish watermarked outputs from natural text. The relevant quantities are not only the nominal valid fraction γ and mean acceptance mass, but also how the context-specific acceptance mass α_t is distributed. When α_t is close to zero, finite-budget generation frequently fails to find a valid candidate. When α_t is close to one, natural candidates already satisfy the rule, leaving little additional selection signal. Intermediate coverage can support both successful sampling and an increase over the natural hit rate.

This distinction is particularly relevant when natural continuations concentrate in a few semantic partitions. For fixed partition masses $p_{t,1}, \ldots, p_{t,K}$, uniformly selecting exactly γK valid partitions gives

$$\mathbb{E}_G[\alpha_t] = \gamma, \qquad \mathrm{Var}_G(\alpha_t) = \gamma(1-\gamma)\frac{K\sum_j p_{t,j}^2 - 1}{K-1}, \tag{27}$$

where $K > 1$ and γK is an integer. Equal partition masses yield $\alpha_t = \gamma$ for every valid set. In contrast, if one partition carries all natural mass, acceptance is either zero or one, depending on whether that partition is selected. Thus, increasing the nominal valid fraction need not produce moderate coverage in individual contexts. This random-set calculation identifies a possible source of uneven coverage; actual variation across contexts must be measured.

Why enlarging the acceptance set is insufficient. The effectiveness of a watermark depends not only on the size of its acceptance set, but also on how this set overlaps with the natural next-sentence distribution. Natural continuations can concentrate in a few context-dependent semantic directions. For SemStamp and K-SemStamp, pseudorandomly selecting more valid partitions does not ensure that these partitions provide appropriate partial coverage of such continuations. Likewise, widening SimMark's fixed similarity interval does not ensure suitable coverage across different contexts.

This mismatch can leave many contexts with little useful watermark signal. When the valid set covers almost all natural probability mass, unwatermarked candidates already satisfy the constraint, so watermark selection adds little evidence. Conversely, when it covers very little natural mass, valid candidates are difficult to obtain, and fallback can weaken watermark embedding. Enlarging the nominal acceptance set may therefore improve sampling efficiency without ensuring that individual contexts contribute discriminative evidence. This does not imply that larger acceptance sets necessarily impair detection; the issue is their alignment with context-dependent natural generation.

Implications for HammingMark. HammingMark organizes valid codes around the preceding sentence's hash, rather than selecting them pseudorandomly or prescribing a fixed continuous-similarity interval. This design aims to retain naturally plausible continuations while preserving a selective bit-matching constraint, allowing more contexts to contribute watermark evidence. Its intended advantage is therefore not simply a larger acceptance mass, but a more contextually aligned arrangement of valid candidates. Uneven contextual coverage offers a possible explanation for the detection degradation of relaxed baselines, although establishing this mechanism requires measuring their context-specific acceptance masses.

C Semantic Narrowing in Offline PMark

We analyze the semantic restriction induced by the offline variant of PMark. Our analysis does not assume that natural semantic candidates are uniformly distributed across watermark partitions, nor does it require independence or balance among different proxy channels. Instead, we directly examine the candidate-selection mechanism used by offline PMark.

Offline watermark partition. Let π denote the current generation context and

$$P_\pi(s) = P_M(s \mid \pi)$$

the natural next-sentence distribution. Given b proxy directions $\{v_j\}_{j=1}^{b}$, offline PMark uses the fixed threshold zero and assigns each sentence a binary semantic representation

$$g(s) = (\mathbb{I}[F_1(s) > 0], \ldots, \mathbb{I}[F_b(s) > 0]),$$

where

$$F_j(s) = \langle v_j, T(s) \rangle.$$

For a fixed watermark seed r, let $E_r(s)$ denote the watermark evidence assigned to sentence s. A larger $E_r(s)$ indicates that the sentence satisfies more of the prescribed watermark constraints.

Offline PMark first independently samples a candidate set

$$W = \{X_1, \ldots, X_N\}, \qquad X_i \overset{\text{i.i.d.}}{\sim} P_\pi,$$

and then retains only candidates with the maximum watermark evidence:

$$\mathcal{B}(W, r) = \left\{ X_i \in W : E_r(X_i) = \max_{X_j \in W} E_r(X_j) \right\}.$$

The final sentence is sampled from $\mathcal{B}(W, r)$.

Candidate-level semantic restriction. For every sampled candidate set,

$$\mathcal{B}(W, r) \subseteq W.$$

Moreover,

$$\mathcal{B}(W, r) \subsetneq W$$

whenever the candidates in W do not all have identical watermark evidence. Therefore, whenever multiple watermark-evidence levels occur in the natural candidate set, every candidate below the maximum level is excluded from final selection.

Importantly, this exclusion is not determined by the likelihood or semantic preference of the base model. Two candidates that are both plausible under P_π may receive different watermark evidence solely because they fall into different regions induced by the fixed proxy partitions. Consequently, offline PMark removes part of the naturally sampled semantic alternatives whenever their watermark evidence is lower than that of another candidate in the same candidate pool.

Distribution-level concentration. The same effect can be characterized without making any assumption on how natural semantic probability mass is distributed among the partitions. Let

$$Z = E_r(S), \qquad S \sim P_\pi,$$

denote the watermark evidence of a naturally sampled sentence, and define its cumulative distribution

$$F_r(k) = \Pr_{S \sim P_\pi} [E_r(S) \le k].$$

Let Y be the sentence selected by offline PMark and

$$Z^\star = E_r(Y).$$

Since PMark selects a candidate with the maximum watermark evidence among N independent samples,

$$Z^\star = \max_{1 \le i \le N} E_r(X_i).$$

Therefore,

$$\Pr(Z^\star \le k) = \Pr(E_r(X_1) \le k, \ldots, E_r(X_N) \le k) = F_r(k)^N.$$

For $N > 1$,

$$F_r(k)^N \le F_r(k),$$

with strict inequality whenever

$$0 < F_r(k) < 1.$$

Equivalently,

$$\Pr(Z^\star > k) \ge \Pr(Z > k).$$

Thus, the distribution induced by offline PMark is systematically shifted toward semantic regions with higher watermark evidence.

Semantic narrowing. This concentration constitutes semantic narrowing at the candidate-selection level. The base model initially provides multiple semantic alternatives according to P_π, whereas offline PMark ranks these alternatives according to an additional watermark-specific criterion and removes all candidates below the maximal evidence level within each candidate pool.

The result does not rely on any particular amount of natural probability mass occupying a watermark partition. As long as the watermark evidence is non-constant over naturally plausible continuations, i.e.,

$$\exists\, s_a, s_b \in \mathrm{supp}(P_\pi) \quad \text{such that} \quad E_r(s_a) \neq E_r(s_b),$$

maximum-evidence selection assigns different selection preferences to these semantic alternatives. For $N > 1$, this preference is amplified through best-of-N selection, progressively concentrating generation toward watermark-preferred semantic regions.

Therefore, the reduced rejection-sampling cost of offline PMark should not be interpreted as the absence of semantic restriction. Offline PMark replaces explicit rejection with maximum-evidence selection: candidates are generated from the natural distribution, but only the watermark-preferred subset is eligible for final selection. This preserves sampling efficiency while still inducing semantic narrowing.

D Locality and Perturbation Stability of Semantic Hashes

This section formalizes the locality property used by HammingMark. Let $e, e' \in \mathbb{S}^{d-1}$ be two normalized SBERT embeddings, let $R = [r_1^\top; \ldots; r_m^\top] \in \mathbb{R}^{m\times d}$, where the rows are independently sampled from a spherically symmetric continuous distribution (e.g., $r_j \sim \mathcal{N}(0, I_d)$), and define the j-th random-hyperplane hash bit as

$$H_j(e) = \mathbf{1}[r_j^\top e \geq 0]. \tag{28}$$

Write $\rho = e^\top e'$ and $\theta = \arccos(\rho) \in [0, \pi]$.

Proposition 1 (angular locality). For every hash bit,

$$\Pr_R[H_j(e) \neq H_j(e')] = \frac{\theta}{\pi}, \qquad \Pr_R[H_j(e) = H_j(e')] = 1 - \frac{\theta}{\pi}. \tag{29}$$

Consequently,

$$d_H(H(e), H(e')) \sim \mathrm{Binomial}\left(m, \frac{\theta}{\pi}\right), \qquad \mathbb{E}_R\left[\frac{d_H(H(e), H(e'))}{m}\right] = \frac{\theta}{\pi}. \tag{30}$$

Proof. By rotational invariance, only the two-dimensional plane spanned by e and e' matters. A hyperplane separates the two vectors exactly when its normal falls in the angular wedge between them. The measure of this wedge is θ/π. Independence of the rows of R then gives equation 30.

equation 30 states that semantic closeness is preserved in probability rather than deterministically. A standard Hoeffding bound also gives

$$\Pr_R\left[\left|\frac{d_H(H(e), H(e'))}{m} - \frac{\theta}{\pi}\right| \geq \varepsilon\right] \leq 2\exp(-2m\varepsilon^2). \tag{31}$$

Thus, smaller angular distance yields fewer expected bit flips, while a longer hash concentrates more tightly around this expectation.

Corollary 1 (stability under a meaning-preserving edit). Suppose an edit changes a sentence embedding from e to $\widetilde{e}$, with $\delta = \arccos(e^\top \widetilde{e})$. Then

$$\Pr_R[H(e) = H(\widetilde{e})] = \left(1 - \frac{\delta}{\pi}\right)^m, \tag{32}$$

$$\Pr_R[d_H(H(e), H(\widetilde{e})) \leq b] = \sum_{j=0}^{b} \binom{m}{j} \left(\frac{\delta}{\pi}\right)^j \left(1 - \frac{\delta}{\pi}\right)^{m-j}. \tag{33}$$

Moreover, for two original codes h_{t-1}, h_t and their edited versions $\widetilde{h}_{t-1}, \widetilde{h}_t$,

$$\begin{aligned}\left|\operatorname{Match}(h_{t-1}, h_t) - \operatorname{Match}(\widetilde{h}_{t-1}, \widetilde{h}_t)\right| \le d_H(h_{t-1}, \widetilde{h}_{t-1}) \\ + d_H(h_t, \widetilde{h}_t).\end{aligned} \tag{34}$$

This follows from $\operatorname{Match}(a, b) = m - d_H(a, b)$ and the triangle inequality for Hamming distance. In particular, a transition satisfying $\operatorname{Match}(h_{t-1}, h_t) \ge T + \gamma$ remains valid whenever the two edited sentences jointly flip at most γ bits. This margin argument connects angular locality to robustness of the transition statistic used by HammingMark.

The probability in the results above is over the random construction of R. Once R is fixed, an adversarially chosen embedding can lie close to a hyperplane, so no deterministic collision guarantee is claimed.

E Semantic Diversity within a Hamming Neighborhood

For a code $h \in \{0,1\}^m$, define its normalized semantic preimage by

$$\mathcal{C}(h) = \{e \in \mathbb{S}^{d-1} : H(e) = h\}. \tag{35}$$

Because $m \ll d$, H is highly non-injective. The following construction shows that this is not merely a counting argument.

Proposition 2 (large angular diameter of a hash cell). Assume $\dim \ker(R) \ge 2$ and take a unit vector e whose projections Re are all nonzero. There exists a unit vector $v \in \ker(R) \cap e^{\perp}$. For any $\tau \ge 0$, define

$$e_+(\tau) = \frac{e + \tau v}{\sqrt{1+\tau^2}}, \qquad e_-(\tau) = \frac{e - \tau v}{\sqrt{1+\tau^2}}. \tag{36}$$

Then

$$H(e_+(\tau)) = H(e) = H(e_-(\tau)) \tag{37}$$

for every finite τ, while

$$e_+(\tau)^{\top} e_-(\tau) = \frac{1-\tau^2}{1+\tau^2} \longrightarrow -1 \quad \text{as } \tau \to \infty. \tag{38}$$

Proof. The dimension assumption guarantees a nonzero vector in $\ker(R) \cap e^{\perp}$. Since $Rv = 0$, $Re_+(\tau) = Re_-(\tau) = Re/\sqrt{1+\tau^2}$. The positive normalization factor does not change any projection sign, proving equation 37. equation 38 follows by direct calculation. □

For a Gaussian random projection with $m < d$, R has full row rank almost surely, so $\dim \ker(R) = d - m$. In our setting, $d = 768$ and $m = 8$, leaving a 760-dimensional null space. Moreover, full row rank makes $R : \mathbb{R}^d \to \mathbb{R}^m$ surjective; hence every strict sign pattern is realizable in the ambient embedding space.

Given a preceding code h_{t-1} and radius $r = m - T$, the complete admissible semantic set is

$$\mathcal{A}_T(h_{t-1}) = H^{-1}(\mathcal{B}_r(h_{t-1})) = \bigcup_{h \in \mathcal{B}_r(h_{t-1})} \mathcal{C}(h), \tag{39}$$

where

$$|\mathcal{B}_r(h_{t-1})| = \sum_{j=0}^{r} \binom{m}{j}. \tag{40}$$

With $m = 8$ and $T = 6$, $r = 2$ and the acceptance set contains $1 + 8 + 28 = 37$ different hash cells. Each cell contains a continuous family of ambient semantic vectors, so the accepted set can cover distinct semantic directions even though all its codes remain close to h_{t-1}.

Proposition 2 concerns the geometry of the ambient SBERT representation space and does not imply that every constructed vector corresponds to a realizable natural-language sentence. We therefore interpret it only as a geometric explanation for why short semantic hashes need not define a narrow

Table 6: Distribution of watermark-valid candidates across eight semantic clusters.

Method	C1	C2	C3	C4	C5	C6	C7	C8	Total
SemStamp	1	0	0	0	0	0	0	0	1
K-SemStamp	1	1	0	0	0	0	0	0	2
SimMark	7	5	4	0	0	0	0	0	16
PMark	10	8	6	4	3	0	0	0	31
HammingMark	24	20	17	15	12	7	4	0	99

region in representation space. Our empirical results provide complementary, aggregate evidence: among independently sampled natural continuations, HammingMark accepts a substantially broader subset than the compared semantic watermarking methods, as illustrated in Fig. 2 and reflected by the higher natural-candidate acceptance rate. These observations are consistent with, but do not by themselves prove, greater semantic diversity on the natural-language manifold.

F Semantic Coverage of Watermark-Valid Candidates

To further examine whether higher acceptance corresponds to broader semantic preservation, we sample 300 natural candidate continuations from the same prompt. To avoid evaluating semantic coverage directly in the representation space used by HammingMark for watermark generation and detection, we encode these candidates using the frozen `sentence-transformers/all-mpnet-base-v2` evaluation model and cluster their normalized embeddings into $K = 8$ semantic regions, matching the partition granularity used in SemStamp and K-SemStamp. We then apply each watermarking method's original constraint and encoder to the same candidate pool and examine the distribution of watermark-valid candidates across the externally constructed semantic clusters. The external evaluation encoder is used only for constructing the clusters and does not participate in candidate acceptance for any watermarking method.

As shown in Table 6, existing watermark constraints retain candidates from only a limited number of semantic regions. In contrast, HammingMark accepts candidates spanning seven of the eight clusters. Notably, even several low-frequency semantic modes remain accessible. This indicates that HammingMark does not merely increase the number of watermark-valid candidates, but preserves a broader portion of the semantic modes present in natural generation, providing more direct evidence of alleviated semantic narrowing.

G Semantic Diversity under Fixed Contexts

Natural acceptance measures how much probability mass of the original generation distribution remains compatible with a watermark constraint, but it does not directly characterize whether the resulting continuations remain semantically diverse. We therefore evaluate semantic diversity under fixed contexts. Using the same 100 prompts as in the natural-acceptance experiment, we independently generate 300 next-sentence continuations for each method from the same context.

To avoid measuring diversity directly in the representation space used by HammingMark for watermark generation and detection, we use a separate, frozen sentence-embedding model, `sentence-transformers/all-mpnet-base-v2`, solely as the evaluation encoder. This checkpoint is not used by any watermarking method during generation or detection. Let

$$z_i = \frac{E_{\text{eval}}(S_i)}{\|E_{\text{eval}}(S_i)\|_2}$$

denote the normalized evaluation embedding of continuation S_i, where E_{eval} is `all-mpnet-base-v2`. We define pairwise semantic diversity as

$$D_{\text{pair}} = \frac{2}{N(N-1)} \sum_{1 \le i < j \le N} \left(1 - z_i^\top z_j\right), \tag{41}$$

where $N = 300$. A larger value indicates that repeated generations span a broader range of semantic realizations according to the held-out evaluation encoder. We further report the diversity retention

Table 7: Semantic diversity under fixed contexts. Each method independently generates 300 continuations for each of 100 prompts. Higher values indicate greater semantic diversity.

Method	Pairwise diversity ↑	Diversity retention ↑
No watermark	0.284	1.000
SimMark	0.224	0.789
SemStamp	0.142	0.500
K-SemStamp	0.158	0.556
PMark	0.201	0.708
Random Code	0.184	0.648
HammingMark	**0.269**	**0.947**

ratio

$$R_{\mathrm{div}} = \frac{D_{\mathrm{pair}}^{\mathrm{WM}}}{D_{\mathrm{pair}}^{\mathrm{NoWM}}}, \tag{42}$$

which measures the fraction of unwatermarked semantic diversity preserved after watermarking under the same evaluation metric.

As shown in Table 7, semantic watermarking generally reduces the diversity of continuations generated from the same context, but the degree of reduction differs substantially across methods. SemStamp and K-SemStamp exhibit the strongest concentration, retaining only 50.0% and 55.6% of the semantic diversity of unwatermarked generation. This is consistent with their region-based acceptance mechanisms, which restrict generation to designated portions of the semantic space. SimMark retains more diversity, but still reduces the pairwise distance from 0.284 to 0.224. PMark similarly shows a noticeable reduction, retaining approximately 70.8% of the unwatermarked diversity.

In contrast, HammingMark achieves a pairwise diversity of 0.269, corresponding to 94.7% diversity retention and remaining close to unwatermarked generation according to the held-out evaluation encoder. This result complements the natural-acceptance analysis: a high acceptance mass alone could still arise if the accepted continuations were concentrated around a small number of similar semantic realizations, whereas the present results show that HammingMark also preserves substantial variation among independently generated continuations. This provides additional evidence that the realized-sentence-centered Hamming constraint preserves not only more natural probability mass, but also a broader range of semantic alternatives.

The comparison with Random Code further isolates the effect of constraint placement. Although Random Code uses the same code-space coverage as HammingMark, it retains only 64.8% of the unwatermarked diversity. Thus, the advantage of HammingMark cannot be explained solely by admitting a larger set of hash codes. Rather, centering the Hamming neighborhood on the realized preceding sentence allows the admissible region to better follow the local semantic structure of natural generation, while the coarse many-to-one hash mapping leaves multiple semantic realizations simultaneously valid.

H Deriving the Matching Threshold from Natural Generation

This section explains how the matching threshold T is determined by the natural matching behavior of unwatermarked generation and the desired watermark strength. We first derive an exact rule that holds for an arbitrary natural match-count distribution. We then obtain a compact approximation in terms of the hash length m and the natural expected number of matching bits, and finally instantiate the result for our default setting $m = 8$.

H.1 Exact Threshold from the Natural Match Distribution

Consider one sentence-generation step and condition on the realized context π. To simplify notation, we omit the sentence-step index throughout this section. Let

$$P(S \mid \pi)$$

denote the base-model distribution over the next candidate sentence before watermarking, and let $h_{\text{prev}} \in \{0,1\}^m$ be the semantic hash of the preceding sentence.

For a candidate sentence S, define its number of matching hash bits with the preceding sentence as

$$M(S) = \text{Match}\left(h_{\text{prev}},\, H(F(S))\right) \in \{0, \ldots, m\}. \tag{43}$$

When $S \sim P(\cdot \mid \pi)$, M is therefore a discrete random variable describing the matching behavior that arises naturally before any watermark constraint is applied.

For a threshold T, HammingMark accepts a candidate if

$$M(S) \geq T.$$

The probability that an unwatermarked candidate already satisfies this condition is

$$a(T) = \Pr_{S \sim P(\cdot|\pi)}[M(S) \geq T]. \tag{44}$$

This quantity is central to the threshold selection: it measures how much of the natural next-sentence probability mass remains admissible under the watermark constraint.

Assuming rejection sampling continues until an admissible candidate is found, the distribution of the accepted sentence is simply the base distribution conditioned on $M(S) \geq T$:

$$Q_T(S \mid \pi) = \frac{P(S \mid \pi)\, \mathbf{1}[M(S) \geq T]}{a(T)}. \tag{45}$$

For every sentence in the accepted region,

$$\frac{Q_T(S \mid \pi)}{P(S \mid \pi)} = \frac{1}{a(T)}.$$

Hence the KL divergence from the natural distribution is

$$\begin{aligned} D_{\text{KL}}(Q_T \| P) &= \sum_{S: M(S) \geq T} Q_T(S \mid \pi) \log \frac{Q_T(S \mid \pi)}{P(S \mid \pi)} \\ &= \sum_{S: M(S) \geq T} Q_T(S \mid \pi) \log \frac{1}{a(T)} \\ &= \log \frac{1}{a(T)} = -\log a(T). \end{aligned} \tag{46}$$

Distributional watermark strength. We characterize the *per-transition watermark strength* through the distributional bias introduced by enforcing the watermark acceptance event. Specifically, under a fixed context π, we define

$$\lambda(T; \pi) \triangleq \frac{1}{\ln 2} D_{\text{KL}}\left(Q_T(\cdot \mid \pi) \| P(\cdot \mid \pi)\right) = -\log_2 a_\pi(T), \tag{47}$$

where

$$a_\pi(T) = \Pr_{S \sim P(\cdot|\pi)}[M(S) \geq T]$$

is the probability that a naturally generated candidate already satisfies the watermark constraint.

Intuitively, $\lambda(T; \pi)$ measures how unlikely the enforced watermark event is under natural generation. A smaller natural acceptance probability corresponds to a stronger watermark-specific selection bias and, in general, provides stronger per-transition evidence for distinguishing watermarked from unwatermarked generation. We therefore use λ as a *distributional proxy for watermark strength*, rather than as a direct measure of sequence-level detection performance, which also depends on factors such as text length, dependence across transitions, the detector statistic, and possible perturbations.

In particular, $\lambda(T; \pi) = 1$ bit corresponds to an acceptance event with natural probability $1/2$. More generally, requiring at least λ_0 bits of distributional watermark strength is equivalent to

$$\lambda(T; \pi) \geq \lambda_0 \quad \Longleftrightarrow \quad a_\pi(T) \leq 2^{-\lambda_0}. \tag{48}$$

This relation directly yields a threshold-selection rule. Since

$$a_\pi(T+1) = a_\pi(T) - \Pr[M = T \mid \pi] \le a_\pi(T),$$

the natural acceptance probability is non-increasing in T, and consequently $\lambda(T;\pi) = -\log_2 a_\pi(T)$ is non-decreasing in T. Under ideal rejection sampling, each independent candidate is accepted with probability $a_\pi(T)$, so the expected number of sampled candidates required for one accepted sentence is

$$\mathbb{E}[N_T \mid \pi] = \frac{1}{a_\pi(T)}. \tag{49}$$

Thus, increasing T generally induces a stronger watermark-specific distributional bias, but also rejects more natural probability mass and increases the expected sampling cost.

Accordingly, among all thresholds that achieve at least λ_0 bits of distributional watermark strength, the least restrictive choice is

$$\boxed{T^\star = \min\left\{T \in \{0,\dots,m\} : \Pr[M \ge T \mid \pi] \le 2^{-\lambda_0}\right\}.} \tag{50}$$

This choice satisfies the desired distributional-strength requirement while maximizing the retained natural acceptance probability, or equivalently minimizing the expected rejection-sampling cost.

equation 50 is exact for the idealized unlimited-budget rejection-sampling procedure and makes no parametric assumption about the natural distribution of M. In practice, $T^\star$ can therefore be estimated directly from the empirical match-count distribution of unwatermarked continuations.

H.2 Estimating the Matching Threshold from Natural Statistics

The natural match-count distribution depends on both the semantic relation between adjacent sentences and the fixed random projection used by the watermark key. Therefore, we do not assume that the marginal distribution of M follows an exact parametric form. Instead, we use a simple moment-matched distribution only to obtain an interpretable estimate of the threshold scale.

Let

$$\mu = \mathbb{E}[M]$$

denote the mean number of matching bits observed from naturally generated sentence pairs. Since $M \in \{0,\dots,m\}$, we summarize the average bit-wise matching level by

$$\bar{p} = \frac{\mu}{m}.$$

We then introduce the following moment-matched proxy:

$$\widetilde{M} \sim \text{Binomial}\left(m, \frac{\mu}{m}\right). \tag{51}$$

This approximation matches the empirical mean, $\mathbb{E}[\widetilde{M}] = \mu$, while ignoring the heterogeneity of semantic similarities across contexts. It is used only for estimating a reasonable operating threshold rather than for characterizing the exact natural match-count distribution.

Under this approximation, the estimated natural acceptance probability for threshold T is

$$\widetilde{a}(T) = \Pr[\widetilde{M} \ge T] = \sum_{k=T}^{m} \binom{m}{k} \left(\frac{\mu}{m}\right)^k \left(1 - \frac{\mu}{m}\right)^{m-k}. \tag{52}$$

Following the definition of watermark strength in the previous section, we define the corresponding estimated strength as

$$\widetilde{\lambda}(T) = -\log_2 \widetilde{a}(T). \tag{53}$$

Here, $\widetilde{\lambda}(T)$ should be interpreted only as an estimate of the distributional strength associated with the selected threshold. The actual strength remains determined by the true natural acceptance probability $a(T) = \Pr[M \ge T]$.

For a desired strength level λ_0, the same approximation provides a direct estimate of the required matching threshold. Let

$$F_{m,p}(k) = \Pr[X \le k], \qquad X \sim \text{Binomial}(m,p),$$

and define the discrete quantile

$$F_{m,p}^{-1}(u) = \min\{k : F_{m,p}(k) \geq u\}.$$

Since

$$\Pr[\widetilde{M} \geq T] = 1 - F_{m,\mu/m}(T-1),$$

the estimated threshold is

$$\widehat{T}(m, \mu, \lambda_0) = 1 + F_{m,\mu/m}^{-1}\left(1 - 2^{-\lambda_0}\right). \tag{54}$$

equation 54 provides a convenient mapping from the observed natural matching level μ and a desired watermark strength λ_0 to an estimated threshold. It should not be interpreted as an exact solution for the true generation distribution, since natural sentence pairs may exhibit substantial variation in semantic similarity and hence a non-binomial marginal distribution of M.

For additional intuition, applying a normal approximation to. equation 51 gives

$$\widetilde{M} \approx \mathcal{N}\left(\mu, \mu\left(1 - \frac{\mu}{m}\right)\right).$$

With a continuity correction, this yields the rough closed-form estimate

$$\widehat{T} \approx \left\lceil \mu + \sqrt{\mu\left(1 - \frac{\mu}{m}\right)}\, \Phi^{-1}\left(1 - 2^{-\lambda_0}\right) + \frac{1}{2} \right\rceil, \tag{55}$$

where Φ^{-1} is the inverse standard-normal CDF. We use this expression only for interpretation; all numerical estimates below are computed from the binomial tail in equation 52.

H.3 Estimated Operating Point for the Default Configuration

We next apply the above approximation to the default HammingMark configuration. We use a semantic hash length of

$$m = 8,$$

and natural unwatermarked sentence pairs exhibit an average match count of approximately

$$\mu \approx 4.8.$$

The moment-matched proxy is therefore

$$\widetilde{M} \sim \mathrm{Binomial}(8, 0.60). \tag{56}$$

We use $\lambda_0 = 1$ bit as a convenient reference strength. Under the definition

$$\lambda = -\log_2 a,$$

one bit corresponds to an acceptance event whose natural probability is approximately one half. Substituting $\lambda_0 = 1$ into equation 54 gives

$$\widehat{T} = 1 + F_{8,0.60}^{-1}(0.5) = 6. \tag{57}$$

The same result can be seen directly from the estimated upper-tail probabilities:

$$\widetilde{a}(5) = \Pr[\widetilde{M} \geq 5] \approx 0.5941, \tag{58}$$

$$\widetilde{a}(6) = \Pr[\widetilde{M} \geq 6] \approx 0.3154. \tag{59}$$

The corresponding estimated watermark strengths are

$$\widetilde{\lambda}(5) = -\log_2(0.5941) \approx 0.75 \text{ bits}, \tag{60}$$

$$\widetilde{\lambda}(6) = -\log_2(0.3154) \approx 1.66 \text{ bits}. \tag{61}$$

Thus, under the moment-matched approximation, the one-bit reference point lies between $T = 5$ and $T = 6$, making $T = 6$ a natural estimated operating point for $m = 8$. This estimate also has a simple interpretation through equation 55. When $\lambda_0 = 1$,

$$\Phi^{-1}(1 - 2^{-1}) = \Phi^{-1}(0.5) = 0,$$

Table 8: Empirical fallback rates under the maximum resampling budget $B = 16$.

Dataset	Generator	Fallback rate (%)
C4	Mistral-7B	1.3
BookSum	Mistral-7B	0.8
C4	Qwen2.5-7B	1.2
BookSum	Qwen2.5-7B	0.6
ELI5	Mistral-7B	2.3
Multi-News	Mistral-7B	2.7

and the approximate threshold reduces to

$$\widehat{T} \approx \left\lceil \mu + \frac{1}{2} \right\rceil.$$

For $\mu = 4.8$, this again gives

$$\widehat{T} \approx \lceil 5.3 \rceil = 6.$$

We emphasize that these calculations are intended only to estimate the appropriate threshold scale. The true natural match-count distribution is context dependent and need not follow the moment-matched binomial proxy. Accordingly, the values $\widetilde{a}(T)$ and $\widetilde{\lambda}(T)$ above should not be interpreted as measured acceptance rates or exact watermark strengths. Empirical acceptance and detection performance are evaluated separately in the experiments. The analysis here only provides an intuitive explanation for why a threshold around $T = 6$ is a reasonable default for an 8-bit semantic hash.

Finite-budget implementation. The analysis above characterizes the idealized rejection-sampling procedure that continues until a valid candidate is obtained. In practice, we impose a finite resampling budget for computational efficiency. Therefore, Eqs. (39)–(44) should be interpreted as the exact characterization of the unlimited-budget procedure, while our implementation provides a finite-budget approximation to it.

Empirical fallback frequency. In our implementation, the maximum resampling budget is $B = 16$. When no candidate satisfies the watermark constraint within this budget, the last sampled candidate is retained. We additionally measure the empirical fallback rate as

$$r_{\mathrm{fb}} = \frac{\#\{\text{sentence steps exhausting all } B \text{ attempts}\}}{\#\{\text{generated sentence steps}\}}.$$

Whenever fallback does not occur, the finite-budget procedure coincides with the ideal rejection-sampling procedure analyzed above. Across all evaluated settings, fallback occurs in only 0.6%–2.7% of sentence steps, as shown in Table 8. Thus, the ideal conditional-distribution analysis provides a close description of the generation procedure for the large majority of sentence steps.

I Effect of Semantic Hash Length

We study the effect of the semantic hash length M by evaluating $M \in \{4, 8, 16, 32\}$ on C4 with Mistral-7B. For all settings, we fix the required matching ratio to 0.75, giving thresholds $T \in \{3, 6, 12, 24\}$, respectively. All other generation and detection settings remain unchanged. Robustness is evaluated under Parrot and DIPPER rewriting attacks.

As shown in Table 9, $M = 4$ provides insufficient discriminative resolution: with only four bits, unwatermarked transitions can also frequently satisfy the $3/4$ matching requirement, resulting in substantial overlap between watermarked and unwatermarked scores even without attacks. For example, under an independent balanced-bit approximation, the probability of matching at least three bits is already $5/16 = 31.25\%$.

Increasing M improves hash resolution but gradually reduces robustness under Parrot and DIPPER. Longer hashes provide a more precise estimate of semantic similarity and therefore reduce the quantization slack of short codes. After paraphrasing perturbs the sentence embeddings, more hyperplane

Table 9: Effect of semantic hash length on clean detection and attack robustness. Each entry reports TPR@1%/TPR@5%/AUROC.

M	T	Clean	Parrot	DIPPER
4	3	0.76/0.86/0.97	0.75/0.86/0.97	0.74/0.83/0.96
8	6	**0.98/1.00/1.00**	**0.94/0.99/1.00**	**0.93/0.98/0.99**
16	12	0.98/1.00/1.00	0.78/0.87/0.98	0.73/0.81/0.96
32	24	0.99/1.00/1.00	0.63/0.73/0.90	0.64/0.74/0.91

signs may change, making transitions near the 0.75 boundary less likely to remain valid. Overall, $M = 8$ achieves the best balance between clean detectability and tolerance to meaning-preserving perturbations.

J Reproducibility Details

Tasks and models. We evaluate HammingMark on C4 and BookSum for open-ended generation, using Mistral-7B-Instruct and Qwen2.5-7B-Instruct as generators. We further evaluate Multi-News summarization and ELI5 long-form question answering as more constrained generation tasks.

Baselines and configuration. We compare HammingMark with token-level methods (KGW, SynthID, MorphMark, and SIR) and sentence-level semantic watermarks (SemStamp, K-SemStamp, SimMark, and PMark). HammingMark uses an $m = 8$-bit semantic hash and accepts a transition when at least $T = 6$ bits match.

Evaluation protocol. For KGW, SynthID, MorphMark, SIR, SemStamp, and K-SemStamp, we follow their respective configurations in MarkLLM (Pan et al., 2024) without modification, including all algorithm-specific hyperparameters and, where applicable, sentence-sampling procedures, resampling budgets, and fallback strategies when no valid candidate is found. For PMark, we evaluate the offline variant using its default parameter settings. For SimMark, we use the default parameter settings reported in its original paper.

For detection, we calibrate the threshold $\tau_d(\alpha)$ separately for each method and evaluation dataset at a target false-positive rate α. Specifically, for each dataset, we randomly sample 500 examples disjoint from its test set to construct the calibration set. All methods use the same calibration examples and threshold-selection procedure, with method-specific thresholds determined from their respective detection scores. Each calibrated threshold is then fixed and applied to the corresponding test set. We report $\mathrm{TPR@1\%}$, $\mathrm{TPR@5\%}$, and AUROC on both clean and attacked texts. Robustness is evaluated under Parrot (Damodaran, 2021) and Pegasus paraphrasing (Zhang et al., 2020), DIPPER rewriting (Krishna et al., 2023), and API-based rewriting with DeepSeek-V4-Pro (Xu et al., 2026).

We adopt dataset-specific detection threshold calibration to ensure a controlled comparison across methods at the same target false-positive rate. Nevertheless, a fixed threshold would simplify deployment when representative calibration data are unavailable. Empirically, for the Global Bits detector evaluated on clean text, the calibrated sequence-level thresholds across the evaluated datasets remain within 5% of 0.75 in relative terms. Under a common measure of relative threshold variation, HammingMark exhibits the smallest cross-dataset variation among the compared methods, whereas KGW-family detectors exhibit variations exceeding 30%. These observations support the stability of the Global Bits detection threshold across the evaluated clean-text datasets. Its concentration around 0.75 is also consistent with the generation rule, which requires at least six matching bits out of eight for each accepted transition. Together, these results suggest that 0.75 is a reasonable default decision threshold for Global Bits in comparable clean-text settings. However, the generation acceptance condition and the sequence-level detection threshold serve different purposes: their numerical proximity does not guarantee that a fixed threshold will maintain the target false-positive rate on unseen distributions. We therefore retain dataset-specific calibration for the reported comparisons and regard 0.75 as an empirically motivated deployment default whose false-positive rate should be validated in the intended setting.

Datasets and generation. We randomly select 500 samples from C4 and BookSum, respectively, and 100 samples from ELI5 and Multi-News. The target generation length is set to 200 tokens

Table 10: Effect of the matching threshold T on C4 with Mistral-7B using the Global Bits detector. Detection results are reported as TPR@1% / TPR@5% / AUROC. Lower sampling cost is better.

T	Clean	Parrot	DIPPER	PPL ↓	Samples / sent. ↓
5	0.71 / 0.83 / 0.95	0.32 / 0.56 / 0.81	0.27 / 0.40 / 0.74	4.61	1.3
6	0.98 / 1.00 / 1.00	0.94 / 0.99 / 1.00	0.93 / 0.98 / 0.99	4.61	2.2
7	0.98 / 1.00 / 1.00	0.98 / 0.99 / 1.00	0.98 / 0.99 / 1.00	5.33	6.4

for all tasks. Unless otherwise specified, all decoding parameters follow the default configurations provided by MarkLLM. Sentence boundaries are identified using the NLTK sentence tokenizer integrated into MarkLLM.

HammingMark implementation. HammingMark uses the SBERT encoder released with the SemStamp implementation. The default hash length and matching threshold are $m = 8$ and $T = 6$, respectively. The random projection matrix is initialized from the corresponding experimental seed. The maximum number of resampling attempts is set to 16. If no candidate satisfies the watermark constraint within this budget, the latest candidate is used as a fallback and remains included in subsequent watermark detection.

Calibration, randomness, and hardware. Detection thresholds and attack configurations follow the standardized MarkLLM evaluation pipeline and are kept consistent across all baselines. Reported results are averaged over three random seeds, with detection rates varying by less than one percentage point across runs. All experiments are conducted on a single NVIDIA H200 GPU.

K Effect of the Matching Threshold

We further study the effect of the matching threshold T while fixing the semantic hash length to $m = 8$. We evaluate $T \in \{5, 6, 7\}$ on C4 with Mistral-7B using the Global Bits detector. The results are summarized in Table 10.

The matching threshold directly controls the strength of the transition-level watermark constraint. When $T = 5$, the acceptance condition is relatively weak: a large fraction of naturally generated transitions already satisfy the watermark rule. This substantially reduces the sampling cost to only 1.3 candidates per accepted sentence and leaves generation quality nearly unchanged. However, because the accepted transitions are less distinguishable from those occurring naturally in unwatermarked text, both clean detection and robustness under paraphrasing and rewriting decrease.

Increasing the threshold to $T = 7$ has the opposite effect. The stricter matching requirement makes accepted transitions more strongly biased toward the watermark condition. Clean-text detection remains comparable to $T = 6$, while robustness under both Parrot and DIPPER improves. However, this stronger constraint considerably reduces the natural acceptance probability, increasing the average sampling cost from 2.2 to 6.4 candidates per accepted sentence. We also observe a substantially larger change in perplexity relative to the unwatermarked generation, indicating that the stronger filtering perturbs the original generation distribution more noticeably.

Overall, $T = 6$ provides the best empirical trade-off among watermark strength, robustness, generation fidelity, and sampling efficiency. A smaller threshold admits too many naturally occurring transitions and therefore provides insufficient watermark separation, whereas a larger threshold produces a stronger and more robust watermark at the cost of substantially more restrictive sampling. These results are consistent with our use of $T = 6$ as the default configuration throughout the main experiments.

L Cross-Key Detection and Key Specificity

Motivation and setup. HammingMark uses a randomly initialized projection matrix as part of the watermark key. We examine whether the detection signal is specific to the matrix used during generation. Watermarked text is generated on C4 with Mistral-7B using the default configuration

Table 11: Cross-key detection on C4 with Mistral-7B. Watermarked text is generated using $R^\star$. TPR is measured at a target FPR of 1%.

Detection key	TPR@1% ↑
Correct key $R^\star$	0.99
Independent wrong key R'	0.08

($m = 8, T = 6$) and a secret projection matrix $R^\star$. During detection, we compare the correct key $R^\star$ with an independently sampled wrong key $R' \neq R^\star$. Unwatermarked generations are used as negatives, and the detection threshold is calibrated at a target false-positive rate of 1%.

Results. As shown in Table 11, replacing the generation key with an independent projection matrix reduces TPR@1% from 0.99 to 0.08. The resulting 91-percentage-point gap indicates that the dominant watermark signal is strongly specific to the secret projection matrix used during generation. The non-zero wrong-key detection rate nevertheless suggests a weaker key-independent effect.

Discussion. The cross-key results provide strong evidence that HammingMark's detection signal is highly key-specific. Replacing the correct projection matrix $R^\star$ with an independently sampled matrix R' reduces TPR@1% from 0.99 to 0.08, corresponding to a 91-percentage-point drop. This large degradation indicates that high detection performance cannot be explained primarily by a generic increase in inter-sentence semantic continuity.

A small residual signal under the wrong key is nevertheless expected. Because HammingMark centers its acceptance region on the preceding sentence, it introduces a mild preference for semantically coherent transitions. Such transitions may exhibit above-random hash agreement under multiple random-hyperplane projections, allowing an independent matrix R' to capture a limited key-independent component of the signal. However, R' does not reproduce the specific hyperplane partition or the bit-level Hamming constraints imposed during generation under $R^\star$. Only the correct detector evaluates each transition in the same secret projection geometry that governed candidate acceptance, allowing key-specific evidence to accumulate consistently across the sequence. The substantial gap between correct-key and wrong-key detection therefore shows that, while generic semantic continuity contributes modestly, the dominant detection signal is tied to the secret projection matrix. Finally, the wrong-key TPR should not be interpreted as a false-positive rate, because the evaluated samples are genuinely watermarked texts rather than unwatermarked negatives.

M Comparison with Continuous Similarity-Based Acceptance

To determine whether the improvement of HammingMark arises merely from using the preceding sentence as a relational reference or from the discrete Hamming neighborhood itself, we construct a continuous-similarity baseline. This baseline uses the same SBERT semantic encoder as HammingMark and directly computes the cosine similarity between a candidate sentence and its preceding sentence. A candidate is accepted when its similarity exceeds a predefined threshold and is otherwise resampled. This design is conceptually related to inter-sentence similarity-based watermarking methods such as SimMark, since both select candidates directly according to their relations in a continuous semantic space. It therefore provides a controlled comparison between continuous similarity constraints and Hamming-code-based acceptance.

We use the same 100 prompts as in the natural-acceptance experiment and set the cosine-similarity threshold to 0.63. This produces an average natural acceptance rate of approximately 0.58, matching that of HammingMark. Despite accepting a similar fraction of natural candidates, the continuous baseline achieves only 0.68 TPR@1%, 0.76 TPR@5%, and 0.93 AUROC on clean text, substantially below HammingMark. Its fallback rate also increases from 1.3% for HammingMark to 7.5%.

These results reveal two limitations of direct continuous-similarity acceptance. First, naturally generated adjacent sentences already tend to be locally coherent and semantically similar. Consequently, a substantial fraction of unwatermarked transitions also exceeds the cosine threshold. When inter-sentence similarity is used directly as the watermark signal, the score distributions of watermarked and unwatermarked texts remain relatively close, limiting clean-text detectability. This illustrates

a fundamental trade-off for continuous similarity-based methods: increasing the threshold may strengthen the watermark signal, but it also imposes a more restrictive semantic constraint and increases the sampling cost.

Second, the average natural acceptance rate does not capture variation across contexts. Although most adjacent sentences can readily satisfy the threshold of 0.63 because of local semantic continuity, some valid discourse transitions naturally exhibit lower cosine similarity. Examples include contrast, topic shifts, the introduction of new facts, and transitions from background information to a conclusion. In such contexts, a hard continuous threshold may repeatedly reject otherwise appropriate candidates, eventually exhausting the sampling budget and triggering fallback. Thus, two methods with the same average natural acceptance rate can have substantially different numbers of low-acceptance contexts. The considerably higher fallback rate of the continuous baseline is consistent with this context-dependent effect. Because a fallback candidate is not guaranteed to satisfy the watermark constraint, these events further reduce the consistency of the accumulated sequence-level signal.

HammingMark alleviates these limitations by defining its acceptance neighborhood in a compact semantic hash space. Semantically similar sentences are more likely to receive similar hash codes, but Hamming agreement is not a deterministic hard cutoff on continuous cosine similarity. Owing to the coarse, many-to-one nature of a short semantic hash, candidates with different continuous similarity values may share the same or neighboring hash codes. As a result, some candidates with relatively low cosine similarity but valid discourse functions can still satisfy the Hamming constraint. Unlike a direct similarity threshold, the resulting acceptance region does not correspond to a single interval in the continuous semantic space. This quantization slack leaves more flexibility for contrast, information introduction, and topic transitions.

The discrete representation also provides more stable sequence-level evidence. HammingMark accumulates agreement over multiple projected bits, and the Global Bits detector retains partial evidence even when a modified transition no longer satisfies the complete generation threshold. In contrast, the continuous baseline relies primarily on generic inter-sentence similarity that is already present in natural text, making its detection signal more likely to overlap with the unwatermarked distribution. The random projection matrix additionally makes the HammingMark signal key-specific, whereas raw cosine similarity mainly captures key-independent semantic coherence.

Overall, this experiment shows that the advantage of HammingMark cannot be attributed solely to using the preceding sentence as the semantic reference. Even after matching the average natural acceptance rate, direct continuous similarity acceptance produces weaker detection and a substantially higher fallback rate. Compared with similarity-based selection mechanisms related to SimMark, the short Hamming representation provides a less rigid quantized acceptance region, more stable sampling behavior across contexts, and key-specific evidence that can be accumulated at the sequence level.