UNICOMARK: UNBIASED AND ROBUST MULTI-BIT LLM WATERMARKING VIA CO-DESIGN OF ERROR-RESILIENT RECOVERY AND SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-bit watermarking for large language models (LLMs) aims to embed decodable source identifiers into generated text, but exact and robust recovery remains difficult while preserving text quality in normal and adversarial settings. We propose `UniCoMark`, which co-designs two tightly coupled components: an embedding–extraction process for reliable watermark message recovery and a matched sampling process for text generation. `UniCoMark` combines a fixed-length linear code with soft-information decoding and a sampling rule tailored to the recovery objective. To ensure high text quality, we introduce an unbiasedness constraint into the sampling design, enabling the sampling process to be formulated as an optimal transport problem. We further establish a recovery theorem for `UniCoMark`, giving a message-accuracy bound that characterizes robustness under textual attacks. Extensive experiments validate the effectiveness of `UniCoMark`. In the normal setting, with 256 tokens generated by base LLMs and a 32-bit payload, `UniCoMark` achieves 94.9% message accuracy, significantly outperforming the strongest baseline at 60.2%, while also attaining lower perplexity (10.10 vs. 11.22). Under adversarial conditions, `UniCoMark` remains highly robust: for a 512-token sequence and a 32-bit payload, it achieves 88.0% message accuracy under 15% token-level mixed edits and 87.7% under 30% sentence-level mixed edits, exceeding the best baseline by 64.3% and 52.1%, respectively. An anonymized implementation is available at `https://anonymous.4open.science/r/watermarking-31C6`.

1 INTRODUCTION

Large language models (LLMs) (Achiam et al., 2023; Team et al., 2024; Touvron et al., 2023; Bai et al., 2023) are increasingly used to generate public-facing text, creating new needs for provenance, accountability, and attribution (Pudasaini et al., 2025). Inference-time watermarking addresses this need by modifying autoregressive sampling so that generated text carries a recoverable signal, without retraining the model (Cox et al., 2002; Liu et al., 2024; Zhao et al., 2025; Liang et al., 2026). While early work mainly studies zero-bit detection (i.e., determining whether a text contains a watermark) (Kirchenbauer et al., 2023; Dathathri et al., 2024), many applications, including tracing leaked outputs, distinguishing deployment endpoints, and user-level attribution, require multi-bit watermarking, where the detector recovers a source identifier or other payload from the generated text (Klasén et al., 2024; Fernandez et al., 2023; Yoo et al., 2023; Feng et al., 2025; Qu et al., 2025).

Multi-bit watermarking is harder than zero-bit detection: the detector must recover the encoded message from a large message space, and text perturbations can corrupt the watermark information needed for exact recovery (Fernandez et al., 2023). Therefore, *message accuracy*, which measures the fraction of messages that can be exactly recovered, is a key metric for evaluating multi-bit watermarking. A multi-bit watermarking design should achieve multiple properties simultaneously: i) **effectiveness** (i.e., achieving high message accuracy without attacks), ii) **robustness** (i.e., remaining high message accuracy under textual attacks such as word insertion, deletion, substitution, and their combinations) and iii) **unbiasedness** (i.e., preserving the host LLM's native sampling distribution (Hu et al., 2024; Dathathri et al., 2024)). Existing designs satisfy only partial properties; see Table 1 for more details. Achieving all three properties simultaneously is challenging. Under textual attacks, the detector must distinguish the true message from competing messages by accumulating message-

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.

dependent evidence across surviving token observations. This requires a sequence-level recovery objective, whereas the generator controls only one token at each step. Meanwhile, unbiasedness constrains the marginal distribution of each sampled token, preventing the generator from simply forcing tokens to match the encoded message. Therefore, the recovery objective must both aggregate evidence across the sequence and decompose into local terms that can guide generation while preserving the host distribution.

We propose `UniCoMark` to address these challenges through a co-design of embedding-extraction and sampling. Our key insight is to formulate the evidence to distinguish candidate messages as an additive decoding metric. The detector accumulates this metric over the observed text, while each per-token term provides a local sampling objective at the corresponding generation step. Thus, recovery and sampling optimize the same mismatch: the detector uses its accumulated value to recover the message, while the sampler minimizes its per-token contribution under the unbiasedness constraint.

Table 1: Comparison with prior work. "Focus" denotes embedding–extraction (E), sampling (S), or joint (E+S) design. "Effective"/"Robust" indicate high message accuracy without/under attacks; "Unbiased" indicates preservation of the host distribution. ✓: satisfied; △: partial; ×: not satisfied.

Method	Focus	Effective	Robust	Unbiased
MPAC (Yoo et al., 2023)	–	×	△	×
SegMark (Qu et al., 2025)	E	△	✓	×
BiMark (Feng et al., 2025)	S	△	△	✓
`UniCoMark`	E+S	✓	✓	✓

To realize this design, we encode the payload as an error-correcting codeword and define an additive candidate-message decoding metric that accumulates seed-routed token-label evidence across the observed text. This construction enables complete message recovery even when some observations are corrupted or removed. Since the decoding metric decomposes over token positions, the contribution of each sampled token becomes a Hamming mismatch between its token label and the corresponding target label. At generation time, our label-level optimal-transport sampler minimizes the expected Hamming mismatch while preserving the host-induced token-label marginal. Thus, the recovery objective directly induces the sampling objective used during generation. We further rigorously analyze message accuracy of `UniCoMark` by bounding the exact-recovery error probability, capturing the joint effects of payload size, codeword separation, sampling-induced noise, and textual attacks.

Across two base LLMs and two datasets, `UniCoMark` consistently outperforms baselines in message accuracy while preserving downstream quality. With 256 generated tokens and a 32-bit payload, it achieves 94.9% message accuracy at PPL 10.10, versus 60.2% (PPL 11.22) for the strongest baseline and 35.0% (PPL 10.58) for the unbiased baseline on Llama-3.1-8B and C4 RealNewsLike. Under textual attacks, `UniCoMark` remains robust for 32-bit messages at a generation length of 512 tokens, achieving 88.0% message accuracy under 15% token-level mixed edits and 87.7% under 30% sentence-level mixed edits, exceeding the best baseline by 64.3% and 52.1%, respectively.

Contributions. We summarize the main contributions below.

1. We design `UniCoMark`, a multi-bit watermarking scheme that co-designs embedding–extraction and generation-time sampling to achieve effectiveness, robustness, and unbiasedness.
2. We bound `UniCoMark`'s exact-recovery error probability, capturing the effects of payload size, codeword separation, sampling noise, and textual attacks.
3. Experiments show effective and robust message recovery of `UniCoMark` across both benign and adversarial settings while preserving text quality.

2 BACKGROUND

Autoregressive LLM and inference-time watermarking. Let $\mathcal{V}$ denote the token vocabulary. When distinguishing a stochastic quantity from its realization, we use uppercase letters for random variables and lowercase letters for realized values. Given the prefix $t_{1:i-1}$ and the prompt, an autoregressive LLM defines the next-token distribution $P_i(t) = \Pr(T_i = t \mid t_{1:i-1}, \text{prompt})$, $t \in \mathcal{V}$. At a fixed generation position, we also write this distribution as P_T. Inference-time watermarking leaves the model weights unchanged and modifies only the sampling step, so that the final token sequence can later be tested for a watermark or decoded into an embedded message.

Following prior inference-time watermarking schemes (Feng et al., 2025; Dathathri et al., 2024; Kirchenbauer et al., 2023), each generation step uses a detector-reproducible seed computed by a

secret-keyed hash of the preceding $(n-1)$-token prefix: $s_i = F_{\text{key}}(t_{i-n+1}, \ldots, t_{i-1})$, where F_{key} denotes a hash function instantiated with the secret key key. We use S_i to denote the corresponding seed random variable with realization s_i. The generator uses s_i to determine the local watermark target for t_i, and detector recomputes the same seed from the observed text and the key.

Multi-bit watermarking. In multi-bit watermarking, the sampler uses the prompt, the secret key, and a watermark message m to generate $\hat{t}_{1:L} \leftarrow \text{Sample}(\text{prompt}, \text{key}, m)$. The detector receives only $\hat{t}_{1:L}$ and the key, and outputs $\hat{m} = \text{Dec}_{\text{key}}(\hat{t}_{1:L})$, with the goal of *exact recovery* $\hat{m} = m$, possibly after textual attacks or other advanced attacks such as paraphrasing.

Unbiased watermarking. Though the sampler may depend on seed and message, the token marginal averaged over the seed matches the host distribution, i.e., $\mathbb{E}_{S_i}\left[P_{T|S_i,m}(t)\right] = P_T(t), \forall t \in \mathcal{V}$.

3 UniCoMark

3.1 Overview

The workflow of UniCoMark is shown in Figure 1. The single-line arrows represent embedding–extraction, and the double-line arrows represent sampling. The generator knows the LLM's next-token distribution, whereas the detector receives only the text and secret key. Our co-design therefore first specifies embedding–extraction rules that do not require these probabilities, then addresses distribution-dependent sampling. The rules define an additive decoding metric whose per-token terms determine the sampling objective under the unbiasedness constraint.

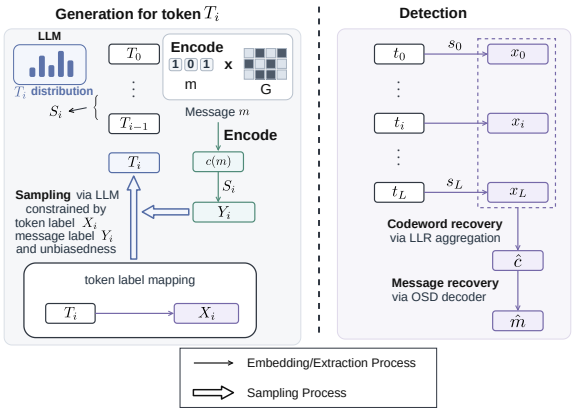


Figure 1: Workflow of UniCoMark

We first encode the message m as a fixed error-correcting codeword $c(m)$ (Lin et al., 2001). To generate the i-th token T_i, a seed random variable S_i is derived from its prefix. The seed maps coordinates of $c(m)$ to a k-bit message-label random variable Y_i. A fixed labeling map $\phi : \mathcal{V} \to \{0,1\}^k$ assigns each candidate token $t \in \mathcal{V}$ a k-bit token label; for the sampled token T_i, we write $X_i = \phi(T_i)$. Uppercase symbols denote random variables, and their realizations are denoted by s_i, t_i, x_i, and y_i.

For the detector-readable sequence $u_{n:L} = ((s_n, t_n), \ldots, (s_L, t_L))$, let $x_i = \phi(t_i)$ be the observed token label and let $y_i(m)$ be the message label determined by candidate message m and seed s_i. We define the candidate-message decoding metric

$$D(u_{n:L}; m) = \sum_{i=n}^{L} d_H(x_i, y_i(m)), \tag{1}$$

where d_H denotes Hamming distance. The metric accumulates the mismatch between the observed text and a candidate message, with a smaller value indicating stronger agreement. Its routed-bit expansion and connection to the recovery margin are given in Appendix C.

Eq. (1) provides the connection between recovery and sampling. The detector uses its accumulated value to compare complete messages, while its contribution at position i, $d_H(X_i, Y_i)$, provides the sampling cost at that generation step. Therefore, the recovery criterion directly determines the objective optimized by the sampler under the unbiasedness constraint.

(1) Embedding and Extraction. The fixed codeword provides a common recovery target for all token observations. Seed-dependent routing maps each token label to selected codeword coordinates, distributing repeated observations across the text. During detection, the same seeds and token labels are reconstructed and aggregated into coordinate-wise relative log-likelihood ratios (LLRs) (Fossorier & Lin, 1995). Local edits only corrupt/remove part of these observations, while remaining evidence continues to support the same codeword. The decoder uses the relative LLRs to recover the message.

(2) Sampling. At each position, the host next-token distribution fixes the marginal distribution of X_i, but leaves freedom in how X_i is coupled with Y_i. Our label-level optimal-transport sampler selects the coupling that minimizes $\mathbb{E}[d_H(X_i, Y_i)]$, the expected per-token contribution to Eq. (1), while preserving the required host-induced marginal.

3.2 Embedding and Extraction

We detail the embedding and extraction processes in this section, which are illustrated in Figure 2.

3.2.1 Embedding

Error-Correcting Code. We use a fixed-length binary linear ECC (Lin et al., 2001): $c(m) = mG \in \{0,1\}^{N_c}$ for $m \in \{0,1\}^r$, where G is a binary generator matrix.

Seeds. We use the seed s_i defined in Sec. 2, and denote the corresponding random variable by S_i.

Token Label. Let $\phi : \mathcal{V} \to \{0,1\}^k$ be a fixed token-labeling map shared by the generator and detector. For each token $t \in \mathcal{V}$, $\phi(t)$ is its k-bit token label. ϕ is fixed across the generation positions, seeds, and messages, and we use a hash function to instantiate it. In practice, k is small, typically $k < 8$, as limited next-token entropy makes larger labels offer little extra benefit.

For the sampled token T_i at position i, we write its token-label as $X_i := \phi(T_i)$. Our key idea is that the k bits of this token label carry information about certain coordinates of the codeword. Therefore, at the same time, the seed random variable S_i determines the indices of these coordinates, denoted as $R_{S_i} = (j_{i,1}, \ldots, j_{i,k}) \in [N_c]^k$. We will see how to use R_{S_i} below.

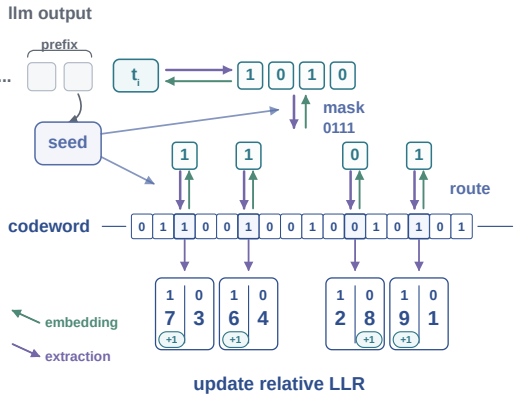


Figure 2: Embedding–Extraction Process. Green arrows show the embedding after encoding: routed bits are masked into a message label. Purple arrows show the extraction before decoding: token labels are unmasked, routed back to codeword coordinates, and accumulated as relative LLRs.

Message Label. Given the codeword $c(m)$ and the coordinate indices R_{S_i} determined by the seed S_i, the generator obtains $c(m)_{R_{S_i}} = (c(m)_{j_{i,1}}, \ldots, c(m)_{j_{i,k}})$ called routed bits, which is the information that the token T_i is expected to carry. Further, to prevent a fixed message from favoring the same token groups across contexts, the seed S_i generates a mask $a_{S_i} \in \{0,1\}^k$. Finally, the message-label random variable at position i is defined as $Y_i := c(m)_{R_{S_i}} \oplus a_{S_i}$. Watermark sampling then chooses the next token so that the token label X_i is close to the message label Y_i. The details of the sampling rule are given in Section 3.3.

3.2.2 Extraction

Codeword Recovery. Given the token sequence $(t_1, \cdots, t_L)$, for each position i, the detector recomputes the seed s_i, from which the detector further obtains the mask a_{s_i} and the coordinate indices $R_{s_i} = (j_{i,1}, \ldots, j_{i,k}) \in [N_c]^k$. At the same time, the detector obtains the corresponding token label $\phi(t_i)$ for each token t_i, and further removes the mask a_{s_i} from it, obtaining $z_i = \phi(t_i) \oplus a_{s_i}$. We denote the q-th bit of z_i as $z_{i,q}$. According to R_{s_i}, $z_{i,q}$ carries the codeword coordinate of index $j_{i,q}$. We can see that each codeword coordinate could be carried by multiple tokens. Next, we will show how to recover the codeword by using $\{z_1, \cdots, z_L\}$.

For each coordinate of the codeword, we do not immediately convert to a hard bit as in SegMark (Qu et al., 2025). Instead, we keep a relative Log-Likelihood Ratio (LLR) that reflects both the majority value and its reliability. Specifically, for each coordinate index $\ell \in [N_c]$, we define

$$N_\ell(1) = \#\{(i,q) : j_{i,q} = \ell,\ z_{i,q} = 1\}, \qquad N_\ell(0) = \#\{(i,q) : j_{i,q} = \ell,\ z_{i,q} = 0\}. \tag{2}$$

If each bit observation is approximated as independent with the same effective error rate, then the coordinate-level LLR is proportional to $N_\ell(1) - N_\ell(0)$ (*See Lemma 1 in Appendix B for the*

derivation). We therefore define $\lambda_\ell = N_\ell(1) - N_\ell(0)$ and obtain $\lambda = (\lambda_1, \ldots, \lambda_{N_c}) \in \mathbb{Z}^{N_c}$. If λ_ℓ is positive, the ℓ-th coordinate will be decided as 1, otherwise 0, with a reliability of $|\lambda_\ell|$.

Message Recovery. The vector λ is then used as a soft observation of the codeword. For each index ℓ, the sign of λ_ℓ gives a tentative bit value $\hat{c}_\ell = \mathbf{1}\{\lambda_\ell > 0\}$, while its magnitude $\rho_\ell = |\lambda_\ell|$ indicates the corresponding reliability. We recover the message by applying ordered statistics decoding (OSD) (Fossorier & Lin, 1995) to the soft observation: $\hat{m} = \mathrm{Dec}_{\mathrm{OSD}}(\lambda)$. In brief, OSD uses the reliabilities ρ_ℓ to prioritize more reliable coordinates and searches over low-order corrections on selected reliable codeword positions. The resulting valid codeword is then mapped back to the recovered message $\hat{m}$. Details are provided in Appendix B.

Remark. The OSD decoder returns a candidate message for *any* input, including text not containing a watermark (i.e., *False Attribution*). Therefore, in some scenarios where the detector does not know whether the input is watermarked beforehand, we additionally apply a watermark-presence test after recovery: the detector accepts $\hat{m}$ only when the minimum normalized decoding metric among the OSD candidates does not exceed a calibrated threshold, and returns $\perp$ otherwise (see Appendix F).

3.3 Unbiased Sampling

At position i, the host next-token distribution P_{T_i} induces a marginal over token label X_i, $P_{X_i}(x) = \sum_{t:\phi(t)=x} P_{T_i}(t)$, while the seed and message induce a marginal $P_{Y_i|m}$ over message labels Y_i. Thus, sampling can be viewed at the label level as matching the message label Y_i with X_i. By Eq. (1), each position contributes a local Hamming mismatch $d_H(X_i, Y_i)$ to the decoding metric. We therefore use $C(y, x) = d_H(y, x)$ as the Optimal Transport (OT) cost.

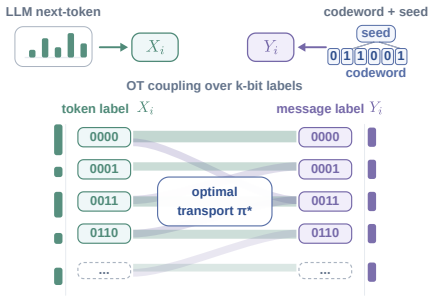


Figure 3: Label-level OT sampling. The optimal coupling matches message label Y_i to token label X_i while preserving the LLM-induced marginal P_{X_i}.

Unbiasedness requires the host token marginal to be preserved. At the label level, this means that the sampled X_i must still have marginal P_{X_i}, while the random mask a_{S_i} makes the marginal $P_{Y_i|m}$ of Y_i uniform. The remaining design choice is therefore how to couple these two label distributions so that matched labels are as close as possible. Formally,

$$\pi^\star = \arg\min_{\pi} \sum_{y,x} \pi(y,x) d_H(y,x), \text{ s.t. } \sum_x \pi(y,x) = P_{Y_i|m}(y), \ \sum_y \pi(y,x) \quad = P_{X_i}(x). \tag{3}$$

The above optimization problem is a discrete OT problem (Peyré & Cuturi, 2019), which is reviewed in Appendix D and illustrated in Figure 3. The transport matrix has $2^k \times 2^k = 2^{2k}$ entries; we use $k = 4$ by default in our experiments, resulting in a 16×16 matrix. It can be solved via linear programming (Peyré & Cuturi, 2019) or entropically regularized Sinkhorn iterations (Cuturi, 2013).

Given the optimal coupling $\pi^\star$, the label-level sampling rule and the token distribution are given by:

$$P_{X_i|Y_i=y,m}(x) = \frac{\pi^\star(y,x)}{P_{Y_i|m}(y)}, \quad P_{T_i|Y_i=y,m}(t) = P_{X_i|Y_i=y,m}(\phi(t)) \frac{P_{T_i}(t)}{P_{X_i}(\phi(t))}. \tag{4}$$

Algorithm 1–Algorithm 3 in Appendix A detail `UniCoMark`'s embedding and generation, matched OT sampling, and message recovery.

4 Message Accuracy Analysis

In this section, we rigorously analyze the message accuracy achieved by `UniCoMark` in both benign and adversarial settings. We begin by defining the sampling-induced flip rate and distinguishing tokens that carry watermark information from those that do not. Based on these definitions, we establish Theorem 1, which bounds the exact-recovery error probability. We then show how this theorem can be used to bound message accuracy in both benign and adversarial settings.

4.1 Matched Sampling and Label Flip Rate

With the embedding–extraction rule fixed, the remaining sampling problem is to reduce the noise between the message label Y_i and token label X_i at each position i. From the decoder's perspective, X_i is a noisy observation of Y_i. The sampler is optimized under the constraint that the marginal token distribution remains the host distribution P_{T_i}.

We summarize the mismatch between Y_i and X_i by the per-position average bit-flip rate p_i. The average p is taken over the watermarked tokens; we denote their number by A. After re-indexing these tokens as $i = 1, \ldots, A$, we write

$$p_i = \frac{1}{k}\mathbb{E}_{\pi_i^\star}\big[d_H(Y_i, X_i)\big], \qquad p = \frac{1}{A}\sum_{i=1}^{A} p_i. \tag{5}$$

For a generated text of length L under no attack, the first $(n-1)$ tokens do not have a full prefix for seed construction, so $A = L - (n-1)$. Here $\pi_i^\star$ is the matched coupling chosen by the OT sampling rule at position i. Thus p is the average routed-bit flip rate induced by the sampler relative to the embedding target. The OT sampling rule chooses the coupling between Y_i and X_i that minimizes this average mismatch while preserving the host token marginal.

4.2 Recovery Bound under Textual Attacks

The message m is encoded as the codeword $c(m)$. During detection, token labels are unmasked and routed by the seed to codeword coordinates, and the routed bits are aggregated into the relative LLR vector λ, enabling recovery from accumulated evidence on the codeword.

Under textual attacks, we use A to denote the number of watermarked tokens that are not affected by the attack, and B to denote the number of tokens that do not provide the watermark information. Under attack, B increases when textual edits introduce tokens or break the original seed-dependent watermark relation of later tokens. On the A tokens, the sampled token label X_i still follows the original target message label Y_i, with average bit-flip rate p. On the B tokens, X_i no longer follows the original Y_i; we treat these observations as pure noise, equivalently with effective flip rate $1/2$.

The following bound summarizes the resulting exact-recovery guarantee. For an interpretable analysis, we adopt a simplified model in which the seed-derived routes R_{S_i} distribute the k token-label bits uniformly across codeword coordinates, and decoding exactly minimizes the derived metric. In practice, deterministic keyed-hash routing and order-2 OSD approximate these analytical conditions, respectively. Their theory-to-practice correspondence is evaluated in Appendix E.3.

Theorem 1 (Recovery bound under uniform routing)**.** *Let r be the message length and N_c be the codeword length. For $\mathcal{C} = \{c(m) : m \in \{0,1\}^r\}$, define $d_{\min} := \min_{m \neq v} d_H(c(m), c(v))$ and $\Delta_{\min} := d_{\min}/N_c$. Let A and B be the numbers defined above, and define $L_{\text{eff}} = k(1-2p)^2 \frac{A^2}{A+B}$. Under the uniform routing and assumptions in Appendix E.3, the exact-recovery error probability satisfies*

$$\Pr(\hat{m}_{\text{exact}} \neq m) \leq (2^r - 1)\exp\left(-\frac{1}{2}\Delta_{\min} L_{\text{eff}}\right). \tag{6}$$

4.3 Effectiveness and Robustness from Theorem 1

Theorem 1 provides a compact characterization of exact recovery. Here $\Pr(\hat{m}_{\text{exact}} \neq m)$ is the exact-message error probability, and $1 - \Pr(\hat{m}_{\text{exact}} \neq m)$ is the corresponding message accuracy. The payload length r controls the number of competing messages, the relative distance $\Delta_{\min}$ controls codeword separation, and the aligned/noisy token counts A, B determine the effective amount of usable evidence. The sampler affects the bound through p: a smaller flip rate increases $(1-2p)^2$ and hence L_{eff}, making the error bound smaller.

Attacks affect the bound through A and B: the A tokens still carry useful information, while the B tokens introduce noise. Under the uniform-routing assumption, recovery remains likely when enough tokens counted by A survive, equivalently when $\Delta_{\min} L_{\text{eff}}$ is large relative to r.

The no-attack case is obtained by taking $A = L - (n-1)$ and $B = 0$, yielding $L_{\text{eff}} = k(1-2p)^2(L-n+1)$, and the same bound characterizes effectiveness from L received tokens.

Table 2: Payload sweep at 256 tokens. Message accuracy is reported for 16–64-bit payloads; PPL uses the 32-bit reference in the appendix. The best two results per column are highlighted.

Model	Method	ASAP-essay					C4 RealNewsLike				
		PPL	Message accuracy (%)				PPL	Message accuracy (%)			
			16 bit	32 bit	48 bit	64 bit		16 bit	32 bit	48 bit	64 bit
Llama-3.1-8B	No watermark	9.05	–	–	–	–	8.24	–	–	–	–
	MPAC $\delta = 1$	10.10	22.3	0.6	0.0	0.0	9.20	13.3	0.2	0.0	0.0
	MPAC $\delta = 2$	13.80	89.1	29.7	5.1	0.3	11.62	77.9	13.7	1.8	0.0
	BiMark	11.99	93.0	48.4	13.7	0.5	10.58	88.7	35.0	9.0	0.0
	SegMark $\delta = 1$	**9.85**	16.8	1.2	0.0	0.0	**8.91**	16.4	0.4	0.0	0.0
	SegMark $\delta = 2$	12.92	82.4	56.5	18.0	1.2	11.22	82.4	60.2	16.4	1.9
	UniCoMark $s = 0.6$	10.10	95.9	76.8	50.1	6.8	9.20	93.8	67.0	41.0	3.9
	UniCoMark $s = 1.0$	11.44	**98.4**	**96.3**	**94.1**	**80.7**	10.10	**97.9**	**94.9**	**92.4**	**78.5**
Gemma-2-9B	No watermark	9.49	–	–	–	–	8.64	–	–	–	–
	MPAC $\delta = 1$	10.58	20.9	0.4	0.0	0.0	9.49	18.0	0.0	0.0	0.0
	MPAC $\delta = 2$	13.80	89.5	30.3	4.4	0.3	11.62	80.5	19.7	3.6	0.0
	BiMark	12.57	94.5	50.8	16.8	0.3	10.92	91.6	36.1	9.5	0.0
	SegMark $\delta = 1$	**9.98**	19.3	1.7	0.0	0.0	**9.03**	17.1	0.8	0.0	0.0
	SegMark $\delta = 2$	12.25	89.7	59.2	21.4	2.0	11.47	85.7	59.9	17.7	2.5
	UniCoMark $s = 0.6$	10.26	98.8	81.6	54.7	7.3	9.20	96.3	67.4	36.4	2.8
	UniCoMark $s = 1.0$	11.27	**100.0**	**98.8**	**97.6**	**91.2**	9.94	**100.0**	**99.0**	**97.2**	**80.5**

Table 3: Length sweep for a 32-bit payload. PPL and message accuracy are reported at 128, 256, and 512 tokens; the best result per column is highlighted.

Model	Method	ASAP-essay						C4 RealNewsLike					
		128 tokens		256 tokens		512 tokens		128 tokens		256 tokens		512 tokens	
		PPL	Acc.	PPL	Acc.	PPL	Acc.	PPL	Acc.	PPL	Acc.	PPL	Acc.
Llama-3.1-8B	MPAC $\delta = 2$	14.46	2.5	13.80	29.7	12.56	73.5	12.37	0.6	11.62	13.7	11.26	58.7
	BiMark	12.97	9.8	11.99	48.4	10.58	80.3	10.92	4.7	10.58	35.0	10.10	82.6
	SegMark $\delta = 2$	13.47	7.4	12.92	56.5	11.71	83.3	12.18	10.5	11.22	60.2	10.57	82.7
	UniCoMark $s = 0.6$	**10.75**	24.4	**10.10**	76.8	**8.77**	90.4	**9.71**	13.7	**9.20**	67.0	**8.91**	93.8
	UniCoMark $s = 1.0$	11.99	**94.1**	11.44	**96.3**	10.09	**96.8**	10.58	**81.1**	10.10	**94.9**	9.86	**97.6**
Gemma-2-9B	MPAC $\delta = 2$	14.46	2.9	13.80	30.3	12.76	78.6	12.65	2.7	11.62	19.7	11.09	72.0
	BiMark	13.38	9.8	12.57	50.8	11.27	84.4	12.00	4.7	10.92	36.1	10.42	82.6
	SegMark $\delta = 2$	13.17	9.8	12.25	59.2	10.98	86.7	12.11	10.7	11.47	59.9	10.58	88.9
	UniCoMark $s = 0.6$	**11.44**	26.8	**10.26**	81.6	**9.19**	96.0	**9.45**	13.7	**9.20**	67.4	**8.74**	98.3
	UniCoMark $s = 1.0$	12.18	**96.1**	11.27	**98.8**	10.42	**99.2**	10.42	**85.9**	9.94	**99.0**	9.49	**100.0**

5 EXPERIMENTS

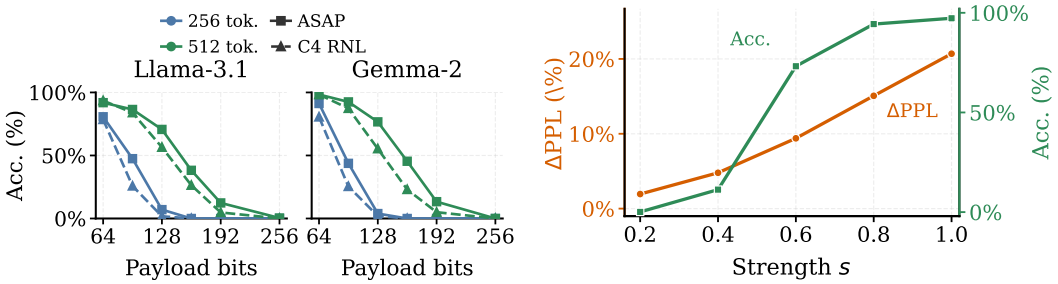
Setup. We evaluate watermarking performance on Gemma-2-9B (Team et al., 2024) and Llama-3.1-8B (Grattafiori et al., 2024). Generation prefixes are drawn from ASAP-essay (Shermis, 2014) and a random subset of C4 RealNewsLike (Raffel et al., 2020). Generated continuations contain 128–512 token length. We use *message accuracy* to measure exact recovery: the fraction of samples whose decoded message has a 100% bit match with the embedded message. Message accuracy is evaluated both with and without attacks, along with perplexity (PPL) to measure text quality.

We compare UniCoMark with MPAC (Yoo et al., 2023), SegMark (Qu et al., 2025), and BiMark (Feng et al., 2025). In MPAC and SegMark, $\delta \in \{1, 2\}$ controls the sampling-bias strength. In UniCoMark, the trade-off sweep varies s in $P_s = (1 - s)P_T + sP_{\mathrm{OT}}$, where $s = 0$ gives ordinary LLM sampling and $s = 1$ gives the full OT sampler. For downstream quality, we follow prior works (Feng et al., 2025; Chen et al., 2025) and evaluate summarization on CNN/DailyMail (See et al., 2017) using `bart-large-cnn` (Lewis et al., 2020), and translation on WMT16 En→Cs (Bojar et al., 2016) using `mbart-large-50-many-to-many-mmt` (Tang et al., 2020). **More experimental settings and results are deferred to Appendix H.**

5.1 RESULTS UNDER NO ATTACK

Recovery under no attack. Table 2 shows that UniCoMark gives the strongest message accuracy across both models and prompt sources while keeping PPL competitive. At 32 bits, UniCoMark reaches 94.9–99.0% accuracy; at 64 bits, UniCoMark with $s = 1.0$ still reaches 78.5–91.2%.

Table 3 shows the same trend under shorter decoding lengths. For 32-bit payloads, UniCoMark with $s = 1.0$ reaches 81.1–96.1% accuracy at 128 tokens and 96.8–100.0% at 512 tokens. Overall, UniCoMark remains reliable at 32 and 48 bits, remains usable at 64 bits, and reaches high accuracy with shorter decoding lengths. Appendix I.3 reports the full message-length × decoding-length sweeps for both message accuracy and bit accuracy.



(a) **Payload capacity under fixed decoding lengths.** Message accuracy as a function of payload length for `UniCoMark` at $s = 1$.

(b) **Strength–quality trade-off.** Message accuracy and relative PPL vs. s (Llama-3.1-8B, C4 RealNewsLike; 32-bit payload; 256 tokens).

Figure 4: **Recovery under no attacks and quality trade-offs.** Left: payload scaling under fixed decoding lengths. Right: recovery–quality trade-off as watermark strength increases.

Table 4: Downstream task quality under reproduced evaluation pipelines. Left: CNN/DailyMail summarization with BART-large-CNN. Right: WMT16 En→Cs translation with mBART-large-50.

Summarization

Method	ROUGE-1 F1 (%)	BERTScore F1 (%)
No watermark	38.9 ± 12.7	87.6 ± 2.3
MPAC $\delta = 1$	38.1 ± 11.8	87.4 ± 2.1
MPAC $\delta = 2$	36.7 ± 11.1	87.1 ± 2.0
BiMark	37.5 ± 11.0	87.3 ± 2.0
SegMark $\delta = 1$	38.3 ± 11.7	87.1 ± 2.1
SegMark $\delta = 2$	36.0 ± 11.2	87.4 ± 2.3
`UniCoMark` $s = 1.0$	38.0 ± 11.5	87.4 ± 2.1
`UniCoMark` $s = 0.8$	38.8 ± 11.9	87.5 ± 2.2
`UniCoMark` $s = 0.6$	37.9 ± 11.6	87.4 ± 2.2

Translation

Method	Sentence BLEU	Corpus BLEU
No watermark	16.1008 ± 14.9396	15.5255
MPAC $\delta = 1$	14.6532 ± 12.9139	14.0274
MPAC $\delta = 2$	11.9092 ± 11.4164	10.9240
BiMark	13.8294 ± 12.5706	13.1443
SegMark $\delta = 1$	14.8932 ± 13.9812	13.9174
SegMark $\delta = 2$	12.0072 ± 12.1942	11.1735
`UniCoMark` $s = 1.0$	14.6087 ± 13.3300	14.2696
`UniCoMark` $s = 0.8$	16.1448 ± 14.4899	15.4651
`UniCoMark` $s = 0.6$	16.0943 ± 14.4106	15.4893

Figure 4a visualizes message accuracy vs. payload size: accuracy remains high when payload is up to one quarter of the decoding length, and decreases approximately linearly as the size increases.

Recovery–quality trade-off. Figure 4b shows that increasing s improves recovery but raises relative PPL, revealing a trade-off between recovery and generation quality under unbiased sampling.

Downstream task quality. Table 4 shows that `UniCoMark` achieves task quality close to the no-watermark baseline. On CNN/DailyMail summarization, `UniCoMark` with $s = 0.8$ gives ROUGE-1 38.8 and BERTScore 87.5, compared with 38.9 and 87.6 without watermarking. On WMT16 En→Cs translation, `UniCoMark` with $s = 0.6/s = 0.8$ gives BLEU 15.4893/15.4651, compared with 15.5255 without watermarking. Thus, our `UniCoMark` does not decrease downstream quality.

False attribution. Table 5 shows the results on Llama-3.1-8B and C4. Using the minimum normalized decoding metric with a threshold calibrated at FPR=1%, the detector identifies 97.6% of watermarked outputs. Overall, 93.2% of the watermarked outputs are both identified and decoded correctly.

Table 5: False attribution at 1% FPR.

Metric	**Rate**
Watermark detected	97.6%
Detected and decoded correctly	93.2%

Ablation study. Table 6 combines BiMark and `UniCoMark` sampling (S_B, S_U) with BiMark and `UniCoMark` recovery (E_B, E_U); E_U uses relative-LLR aggregation and order-2 OSD. $S_U + E_U$ achieves the highest message accuracy in all settings. At fixed recovery, S_U exceeds S_B; at fixed sampling, E_U exceeds E_B. Thus, both the sampling and recovery components contribute to improved message accuracy.

Table 6: Ablation of the sampling and embedding–extraction components. Results show message accuracy (%) at 256 generated tokens with a 32-bit payload.

Model	Dataset	$S_B + E_B$	$S_B + E_U$	$S_U + E_B$	$S_U + E_U$
Llama-3.1-8B	ASAP	48.4	90.2	78.9	**96.3**
Llama-3.1-8B	C4	35.0	83.2	67.8	**94.9**
Gemma-2-9B	ASAP	50.8	93.6	84.0	**98.8**
Gemma-2-9B	C4	36.1	83.0	71.7	**99.0**

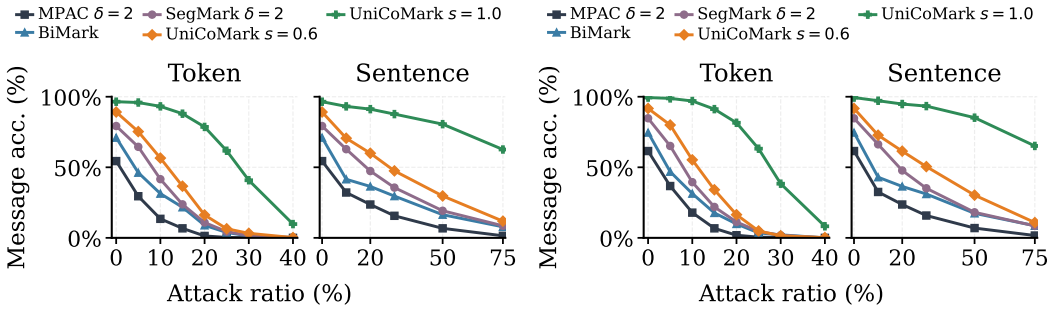
5.2 Robustness Under Attack

Mixed edit attacks select insertion, deletion, or replacement by a pseudorandom seed for each attacked unit. For paraphrase robustness, we use DIPPER (Krishna et al., 2023). In DIPPER, $lXoY$ denotes lexical diversity X and order diversity Y; larger values indicate stronger paraphrasing.

Table 7: Robustness under DIPPER attack (32-bit payload; 512-token generations; message acc, %).

Llama-3.1-8B						
Method	Orig.	l20/o0	l20/o20	l40/o0	l40/o40	l80/o80
MPAC $\delta = 2$	54.4	0.3	0.6	0.0	0.0	0.0
BiMark	71.0	0.9	1.0	0.0	0.0	0.0
SegMark $\delta = 2$	79.3	1.4	1.2	0.0	0.0	0.0
UniCoMark $s = 0.6$	89.2	3.4	2.7	0.4	0.2	0.0
UniCoMark $s = 1.0$	**96.5**	**35.3**	**33.0**	**3.1**	**1.0**	0.0

Gemma-2-9B						
Method	Orig.	l20/o0	l20/o20	l40/o0	l40/o40	l80/o80
MPAC $\delta = 2$	61.5	0.4	0.0	0.0	0.0	0.0
BiMark	74.6	1.6	0.0	0.0	0.0	0.0
SegMark $\delta = 2$	84.7	1.7	0.0	0.0	0.0	0.0
UniCoMark $s = 0.6$	91.7	1.6	0.0	0.0	0.0	0.0
UniCoMark $s = 1.0$	**99.4**	**26.1**	**1.2**	0.0	0.0	0.0



(a) Llama-3.1-8B mixed edit robustness.

(b) Gemma-2-9B mixed edit attacks.

Figure 5: Robustness under mixed edit attacks (32-bit payload; 512-token generations).

Figure 5 shows that UniCoMark is more robust under token- and sentence-level edits. At 15% token-level mixed edits and 30% sentence-level mixed edits, UniCoMark with $s = 1.0$ keeps about 88% accuracy. This matches the robustness analysis in section 4.2: local edits remove or corrupt detector-visible observations, while relative LLRs aggregate the remaining evidence for recovery.

Table 7 shows DIPPER (Krishna et al., 2023) is a much stronger attack than textual editing. All tested baselines are ineffective under paraphrasing, but UniCoMark remains the strongest under mild and medium settings, especially at $s = 1.0$. For $l20o0$/$l20o20$, UniCoMark reaches 35.3%/33.0% accuracy. Stronger paraphrasing, e.g., $l40o40$ largely defeats all methods. This is the main remaining limitation: strong paraphrasing can replace or reorder the local evidence used by the detector.

6 Related Work

Zero-bit and unbiased watermarking. Zero-bit methods embed and detect a keyed signal, with follow-up work improving power, robustness, and deployability (Kirchenbauer et al., 2023; Zhao et al., 2024; Dathathri et al., 2024). Unbiased methods preserve the host distribution through Gumbel sampling, reweighting, coupling, or optimal transport (Hu et al., 2024; Wu et al., 2024; Chen et al., 2025; Tsur et al., 2025); they motivate our marginal constraint but not robust exact multi-bit recovery.

Multi-bit watermarking. MPAC assigns tokens to message positions (Yoo et al., 2023), SegMark uses pseudo-random segments and robust aggregation (Qu et al., 2025), and BiMark extends unbiased reweighting (Feng et al., 2025); Table 1 compares them. We omit the code-unavailable StealthInk (Jiang et al., 2025), whose reported bit accuracy (Table 4) is below MPAC and SegMark. ArcMark (Gilani et al., 2026) also uses OT but focuses on rate–capacity tradeoffs, whereas we emphasize robust exact-message recovery under textual attacks. The different evaluation objectives preclude a direct comparison, and its implementation is currently unavailable.

Orthogonal directions. These include cryptographically verifiable watermarks (Christ et al., 2024; Christ & Gunn, 2024; Cohen et al., 2025); in-context, semantic, or sentence-level watermarking (Liu et al., 2025; Lu et al., 2024; Liu et al., 2023; Huo et al., 2026; Liu & Bu, 2024; Hou et al., 2024; Munyer et al., 2024); and studies of paraphrasing, stealing, spoofing, and benchmark-based evaluation (Giboulot & Furon, 2024; Jovanović et al., 2024; Tu et al., 2024; Piet et al., 2025; Liang et al., 2024; Pang et al., 2024; Omidi et al., 2026).

7 Conclusion

We present UniCoMark, an unbiased and robust multi-bit LLM watermarking method that co-designs code-based embedding–extraction with unbiased OT sampling. Our analysis gives an exact-recovery guarantee under sampling noise and textual attacks. Experiments show robust recovery in clean and adversarial settings, outperforming strong baselines while preserving text quality.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, manuscript restructuring, and summarizing author-provided experimental results in the appendix. The authors reviewed the AI-assisted text and numerical statements against their research records and take responsibility for the final content, claims, and artifacts in this work.

REFERENCES

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

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, et al. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pp. 131–198, 2016.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Improved unbiased watermark for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20587–20601, 2025.

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

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

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

Ingemar Cox, Matthew Miller, Jeffrey Bloom, and Chris Honsinger. Digital watermarking. *Journal of Electronic Imaging*, 11(3):414–414, 2002.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

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*, 2024.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. Bimark: Unbiased multilayer watermarking for large language models. In *International Conference on Machine Learning*, 2025.

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

Marc PC Fossorier and Shu Lin. Soft-decision decoding of linear block codes based on ordered statistics. *IEEE Transactions on information Theory*, 41(5):1379–1396, 1995.

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

Atefeh Gilani, Carol Xuan Long, Sajani Vithana, Oliver Kosut, Lalitha Sankar, and Flavio P Calmon. Arcmark: Multi-bit llm watermark via optimal transport. *arXiv preprint arXiv:2602.07235*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 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 *2024 Conference of the North American Chapter of the Association for Computational Linguistics*, 2024.

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*, 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*, 2026.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. Stealthink: A multi-bit and stealthy watermark for large language models. In *International Conference on Machine Learning*, 2025.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. In *International Conference on Machine Learning*, 2024.

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*, 2023.

Lena Klasén, Niclas Fock, and Robert Forchheimer. The invisible evidence: Digital forensics as key to solving crimes in the digital age. *Forensic science international*, 362:112133, 2024.

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020.

Jiacheng Liang, Zian Wang, Spencer Hong, Shouling Ji, and Ting Wang. Watermark under fire: A robustness evaluation of llm watermarking. *arXiv preprint arXiv:2411.13425*, 2024.

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.

Shu Lin, Daniel J Costello, et al. *Error control coding*, volume 2. Prentice hall Scarborough, 2001.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. *arXiv preprint arXiv:2310.06356*, 2023.

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

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

Yepeng Liu, Xuandong Zhao, Christopher Kruegel, Dawn Song, and Yuheng Bu. In-context watermarks for large language models. *arXiv preprint arXiv:2505.16934*, 2025.

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.

Travis Munyer, Abdullah All Tanvir, Arjon Das, and Xin Zhong. Deeptextmark: a deep learning-driven text watermarking approach for identifying large language model generated text. *Ieee Access*, 12:40508–40520, 2024.

Romina Omidi, Yun Dong, and Binghui Wang. Google's llm watermarking system is vulnerable to layer inflation attack. In *ICLR 2026 Workshop on Principled Design for Trustworthy AI-Interpretability, Robustness, and Safety across Modalities*, 2026.

Qi Pang, Shengyuan Hu, Wenting Zheng, and Virginia Smith. Attacking llm watermarks by exploiting their strengths. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*, 2024.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. Markmywords: Analyzing and evaluating language model watermarks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning*, 2025.

Shushanta Pudasaini, Luis Miralles-Pechuán, David Lillis, and Marisa Llorens Salvador. Survey on ai-generated plagiarism detection: The impact of large language models on academic integrity: S. pudasaini et al. *Journal of Academic Ethics*, 23(3):1137–1170, 2025.

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

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

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

Mark D Shermis. State-of-the-art automated essay scoring: Competition, results, and future directions from a united states demonstration. *Assessing Writing*, 20:53–76, 2014.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*, 2020.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Dor Tsur, Carol Xuan Long, Claudio Mayrink Verdun, Sajani Vithana, Hsiang Hsu, Chun-Fu Chen, Haim H Permuter, and Flavio Calmon. Heavywater and simplexwater: Distortion-free llm watermarks for low-entropy distributions. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. Waterbench: Towards holistic evaluation of watermarks for large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

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

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

KiYoon Yoo, Wonhyuk Ahn, Jiho Jang, and Nojun Kwak. Robust multi-bit natural language watermarking through invariant features. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2092–2115, 2023.

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

Xuandong Zhao, Sam Gunn, Miranda Christ, Jaiden Fairoze, Andres Fabrega, Nicholas Carlini, Sanjam Garg, Sanghyun Hong, Milad Nasr, Florian Tramer, et al. Sok: Watermarking for ai-generated content. In *IEEE Symposium on Security and Privacy*, 2025.

A ALGORITHMS

Algorithm 1 `UniCoMark` embedding and generation

Input: Payload $m \in \{0,1\}^r$, prompt, secret key, generator matrix $G \in \mathbb{F}_2^{r \times N_c}$, token-label map $\phi : V \to \{0,1\}^k$, generation length L, strength $s \in [0,1]$.
Output: Watermarked token sequence $\widehat{t}_{1:L}$.

$c(m) \leftarrow mG \pmod 2$
for $i = 1, \ldots, L$ **do**
  Obtain the host next-token distribution $P_{T_i}(\cdot) = \Pr(T_i = \cdot \mid \text{prompt}, \widehat{t}_{1:i-1})$.
  if $i < n$ **then**
    Sample $\widehat{t}_i \sim P_{T_i}$.
  else
    Compute the seed $s_i \leftarrow F_{\text{key}}(\widehat{t}_{i-n+1}, \ldots, \widehat{t}_{i-1})$.
    $\widehat{t}_i \leftarrow$ MATCHEDOTSAMPLING$(P_{T_i}, s_i, c(m), \phi, s)$.
  end if
end for
return $\widehat{t}_{1:L}$.

Algorithm 2 MATCHEDOTSAMPLING: matched OT sampling

Input: Host distribution $P_T \in \Delta(V)$, seed s_i, codeword $c(m) \in \{0,1\}^{N_c}$, token-label map $\phi : V \to \{0,1\}^k$, strength $s \in [0,1]$.
Output: Sampled token t_i.

Derive the route $R_{s_i} = (j_{i,1}, \ldots, j_{i,k}) \in [N_c]^k$ and mask $a_{s_i} \in \{0,1\}^k$ from s_i.
Form the message label $y_i \leftarrow (c(m)_{j_{i,1}}, \ldots, c(m)_{j_{i,k}}) \oplus a_{s_i}$.
for all $x \in \{0,1\}^k$ **do**
  $P_X(x) \leftarrow \sum_{t:\phi(t)=x} P_T(t)$.
  $P_Y(x) \leftarrow 2^{-k}$.
end for
Solve the label-level optimal transport problem $\pi^\star = \arg\min_{\pi \in \Pi(P_Y, P_X)} \sum_{y,x \in \{0,1\}^k} \pi(y,x) d_H(y,x)$.
for all $x \in \{0,1\}^k$ **do**
  $P_{X|Y=y_i}(x) \leftarrow \pi^\star(y_i, x) / P_Y(y_i)$.
end for
for all $t \in \text{supp}(P_T)$ **do**
  $P_{\text{OT}}(t) \leftarrow P_{X|Y=y_i}(\phi(t)) P_T(t) / P_X(\phi(t))$.
end for
$P_s \leftarrow (1-s) P_T + s P_{\text{OT}}$.
Sample $t_i \sim P_s$.
return t_i.

Algorithm 3 `UniCoMark` extraction and message recovery

Input: Observed token sequence $\widetilde{t}_{1:L}$, secret key, token-label map $\phi : V \to \{0,1\}^k$, generator matrix $G \in \mathbb{F}_2^{r \times N_c}$.
Output: Recovered watermark message $\widehat{m}$.

for $\ell = 1, \ldots, N_c$ **do**
  $N_\ell(0) \leftarrow 0$, $N_\ell(1) \leftarrow 0$.
end for
for $i = n, \ldots, L$ **do**
  Recompute the seed $s_i \leftarrow F_{\text{key}}(\widetilde{t}_{i-n+1}, \ldots, \widetilde{t}_{i-1})$.
  Derive the route $R_{s_i} = (j_{i,1}, \ldots, j_{i,k})$ and mask a_{s_i} from s_i.
  Extract the unmasked token label $z_i \leftarrow \phi(\widetilde{t}_i) \oplus a_{s_i}$.
  for $q = 1, \ldots, k$ **do**
    $\ell \leftarrow j_{i,q}$.
    $N_\ell(z_{i,q}) \leftarrow N_\ell(z_{i,q}) + 1$.
  end for
end for
for $\ell = 1, \ldots, N_c$ **do**
  $\lambda_\ell \leftarrow N_\ell(1) - N_\ell(0)$.
end for
Recover the payload by OSD: $\widehat{m} \leftarrow \mathrm{OSD}(\lambda, G)$.
return $\widehat{m}$.

B Relative LLR aggregation and OSD Decoding

For a seed s_i, let

$$R_{s_i} = (j_{i,1}, \ldots, j_{i,k}) \in [N_c]^k$$

be the route into outer-code coordinates, and let $a_{s_i} \in \{0,1\}^k$ be the mask. For token t_i, define the unmasked token label

$$z_i = \phi(t_i) \oplus a_{s_i}. \tag{7}$$

Thus each component $z_{i,q}$, together with its route $j_{i,q}$, is a routed bit assigned to an outer-code coordinate. For coordinate index $\ell \in [N_c]$ and bit $b \in \{0,1\}$, define the number of routed bits with value b at coordinate ℓ:

$$N_\ell(b) = \sum_{i=n}^{L} \sum_{q=1}^{k} \mathbf{1}\{j_{i,q} = \ell,\ z_{i,q} = b\}. \tag{8}$$

The relative LLR, hard decision, and reliability are

$$\lambda_\ell = N_\ell(1) - N_\ell(0), \qquad \hat{y}_\ell = \mathbf{1}\{\lambda_\ell > 0\}, \qquad \rho_\ell = |\lambda_\ell|. \tag{9}$$

When $\lambda_\ell = 0$, the hard decision is arbitrary because $\rho_\ell = 0$.

Lemma 1 (Additive relative LLR under a binary symmetric channel(BSC) abstraction)**.** *Fix a codeword coordinate index ℓ, and let*

$$\mathcal{O}_\ell = \{(i,q) : j_{i,q} = \ell\}, \qquad \mathcal{Z}_\ell = \{z_{i,q} : (i,q) \in \mathcal{O}_\ell\}. \tag{10}$$

Conditioned on the true codeword bit c_ℓ, suppose the routed bits in $\mathcal{Z}_\ell$ are independent and pass through the same BSC(p):

$$\Pr(z = c_\ell) = 1 - p, \qquad \Pr(z \neq c_\ell) = p, \qquad 0 < p < \frac{1}{2}. \tag{11}$$

Then the coordinate-level log-likelihood ratio is

$$\log \frac{\Pr(\mathcal{Z}_\ell \mid c_\ell = 1)}{\Pr(\mathcal{Z}_\ell \mid c_\ell = 0)} = \big(N_\ell(1) - N_\ell(0)\big) \log \frac{1-p}{p}. \tag{12}$$

Equivalently,

$$\log \frac{\Pr(\mathcal{Z}_\ell \mid c_\ell = 0)}{\Pr(\mathcal{Z}_\ell \mid c_\ell = 1)} = \big(N_\ell(1) - N_\ell(0)\big) \log \frac{p}{1-p}. \tag{13}$$

Thus $\lambda_\ell = N_\ell(1) - N_\ell(0)$ is a positive rescaling of the 1-versus-0 LLR in Eq. (12); its sign gives the hard decision and $|\lambda_\ell|$ gives the reliability.

Proof. Under Eq. (11),

$$\Pr(\mathcal{Z}_\ell \mid c_\ell = 0) = (1-p)^{N_\ell(0)} p^{N_\ell(1)}$$

and

$$\Pr(\mathcal{Z}_\ell \mid c_\ell = 1) = p^{N_\ell(0)} (1-p)^{N_\ell(1)}.$$

Taking the log ratio gives Eq. (12); reversing the ratio gives Eq. (13). □

The vector $\lambda = (\lambda_1, \ldots, \lambda_{N_c})$ is used as the soft input to the outer code. With hard decisions $\hat{y}_\ell$ and reliabilities ρ_ℓ, the corresponding soft-decision decoding objective is the weighted nearest-codeword problem

$$\min_{m \in \{0,1\}^r} \sum_{\ell=1}^{N_c} \rho_\ell \mathbf{1}\{c_\ell(m) \neq \hat{y}_\ell\}. \tag{14}$$

We use ordered statistics decoding (OSD) (Fossorier & Lin, 1995) to approximately minimize Eq. (14).

Let $G \in \mathbb{F}_2^{r \times N_c}$ be the generator matrix. We first scan coordinates in decreasing order of reliability ρ_ℓ and select r linearly independent columns of G. We place these columns first and apply the same reordering to the hard decisions. Let σ denote the resulting coordinate order and Π_σ its permutation matrix. The permuted inputs are

$$\hat{y}^\sigma = (\hat{y}_{\sigma(1)}, \ldots, \hat{y}_{\sigma(N_c)}), \qquad G^\sigma = G\Pi_\sigma. \tag{15}$$

Since the first r columns are linearly independent, Gaussian elimination transforms G^σ into systematic form:

$$\widetilde{G}^\sigma = [I_r \; P]. \tag{16}$$

The hard decisions at these selected positions are

$$\hat{b} = (\hat{y}_1^\sigma, \ldots, \hat{y}_r^\sigma). \tag{17}$$

Order-p_{OSD} OSD tests changes to at most p_{OSD} of these bits. It enumerates the patterns

$$e \in \{0,1\}^r, \qquad \mathrm{wt}_H(e) \leq p_{\mathrm{OSD}}. \tag{18}$$

For each pattern, we change the selected bits and re-encode them to obtain a candidate codeword, then restore the original coordinate order:

$$b^{(e)} = \hat{b} \oplus e, \qquad c^{(e),\sigma} = b^{(e)} \widetilde{G}^\sigma, \qquad c^{(e)} = c^{(e),\sigma} \Pi_\sigma^{-1}. \tag{19}$$

Each candidate is scored by the weighted mismatch in Eq. (14):

$$D_{\mathrm{OSD}}^{(e)} = \sum_{\ell=1}^{N_c} \rho_\ell \mathbf{1}\{c_\ell^{(e)} \neq \hat{y}_\ell\}. \tag{20}$$

We select the codeword with the smallest score and recover its corresponding message. The number of candidates is

$$\sum_{j=0}^{p_{\mathrm{OSD}}} \binom{r}{j}. \tag{21}$$

We use $p_{\mathrm{OSD}} = 2$ by default.

OSD approximation check. OSD searches only its generated candidates, whereas exact decoding searches all valid codewords.

We perform a small exact-enumeration check at 16- and 24-bit payloads, where exhaustive maximum-likelihood decoding remains feasible. As shown in Table 8, OSD-2, OSD-3, and exact decoding recover the message in all evaluated cases at both payload sizes. This check verifies the behavior of OSD at these payload sizes; it is not a worst-case guarantee for OSD.

Table 8: Exact-enumeration check for OSD. Results are message accuracy (%).

Payload	OSD-2	OSD-3	Exact ML
16 bits	100.00	100.00	100.00
24 bits	100.00	100.00	100.00

C Decoding-Margin View of Matched Unbiased Sampling

This section expands the main-text decoding metric for the sampling analysis and noisy recovery proof. The metric is not an additional decoder and is not needed to implement OSD. It assigns, for a fixed final text and a candidate message, the total local mismatch between the candidate codeword and the routed bits constructed by the detector.

C.1 From recovery margin to local Hamming cost

For a seed s, let

$$R_s = (j_{s,1}, \dots, j_{s,k}) \in [N_c]^k, \qquad a_s \in \{0,1\}^k.$$

For token t, define the unmasked token label

$$z(s,t) = \phi(t) \oplus a_s. \tag{22}$$

For a candidate message m, let $c(m) \in \mathcal{C} \subseteq \{0,1\}^{N_c}$ be its codeword and define the message label

$$q_s(m) = c(m)_{R_s} \oplus a_s. \tag{23}$$

The local candidate-message decoding metric is

$$f_s(t;m) = d_H(\phi(t), q_s(m)) = d_H(z(s,t), c(m)_{R_s}). \tag{24}$$

For each position i, the candidate message label in the main-text metric is $y_i(m) = q_{s_i}(m)$. For the detector-readable sequence $u_{n:L} = ((s_n,t_n),\dots,(s_L,t_L))$, write $R_{s_i} = (j_{i,1},\dots,j_{i,k})$ and $z_i = z(s_i,t_i)$. Then the main-text metric in Eq. (1) can be expanded as

$$D(u_{n:L};m) = \sum_{i=n}^{L} f_{s_i}(t_i;m) = \sum_{i=n}^{L}\sum_{q=1}^{k} \mathbf{1}\{z_{i,q} \neq c_{j_{i,q}}(m)\}. \tag{25}$$

For a true message m and a competing message v, define the recovery margin

$$\Delta_{m,v}(u_{n:L}) = D(u_{n:L};v) - D(u_{n:L};m). \tag{26}$$

A negative margin means that the competitor v has smaller loss than the true message. A smooth loss that penalizes small margins is

$$L(u_{n:L},m) = \log\left(\frac{1}{|\mathcal{M}|-1}\sum_{v\neq m} e^{-\Delta_{m,v}(u_{n:L})}\right). \tag{27}$$

At one detector-readable unit (s,t), let V be a uniformly sampled competing message and define the one-step margin loss

$$\psi(s,t;m) = f_s(t;m) + \log \mathbb{E}_V[e^{-f_s(t;V)}]. \tag{28}$$

Lemma 2 (Reduction to local Hamming distance)**.** *If the competing target pattern*

$$c(V)_{R_s} \oplus a_s \tag{29}$$

is uniform on $\{0,1\}^k$*, then* $\psi(s,t;m)$ *differs from*

$$d_H(\phi(t), q_s(m)) \tag{30}$$

by a constant independent of t *and* m*.*

Proof. Under the uniformity assumption,

$$f_s(t;V) = d_H(\phi(t), c(V)_{R_s} \oplus a_s)$$

has the same distribution as the Hamming distance between a fixed k-bit vector and a uniform k-bit vector. This distribution depends only on k, so

$$\mathbb{E}_V[e^{-f_s(t;V)}]$$

is constant in t and m. The only term depending on the true message and the sampled token label is therefore

$$f_s(t;m) = d_H(\phi(t), q_s(m)).$$

□

Thus, at a generation step, minimizing the one-step recovery loss reduces to minimizing

$$d_H(Y,X), \qquad Y = q_S(m), \qquad X = \phi(T). \tag{31}$$

This gives the Hamming cost used by the main-text OT sampler.

D Optimal Transport

Optimal transport (Peyré & Cuturi, 2019) studies how to match two probability distributions while minimizing an expected transport cost. In the finite discrete case, let μ and ν be distributions on finite sets $\mathcal{A}$ and $\mathcal{B}$, and let $C(a, b)$ be the cost of matching $a \in \mathcal{A}$ with $b \in \mathcal{B}$. A coupling π is a joint distribution on $\mathcal{A} \times \mathcal{B}$ whose marginals are μ and ν:

$$\sum_b \pi(a, b) = \mu(a), \qquad \sum_a \pi(a, b) = \nu(b).$$

The discrete optimal transport problem is

$$\min_{\pi \in \Pi(\mu,\nu)} \sum_{a,b} \pi(a, b)C(a, b),$$

where $\Pi(\mu, \nu)$ denotes the set of all couplings with these prescribed marginals (Peyré & Cuturi, 2019). Thus, optimal transport gives a minimum-cost way to couple two distributions while keeping both marginals fixed.

Let $d = 2^k$. The discrete problem has $d^2 = 2^{2k}$ transport variables and $2d$ marginal constraints. We solve its entropically regularized form with GPU Sinkhorn iterations (Cuturi, 2013; Peyré & Cuturi, 2019); at the experimental setting $k = 4$, each transport matrix is 16×16. Measured end-to-end generation throughput is reported in Table 10.

E Recovery Theorem

The noisy recovery analysis uses the decoding metric D in Eq. (1). This keeps the proof aligned with the detector-side candidate-message loss: recovery succeeds when the true message has smaller metric value than every incorrect message. For a true message m and a deterministic competitor $v \neq m$, use the metric margin $M_{m,v} := \Delta_{m,v}(u_{n:L})$ from Eq. (26). When $M_{m,v} > 0$, the true message m beats competitor v. A decoding error can occur only if some deterministic competitor $v \neq m$ has $M_{m,v} \leq 0$.

Perturbations change the routed bits available to the decoder. Deletion removes routed bits. Replacement removes biased routed bits and may add zero-mean routed bits. Sampling mismatch flips routed bits. Throughout this section, a routed bit is indexed by $o = (i, q)$, with routed coordinate $\ell_o = j_{i,q}$ and value $z_o = z_{i,q}$.

E.1 Pairwise margin decomposition

Lemma 3 (Pairwise margin decomposition)**.** *Fix a true message $m \in \mathcal{M}$ and a deterministic competitor $v \in \mathcal{M} \setminus \{m\}$. Let*

$$\mathcal{J}_{m,v} = \{\ell \in [N_c] : c_\ell(m) \neq c_\ell(v)\} \tag{32}$$

be the set of codeword coordinates on which the two messages differ. For each routed bit $o = (i, q)$, define its pairwise margin contribution

$$X_o := \mathbf{1}\{z_o \neq c_{\ell_o}(v)\} - \mathbf{1}\{z_o \neq c_{\ell_o}(m)\}. \tag{33}$$

Then

$$M_{m,v} = \sum_o X_o. \tag{34}$$

Moreover, $X_o = 0$ whenever $\ell_o \notin \mathcal{J}_{m,v}$, and $X_o \in \{-1, +1\}$ whenever $\ell_o \in \mathcal{J}_{m,v}$.

Proof. Expand the two decoding metrics using Eq. (25) and subtract the candidate losses. For each routed bit o, the corresponding term in the difference is exactly X_o. If $\ell_o \notin \mathcal{J}_{m,v}$, then $c_{\ell_o}(m) = c_{\ell_o}(v)$, so the two indicator losses are equal and $X_o = 0$. If $\ell_o \in \mathcal{J}_{m,v}$, the candidate bits differ, so z_o agrees with exactly one of them, giving contribution $+1$ or -1. □

Only routed bits with $\ell_o \in \mathcal{J}_{m,v}$ can affect the pairwise margin. In the concentration bound below, $a_{m,v}$ denotes the number of biased routed bits among these routed bits, and $b_{m,v}$ denotes the number of zero-mean routed bits among these routed bits. Thus $a_{m,v}$ and $b_{m,v}$ are pair-specific counts; they do not count routed bits on coordinates where $c(m)$ and $c(v)$ agree.

E.2 PAIRWISE CONCENTRATION

Lemma 4 (Pairwise margin bound)**.** *Fix a true message $m \in \mathcal{M}$ and a deterministic competitor $v \in \mathcal{M} \setminus \{m\}$. Suppose that $a_{m,v}$ biased routed bits and $b_{m,v}$ zero-mean routed bits are routed to $\mathcal{J}_{m,v}$. A biased routed bit contributes $+1$ with probability $1-p$ and -1 with probability p, where p is the routed-bit flip rate induced by matched sampling, with $0 \le p \le 1/2$. A zero-mean routed bit contributes $+1$ and -1 with equal probability. If these routed-bit contributions are conditionally independent, then*

$$\mathbb{E}[M_{m,v}] = (1-2p)a_{m,v} \tag{35}$$

and, when $a_{m,v} + b_{m,v} > 0$,

$$\Pr(M_{m,v} \le 0) \le \exp\left(-\frac{(1-2p)^2 a_{m,v}^2}{2(a_{m,v}+b_{m,v})}\right). \tag{36}$$

If $a_{m,v} + b_{m,v} = 0$, then no routed bit separates this pair and the bound is understood as the trivial bound $\Pr(M_{m,v} \le 0) \le 1$.

Proof. By Lemma 3, routed bits outside $\mathcal{J}_{m,v}$ contribute zero to the margin. For a biased routed bit inside $\mathcal{J}_{m,v}$, $\mathbb{E}[X_o] = 1-2p$. For a zero-mean routed bit inside $\mathcal{J}_{m,v}$, $\mathbb{E}[X_o] = 0$. Therefore

$$\mathbb{E}[M_{m,v}] = (1-2p)a_{m,v}.$$

Each nonzero routed-bit contribution satisfies $X_o \in [-1,1]$. Hoeffding's inequality (Vershynin, 2018) gives

$$\Pr(M_{m,v} \le 0) \le \exp\left(-\frac{\mathbb{E}[M_{m,v}]^2}{2(a_{m,v}+b_{m,v})}\right),$$

which proves the result after substituting the mean. □

E.3 PROOF OF THEOREM 1

Theorem 1 restated. Let $\mathcal{M} = \{0,1\}^r$, and let

$$\mathcal{C} = \{c(m) : m \in \mathcal{M}\}.$$

Let A be the number of watermarked token positions whose seed-dependent watermark relation is preserved after perturbation. On these tokens, the sampled token label still follows the original target message label with average bit-flip rate p, so the corresponding routed-bit contributions to the decoding margin have mean $1-2p$. Since each token label contains k routed bits, these positions provide kA biased routed-bit observations before restricting to a pairwise codeword difference set. Let B be the number of token positions whose seed-dependent watermark relation is broken by perturbation, or that otherwise do not provide watermark information. On these tokens, the corresponding routed-bit contributions are treated as zero-mean, giving kB zero-mean routed-bit observations before the same restriction.

The idealized uniform-routing model assumes exactly balanced counts: for every competitor $v \ne m$, the pair-specific numbers of biased and zero-mean routed bits satisfy

$$a_{m,v} = \frac{d_H(c(m),c(v))}{N_c} kA, \qquad b_{m,v} = \frac{d_H(c(m),c(v))}{N_c} kB. \tag{37}$$

Define the effective routed-bit length

$$L_{\text{eff}} = k(1-2p)^2 \frac{A^2}{A+B}. \tag{38}$$

If the codeword set has minimum distance

$$d_{\min} = \min_{m \ne v} d_H(c(m),c(v)), \qquad \Delta_{\min} = \frac{d_{\min}}{N_c}, \tag{39}$$

then, under the noise model in Lemma 4, the exact decoding-metric minimizer

$$\hat{m}_{\text{exact}} = \arg\min_{v \in \mathcal{M}} D(u_{n:L}; v) \tag{40}$$

satisfies

$$\Pr(\hat{m}_{\text{exact}} \ne m) \le (2^r - 1)\exp\left(-\frac{1}{2}\Delta_{\min} L_{\text{eff}}\right). \tag{41}$$

Proof. Substituting Eq. (37) into Lemma 4 gives, for each $v \neq m$,

$$\Pr(M_{m,v} \leq 0) \leq \exp\left(-\frac{d_H(c(m), c(v))}{2N_c} L_{\text{eff}}\right).$$

If $\hat{m}_{\text{exact}} \neq m$, then there exists at least one $v \neq m$ such that $D(u_{n:L}; v) \leq D(u_{n:L}; m)$, equivalently $M_{m,v} \leq 0$. A union bound therefore gives

$$\Pr(\hat{m}_{\text{exact}} \neq m) \leq \sum_{v \neq m} \exp\left(-\frac{d_H(c(m), c(v))}{2N_c} L_{\text{eff}}\right).$$

Using $d_H(c(m), c(v)) \geq d_{\min}$ and $|\mathcal{M}| - 1 = 2^r - 1$ yields Eq. (41). □

The bound in Eq. (41) is the main-text recovery guarantee for exact metric decoding under the uniform-routing model. It shows how the four quantities in the analysis enter exact recovery: r controls the number of competing messages, $\Delta_{\min}$ controls codeword separation, p is the routed-bit flip rate induced by matched sampling on the preserved token positions, and A, B determine the effective routed-bit length through the numbers of preserved and uninformative token positions after perturbation. The uniform-routing model is the analysis condition corresponding to the approximately uniform spreading targeted by the keyed routes used in implementation. The implemented decoder is OSD, and all reported experiments evaluate this implemented routing-and-decoding pipeline directly.

Theory-to-practice validation. The theorem predicts that recovery is governed by the effective evidence length L_{eff}. The implemented pipeline realizes the analytical setting approximately: keyed pseudorandom hashes provide near-uniform routing, actual edits introduce correlated errors, and OSD-2 approximates exact metric decoding. We evaluate whether the dependence predicted by the theorem remains explanatory under these practical conditions.

For each condition g, we compute

$$\overline{L}_{\text{eff},g} = k(1 - 2\hat{p}_g)^2 \frac{A^2}{A + B}, \tag{42}$$

where $\hat{p}_g$ is estimated directly from the attack record without using decoding correctness. We regress the smoothed error rate $\widetilde{e}_g = (E_g + 0.5)/(n_g + 1)$ as

$$-\log \widetilde{e}_g = \alpha + \beta \overline{L}_{\text{eff},g} + \epsilon_g. \tag{43}$$

Table 9: Condition-level explanatory power under keyed routing, correlated edits, and OSD-2 decoding.

Predictor	Spearman with exact recovery	Linearized-error R^2
Nominal attack ratio q	-0.308	0.133
$A^2/(A+B)$	0.349	0.177
Full $\overline{L}_{\text{eff}}$	**0.976**	**0.983**

As shown in Table 9, the full predictor achieves a Spearman correlation of 0.976 and an R^2 of 0.983, substantially exceeding either nominal attack ratio or evidence quantity alone. Its fitted slope, $\hat{\beta} \approx 0.0152 > 0$, also matches the direction predicted by the theorem. These results show that the effective-evidence dependence derived by the theorem closely explains recovery in the implemented pipeline, even when its analytical conditions are realized only approximately.

F False Attribution

Both watermarked and no-watermark texts are passed through the same decoder. For detector observations u, let $\mathcal{M}_{\text{OSD}}(u)$ be the candidate messages enumerated by OSD. Define the decoded message and its normalized metric value as

$$\widehat{m}(u) = \arg \min_{m \in \mathcal{M}_{\text{OSD}}(u)} D(u; m), \qquad D^\star(u) = \frac{D\big(u; \widehat{m}(u)\big)}{k|u|}, \tag{44}$$

where D is the decoding metric in Eq. (1), and $|u|$ is the number of detector-readable token observations. A threshold τ is calibrated on no-watermark text for an FPR of 1%. Presence is then decided by

$$\widehat{W}(u) = \mathbf{1}\{D^\star(u) \le \tau\}\,. \tag{45}$$

Thus decoding supplies the presence statistic for either input type, while the threshold separates watermarked from no-watermark text.

G Limitation under Strong Paraphrasing

`UniCoMark` uses a token-level carrier whose extracted bits and routes depend on local token identities and prefixes. Strong semantic paraphrasing can replace most tokens and reorder the text, changing both the extracted token bits and the prefix-dependent routes; the original watermark evidence then disappears. This failure cannot be repaired by changing an existing decoder parameter, because the carrier itself is no longer preserved. The present method therefore does not claim robustness to strong paraphrasing. It nevertheless gives the best recovery among the evaluated baselines under the milder paraphrasing settings in Table 7; stronger settings substantially reduce recovery for all evaluated token-level methods.

H Detailed Experimental Settings

Generation settings. Open-ended generation uses temperature 1.0 and top-$k = 64$, with `min_new_tokens`=128 and `max_new_tokens`=512. We use the same LLM generation settings, prefix length 2, and prefix-repetition masking for all methods. Except for the generation-overhead benchmark, all experiments were run on a single NVIDIA A100 80GB GPU.

Baseline and method settings. All baselines use the authors' official implementations. Method-specific settings are $\gamma = 0.5$ and $\delta \in \{1, 2\}$ for MPAC (Yoo et al., 2023) and SegMark (Qu et al., 2025), 10 layers for BiMark (Feng et al., 2025). For `UniCoMark`, the trade-off sweep varies s in $P_s = (1-s)P_T + sP_{\mathrm{OT}}$, where P_{OT} is the OT-sampling distribution in Eq. (4). We set $k = 4$, use a random linear-code generator matrix G with codeword length $N_c = 2r$, and run order-2 OSD. The GPU OT implementation uses Sinkhorn iterations with at most 200 steps.

PPL evaluation. PPL is computed with a separate Qwen3-14B base model (Yang et al., 2025); for the main payload sweep, we report median PPL at 256 tokens.

I Additional Experiments

I.1 Generation overhead

We measure end-to-end throughput on an NVIDIA RTX 5080 for 128 generated tokens, with top-$k = 64$, temperature 1.0, one warmup, and three timed repeats. Table 10 reports the direct measurements.

Table 10: End-to-end generation throughput and OT slowdown (batch sizes 1, 4, and 8).

Model	Batch	No watermark tok/s	OT tok/s	Slowdown
Qwen3-4B-Base	1	60.31	56.36	6.55%
Qwen3-4B-Base	4	200.79	189.68	5.54%
Qwen3-4B-Base	8	374.75	350.42	6.49%
Llama-3.1-8B (4-bit)	1	71.25	67.73	4.93%
Llama-3.1-8B (4-bit)	4	84.94	82.08	3.36%
Llama-3.1-8B (4-bit)	8	165.45	160.60	2.94%

I.2 Impact of Parameters

Impact of parameter k. We vary k on Llama-3.1-8B/C4 and show results in Table 11. We see that accuracy remains between 90% and 94% for $k \geq 4$. We use $k = 4$.

Table 11: Sensitivity to k.

k	Message accuracy	Median PPL
2	73.25%	9.196
4	91.00%	10.259
6	90.00%	10.259
8	94.00%	11.092

Impact of codeword length N_c. We vary N_c for a 32-bit payload on 256-token Llama-3.1-8B/C4 generations. Results in Table 12 show the default $N_c = 2r = 64$ gives the highest accuracy.

Table 12: Sensitivity to N_c.

N_c	Message accuracy
48	95.49%
64	**96.08%**
96	95.69%

I.3 Full Raw Payload Tables

The following Tables 13-16 report the full fixed-length payload grid over target decoding lengths $\{64, 128, 256, 512\}$ and payloads $\{16, 32, 48, 64\}$. They keep only message accuracy and bit accuracy, reported in percent; the corresponding PPL values are collected separately in Table 17. No-watermark rows are omitted here because they do not carry decoding metrics. Bit accuracy around 50% is close to random guessing over bits.

I.4 PPL Reference Table

Table 17 collects the PPL values for all four model–dataset combinations and all methods. These PPL values follow the main-text convention and are measured under the 32-bit setting.

I.5 Full Basic Attack Tables

Tables in this section report message accuracy in percent under token- and sentence-level edit attacks on the 32-bit source runs generated with requested maximum length 512 and minimum length 256. Higher values are better. Table 18 shows the results under token-level edit attacks, while Table 19 shows the results under sentence-level edit attacks.

Table 13: Full raw payload grid for Llama-3.1-8B on ASAP-essay. Each token block reports results at the indicated decoding length. Entries are message accuracy / bit accuracy in percent.

Token	Method	16 bit		32 bit		48 bit		64 bit	
		Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)
64	MPAC $\delta = 1$	1.4	74.5	0.0	67.0	0.0	63.0	0.0	60.9
	MPAC $\delta = 2$	17.0	88.8	0.0	79.4	0.0	74.0	0.0	69.6
	BiMark	34.2	92.6	0.4	82.3	0.0	73.9	0.0	67.9
	SegMark $\delta = 1$	0.4	57.7	0.0	50.9	0.0	49.6	0.0	49.7
	SegMark $\delta = 2$	17.6	71.3	0.4	57.8	0.0	52.0	0.0	49.7
	UniCoMark $s = 0.2$	0.0	57.6	0.0	56.3	0.0	54.8	0.0	54.3
	UniCoMark $s = 0.4$	7.0	68.6	0.0	62.2	0.0	60.4	0.0	58.5
	UniCoMark $s = 0.6$	41.4	82.2	2.0	68.9	0.0	65.6	0.0	63.2
	UniCoMark $s = 0.8$	78.9	94.4	19.1	77.0	1.6	70.1	0.0	67.1
	UniCoMark $s = 1.0$	93.6	98.2	47.7	86.3	9.6	74.6	0.8	70.1
128	MPAC $\delta = 1$	3.7	81.5	0.0	73.9	0.0	68.9	0.0	66.1
	MPAC $\delta = 2$	48.8	95.5	2.5	88.5	0.0	83.0	0.0	78.9
	BiMark	72.3	97.6	9.8	91.6	0.4	84.0	0.0	77.6
	SegMark $\delta = 1$	4.7	64.6	0.0	54.6	0.0	50.7	0.0	49.8
	SegMark $\delta = 2$	54.7	84.0	7.4	69.2	0.0	57.8	0.0	52.6
	UniCoMark $s = 0.2$	1.2	62.5	0.0	58.9	0.0	56.8	0.0	56.5
	UniCoMark $s = 0.4$	35.0	79.5	1.6	68.1	0.0	64.8	0.0	62.0
	UniCoMark $s = 0.6$	82.0	94.9	24.4	78.7	2.1	70.5	0.0	66.4
	UniCoMark $s = 0.8$	97.5	99.2	76.0	93.4	28.9	80.1	6.1	72.5
	UniCoMark $s = 1.0$	98.4	99.5	94.1	98.4	69.3	91.6	29.5	80.6
256	MPAC $\delta = 1$	22.3	90.0	0.6	81.2	0.0	77.9	0.0	72.6
	MPAC $\delta = 2$	89.1	99.1	29.7	95.8	5.1	93.3	0.3	87.8
	BiMark	93.0	99.3	48.4	96.9	13.7	94.4	0.5	87.4
	SegMark $\delta = 1$	16.8	72.8	1.2	60.2	0.0	52.6	0.0	50.1
	SegMark $\delta = 2$	82.4	94.0	56.5	89.0	18.0	73.9	1.2	64.5
	UniCoMark $s = 0.2$	6.2	67.0	0.2	61.8	0.0	60.4	0.0	58.8
	UniCoMark $s = 0.4$	74.0	92.5	16.8	75.6	2.3	70.4	0.0	65.8
	UniCoMark $s = 0.6$	95.9	98.7	76.8	93.6	50.1	86.6	6.8	73.5
	UniCoMark $s = 0.8$	98.8	99.6	94.9	98.5	88.6	96.8	57.4	88.6
	UniCoMark $s = 1.0$	98.4	99.5	96.3	98.9	94.1	98.0	80.7	94.3
512	MPAC $\delta = 1$	49.8	94.7	6.7	88.3	0.0	83.3	0.0	79.8
	MPAC $\delta = 2$	98.2	99.6	73.5	98.6	35.6	97.2	9.1	95.1
	BiMark	95.5	99.4	80.3	98.4	50.2	96.8	18.2	95.3
	SegMark $\delta = 1$	45.6	83.9	8.1	74.2	0.0	68.4	0.0	65.7
	SegMark $\delta = 2$	88.7	95.6	83.3	94.8	54.8	91.6	32.9	88.7
	UniCoMark $s = 0.2$	18.3	72.8	0.8	65.0	0.0	61.8	0.0	61.3
	UniCoMark $s = 0.4$	88.0	96.1	59.3	87.8	22.4	77.8	4.1	71.4
	UniCoMark $s = 0.6$	96.4	99.5	90.4	97.0	81.6	94.4	62.1	89.7
	UniCoMark $s = 0.8$	98.9	99.6	95.6	98.9	92.2	97.7	86.8	96.3
	UniCoMark $s = 1.0$	98.9	99.6	96.8	98.8	94.8	98.5	91.9	97.3

Table 14: Full raw payload grid for Llama-3.1-8B on C4 RealNewsLike. Each token block reports results at the indicated decoding length. Entries are message accuracy / bit accuracy in percent.

Token	Method	16 bit		32 bit		48 bit		64 bit	
		Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)
64	MPAC $\delta = 1$	0.8	72.7	0.0	65.4	0.0	62.1	0.0	59.7
	MPAC $\delta = 2$	9.0	85.3	0.0	75.6	0.0	71.2	0.0	67.8
	BiMark	19.5	89.9	0.0	79.7	0.0	73.3	0.0	67.9
	SegMark $\delta = 1$	0.8	58.9	0.0	51.2	0.0	49.8	0.0	50.7
	SegMark $\delta = 2$	14.1	72.5	0.0	57.3	0.0	51.4	0.0	50.4
	UniCoMark $s = 0.2$	0.0	58.3	0.0	56.0	0.0	54.3	0.0	54.3
	UniCoMark $s = 0.4$	6.2	65.9	0.0	61.1	0.0	59.4	0.0	58.8
	UniCoMark $s = 0.6$	30.1	77.9	1.4	67.0	0.0	64.2	0.0	62.7
	UniCoMark $s = 0.8$	64.1	88.9	10.9	73.4	0.2	68.1	0.0	65.4
	UniCoMark $s = 1.0$	77.5	93.0	28.3	79.4	4.5	71.4	0.0	68.5
128	MPAC $\delta = 1$	3.1	80.2	0.0	72.0	0.0	67.4	0.0	64.6
	MPAC $\delta = 2$	38.9	93.1	0.6	84.9	0.0	80.6	0.0	76.5
	BiMark	62.9	96.7	4.7	89.7	0.0	83.4	0.0	77.8
	SegMark $\delta = 1$	4.3	66.0	0.0	53.2	0.0	50.4	0.0	50.4
	SegMark $\delta = 2$	51.9	86.1	10.5	69.4	0.0	57.3	0.0	51.6
	UniCoMark $s = 0.2$	0.8	61.5	0.0	58.0	0.0	56.7	0.0	55.7
	UniCoMark $s = 0.4$	22.1	74.4	0.2	65.0	0.0	62.4	0.0	61.7
	UniCoMark $s = 0.6$	71.7	91.8	13.7	74.7	1.6	69.6	0.0	66.2
	UniCoMark $s = 0.8$	91.8	97.4	60.7	88.9	18.6	77.4	2.1	70.9
	UniCoMark $s = 1.0$	95.1	98.3	81.1	94.3	46.9	85.5	15.8	76.5
256	MPAC $\delta = 1$	13.3	87.5	0.2	79.3	0.0	76.4	0.0	71.7
	MPAC $\delta = 2$	77.9	98.1	13.7	93.0	1.8	90.9	0.0	85.8
	BiMark	88.7	99.0	35.0	96.0	9.0	93.9	0.0	88.5
	SegMark $\delta = 1$	16.4	75.0	0.4	59.3	0.0	51.5	0.0	50.3
	SegMark $\delta = 2$	82.4	94.4	60.2	90.3	16.4	75.9	1.9	62.1
	UniCoMark $s = 0.2$	4.9	65.8	0.0	60.8	0.0	59.4	0.0	58.3
	UniCoMark $s = 0.4$	62.7	88.9	9.2	72.1	0.9	69.2	0.0	66.6
	UniCoMark $s = 0.6$	93.8	98.0	67.0	90.8	41.0	84.4	3.9	72.3
	UniCoMark $s = 0.8$	96.7	98.9	90.4	97.1	81.9	95.0	45.6	85.6
	UniCoMark $s = 1.0$	97.9	99.4	94.9	98.5	92.4	97.7	78.5	94.2
512	MPAC $\delta = 1$	50.2	94.9	3.4	88.1	0.0	83.2	0.0	79.8
	MPAC $\delta = 2$	96.7	99.6	58.7	98.0	20.7	95.9	2.9	93.4
	BiMark	98.2	99.8	82.6	99.0	49.5	97.8	13.9	95.9
	SegMark $\delta = 1$	58.3	88.4	4.4	74.3	0.0	70.4	0.0	65.5
	SegMark $\delta = 2$	91.0	96.1	82.7	94.8	59.9	92.1	30.9	89.1
	UniCoMark $s = 0.2$	17.1	72.7	0.0	64.4	0.0	62.8	0.0	61.7
	UniCoMark $s = 0.4$	90.7	97.3	55.4	87.8	15.7	76.2	1.8	71.5
	UniCoMark $s = 0.6$	97.6	99.2	93.8	98.2	83.2	95.4	57.4	88.7
	UniCoMark $s = 0.8$	98.0	99.3	97.3	99.0	95.2	98.5	90.1	97.3
	UniCoMark $s = 1.0$	98.6	99.4	97.6	99.2	96.6	98.9	94.2	98.3

Table 15: Full raw payload grid for Gemma-2-9B on ASAP-essay. Each token block reports results at the indicated decoding length. Entries are message accuracy / bit accuracy in percent.

Token	Method	16 bit		32 bit		48 bit		64 bit	
		Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)
64	MPAC $\delta = 1$	1.4	74.5	0.0	67.3	0.0	63.9	0.0	61.6
	MPAC $\delta = 2$	14.6	88.9	0.0	79.7	0.0	74.3	0.0	70.3
	BiMark	32.0	92.9	0.2	82.5	0.0	74.2	0.0	68.3
	SegMark $\delta = 1$	0.7	59.3	0.0	52.5	0.0	51.7	0.0	50.4
	SegMark $\delta = 2$	17.3	70.9	0.6	56.5	0.0	50.9	0.0	49.8
	UniCoMark $s = 0.2$	0.4	59.2	0.0	56.3	0.0	55.6	0.0	54.6
	UniCoMark $s = 0.4$	9.0	68.6	0.0	63.0	0.0	60.4	0.0	59.2
	UniCoMark $s = 0.6$	39.8	81.9	1.8	68.8	0.0	65.5	0.0	63.3
	UniCoMark $s = 0.8$	84.6	95.3	20.1	77.3	1.2	69.6	0.0	67.2
	UniCoMark $s = 1.0$	94.5	98.7	48.0	86.7	9.6	74.5	0.0	70.1
128	MPAC $\delta = 1$	3.7	82.4	0.0	74.4	0.0	70.1	0.0	66.9
	MPAC $\delta = 2$	57.0	96.5	2.9	89.3	0.0	83.9	0.0	80.1
	BiMark	75.4	98.2	9.8	92.1	0.2	84.4	0.0	77.9
	SegMark $\delta = 1$	4.5	63.8	0.0	55.1	0.0	51.2	0.0	48.9
	SegMark $\delta = 2$	62.7	90.0	9.8	73.2	0.1	56.9	0.0	53.3
	UniCoMark $s = 0.2$	1.0	62.8	0.0	59.3	0.0	57.1	0.0	56.2
	UniCoMark $s = 0.4$	31.6	79.5	1.0	67.5	0.0	64.4	0.0	62.1
	UniCoMark $s = 0.6$	85.0	95.7	26.8	79.5	1.4	70.6	0.0	67.3
	UniCoMark $s = 0.8$	98.4	99.6	78.5	94.7	28.7	81.3	3.9	72.7
	UniCoMark $s = 1.0$	99.4	99.8	96.1	99.1	72.3	92.9	26.0	81.1
256	MPAC $\delta = 1$	20.9	90.6	0.4	82.8	0.0	79.1	0.0	73.6
	MPAC $\delta = 2$	89.5	99.3	30.3	96.2	4.4	93.7	0.3	89.0
	BiMark	94.5	99.6	50.8	97.4	16.8	95.1	0.3	87.8
	SegMark $\delta = 1$	19.3	78.5	1.7	58.8	0.0	52.9	0.0	50.7
	SegMark $\delta = 2$	89.7	96.1	59.2	89.8	21.4	76.2	2.0	65.3
	UniCoMark $s = 0.2$	6.2	67.0	0.0	61.9	0.0	60.3	0.0	58.3
	UniCoMark $s = 0.4$	73.8	92.3	11.7	74.3	1.8	70.1	0.0	66.8
	UniCoMark $s = 0.6$	98.8	99.7	81.6	95.4	54.7	88.4	7.3	74.3
	UniCoMark $s = 0.8$	100.0	100.0	98.2	99.5	92.4	98.1	58.5	89.4
	UniCoMark $s = 1.0$	100.0	100.0	98.8	99.7	97.6	99.4	91.2	97.5
512	MPAC $\delta = 1$	55.7	95.8	5.4	90.1	0.9	84.5	0.0	81.2
	MPAC $\delta = 2$	98.7	99.9	78.6	99.2	34.5	97.8	9.2	96.0
	BiMark	95.4	99.7	84.4	98.8	50.4	97.6	21.8	95.7
	SegMark $\delta = 1$	51.6	86.5	9.7	76.5	0.0	68.2	0.0	64.8
	SegMark $\delta = 2$	91.8	96.7	86.7	95.3	58.3	93.1	35.8	90.5
	UniCoMark $s = 0.2$	21.6	74.9	0.8	65.2	0.0	62.3	0.0	61.0
	UniCoMark $s = 0.4$	94.9	98.2	55.1	87.7	17.3	76.9	1.9	71.3
	UniCoMark $s = 0.6$	98.7	99.6	96.0	99.0	88.4	96.8	68.8	91.9
	UniCoMark $s = 0.8$	100.0	100.0	98.4	99.6	96.3	98.8	93.9	98.3
	UniCoMark $s = 1.0$	100.0	100.0	99.2	99.7	98.3	99.6	97.5	99.3

Table 16: Full raw payload grid for Gemma-2-9B on C4 RealNewsLike. Each token block reports results at the indicated decoding length. Entries are message accuracy / bit accuracy in percent.

Token	Method	16 bit		32 bit		48 bit		64 bit	
		Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)	Msg. Acc. (%)	Bit Acc. (%)
64	MPAC $\delta = 1$	0.4	72.1	0.0	65.6	0.0	62.7	0.0	60.4
	MPAC $\delta = 2$	11.1	86.5	0.0	76.8	0.0	72.0	0.0	68.7
	BiMark	22.5	89.9	0.2	79.9	0.0	73.1	0.0	68.0
	SegMark $\delta = 1$	0.7	59.2	0.0	52.2	0.0	49.3	0.0	49.5
	SegMark $\delta = 2$	13.5	71.6	0.0	56.2	0.0	52.1	0.0	50.1
	UniCoMark $s = 0.2$	0.4	58.1	0.0	55.7	0.0	54.8	0.0	54.4
	UniCoMark $s = 0.4$	5.1	67.1	0.0	62.3	0.0	59.7	0.0	58.7
	UniCoMark $s = 0.6$	27.3	77.5	0.8	66.3	0.0	64.2	0.0	62.1
	UniCoMark $s = 0.8$	63.1	89.2	8.8	72.8	0.2	68.1	0.0	66.5
	UniCoMark $s = 1.0$	84.6	95.5	29.9	80.6	5.7	72.2	0.4	69.2
128	MPAC $\delta = 1$	3.7	80.8	0.0	73.2	0.0	68.6	0.0	65.6
	MPAC $\delta = 2$	39.8	94.1	2.7	86.6	0.0	81.2	0.0	77.4
	BiMark	64.1	96.9	4.7	89.8	0.0	83.6	0.0	77.9
	SegMark $\delta = 1$	5.0	67.2	0.0	53.9	0.0	51.2	0.0	49.9
	SegMark $\delta = 2$	53.8	87.8	10.7	70.5	0.0	57.2	0.0	51.1
	UniCoMark $s = 0.2$	1.0	61.5	0.0	58.6	0.0	56.4	0.0	56.0
	UniCoMark $s = 0.4$	23.2	75.8	0.2	65.4	0.0	63.0	0.0	60.8
	UniCoMark $s = 0.6$	72.3	92.1	13.7	74.4	0.6	68.6	0.0	65.6
	UniCoMark $s = 0.8$	94.9	98.3	57.0	88.5	15.0	76.7	2.0	71.5
	UniCoMark $s = 1.0$	98.4	99.5	85.9	96.4	49.4	86.9	12.5	76.8
256	MPAC $\delta = 1$	18.0	89.1	0.0	80.5	0.0	77.2	0.0	72.3
	MPAC $\delta = 2$	80.5	98.6	19.7	94.2	3.6	91.5	0.0	86.6
	BiMark	91.6	99.4	36.1	96.4	9.5	94.2	0.0	88.6
	SegMark $\delta = 1$	17.1	76.7	0.8	65.1	0.0	52.7	0.0	49.4
	SegMark $\delta = 2$	85.7	94.9	59.9	91.2	17.7	77.4	2.5	64.5
	UniCoMark $s = 0.2$	3.9	66.9	0.2	61.7	0.0	60.3	0.0	57.9
	UniCoMark $s = 0.4$	61.1	88.1	7.4	72.3	0.0	68.6	0.0	65.4
	UniCoMark $s = 0.6$	96.3	99.0	67.4	91.1	36.4	83.0	2.8	72.2
	UniCoMark $s = 0.8$	99.6	99.9	93.8	98.4	87.0	96.7	46.9	86.6
	UniCoMark $s = 1.0$	100.0	100.0	99.0	99.7	97.2	99.2	80.5	95.1
512	MPAC $\delta = 1$	51.3	95.6	2.3	88.1	0.0	83.6	0.0	80.0
	MPAC $\delta = 2$	97.7	99.8	72.0	98.8	24.3	96.7	4.7	94.3
	BiMark	99.7	100.0	82.6	99.4	45.5	98.0	14.4	96.3
	SegMark $\delta = 1$	57.5	86.5	4.3	72.8	0.0	69.9	0.0	65.7
	SegMark $\delta = 2$	95.8	98.5	88.9	96.3	67.8	94.3	35.7	91.5
	UniCoMark $s = 0.2$	19.5	75.5	0.4	65.1	0.0	63.4	0.0	61.5
	UniCoMark $s = 0.4$	93.3	98.1	58.3	88.9	15.2	76.5	1.1	71.4
	UniCoMark $s = 0.6$	99.3	99.8	98.3	99.5	86.8	96.7	56.4	89.1
	UniCoMark $s = 0.8$	100.0	100.0	100.0	100.0	98.2	99.6	96.5	99.1
	UniCoMark $s = 1.0$	100.0	100.0	100.0	100.0	100.0	100.0	99.0	99.8

Table 17: PPL reference table over fixed decoding lengths $\{64, 128, 256, 512\}$.

Model	Dataset	Method	64	128	256	512
Llama-3.1-8B	ASAP-essay	No watermark	10.42	9.79	9.05	7.87
		MPAC $\delta = 1$	12.98	10.92	10.10	9.34
		MPAC $\delta = 2$	15.64	14.46	13.80	12.56
		BiMark	13.80	12.97	11.99	10.58
		SegMark $\delta = 1$	11.84	10.26	9.85	9.03
		SegMark $\delta = 2$	14.59	13.47	12.92	11.71
		UniCoMark $s = 0.2$	10.58	9.79	9.20	7.86
		UniCoMark $s = 0.4$	11.09	10.26	9.49	8.37
		UniCoMark $s = 0.6$	11.62	10.75	10.10	8.77
		UniCoMark $s = 0.8$	12.27	11.44	10.58	9.34
		UniCoMark $s = 1.0$	12.96	11.99	11.44	10.09
	C4 RealNewsLike	No watermark	9.05	8.77	8.24	7.87
		MPAC $\delta = 1$	10.42	9.63	9.20	8.77
		MPAC $\delta = 2$	13.44	12.37	11.62	11.26
		BiMark	11.35	10.92	10.58	10.10
		SegMark $\delta = 1$	10.26	9.49	8.91	8.25
		SegMark $\delta = 2$	13.01	12.18	11.22	10.57
		UniCoMark $s = 0.2$	9.05	8.63	8.50	7.86
		UniCoMark $s = 0.4$	9.27	8.91	8.77	8.31
		UniCoMark $s = 0.6$	10.10	9.71	9.20	8.91
		UniCoMark $s = 0.8$	10.42	10.10	9.64	9.12
		UniCoMark $s = 1.0$	10.92	10.58	10.10	9.86
Gemma-2-9B	ASAP-essay	No watermark	10.92	10.26	9.49	8.50
		MPAC $\delta = 1$	11.93	11.54	10.58	9.63
		MPAC $\delta = 2$	15.04	14.46	13.80	12.76
		BiMark	14.47	13.38	12.57	11.27
		SegMark $\delta = 1$	12.01	11.48	9.98	8.63
		SegMark $\delta = 2$	14.32	13.17	12.25	10.98
		UniCoMark $s = 0.2$	10.92	10.26	9.64	8.64
		UniCoMark $s = 0.4$	11.01	10.18	9.94	8.91
		UniCoMark $s = 0.6$	11.62	11.44	10.26	9.19
		UniCoMark $s = 0.8$	12.18	11.62	10.75	9.94
		UniCoMark $s = 1.0$	12.67	12.18	11.27	10.42
	C4 RealNewsLike	No watermark	9.56	9.05	8.64	8.31
		MPAC $\delta = 1$	10.10	9.79	9.49	8.91
		MPAC $\delta = 2$	13.59	12.65	11.62	11.09
		BiMark	11.81	12.00	10.92	10.42
		SegMark $\delta = 1$	9.72	9.43	9.03	8.75
		SegMark $\delta = 2$	12.52	12.11	11.47	10.58
		UniCoMark $s = 0.2$	9.49	8.91	8.77	8.24
		UniCoMark $s = 0.4$	9.71	9.12	8.91	8.24
		UniCoMark $s = 0.6$	9.48	9.45	9.20	8.74
		UniCoMark $s = 0.8$	10.58	9.94	9.79	9.27
		UniCoMark $s = 1.0$	10.92	10.42	9.94	9.49

Table 18: Token-level edit attacks on the 32-bit source runs generated with requested maximum length 512 and minimum length 256. Left block: Llama-3.1-8B. Right block: Gemma-2-9B. Values are message accuracy in percent, averaged over ASAP-essay and C4 RealNewsLike for each model.

Attack	Method	Llama-3.1-8B							Gemma-2-9B						
		Message accuracy (%)							Message accuracy (%)						
		5%	10%	15%	20%	25%	30%	40%	5%	10%	15%	20%	25%	30%	40%
Insert	MPAC $\delta = 1$	1.1	0.2	0.0	0.0	0.0	0.0	0.0	0.6	0.1	0.0	0.0	0.0	0.0	0.0
	MPAC $\delta = 2$	34.0	16.0	7.2	2.2	1.3	0.3	0.0	39.5	21.7	9.2	3.5	1.1	0.4	0.2
	BiMark	47.5	32.8	21.7	11.2	6.1	2.0	0.3	43.9	31.8	20.1	10.5	5.0	1.6	0.3
	SegMark $\delta = 1$	1.2	0.3	0.0	0.0	0.0	0.0	0.0	1.4	0.0	0.0	0.0	0.0	0.0	0.0
	SegMark $\delta = 2$	66.0	44.3	25.4	11.5	5.6	1.7	0.0	68.2	48.3	24.6	10.7	4.9	1.3	0.0
	UniCoMark $s = 0.2$	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.4$	23.1	9.4	4.5	1.3	0.2	0.0	0.0	19.6	6.8	2.0	0.6	0.3	0.0	0.0
	UniCoMark $s = 0.6$	78.2	60.8	41.1	23.7	10.4	4.9	0.5	81.2	62.8	41.4	21.8	8.7	3.4	0.2
	UniCoMark $s = 0.8$	91.7	86.1	78.4	63.1	45.8	27.2	4.3	95.6	91.6	81.2	67.4	46.2	24.8	3.6
	UniCoMark $s = 1.0$	94.9	93.2	89.8	82.1	71.7	53.4	19.3	98.8	97.4	93.7	87.8	73.7	53.0	16.0
Delete	MPAC $\delta = 1$	0.7	0.4	0.0	0.0	0.0	0.0	0.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0
	MPAC $\delta = 2$	32.7	14.4	5.1	2.0	0.2	0.1	0.0	38.9	18.8	5.6	1.9	0.1	0.2	0.0
	BiMark	46.7	28.8	18.1	8.3	4.2	0.9	0.0	46.9	30.1	16.3	6.9	2.8	0.4	0.0
	SegMark $\delta = 1$	0.9	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
	SegMark $\delta = 2$	65.9	36.5	21.3	11.0	5.9	1.6	0.1	65.6	40.9	20.7	9.9	4.7	0.7	0.0
	UniCoMark $s = 0.2$	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.4$	20.3	7.9	2.1	0.5	0.0	0.0	0.0	16.9	6.3	1.8	0.6	0.1	0.0	0.0
	UniCoMark $s = 0.6$	77.1	55.4	30.5	11.9	3.7	0.7	0.0	78.7	54.4	29.0	12.7	3.5	0.6	0.1
	UniCoMark $s = 0.8$	92.0	84.0	70.8	49.4	25.3	9.2	0.3	95.5	89.5	72.0	47.9	23.3	5.6	0.2
	UniCoMark $s = 1.0$	95.3	93.5	86.4	75.2	53.1	27.0	1.3	98.5	96.4	90.4	75.9	50.9	25.3	1.0
Replace	MPAC $\delta = 1$	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.7	0.0	0.0	0.0	0.0	0.0	0.0
	MPAC $\delta = 2$	27.4	10.0	2.0	0.4	0.0	0.0	0.0	33.2	12.5	2.7	0.3	0.1	0.1	0.0
	BiMark	45.3	25.4	12.6	3.4	0.8	0.4	0.0	41.8	21.8	10.4	2.6	0.5	0.2	0.0
	SegMark $\delta = 1$	1.4	0.0	0.0	0.0	0.0	0.0	0.0	0.9	0.0	0.0	0.0	0.0	0.0	0.0
	SegMark $\delta = 2$	63.9	27.8	10.9	2.7	0.0	0.0	0.0	62.2	31.1	10.2	1.9	0.1	0.0	0.0
	UniCoMark $s = 0.2$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.1	0.0	0.0	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.4$	18.9	4.8	0.9	0.1	0.0	0.0	0.0	15.6	3.9	0.7	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.6$	73.9	48.0	20.7	5.5	1.0	0.0	0.0	75.5	49.4	18.9	4.1	1.3	0.1	0.0
	UniCoMark $s = 0.8$	91.2	80.6	63.1	33.4	10.4	1.7	0.0	94.4	84.8	60.5	29.7	8.7	2.0	0.0
	UniCoMark $s = 1.0$	94.7	91.4	81.8	61.3	28.9	8.3	0.1	98.8	95.2	85.2	60.9	28.8	6.2	0.1
Mixed	MPAC $\delta = 1$	0.5	0.1	0.0	0.0	0.0	0.0	0.0	0.7	0.1	0.0	0.0	0.0	0.0	0.0
	MPAC $\delta = 2$	29.5	13.5	6.7	1.4	0.3	0.1	0.0	36.7	17.9	6.7	1.9	0.5	0.3	0.0
	BiMark	46.2	31.3	21.7	8.9	3.8	1.4	0.2	46.9	31.3	17.7	9.9	3.5	2.1	0.0
	SegMark $\delta = 1$	1.2	0.2	0.0	0.0	0.0	0.0	0.0	1.0	0.3	0.0	0.0	0.0	0.0	0.0
	SegMark $\delta = 2$	64.5	41.6	23.7	10.9	4.4	1.6	0.3	65.1	39.5	21.9	11.2	4.6	1.8	0.0
	UniCoMark $s = 0.2$	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.4$	20.8	9.6	2.3	1.1	0.0	0.0	0.0	17.9	6.3	1.5	0.3	0.1	0.0	0.0
	UniCoMark $s = 0.6$	75.3	56.5	36.6	16.2	6.4	3.2	0.2	80.0	55.3	34.1	16.4	4.9	1.5	0.2
	UniCoMark $s = 0.8$	90.6	84.3	73.1	57.0	35.4	17.0	2.2	96.1	89.1	75.4	56.9	31.5	14.6	1.9
	UniCoMark $s = 1.0$	95.9	93.2	88.0	78.6	61.7	40.8	9.8	98.8	97.0	91.2	81.5	63.2	38.4	8.2

Table 19: Sentence-level edit attacks on the 32-bit source runs generated with requested maximum length 512 and minimum length 256. Left block: Llama-3.1-8B. Right block: Gemma-2-9B. Values are message accuracy in percent, averaged over ASAP-essay and C4 RealNewsLike for each model.

Attack	Method	Llama-3.1-8B					Gemma-2-9B				
		Message accuracy (%)					Message accuracy (%)				
		10%	20%	30%	50%	75%	10%	20%	30%	50%	75%
Insert	MPAC $\delta = 1$	0.3	1.0	0.7	0.5	0.2	1.2	0.2	0.1	0.1	0.0
	MPAC $\delta = 2$	36.7	30.8	23.8	17.8	10.4	39.1	31.4	26.0	18.1	11.0
	BiMark	45.9	40.3	34.9	27.7	20.9	47.2	43.6	36.3	28.1	22.7
	SegMark $\delta = 1$	1.0	0.7	0.5	0.2	0.0	2.0	1.1	0.2	0.0	0.0
	SegMark $\delta = 2$	64.8	48.9	40.7	32.5	24.2	69.3	61.5	49.4	37.9	30.1
	UniCoMark $s = 0.2$	0.3	0.1	0.2	0.1	0.0	0.0	0.1	0.0	0.0	0.0
	UniCoMark $s = 0.4$	22.2	18.3	13.2	7.7	3.3	19.0	12.9	10.2	6.8	3.7
	UniCoMark $s = 0.6$	76.0	69.1	62.5	50.5	38.3	75.3	70.6	66.8	53.2	35.9
	UniCoMark $s = 0.8$	88.7	87.5	84.9	80.1	70.7	92.6	91.1	88.6	82.5	76.1
	UniCoMark $s = 1.0$	94.1	92.6	92.1	89.6	84.4	97.4	96.2	96.4	93.2	90.1
Delete	MPAC $\delta = 1$	0.4	0.2	0.1	0.1	0.0	0.5	0.3	0.2	0.0	0.0
	MPAC $\delta = 2$	34.6	24.5	15.5	4.4	0.2	36.2	24.4	16.0	4.4	0.4
	BiMark	43.8	38.5	31.3	15.0	3.1	46.4	37.6	31.2	16.8	4.3
	SegMark $\delta = 1$	0.4	0.4	0.3	0.2	0.1	0.6	0.2	0.2	0.0	0.0
	SegMark $\delta = 2$	61.7	46.5	35.2	18.9	3.7	66.0	48.3	35.6	19.0	2.7
	UniCoMark $s = 0.2$	0.1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.4$	20.1	12.0	5.3	1.1	0.0	16.7	9.4	5.1	1.7	0.0
	UniCoMark $s = 0.6$	70.9	61.1	48.4	22.4	3.0	73.3	62.3	51.1	23.1	1.7
	UniCoMark $s = 0.8$	88.8	85.4	78.7	60.4	11.6	92.2	89.4	82.1	60.4	12.2
	UniCoMark $s = 1.0$	94.3	92.8	89.7	79.0	30.3	96.7	95.7	93.4	83.5	31.9
Replace	MPAC $\delta = 1$	0.5	0.2	0.1	0.0	0.0	0.6	0.2	0.0	0.0	0.0
	MPAC $\delta = 2$	26.4	14.2	5.6	1.2	0.0	25.5	11.6	5.8	0.8	0.0
	BiMark	38.1	26.2	13.6	2.1	0.2	38.0	26.1	15.0	3.6	0.0
	SegMark $\delta = 1$	1.1	0.6	0.4	0.1	0.0	0.9	0.5	0.0	0.0	0.0
	SegMark $\delta = 2$	57.5	30.3	15.3	2.4	0.0	59.2	30.4	18.4	2.1	0.0
	UniCoMark $s = 0.2$	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	UniCoMark $s = 0.4$	14.7	5.3	1.8	0.1	0.0	10.8	4.3	0.9	0.0	0.0
	UniCoMark $s = 0.6$	63.1	44.7	23.0	3.4	0.0	65.6	43.6	23.4	1.9	0.0
	UniCoMark $s = 0.8$	84.9	75.7	58.9	15.5	0.2	89.0	80.4	61.0	18.4	0.4
	UniCoMark $s = 1.0$	92.2	87.1	76.7	37.1	0.7	95.4	91.8	82.1	38.7	1.2
Mixed	MPAC $\delta = 1$	0.6	0.1	0.1	0.0	0.0	0.4	0.1	0.0	0.2	0.0
	MPAC $\delta = 2$	32.1	23.6	15.6	6.7	1.5	32.5	23.6	15.8	6.9	1.7
	BiMark	41.5	36.4	29.6	16.4	7.8	43.0	36.4	31.1	17.4	8.6
	SegMark $\delta = 1$	0.5	0.2	0.3	0.0	0.0	0.9	0.5	0.2	0.1	0.0
	SegMark $\delta = 2$	62.8	47.3	35.6	19.2	8.5	66.2	47.8	35.1	18.1	8.5
	UniCoMark $s = 0.2$	0.1	0.0	0.0	0.0	0.0	0.1	0.0	0.1	0.0	0.0
	UniCoMark $s = 0.4$	19.0	11.1	6.6	2.4	0.6	15.4	8.9	4.4	2.4	0.7
	UniCoMark $s = 0.6$	70.6	60.0	47.5	29.7	11.8	72.8	61.4	50.5	30.3	10.8
	UniCoMark $s = 0.8$	88.0	84.0	79.5	63.5	39.2	91.6	87.4	81.8	66.1	42.2
	UniCoMark $s = 1.0$	93.2	91.2	87.7	80.6	62.6	97.2	94.8	93.4	85.2	65.1

J WATERMARKED GENERATION EXAMPLE UNDER LLAMA-3.1-8B

Prompt(C4 RealNewsLike)

In all the hubbub over what shape the new Peace Bridge should take, the Niagara Falls Bridge Commission thinks there's an important fact being overlooked. The Peace Bridge is only one of four bridges in the region carrying traffic between the United States and Canada, and the commission runs the other three.

No-Watermark Generation

So how does a bridge commission handle a situation where one bridge is getting a $450 million overhaul and another bridge is in dire need of repair? Let me tell you. This is actually a pretty good question. As the Buffalo News points out, the commission has no money to spend on anything, even if it wanted to. Every dollar the commission makes goes toward maintenance on the four bridges, and in recent years, the bridges have needed a lot of maintenance, mostly in the form of rust removal and repair. So the bridges are in dire need of repair, and the commission has no money to do it. What can they do? Why, just cut people's salary, of course. Now, most of us wouldn't call it a cut, but a reduction in "non-essential overtime and other employment expenses." But really, that's a cut, right? The other bridges in the region are carrying a heavier load of traffic these days than they used to, and the Peace Bridge is the only one with the money to undergo expansion. What the commission has proposed is to put half of the money from the expanded Peace Bridge into the coffers, and to use that money to improve the other three bridges. If everything is approved by the governments, the commission hopes to build…

Watermarked Generation

But all four are in bad shape, and there's no good option for keeping them open and replacing the old structures. In a recent presentation to Mayor Paul Dyster and the Buffalo Common Council, Mark Reesor, the commission's director of facilities, engineering and river projects, outlined a nearly billion-dollar project of massive construction and environmental upgrading. Reesor and his commission counterparts in Lewiston are more than a little nervous about all the attention being lavished on one bridge. "We feel there is a lack of appreciation for the overall impact on us if only the Peace Bridge is replaced," says Lewiston Commission spokesman Larry Pelura. "This is a regional issue," says Reesor. "All four bridges need to be addressed as a whole." In 1999, the commission replaced the tolled Lewiston-Queenston Bridge, between Lewiston and Niagara-on-the-Lake, at a cost of $430 million. It's a massive, 20-lane span designed to last at least 75 years. Its big, heavy cantilevered towers, painted in bold colors, are not unlike the Peace Bridge's own big, heavy cantilevered towers, but the Lewiston-Queenston Bridge's were easier to construct and cost a lot less. "The Peace Bridge…

Prompt(ASAP Essay)

This year I tried out for the @NUM1 grade volleyball team for my school. There were so many girls trying out that I was nervous about what team I was going to get on or if I even would make a team. So I worked my hardest and figured that, since I had experience from the other teams I was on before, I was now confident I would make the team.

No-Watermark Generation

When I made it I had so much faith in myself I knew I could do it again. As the season progressed I played better and better and did everything the coach asked me to.
When the tournament came around I knew it was all about focus and determination. When the tournament got to me I was going to prove to myself that I could do this. Even when my team lost I was still happy for them and we came out second.

Watermarked Generation

I showed up the first day and tried my best, but to my surprise I did not make the team. I was very upset and cried to myself for a little while, but later I got used to not playing with the team. The team was doing very well, and I was really nervous because if I did not get picked to play with them, what was I going to do? Then I got picked to play with them as a substitute. I learned to work my hardest so that I would get to play volleyball this year. I have never worked so hard for something before. I was sad at first and scared that I would not make the team, but when I worked hard I got the opportunity to finally play, and this has been the best year of my volleyball career so far!
I never expected things to turn around the way they did this year. I learned in this process to keep my head up no matter what happens in life. Just because you do not get picked the first time does not mean you cannot get in later. Things will happen for the best! Practice is also the key. If you work your hardest, anything can happen!