MC^{2}MARK: Distortion-Free Multi-Bit Watermarking for Long Messages

Xuehao Cui[1*], **Ruibo Chen**[1*], **Yihan Wu**[1,2], **Weidong Cai**[3], **Heng Huang**[1]
[1]Department of Computer Science, University of Maryland, College Park [2]Meta, Menlo Park
[3]School of Computer Science, University of Sydney, Sydney, NSW, Australia
{cedrus,rbchen,yhwu,heng}@umd.edu tom.cai@sydney.edu.au

Abstract

Large language models now produce text indistinguishable from human writing, which increases the need for reliable provenance tracing. Multi-bit watermarking can embed identifiers into generated text, but existing methods struggle to keep both text quality and watermark strength while carrying long messages. We propose MC^{2}MARK, a distortion-free multi-bit watermarking framework designed for reliable embedding and decoding of long messages. Our key technical idea is Multi-Channel Colored Reweighting, which encodes bits through structured token reweighting while keeping the token distribution unbiased, together with Multi-Layer Sequential Reweighting to strengthen the watermark signal and an evidence-accumulation detector for message recovery. Experiments show that MC^{2}MARK improves detectability and robustness over prior multi-bit watermarking methods while preserving generation quality, achieving near-perfect accuracy for short messages and exceeding the second-best method by nearly 30% for long messages.

1 Introduction

Recent advances in large language models have significantly improved text generation capabilities, but they also raise concerns about misuse and authenticity [1, 2, 3, 4, 5]. To address these issues, statistical watermarking methods [6, 7, 8, 9, 10] have been proposed to detect machine-generated text. These methods introduce hidden statistical patterns into the text generation process using secret keys, enabling later detection through statistical hypothesis testing. However, these methods often distort the model's output distribution, potentially degrading text quality.

To preserve generation quality while maintaining watermark effectiveness, distortion-free watermarking techniques have been developed. Aaronson [6] proposed Gumbel-max sampling to modify token distribution in a distortion-free manner. Subsequent approaches [11, 12, 13, 14], further optimized the trade-off between watermark strength and text quality. Recent methods like DiPmark [15], STA-1 [16], and MCMARK [17] have demonstrated improvements in robustness and detectability while maintaining distortion-freeness. However, these methods output only a zero-bit decision (watermarked or not) and do not provide the information needed for provenance tracing.

Multi-bit watermarking addresses this limitation by embedding additional information, such as model or user identifiers, into the generated text. Early approaches [18, 19] need access to the message space during detection, which limits their use in practice. More recent work, such as MPAC [20] and watermarks adopting error-correction codes [21], improved message encoding but sacrificed distortion-free properties. While BiMark [22] achieved distortion-free multi-bit watermarking, its detectability diminishes with longer messages, posing challenges for real-world deployment. As

*Equal contribution.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

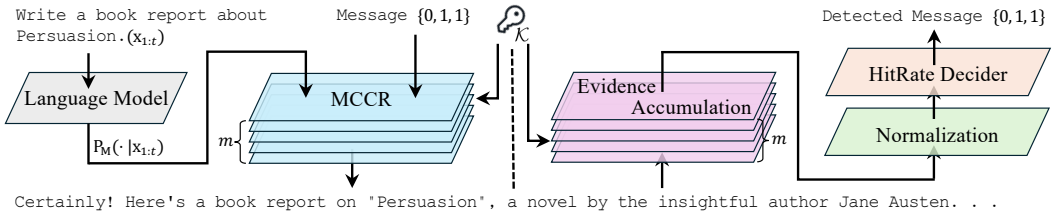


Figure 1: Overview of MC^{2}MARK framework. The generation process (left) employs Multi-Channel Colored Reweighting (MCCR) and Multi-Layer Sequential Reweighting (m layers illustrated) to produce text without distortion. The detection process (right) utilizes evidence accumulation for accurate message extraction.

pointed out in [22], the problem to preserve text quality and achieve large message embedding capacity simultaneously is non-trivial because the two goals trade off.

We address this problem by proposing MC^{2}MARK, a distortion-free multi-bit watermark framework that supports reliable embedding and detection for long messages. MC^{2}MARK uses Multi-Channel Colored Reweighting (MCCR), which dynamically scales token probabilities based on message bits while preserving the original distribution in expectation. We further propose Multi-Layer Sequential Reweighting (MLSR) to strengthen the statistical signals by iteratively reinforcing the watermark across layers. For message recovery, we use a detection method based on evidence accumulation to improve extraction accuracy.

The contribution of our work can be summarized as:

- We propose MC^{2}MARK, a novel distortion-free watermarking framework for multi-bit message embedding and detection that maintains high detectability and robustness even for long messages.
- We introduce Multi-Channel Colored Reweighting to embed messages by adaptively scaling token subsets, Multi-Layer Sequential Reweighting to strengthen watermark signals and a detection scheme based on evidence accumulation to robustly recover messages.
- We empirically demonstrate that MC^{2}MARK consistently outperforms existing multi-bit watermarking methods in detectability and robustness, while maintaining generation quality. For 256-bit messages, MC^{2}MARK achieves at least 91% accuracy, which exceeds previous methods by nearly 30%, setting a new benchmark for long-message multi-bit watermarking.

2 Related Work

Distortion-free watermarks. To preserve the quality of generated text, several studies have developed distortion-free watermarking methods. Aaronson [6] uses prefix n-grams as watermark keys and applies the Gumbel-max trick to modify token distribution while preserving generation quality. Christ et al. [11] determine watermark keys based on token positions and leverage inverse sampling during token sampling. ITS-edit and EXP-edit [12] adjust the token distributions in a distortion-free manner using inverse sampling and Gumbel-max, respectively. Hu et al. [13] introduce δ-reweighting and γ-reweighting techniques, but require prompts and logits from language models in detection. DiPmark [15] further improves the γ-reweighting and introduces a robust detector. Mao et al. [16] show the tradeoff between quality and watermark strength in low-entropy generation settings and propose STA-1 for better quality and watermark strength. SynthID [23] uses tournament-based watermarking to enable distortion-free generation. MCMARK [17] splits token distribution into multiple channels and adjusts the distribution in a distortion-free manner, greatly improving detectability. Advances [24, 25, 26, 27] have also been made in understanding the downstream effects of watermarking.

Multi-bit watermarks. Several studies [18, 19, 20, 21, 22, 28, 29, 30, 31] have developed methods for embedding multi-bit messages in generated text. Fernandez et al. [18] assign each message a unique secret key to control the watermarking process. Fairoze et al. [19] encodes messages by

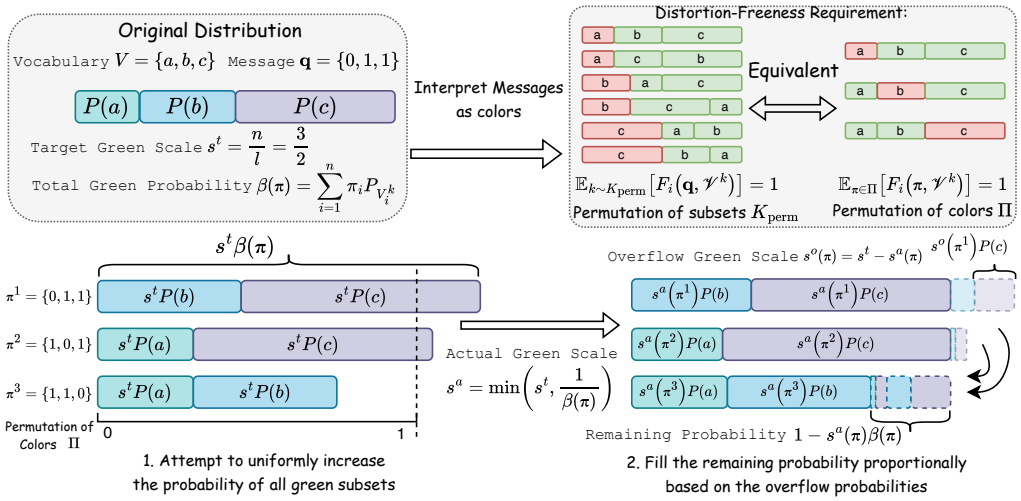


Figure 2: Illustration of the Multi-Channel Colored Reweighting (MCCR) framework. The proposed method interprets the message bit vector $\mathbf{q}$ as color assignments for partitioned vocabulary subsets $\mathcal{V}^k$, where $q_i = 1$ designates a green subset and $q_i = 0$ designates a red subset. The diagram visualizes the reweighting process across the permutation space Π to maintain distortion-free generation. The algorithm first attempts to uniformly amplify green subsets via a target scale s^t, computes the actual feasible green scale $s^a(\boldsymbol{\pi})$ based on the total green probability $\beta(\boldsymbol{\pi})$, and subsequently redistributes the remaining probability mass using the overflow scale $s^o(\boldsymbol{\pi})$.

feeding them into hash functions that determine watermarking constraints during generation. However, these approaches need access to the message space or verification context during detection because the messages are bound to hash values or secret keys in ways that cannot be directly reversed from the watermarked text. Yoo et al. [20] address this limitation by proposing Multi-bit watermark via Position Allocation (MPAC), which first assigns tokens to different parts of the messages and then encodes messages using the zero-bit watermarking scheme of [7]. Qu et al. [21] further improves robustness by incorporating error correction codes into multi-bit watermarking. However, these methods do not guarantee distortion-free generation. BiMark [22] employs a layered bit-flip reweighting mechanism to achieve distortion-free generation, but its detectability degrades rapidly as the message length increases, which limits its application in real-world scenarios.

3 Methodology

3.1 Notation

We follow the notation established in prior studies [13, 15, 17] to describe the token generation process of a large language model. Let V denote the vocabulary of size $N = |V|$, and let $\mathcal{V}$ represent the set of all possible token sequences of arbitrary length, including the empty sequence. Given an input prompt, the model generates tokens autoregressively. We denote the probability of generating the next token $x_{t+1} \in V$, conditioned on the sequence of previously generated tokens $\mathbf{x}_{1:t} = (x_1, \ldots, x_t)$, as $P_M(x_{t+1} \mid \mathbf{x}_{1:t})$. The resulting conditional distribution over the vocabulary lies within $\mathcal{P}$, the probability simplex. For notational brevity, we omit the explicit dependency on the input prompt. Given a private key k drawn from a key space $\mathcal{K}$, the reweighted logit distribution (referred to as a *distribution channel* in [17]) is denoted by $P_{M,w}(\cdot \mid \mathbf{x}_{1:t}, k)$. Existing literature [6, 11, 13, 23] has emphasized the concept of *distortion-free* (or unbiased) watermarks. This property ensures that, in expectation over the watermark key k, the token generation probabilities remain unaltered:

$$\mathbb{E}_{k \sim \mathcal{K}} \left[P_{M,w}(x_{t+1} \mid \mathbf{x}_{1:t}, k) \right] = P_M(x_{t+1} \mid \mathbf{x}_{1:t}). \tag{1}$$

Satisfying this condition implies that the quality of the generated text is better preserved. A summary of the notation used in the paper is provided in Appendix A.

3.2 Multi-Channel Colored Reweighting

We extend the Multi-Channel Watermark framework [17] to encode multi-bit information, proposing the **Multi-Channel Colored Watermark** (MC^{2}MARK). Given an n-bit message (i.e., payload) $\mathbf{q} \in \{0,1\}^n$ and a watermark key k, we first partition the vocabulary V into n disjoint subsets of equal size, denoted as $\mathcal{V}^k = \{V_1^k, V_2^k, \ldots, V_n^k\}$. We then reweight the probability distribution of tokens within each subset using a scaling factor $\alpha_i^k \geq 0$. The watermarked distribution $P_{M,w}$ is defined as:

$$P_{M,w}(x_{t+1} \mid \mathbf{x}_{1:t}, k) = \alpha_i^k P_M(x_{t+1} \mid \mathbf{x}_{1:t}),\ \forall x_{t+1} \in V_i^k. \tag{2}$$

As illustrated in Figure 2, we interpret the bit vector $\mathbf{q}$ as assigning a "color" to each subset V_i^k. Specifically, if $q_i = 1$, we designate V_i^k as a **green** subset and aim to amplify its probability mass (i.e., $\alpha_i^k \geq 1$). Conversely, if $q_i = 0$, we designate V_i^k as a **red** subset and attenuate its mass (i.e., $\alpha_i^k \leq 1$). To determine the optimal scaling factors $\boldsymbol{\alpha}^k \in \mathbb{R}^n$, we formulate an optimization problem where α_i^k is a function of the message and the partition:

$$\alpha_i^k = F_i(\mathbf{q}, \mathcal{V}^k). \tag{3}$$

Our objective is to maximize the expected probability mass of the green subsets:

$$J = \mathbb{E}_{k\sim\mathcal{K}}\left[\sum_{i=1}^{n} \alpha_i^k q_i P_{V_i^k}\right], \tag{4}$$

where $P_{V_i^k} = \sum_{x\in V_i^k} P_M(x \mid \mathbf{x}_{1:t})$. This maximization is subject to the following constraints:

$$\begin{cases} \alpha_i^k \geq 0, \quad \forall i \in \{1,\ldots,n\}, k \in \mathcal{K}, & \text{(5a)} \\ \displaystyle\sum_{i=1}^{n} \alpha_i^k P_{V_i^k} = 1, \quad \forall k \in \mathcal{K}, & \text{(5b)} \\ \mathbb{E}_{k\sim\mathcal{K}}\left[P_{M,w}(x_{t+1} \mid \mathbf{x}_{1:t}, k)\right] = P_M(x_{t+1} \mid \mathbf{x}_{1:t}), \quad \forall x_{t+1} \in V. & \text{(5c)} \end{cases}$$

Here, Eq. (5b) ensures the watermarked distribution is valid (sums to 1), and Eq. (5c) enforces the *distortion-free* property. To satisfy the distortion-free constraint, we leverage the symmetry of the key space. For any $k \in \mathcal{K}$, there exists a set of keys $K_{\text{perm}} \subset \mathcal{K}$ such that the partition structure is identical, but the assignment of subsets is permuted. Specifically, for any $k' \in K_{\text{perm}}$, $P_{V_i^k} = P_{V_{\sigma(i)}^{k'}}$, where σ is a permutation of $\{1,\ldots,n\}$. Consequently, the distortion-free constraint (Eq. (5c)) is satisfied if, for any such set K_{perm}, $\mathbb{E}_{k\sim K_{\text{perm}}}\left[P_{M,w}(x_{t+1} \mid \mathbf{x}_{1:t}, k)\right] = P_M(x_{t+1} \mid \mathbf{x}_{1:t})$, i.e.,

$$\mathbb{E}_{k\sim K_{\text{perm}}}[F_i(\mathbf{q}, \mathcal{V}^k)] = 1, \forall i \in \{1,\ldots,n\}. \tag{6}$$

Due to the symmetry of $\boldsymbol{\alpha}^k$ defined in Eq. (3), permuting the subsets $\{V_i^k\}$ is equivalent to permuting the message bits $\mathbf{c}$. Let $l = \sum_{i=1}^{n} q_i$ be the Hamming weight of the message. We denote the set of all binary vectors with weight l as:

$$\Pi = \left\{\mathbf{x} \in \{0,1\}^n : \sum_{i=1}^{n} x_i = l\right\}. \tag{7}$$

We can thus simplify Eq. (6) to the condition:

$$\mathbb{E}_{\boldsymbol{\pi}\in\Pi}\left[F_i(\boldsymbol{\pi}, \mathcal{V}^k)\right] = 1, \quad \forall i \in \{1,\ldots,n\}. \tag{8}$$

This transformation allows us to convert the original optimization problem into a linear programming task. However, finding the exact optimal solution is computationally prohibitive. We therefore propose a computationally efficient solution. As depicted in Figure 2, each $\boldsymbol{\pi} \in \Pi$ corresponds to a

potential distribution channel. We must define the scaling factors for all $|\Pi| = \binom{n}{l}$ permutations such that Eq. (8) holds.

First, we attempt to uniformly increase the probability of all green subsets. A subset V_i^k is green if $\pi_i = 1$, which occurs in $\binom{n-1}{l-1}$ permutations. To satisfy Eq. (8), the ideal **target green scale** s^t is:

$$s^t = \frac{\binom{n}{l}}{\binom{n-1}{l-1}} = \frac{n}{l}. \tag{9}$$

However, we must also respect the normalization constraint (Eq. (5b)). Let $\beta(\boldsymbol{\pi}) = \sum_{i=1}^{n} \pi_i P_{V_i^k}$ represent the total probability mass of the green subsets for a given permutation $\boldsymbol{\pi}$. We define the **actual green scale** as:

$$s^a(\boldsymbol{\pi}) = \min\left(s^t, \frac{1}{\beta(\boldsymbol{\pi})}\right). \tag{10}$$

The **overflow green scale** $s^o(\boldsymbol{\pi})$, which quantifies the scaling magnitude that cannot be accommodated by the channel, is given by

$$s^o(\boldsymbol{\pi}) = s^t - s^a(\boldsymbol{\pi}). \tag{11}$$

We aggregate this excess mass for each subset V_i^k across all permutations to compute the total overflow probability as $P^o_{V_i^k} = \sum_{\boldsymbol{\pi}\in\Pi} \pi_i s^o(\boldsymbol{\pi}) P_{V_i^k}$. Finally, to fully utilize the probability space, any remaining probability mass $1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi})$ is distributed proportionally based on the overflow probabilities. The resulting reweighting function F_i is:

$$F_i(\boldsymbol{\pi}, \mathcal{V}^k) = \pi_i s^a(\boldsymbol{\pi}) + \frac{(1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi}))P^o_{V_i^k}}{\left(\sum_{j=1}^{n} P^o_{V_j^k}\right) P_{V_i^k}}. \tag{12}$$

We verify that this solution satisfies both Eq. (5b) and Eq. (8) in Appendix B. Substituting the specific message $\mathbf{q}$ into the reweighting function, the final scaling factor α_i^k is:

$$\alpha_i^k = F_i(\mathbf{q}, \mathcal{V}^k) = c_i s^a(\mathbf{q}) + \frac{(1 - s^a(\mathbf{q})\beta(\mathbf{q}))P^o_{V_i^k}}{\left(\sum_{j=1}^{n} P^o_{V_j^k}\right) P_{V_i^k}}. \tag{13}$$

3.3 Generation

The proposed MC^{2}MARK generation process operates sequentially through a multi-layer manner. In this section, we detail the algorithm at a single time step t, given the base language model distribution $P_M(x_{t+1} \mid \mathbf{x}_{1:t})$ and the n-bit message payload $\mathbf{q} \in \{0,1\}^n$ intended for embedding.

Complexity Reduction via Segmentation. Directly computing the reweighting factors for MC^{2}MARK using Eq. (13) requires evaluating $|\Pi| = \binom{n}{l}$ channels. This yields a computational complexity of $O\left(\binom{n}{l}\right)$, which scales to $O(2^n/\sqrt{n})$ when $l \approx n/2$, becoming prohibitively expensive for large n. To mitigate this computational cost, we employ a segmentation strategy. We partition the message $\mathbf{q}$ into g disjoint segments $\mathbf{q}_1, \ldots, \mathbf{q}_g$, where each segment $\mathbf{q}_i \in \{0,1\}^{n'}$ and $n' = n/g$. We assume n is divisible by g without loss of generality.

At each time step t, we derive a pseudo-random seed from the watermark key k_t to sample: a segment index $\text{ind}_t \in \{1, \ldots, g\}$, a masking vector $\mathbf{h}_t \in \{0,1\}^{n'}$ and a vocabulary partition $\mathcal{V}^{k_t} = (V_i^{k_t})_{i=1}^{n'}$. The local payload for this step is computed as $\mathbf{q}'_t = \mathbf{q}_{\text{ind}_t} \oplus \mathbf{h}_t$, where $\oplus$ denotes the bit-wise XOR operation. This masking operation serves as a whitening transform, mitigating bias in $\mathbf{q}$ and ensuring that the expected Hamming weight satisfies $\mathbb{E}_{k\sim\mathcal{K}}[\sum_{i=1}^{n'} (\mathbf{q}'_t)_i] = n'/2$.

Single-Layer Reweighting. Subsequently, we compute the reweighting scales $\alpha_i^{k_t} = F_i(\mathbf{q}'_t, \mathcal{V}^{k_t})$ for all $i \in \{1, \ldots, n'\}$ using Eq. (13) and update the distribution. For all $x_{t+1} \in V_i^{k_t}$, the watermarked probability is:

$$P_{M,w}(x_{t+1} \mid \mathbf{x}_{1:t}, k_t) = \alpha_i^{k_t} P_M(x_{t+1} \mid \mathbf{x}_{1:t}). \tag{14}$$

Multi-Layer Sequential Reweighting. We extend this approach to a multi-layer setting to enhance robustness, following the framework proposed by [23, 32, 22]. We utilize m distinct watermark keys $k_t^1, \ldots, k_t^m$ at each step t. Let $P_0(\cdot \mid \mathbf{x}_{1:t}) = P_M(\cdot \mid \mathbf{x}_{1:t})$ denote the initial distribution. For each layer $j = 1, \ldots, m$, we sequentially apply the MC^{2}MARK reweighting. Specifically, given the partition defined by k_t^j, for any token $x_{t+1} \in V_i^{k_t^j}$, the distribution is updated as:

$$P_j(x_{t+1} \mid \mathbf{x}_{1:t}) = \alpha_i^{k_t^j} P_{j-1}(x_{t+1} \mid \mathbf{x}_{1:t}), \tag{15}$$

where $\alpha_i^{k_t^j}$ is calculated based on the probability mass of the partition in the previous distribution P_{j-1}. The final watermarked distribution is given by:

$$P_{M,w}(\cdot \mid \mathbf{x}_{1:t}, \{k_t^j\}_{j=1}^m) = P_m(\cdot \mid \mathbf{x}_{1:t}). \tag{16}$$

To ensure the watermark is recoverable, the keys k_t^j are generated deterministically using a cryptographic hash of the preceding tokens $\mathbf{x}_{1:t}$, consistent with the settings in previous works [23, 17]. Furthermore, since the single-layer MC^{2}MARK is distortion-free (satisfying Eq. (5c)) and the keys for each layer are sampled i.i.d. from $\mathcal{K}$, the composite multi-layer reweighting preserves the distortion-free property, as demonstrated in [23, 32]. The complete generation procedure is in Algorithm 1.

3.4 Detection

Given only the generated sequence $x_{1:T}$, the detector aims to recover the embedded message $\hat{\mathbf{q}} \in \{0,1\}^n$ without access to the original payload. As summarized in Algorithm 2, detection mirrors the generation procedure: for each position t and layer j, we deterministically reconstruct the same watermark components used in generation and aggregate statistical evidence across time and layers.

Evidence accumulation. For each next-token realization x_{t+1} at time step t, we regenerate the layer-specific watermark key k_t^j, the segment index $\mathrm{ind}_t^j \in \{1, \ldots, g\}$, the mask vector $\mathbf{h}_t^j \in \{0,1\}^{n'}$, and the corresponding vocabulary partition $\mathcal{V}^{k_t^j} = \{V_1^{k_t^j}, \ldots, V_{n'}^{k_t^j}\}$. We then identify the unique active subset index i^* such that $x_{t+1} \in V_{i^*}^{k_t^j}$.

Because MC^{2}MARK increases the sampling probability of *green* subsets, observing $x_{t+1} \in V_{i^*}^{k_t^j}$ provides evidence that subset i^* was colored green at (t, j). Recall that subset i is green if and only if the corresponding local payload bit equals one:

$$\left(\mathbf{q}_{\mathrm{ind}_t^j} \oplus \mathbf{h}_t^j\right)_i = 1. \tag{17}$$

Let $u^* = (\mathrm{ind}_t^j - 1)n' + i^*$ be the global bit index corresponding to segment ind_t^j and local coordinate i^*. Inverting the XOR in Eq. (17) yields the implied hypothesis for q_{u^*}: if $(\mathbf{h}_t^j)_{i^*} = 0$, then green implies $q_{u^*} = 1$; if $(\mathbf{h}_t^j)_{i^*} = 1$, then green implies $q_{u^*} = 0$. We record this evidence in per-bit hit counters $\mathbf{c}_0^{\mathrm{hit}}, \mathbf{c}_1^{\mathrm{hit}} \in \mathbb{Z}_{\geq 0}^n$, incrementing $\mathbf{c}_v^{\mathrm{hit}}(u^*)$ for the corresponding implied value $v \in \{0,1\}$.

Normalization and sequence extraction. A direct hit count can be biased because different bits may be exposed under different mask configurations across (t, j). To correct for this effect, we additionally maintain *total-opportunity* counters $\mathbf{c}_0^{\mathrm{total}}, \mathbf{c}_1^{\mathrm{total}} \in \mathbb{Z}_{\geq 0}^n$. For each reconstructed segment ind_t^j, we update $\mathbf{c}_v^{\mathrm{total}}$ for *all* local indices $i \in \{1, \ldots, n'\}$: if $(\mathbf{h}_t^j)_i = 0$, then a green observation at i would support $q_u = 1$, where $u = (\mathrm{ind}_t^j - 1)n' + i$, and we increment $\mathbf{c}_1^{\mathrm{total}}$ by 1; if $(\mathbf{h}_t^j)_i = 1$, it would support $q_u = 0$, then we increment $\mathbf{c}_0^{\mathrm{total}}$ instead. Thus, $\mathbf{c}_v^{\mathrm{total}}(u)$ counts the number of times the masking configuration at (t, j) would have allowed an observation to contribute evidence in favor of hypothesis $q_u = v$.

Table 1: Comparison of bit-wise accuracy for different message lengths n. [**KEY**: Best]

n	Method	*book_report*	*story*	*fake_news*	*dolly_cw*	*longform*	*finance*	*c4_subset*
16	MPAC(δ =1.0)	89.94%	96.48%	91.94%	79.63%	77.75%	78.06%	88.63%
	MPAC(δ =1.5)	98.31%	99.61%	97.94%	90.81%	90.13%	92.22%	98.19%
	BiMark	99.06%	99.61%	98.81%	94.38%	97.09%	97.13%	99.62%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**98.55%**	**100.00%**	**100.00%**	**100.00%**
32	MPAC(δ =1.0)	78.25%	86.39%	81.16%	75.56%	72.48%	72.86%	81.50%
	MPAC(δ =1.5)	87.41%	97.43%	91.50%	81.38%	78.17%	80.92%	91.28%
	BiMark	94.41%	97.52%	95.44%	93.50%	95.02%	91.41%	97.44%
	MC^{2}MARK	**99.88%**	**100.00%**	**100.00%**	**100.00%**	**99.92%**	**99.77%**	**100.00%**
64	MPAC(δ =1.0)	72.03%	79.00%	74.23%	66.80%	64.43%	66.27%	71.88%
	MPAC(δ =1.5)	81.97%	88.62%	82.67%	73.00%	69.35%	69.96%	81.02%
	BiMark	88.13%	91.67%	88.40%	87.76%	87.43%	86.08%	91.77%
	MC^{2}MARK	**99.94%**	**99.71%**	**98.89%**	**99.39%**	**98.97%**	**98.48%**	**99.80%**
128	MPAC(δ =1.0)	64.74%	70.42%	66.45%	61.62%	59.11%	59.96%	65.45%
	MPAC(δ =1.5)	74.35%	80.14%	72.78%	68.35%	66.48%	66.85%	74.10%
	BiMark	74.82%	79.35%	77.58%	74.99%	75.50%	72.52%	80.04%
	MC^{2}MARK	**99.38%**	**98.71%**	**96.30%**	**98.39%**	**94.49%**	**95.51%**	**99.55%**
256	MPAC(1.0)	60.81%	63.83%	60.72%	57.82%	55.96%	56.48%	60.60%
	MPAC(1.5)	65.81%	70.42%	66.75%	61.86%	60.86%	60.69%	67.25%
	BiMark	57.27%	60.99%	58.74%	57.62%	56.75%	53.87%	60.57%
	MC^{2}MARK	**96.19%**	**97.76%**	**97.41%**	**95.36%**	**95.47%**	**91.34%**	**97.39%**
512	MPAC(1.0)	57.55%	59.37%	57.74%	56.38%	55.77%	55.64%	57.71%
	MPAC(1.5)	60.95%	64.21%	60.88%	57.31%	55.62%	56.31%	60.80%
	BiMark	38.19%	41.18%	39.79%	37.77%	36.68%	36.82%	40.62%
	MC^{2}MARK	**92.70%**	**92.69%**	**92.04%**	**91.70%**	**87.24%**	**87.21%**	**92.39%**

After processing the full sequence, we compute an empirical hit rate for each bit $i \in \{1, \ldots, n\}$ under both hypotheses:

$$\mathrm{HitRate}_v(i) = \frac{\left(\mathbf{c}_v^{\mathrm{hit}}\right)_i}{\max\left(1, \left(\mathbf{c}_v^{\mathrm{total}}\right)_i\right)}, v \in \{0, 1\} \tag{18}$$

We decode each bit by comparing these normalized rates: $\hat{q}_i = 1$ if $\mathrm{HitRate}_1(i) > \mathrm{HitRate}_0(i)$, and $\hat{q}_i = 0$ otherwise. This normalization effectively marginalizes over randomness induced by the model distribution and the masking process, isolating the watermark signal while aggregating evidence across all time steps and layers.

Zero-bit detection. In zero-bit detection, we only test whether the generated sequence is watermarked or not. Under the null hypothesis that the sequence is unwatermarked, the joint count vector $(\mathbf{c}_0^{\mathrm{hit}}, \mathbf{c}_1^{\mathrm{hit}})$ follows a multinomial distribution $\mathrm{Multi}(mT, p)$, where each category is assigned probability $p = \frac{1}{2n}$. We define the following test statistic:

$$y = \frac{1}{mT} \sum_{i=1}^{n} \max\left((\mathbf{c}_0^{\mathrm{hit}})_i, (\mathbf{c}_1^{\mathrm{hit}})_i\right), \tag{19}$$

where m denotes the number of layers, T is the total number of generated tokens, and n is the message length. A larger value of y provides stronger evidence for the presence of a watermark.

4 Experiments

We implemented our pipeline in Python using the PyTorch framework and conducted experiments on four NVIDIA RTX6000ada GPUs. We evaluate our method in the aspects of detectability, robustness and distortion-freeness. The baselines are MPAC [20] and BiMark [22]. We follow their original configurations in the experiments. The number of layers m defaults to 10 in following experiments.

Table 2: Robustness under different attacks for various message lengths n. [**KEY**: Best]

n	Method	Random Token Replacement				Dipper Paraphrasing
		10%	**20%**	**30%**	**50%**	
16	MPAC(δ =1.0)	84.81%	78.31%	73.19%	63.00%	60.06%
	MPAC(δ =1.5)	96.38%	93.81%	87.13%	71.00%	66.19%
	BiMark	98.09%	93.11%	85.78%	65.94%	51.56%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**90.44%**	**72.40%**
32	MPAC(δ =1.0)	75.78%	71.91%	67.16%	59.69%	59.28%
	MPAC(δ =1.5)	85.84%	81.94%	75.78%	62.50%	58.72%
	BiMark	92.93%	85.07%	75.66%	60.26%	45.00%
	MC^{2}MARK	**100.00%**	**99.63%**	**97.97%**	**80.28%**	**67.59%**
64	MPAC(δ =1.0)	68.67%	66.20%	62.03%	56.09%	55.23%
	MPAC(δ =1.5)	76.88%	71.98%	66.50%	57.86%	56.59%
	BiMark	83.89%	75.45%	67.28%	54.22%	59.32%
	MC^{2}MARK	**99.09%**	**96.97%**	**91.52%**	**70.84%**	**62.51%**
128	MPAC(δ =1.0)	62.55%	60.40%	57.00%	53.79%	53.66%
	MPAC(δ =1.5)	70.41%	66.74%	63.34%	57.90%	55.34%
	BiMark	72.62%	64.66%	58.38%	49.48%	47.06%
	MC^{2}MARK	**97.85%**	**93.39%**	**85.34%**	**65.43%**	**58.61%**
256	MPAC(1.0)	58.23%	57.23%	54.79%	52.73%	52.55%
	MPAC(1.5)	63.94%	60.75%	58.67%	54.37%	53.92%
	BiMark	55.00%	50.42%	45.97%	41.01%	51.30%
	MC^{2}MARK	**92.35%**	**84.61%**	**75.52%**	**60.84%**	**54.90%**
512	MPAC(1.0)	56.51%	55.62%	54.37%	53.00%	52.57%
	MPAC(1.5)	58.85%	57.18%	55.75%	52.92%	52.61%
	BiMark	37.21%	34.21%	31.99%	28.50%	44.78%
	MC^{2}MARK	**84.81%**	**76.76%**	**68.84%**	**56.78%**	**54.01%**

Datasets and Models. The datasets used for detectability and robustness evaluation include three MMW datasets [33] (i.e., *book_report*, *story*, and *fake_news*), *dolly_cw* dataset [34], two WaterBench [35] QA datasets (i.e., *longform* and *finance*) and a subset randomly selected from C4 dataset [36], denoted as *c4_subset* in following sections. For unbaisedness evaluation, we use MBart [37] for machine translation and BART [38] for text summarization, following previous works [13, 15]. The language models used for other evaluation are Qwen2.5-3B-Instruct models [39] unless otherwise specified. Additional experiments on more language models are included in Appendix C.

4.1 Detectability

Table 1 reports bit-wise detection accuracy across seven datasets for different message lengths $n \in \{16, 32, 64, 128, 256, 512\}$. The comparison includes MPAC with $\delta = 1.0$ and $\delta = 1.5$, BiMark, and MC^{2}MARK. The text length is 512 for all generation and evaluation. Across all datasets and message lengths, MC^{2}MARK consistently achieves the highest detection accuracy. When the message length is very long, MC^{2}MARK outperforms the second-best method by roughly 30%. These results show that MC^{2}MARK provides consistently strong and stable detectability across diverse domains and message lengths. We also report full-sequence extraction accuracy and provide comparisons with additional baselines in Appendix C.

Detectability vs. text length. Figure 3 further analyzes detectability by illustrating detection accuracy as a function of text length for different message lengths. The comparison is conducted on *c4_subset* and focuses on MC^{2}MARK and BiMark. Across all message lengths and text lengths, MC^{2}MARK exhibits stronger performance than BiMark. As the message length increases, the performance gap between the two methods becomes larger. The accuracy of MC^{2}MARK saturates quickly around 100% as text length increases, suggesting that the embedded signal is evenly distributed across tokens and does not rely on long contexts for detection.

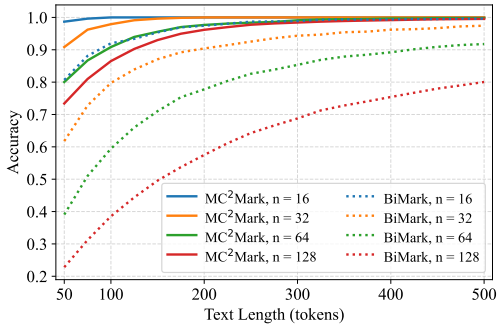


Figure 3: Detection accuracy as a function of text length for different message lengths n on *c4_subset*. Solid lines denote MC^{2}MARK, while dotted lines denote BiMark.

Table 3: Zero-bit detectability in TPR and AUROC at different message lengths. [**KEY**: Best]

n	Method	FPR=1%	FPR=0.1%	AUROC
32	BiMark	97.2%	85.9%	0.978
	Ours	**97.7%**	**97.2%**	**0.988**
64	BiMark	96.0%	89.7%	0.963
	Ours	**97.1%**	**96.8%**	**0.975**

Table 4: Comparison of our solution against the optimal solution when segment length $n' = 8$.

n	64	128	256
Ours	98.85%	97.96%	96.76%
Optimal	98.96%	97.54%	97.03%

Zero-bit detectability. We compare the zero-bit detectability in Table 3. The True Positive Rate (TPR) and AUROC are evaluated using Llama-2-7B-chat-hf [40] on a balanced test set consisting of 1000 watermarked and 1000 unwatermarked sequences. The results demonstrate that our method consistently achieves higher TPR at different FPR levels and a superior AUROC compared to BiMark, confirming the enhanced discriminative power of our proposed detection scheme.

4.2 Robustness

Table 2 reports the robustness of different methods on *c4_subset* under two types of perturbations: random token replacement and Dipper paraphrasing. For random token replacement, a fixed proportion of tokens (10%, 20%, 30%, or 50%) is randomly substituted. For paraphrasing, texts are paraphrased using the Dipper model [41]. Performance is measured by bit-wise accuracy. Under the attack of random token replacement, MC^{2}MARK demonstrates substantially stronger robustness across all message lengths and replacement ratios. MC^{2}MARK also achieves the highest robustness under Dipper paraphrasing attack for all message lengths.

4.3 Distortion-Freeness

Table 5 evaluates whether different watermarking methods introduce systematic bias that degrades task performance. We compare text quality on two representative generation tasks: text summarization and machine translation. The "No Watermark" setting serves as the reference baseline. For text summarization, our method maintains BERTScore, perplexity and ROUGE-1 values that are very close to the non-watermarked baseline, indicating minimal distortion of the summarization behavior. A similar pattern is observed for machine translation, in which our method achieves similar BERTScore and BLEU to the non-watermarked baseline. The results demonstrate that the proposed watermarking approach is unbiased with respect to downstream task quality, preserving both semantic accuracy and fluency while embedding the watermark.

4.4 Ablation Study

Generation methods. Figure 4 reports an ablation study comparing different methods under the same experimental setting on *book_report* dataset. MC^{2}MARK constantly achieves the highest decoding accuracy for all message lengths. Removing the mask bit from MC^{2}MARK leads to a noticeable performance drop when the message length becomes large. Although the accuracy at short message lengths ($n = 16$ and $n = 32$) remains close to that with the mask bit, the performance degrades significantly when $n = 64$ and $n = 128$. This comparison isolates the contribution of the mask bit and shows that it is a key component for preserving decoding accuracy. The vanilla method performs substantially worse than both MC^{2}MARK variants. Note that the detection method for MC^{2}MARK does not apply to methods without the mask bit, in the case of which we use the tail bound of binomial distribution to detect the message. Table 4 compares our solution against the

Table 5: Impact of watermarking methods on text summarization and machine translation quality.

	Text Summarization			Machine Translation	
Method	**BERTScore**	**ROUGE-1**	**Perplexity**	**BERTScore**	**BLEU**
No Watermark	0.3058	0.3772	6.4155	0.5436	20.2038
MPAC(δ =1.0)	0.3045	0.3715	6.7869	0.5358	20.0009
MPAC(δ =1.5)	0.2897	0.3678	7.4193	0.5313	19.0335
BiMark	0.3172	0.3850	5.0777	0.5460	20.2671
MC^{2}MARK	0.3065	0.3778	6.3742	0.5480	20.6516

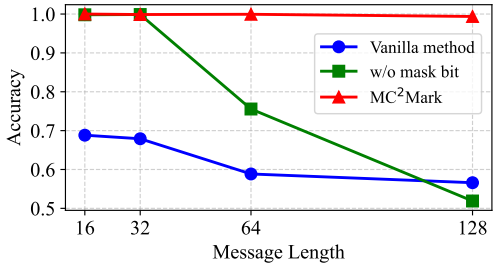


Figure 4: Accuracy comparison of MC^{2}Mark, MC^{2}Mark without the mask bit, and the vanilla method under different message lengths. The vanilla method directly applies MCMARK which amplifies only one channel in reweighting.

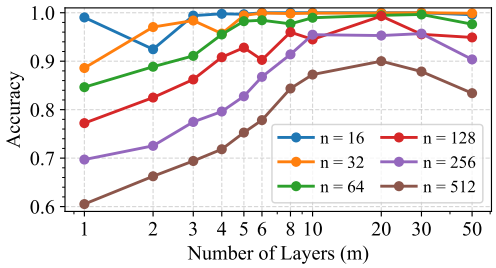


Figure 5: Accuracy as a function of the number of layers m for different message lengths $n \in \{16, 32, 64, 128, 256, 512\}$ on *longform* dataset. The horizontal axis is shown on a logarithmic scale. Performance peaks at $m = 20$.

optimal solution on *c4_subest* with segment length $n' = 8$. The optimal solution is got by directly solving the maximization problem in Eq. (4) under the constraints in Eq. (5). The empirical results show that our efficient method achieves performance nearly identical to the optimal solution.

Number of layers. Figure 5 presents an ablation study for the number of layers m on *longform* dataset. We evaluate MC^{2}MARK with different number of layers m while fixing all other settings, and report accuracy for different message lengths n. For short messages ($n = 16$ and $n = 32$), the accuracy is already high with a small number of layers. Increasing m from 1 to around 10 layers leads to a rapid improvement, after which the performance saturates and remains close to 1.0. Adding more layers beyond this point does not yield noticeable gains and can slightly reduce accuracy in some cases, indicating diminishing returns. For longer messages, shallow MC^{2}MARK perform worse, and accuracy increases more gradually as the number of layers grows. This shows that deeper architectures are required to capture the increased complexity associated with longer messages. For these settings, the performance tends to peak when the number of layers is approximately 20, and further increasing depth may incur slight loss of accuracy.

5 Conclusion

This work introduces MC^{2}MARK, a distortion-free multi-bit watermarking framework that achieves large message embedding capacity while preserving text quality. Theoretical analysis and extensive experiments have shown that MC^{2}MARK supports robust and accurate detection under long-message scenarios. The key lies in the layered multi-channel reweighting techniques and detection scheme based on evidence accumulation, providing a practical solution for deploying multi-bit watermarks. This approach enables scalable provenance tracking and accountability for AI-generated contents, safeguarding their use in real-world scenarios such as entertainment, social media and marketing.

Acknowledgements

This work was partially supported by NSF IIS 2347592, 2348169, DBI 2405416, CCF 2348306, CNS 2347617, RISE 2536663. The authors declare no competing interests.

References

[1] J. A. Goldstein, "Generative language models and automated influence operations: Emerging threats and potential mitigations," *arXiv preprint arXiv:2301.04246*, 2023.

[2] L. Weidinger, J. Mellor, M. Rauh, C. Griffin, J. Uesato, P.-S. Huang, M. Cheng, M. Glaese, B. Balle, A. Kasirzadeh *et al.*, "Ethical and social risks of harm from language models," *arXiv preprint arXiv:2112.04359*, 2021.

[3] P. Mai, X. Gao, Y. Ding, X. Du, W. Liu, and Y. Pang, "Sharedrequest: Privacy-preserving model-agnostic inference for large language models," in *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2026, pp. 7129–7150.

[4] P. Mai, Y. Yang, R. Yan, and Y. Pang, "Nexus scissor: enhance open-access language model safety by connection pruning," *npj Artificial Intelligence*, vol. 2, no. 1, p. 1, 2026.

[5] G. Milis, T. Sander, T. Souček, H. Huang, and P. Fernandez, "Could llm watermark detection be public?" *arXiv preprint arXiv:2610.12106*, 2026.

[6] S. Aaronson, "My AI safety lecture for UT effective altruism," 2022. [Online]. Available: https://scottaaronson.blog/?p=6823

[7] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, "A watermark for large language models," *arXiv preprint arXiv:2301.10226*, 2023.

[8] X. Zhao, P. Ananth, L. Li, and Y.-X. Wang, "Provable robust watermarking for ai-generated text," *arXiv preprint arXiv:2306.17439*, 2023.

[9] A. Liu, L. Pan, X. Hu, S. Meng, and L. Wen, "A semantic invariant robust watermark for large language models," *arXiv preprint arXiv:2310.06356*, 2023.

[10] R. Chen, Z. Hu, D. Lu, X. Cui, G. Milis, Y. Wu, J. Du, and H. Huang, "Ttmark: Pairwise distortion-free watermarking beyond single-token entropy," *arXiv preprint arXiv:2609.36372*, 2026.

[11] M. Christ, S. Gunn, and O. Zamir, "Undetectable watermarks for language models," *arXiv preprint arXiv:2306.09194*, 2023.

[12] R. Kuditipudi, J. Thickstun, T. Hashimoto, and P. Liang, "Robust distortion-free watermarks for language models," *arXiv preprint arXiv:2307.15593*, 2023.

[13] Z. Hu, L. Chen, X. Wu, Y. Wu, H. Zhang, and H. Huang, "Unbiased watermark for large language models," *arXiv preprint arXiv:2310.10669*, 2023.

[14] R. Chen, Y. Wu, X. Cui, J. Zhang, and H. Huang, "More haste, less speed: Weaker single-layer watermark improves distortion-free watermark ensembles," *arXiv preprint arXiv:2602.11793*, 2026.

[15] Y. Wu, Z. Hu, J. Guo, H. Zhang, and H. Huang, "A resilient and accessible distribution-preserving watermark for large language models," *arXiv preprint arXiv:2310.07710*, 2023.

[16] M. Mao, D. Wei, Z. Chen, X. Fang, and M. Chau, "A watermark for low-entropy and unbiased generation in large language models," *arXiv preprint arXiv:2405.14604*, 2024.

[17] R. Chen, Y. Wu, J. Guo, and H. Huang, "Improved unbiased watermark for large language models," *arXiv preprint arXiv:2502.11268*, 2025.

[18] P. Fernandez, A. Chaffin, K. Tit, V. Chappelier, and T. Furon, "Three bricks to consolidate watermarks for large language models," in *2023 IEEE international workshop on information forensics and security (WIFS)*. IEEE, 2023, pp. 1–6.

[19] J. Fairoze, S. Garg, S. Jha, S. Mahloujifar, M. Mahmoody, and M. Wang, "Publicly-detectable watermarking for language models," *arXiv preprint arXiv:2310.18491*, 2023.

[20] K. Yoo, W. Ahn, and N. Kwak, "Advancing beyond identification: Multi-bit watermark for large language models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 2024, pp. 4031–4055.

[21] W. Qu, W. Zheng, T. Tao, D. Yin, Y. Jiang, Z. Tian, W. Zou, J. Jia, and J. Zhang, "Provably robust multi-bit watermarking for ai-generated text," in *34th USENIX Security Symposium*, 2025, pp. 201–220.

[22] X. Feng, H. Zhang, Y. Zhang, L. Y. Zhang, and S. Pan, "Bimark: Unbiased multilayer watermarking for large language models," *arXiv preprint arXiv:2506.21602*, 2025.

[23] S. Dathathri, A. See, S. Ghaisas, P.-S. Huang, R. McAdam, J. Welbl, V. Bachani, A. Kaskasoli, R. Stanforth, T. Matejovicova *et al.*, "Scalable watermarking for identifying large language model outputs," *Nature*, vol. 634, no. 8035, pp. 818–823, 2024.

[24] A. Ajith, S. Singh, and D. Pruthi, "Downstream trade-offs of a family of text watermarks," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 14 039–14 053.

[25] P. Molenda, A. Liusie, and M. Gales, "Waterjudge: Quality-detection trade-off when watermarking large language models," in *Findings of the Association for Computational Linguistics: NAACL 2024*, 2024, pp. 3515–3525.

[26] A. Verma, N. Phan, and S. Trivedi, "Watermarking degrades alignment in language models: Analysis and mitigation," *arXiv preprint arXiv:2506.04462*, 2025.

[27] Y. Wu, X. Cui, R. Chen, and H. Huang, "Analyzing and evaluating unbiased language model watermark," in *International Conference on Learning Representations*, vol. 2026, 2026, pp. 142 418–142 433.

[28] Q. Lin, C. Tang, X. Li *et al.*, "Dermark: A dynamic, efficient and robust multi-bit watermark for large language models," *arXiv preprint arXiv:2502.05213*, 2025.

[29] Y. Jiang, M. K. Boroujeny, S. S. Kumar, and K. Zeng, "Mirrormark: A distortion-free multi-bit watermark for large language models," *arXiv preprint arXiv:2601.22246*, 2026.

[30] Y. Jiang, C. Wu, M. K. Boroujeny, B. Mark, and K. Zeng, "Stealthink: A multi-bit and stealthy watermark for large language models," *arXiv preprint arXiv:2506.05502*, 2025.

[31] Z. Yu, X. Jiang, W. Gu, Y. Wang, Q. Wen, S. Zhang, and W. Ye, "Saemark: Steering personalized multilingual llm watermarks with sparse autoencoders," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

[32] Y. Wu, R. Chen, G. Milis, and H. Huang, "An ensemble framework for unbiased language model watermarking," *arXiv preprint arXiv:2509.24043*, 2025.

[33] J. Piet, C. Sitawarin, V. Fang, N. Mu, and D. Wagner, "Mark my words: Analyzing and evaluating language model watermarks," *arXiv preprint arXiv:2312.00273*, 2023.

[34] M. Conover, M. Hayes, A. Mathur, J. Xie, J. Wan, S. Shah, A. Ghodsi, P. Wendell, M. Zaharia, and R. Xin, "Free dolly: Introducing the world's first truly open instruction-tuned llm," 2023. [Online]. Available: https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm

[35] S. Tu, Y. Sun, Y. Bai, J. Yu, L. Hou, and J. Li, "Waterbench: Towards holistic evaluation of watermarks for large language models," *arXiv preprint arXiv:2311.07138*, 2023.

[36] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of machine learning research*, vol. 21, no. 140, pp. 1–67, 2020.

[37] Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemoyer, "Multilingual denoising pre-training for neural machine translation," *Transactions of the Association for Computational Linguistics*, vol. 8, pp. 726–742, 2020.

[38] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. 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, pp. 7871–7880.

[39] Qwen Team, "Qwen2.5: A party of foundation models," September 2024. [Online]. Available: https://qwen.ai/blog?id=qwen2.5

[40] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[41] K. Krishna, Y. Song, M. Karpinska, J. Wieting, and M. Iyyer, "Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense," *arXiv preprint arXiv:2303.13408*, 2023.

[42] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[43] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl *et al.*, "Phi-3 technical report: A highly capable language model locally on your phone," *arXiv preprint arXiv:2404.14219*, 2024.

A Notation Table

Table 6: Summary of notation.

Symbol	Meaning
V	Model vocabulary
N	Vocabulary size, $N = \|V\|$
$\mathbf{x}_{1:t}$	Token prefix through step t
T	Generated-text length
P_M	Base next-token distribution
$P_{M,w}$	Watermarked next-token distribution
$\mathcal{K}$	Key space
k_t^j	Derived context key at position t, layer j
$\mathbf{q}$	Embedded payload, $\mathbf{q} \in \{0,1\}^n$
q_i	i-th payload bit, $q_i \in \{0,1\}$ for $i \in \{1,\dots,n\}$
n	Payload length
g	Number of payload segments
n'	Segment length, $n' = n/g$
$\mathbf{q}_i$	i-th payload segment, $\mathbf{q}_i \in \{0,1\}^{n'}$ for $i \in \{1,\dots,g\}$
m	Number of sequential reweighting layers
ind_t^j	Segment index at position t, layer j
$\mathbf{h}_t^j$	Mask vector at position t, layer j
$\mathcal{V}^k$	Key-dependent vocabulary partition, $\mathcal{V}^k = \{V_i^k\}_{i=1}^{n'}$
$P_{V_i^k}$	Base probability mass of subset V_i^k
α_i^k	Reweighting scale for subset V_i^k, $\alpha_i^k = F_i(\mathbf{q}, \mathcal{V}^k)$
F_i	Reweighting function for subset V_i^k
l	Local Hamming weight
Π	Set of weight-l binary vectors
$\boldsymbol{\pi}$	A binary vector in Π
$\beta(\boldsymbol{\pi})$	Total probability mass of green subsets for a given $\boldsymbol{\pi}$
s^t	Target green scale
s^a	Feasible actual green scale
s^o	Overflow green scale
$P^o_{V_i^k}$	Overflow probability assigned to subset V_i^k
$\mathbf{c}_v^{\mathrm{hit}}$	Hit counters supporting bit value $v \in \{0,1\}$
$\mathbf{c}_v^{\mathrm{total}}$	Total-opportunity counters supporting bit value $v \in \{0,1\}$
$\hat{\mathbf{q}}$	Recovered payload
y	Watermark-presence statistic

B Distortion-Freeness

First, we verify the satisfaction of Eq. (5b). Summing over all agents $i \in \{1, \ldots, n\}$, we obtain:

$$\begin{aligned}\sum_{i=1}^{n} F_i(\boldsymbol{\pi}, (V_i^k)_{i=1}^n) P_{V_i^k} &= s^a(\boldsymbol{\pi}) \sum_{i=1}^{n} \boldsymbol{\pi}_i P_{V_i^k} + \sum_{i=1}^{n} \frac{(1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi}))P^o_{V_i^k}}{(\sum_{j=1}^n P^o_{V_j^k})P_{V_i^k}} P_{V_i^k} \\ &= s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi}) + (1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi})) \frac{\sum_{i=1}^n P^o_{V_i^k}}{\sum_{j=1}^n P^o_{V_j^k}} \\ &= s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi}) + 1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi}) = 1. \end{aligned} \tag{20}$$

Next, we verify Eq. (8). We begin by expanding the expectation:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi}\in\Pi}\left[F_i(\boldsymbol{\pi}, (V_i^k)_{i=1}^n)\right] &= \frac{1}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \left(\boldsymbol{\pi}_i s^a(\boldsymbol{\pi}) + \frac{(1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi}))P^o_{V_i^k}}{(\sum_{j=1}^n P^o_{V_j^k})P_{V_i^k}} \right) \\ &= \frac{1}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i s^a(\boldsymbol{\pi}) + \frac{1}{|\Pi|} \frac{P^o_{V_i^k}}{(\sum_{j=1}^n P^o_{V_j^k})P_{V_i^k}} \sum_{\boldsymbol{\pi}\in\Pi} (1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi})). \end{aligned} \tag{21}$$

To simplify the second term in Eq. (21), we establish the following identity. Using the relationship $s^a(\boldsymbol{\pi}) = s^t - s^o(\boldsymbol{\pi})$ and the definition of $\beta(\boldsymbol{\pi})$, we derive:

$$\begin{aligned}\sum_{\boldsymbol{\pi}\in\Pi} (1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi})) &= \sum_{\boldsymbol{\pi}\in\Pi} (1 - s^t\beta(\boldsymbol{\pi}) + s^o(\boldsymbol{\pi})\beta(\boldsymbol{\pi})) \\ &= \sum_{\boldsymbol{\pi}\in\Pi} \left(1 - s^t \sum_{i=1}^{n} \boldsymbol{\pi}_i P_{V_i^k}\right) + \sum_{\boldsymbol{\pi}\in\Pi} \sum_{i=1}^{n} \boldsymbol{\pi}_i P_{V_i^k} s^o(\boldsymbol{\pi}) \\ &= |\Pi| - \frac{n}{l} \sum_{i=1}^{n} \left(P_{V_i^k} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i \right) + \sum_{i=1}^{n} P^o_{V_i^k} \\ &= |\Pi| - \frac{n}{l} \sum_{i=1}^{n} P_{V_i^k} \binom{n-1}{l-1} + \sum_{i=1}^{n} P^o_{V_i^k}. \end{aligned} \tag{22}$$

Substituting $|\Pi| = \binom{n}{l} = \frac{n}{l}\binom{n-1}{l-1}$ and recalling that $\sum_{i=1}^n P_{V_i^k} = 1$, the expression simplifies to:

$$\sum_{\boldsymbol{\pi}\in\Pi} (1 - s^a(\boldsymbol{\pi})\beta(\boldsymbol{\pi})) = |\Pi| - |\Pi| \underbrace{\sum_{i=1}^{n} P_{V_i^k}}_{1} + \sum_{i=1}^{n} P^o_{V_i^k} = \sum_{i=1}^{n} P^o_{V_i^k}. \tag{23}$$

Finally, substituting this result back into Eq. (21) yields:

$$\begin{aligned}\mathbb{E}_{\boldsymbol{\pi}\in\Pi}\left[F_i(\boldsymbol{\pi}, (V_i^k)_{i=1}^n)\right] &= \frac{1}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i s^a(\boldsymbol{\pi}) + \frac{1}{|\Pi|} \frac{P^o_{V_i^k}}{(\sum_{j=1}^n P^o_{V_j^k})P_{V_i^k}} \left(\sum_{j=1}^{n} P^o_{V_j^k} \right) \\ &= \frac{1}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i s^a(\boldsymbol{\pi}) + \frac{1}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i s^o(\boldsymbol{\pi}) \\ &= \frac{1}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i (s^a(\boldsymbol{\pi}) + s^o(\boldsymbol{\pi})) \\ &= \frac{s^t}{|\Pi|} \sum_{\boldsymbol{\pi}\in\Pi} \boldsymbol{\pi}_i = \frac{n/l}{\binom{n}{l}} \binom{n-1}{l-1} = 1. \end{aligned} \tag{24}$$

C Additional Results

Results on more language models. We present more results across different datasets on different large language models. Table 8 shows the detectability results on Llama-3.2-3B-Instruct [42], and Table 10 shows the detectability results on Phi-3.5-mini-instruct [43]. These results indicate that MC^{2}MARK can work effectively on different language models.

Full-sequence extraction accuracy. We report the comparison of full-sequence extraction accuracy across different datasets on Llama-3.2-3B-Instruct [42] in Table 9. The full-sequence extraction accuracy is sensitive to even small mistakes in detection, so it can decrease significantly even when bit-wise accuracy only decrease slightly. The results show that MC^{2}MARK constantly achieve better performance when measured in full-sequence extraction accuracy.

Comparison with additional methods. In Table 11, we provide an extended comparison with additional SOTA baselines (i.e., Qu et al. [21] and StealthInk [30]) using Llama-2-7B-chat-hf [40]. All results reflect the mean and standard deviation across four random seeds to ensure statistical reliability. Our method consistently outperforms these baselines in bit-wise extraction accuracy.

Impact of segment length. In Table 7, we evaluate the impact of segment length n' using a fixed total message length $n = 120$ on Llama-2-7b-chat-hf [40]. Benchmarked on a single NVIDIA H200 GPU, the results indicate that as scales, bit-wise accuracy improves at the cost of increased generation latency, representing a clear trade-off between runtime overhead and watermark signal strength.

Table 7: Generation time and bit-wise detection accuracy of MC^{2}MARK for different values of n'.

Method	**Generation Time (s/sequence)**	**Accuracy** (*book_report*)	**Accuracy** (*c4_subset*)
MC^{2}Mark ($n' = 2$)	1.6104636	83.54%	81.27%
MC^{2}Mark ($n' = 4$)	1.6154370	90.63%	91.10%
MC^{2}Mark ($n' = 6$)	1.8376376	97.12%	93.93%
MC^{2}Mark ($n' = 8$)	2.0869092	99.15%	98.98%
MC^{2}Mark ($n' = 10$)	2.1132132	99.92%	99.58%

D MC^{2}MARK Generator

Algorithm 1 MC^{2}MARK Generation Algorithm

Require: Language model M, generation length T, vocabulary V, key space $\mathcal{K}$, message $\mathbf{q} \in \{0,1\}^n$, segments g, layers m.
Ensure: Watermarked sequence $\mathbf{x}_{1:T}$.
Calculate segment length $n' \leftarrow n/g$.
for $t = 1$ **to** T **do**
  Initialize $P_0(\cdot \mid \mathbf{x}_{1:t}) \leftarrow P_M(\cdot \mid \mathbf{x}_{1:t})$
  for $j = 1$ **to** m **do**
    Derive watermark key $k_t^j \in \mathcal{K}$ based on $\mathbf{x}_{1:t}$.
    Generate pseudorandom components using k_t^j: segment index $\text{ind}_t^j \in \{1, \ldots, g\}$, mask vector $\mathbf{h}_t^j \in \{0,1\}^{n'}$, partition $\mathcal{V}^{k_t^j} = \{V_1^{k_t^j}, \ldots, V_{n'}^{k_t^j}\}$
    Compute target message segment: $(\mathbf{q}')_t^j \leftarrow \mathbf{q}_{\text{ind}_t^j} \oplus \mathbf{h}_t^j$.
    Compute scaling factors $\alpha_i^{k_t^j}$ via Eq. (13) using P_{j-1} statistics: $\alpha_i^{k_t^j} \leftarrow F_i((\mathbf{q}')_t^j, \mathcal{V}^{k_t^j}), \quad \forall i \in \{1, \ldots, n'\}$
    Update distribution for all i and all $x_{t+1} \in V_i^{k_t^j}$: $P_j(x_{t+1} \mid \mathbf{x}_{1:t}) \leftarrow \alpha_i^{k_t^j} P_{j-1}(x_{t+1} \mid \mathbf{x}_{1:t})$
  end for
  Sample next token $x_{t+1} \sim P_m(\cdot \mid \mathbf{x}_{1:t})$
end for
return $\mathbf{x}_{1:T}$

E MC^{2}MARK Detector

Algorithm 2 MC^{2}MARK Detection Algorithm

Require: Generated sequence $\mathbf{x}_{1:T}$, Vocabulary V, message length n, segment length n', layers m.
Ensure: Detected message $\hat{\mathbf{q}} \in \{0,1\}^n$.

Calculate number of segments $g \leftarrow n/n'$.
Initialize hit counters $\mathbf{c}_0^{\text{hit}}, \mathbf{c}_1^{\text{hit}} \in \mathbb{Z}^n$ to $\mathbf{0}$.
Initialize total occurrence counters $\mathbf{c}_0^{\text{total}}, \mathbf{c}_1^{\text{total}} \in \mathbb{Z}^n$ to $\mathbf{0}$.
for $t = 1$ to $T-1$ **do**
  for $j = 1$ to m **do**
    Derive watermark key k_t^j, mask $\mathbf{h}_t^j$, segment index ind_t^j, and partition $\mathcal{V}^{k_t^j}$.
    Identify the active partition index i^* such that $x_{t+1} \in V_{i^*}^{k_t^j}$.
    // Update Evidence (Hits)
    Calculate global index $u^* \leftarrow (\text{ind}_t^j - 1)n' + i^*$.
    if $(\mathbf{h}_t^j)_{i^*} = 1$ **then**
      $(\mathbf{c}_0^{\text{hit}})_{u^*} \leftarrow (\mathbf{c}_0^{\text{hit}})_{u^*} + 1$ ▷ Evidence supporting $q_{u^*} = 0$
    else
      $(\mathbf{c}_1^{\text{hit}})_{u^*} \leftarrow (\mathbf{c}_1^{\text{hit}})_{u^*} + 1$ ▷ Evidence supporting $q_{u^*} = 1$
    end if
    // Update Normalization Counts (Total Occurrences)
    for $i = 1$ to n' **do**
      Calculate global index $u \leftarrow (\text{ind}_t^j - 1)n' + i$.
      if $(\mathbf{h}_t^j)_i = 1$ **then**
        $(\mathbf{c}_0^{\text{total}})_u \leftarrow (\mathbf{c}_0^{\text{total}})_u + 1$
      else
        $(\mathbf{c}_1^{\text{total}})_u \leftarrow (\mathbf{c}_1^{\text{total}})_u + 1$
      end if
    end for
  end for
end for
// Decode Message by comparing hit rates
for $i = 1$ to n **do**
  $\text{HitRate}_0 \leftarrow (\mathbf{c}_0^{\text{hit}})_i / \max(1, (\mathbf{c}_0^{\text{total}})_i)$
  $\text{HitRate}_1 \leftarrow (\mathbf{c}_1^{\text{hit}})_i / \max(1, (\mathbf{c}_1^{\text{total}})_i)$
  if $\text{HitRate}_1 > \text{HitRate}_0$ **then**
    $\hat{q}_k \leftarrow 1$
  else
    $\hat{q}_k \leftarrow 0$
  end if
end for
return $\hat{\mathbf{q}}$

Algorithm 3 MC^{2}MARK Zero-Bit Detection Algorithm

Require: Generated sequence $\mathbf{x}_{1:T}$, Vocabulary V, message length n, segment length n', layers m, rejection threshold τ_α.
Ensure: Zero-bit decision $d \in \{\text{WATERMARKED}, \text{UNWATERMARKED}\}$.

Calculate number of segments $g \leftarrow n/n'$.
Initialize hit counters $\mathbf{c}_0^{\text{hit}}, \mathbf{c}_1^{\text{hit}} \in \mathbb{Z}^n$ to $\mathbf{0}$.
Initialize total occurrence counters $\mathbf{c}_0^{\text{total}}, \mathbf{c}_1^{\text{total}} \in \mathbb{Z}^n$ to $\mathbf{0}$.
for $t = 1$ to $T - 1$ **do**
  for $j = 1$ to m **do**
    Derive watermark key k_t^j, mask $\mathbf{h}_t^j$, segment index ind_t^j, and partition $\mathcal{V}^{k_t^j}$.
    Identify the active partition index i^* such that $x_{t+1} \in V_{i^*}^{k_t^j}$.
    // Update Evidence (Hits)
    Calculate global index $u^* \leftarrow (\text{ind}_t^j - 1)n' + i^*$.
    if $(\mathbf{h}_t^j)_{i^*} = 1$ **then**
      $(\mathbf{c}_0^{\text{hit}})_{u^*} \leftarrow (\mathbf{c}_0^{\text{hit}})_{u^*} + 1$ ▷ Evidence supporting $q_{u^*} = 0$
    else
      $(\mathbf{c}_1^{\text{hit}})_{u^*} \leftarrow (\mathbf{c}_1^{\text{hit}})_{u^*} + 1$ ▷ Evidence supporting $q_{u^*} = 1$
    end if
    // Update Normalization Counts (Total Occurrences)
    for $i = 1$ to n' **do**
      Calculate global index $u \leftarrow (\text{ind}_t^j - 1)n' + i$.
      if $(\mathbf{h}_t^j)_i = 1$ **then**
        $(\mathbf{c}_0^{\text{total}})_u \leftarrow (\mathbf{c}_0^{\text{total}})_u + 1$
      else
        $(\mathbf{c}_1^{\text{total}})_u \leftarrow (\mathbf{c}_1^{\text{total}})_u + 1$
      end if
    end for
  end for
end for
// Zero-bit decision using Eq. (18)
$y \leftarrow \dfrac{1}{mT} \displaystyle\sum_{i=1}^{n} \max\big((\mathbf{c}_0^{\text{hit}})_i, (\mathbf{c}_1^{\text{hit}})_i\big)$
if $y \geq \tau_\alpha$ **then**
  return WATERMARKED
else
  return UNWATERMARKED
end if

F Limitation

Although we have shown that MC^{2}MARK is effective through empirical evaluation, we mainly test our method on open-source large language models with the number of parameters ranging from 3B to 7B due to limited resources.

Table 8: Comparison of bit-wise accuracy on Llama-3.2-3B-Instruct across datasets for different message lengths n. [**KEY**: Best]

n	Method	*book_report*	*story*	*fake_news*	*dolly_cw*	*longform*	*finance*	*c4_subset*
16	MPAC(1.0)	77.63%	83.07%	83.81%	69.44%	72.38%	68.84%	80.75%
	MPAC(1.5)	88.06%	93.29%	95.06%	73.38%	87.50%	75.88%	89.69%
	BiMark	93.22%	96.74%	96.44%	93.75%	95.70%	96.79%	97.75%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
32	MPAC(1.0)	69.56%	75.23%	71.97%	62.41%	66.41%	65.77%	74.44%
	MPAC(1.5)	80.91%	87.73%	82.34%	69.25%	74.78%	72.28%	85.84%
	BiMark	88.24%	90.03%	91.02%	83.78%	88.78%	91.78%	94.34%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
64	MPAC(1.0)	67.47%	68.60%	67.95%	61.63%	63.88%	62.67%	70.00%
	MPAC(1.5)	75.00%	80.16%	76.19%	64.86%	69.09%	66.14%	77.31%
	BiMark	78.32%	80.67%	82.40%	75.75%	81.34%	82.88%	87.14%
	MC^{2}MARK	**99.98%**	**100.00%**	**99.97%**	**100.00%**	**100.00%**	**99.97%**	**99.97%**
128	MPAC(1.0)	60.53%	64.97%	60.51%	57.87%	58.08%	57.23%	62.57%
	MPAC(1.5)	67.16%	71.13%	69.86%	62.14%	63.55%	62.85%	70.56%
	BiMark	66.38%	68.86%	69.78%	63.36%	69.74%	69.55%	74.17%
	MC^{2}MARK	**99.64%**	**99.98%**	**99.88%**	**96.80%**	**99.77%**	**99.70%**	**99.94%**
256	MPAC(1.0)	57.55%	57.76%	57.95%	54.02%	54.66%	53.77%	58.27%
	MPAC(1.5)	62.92%	65.10%	62.37%	58.29%	58.28%	59.04%	65.82%
	BiMark	48.91%	52.20%	51.59%	47.47%	51.12%	50.52%	54.73%
	MC^{2}MARK	**98.45%**	**99.43%**	**99.10%**	**96.09%**	**98.40%**	**99.32%**	**99.61%**
512	MPAC(1.0)	55.81%	56.56%	57.19%	53.89%	54.10%	54.33%	56.13%
	MPAC(1.5)	59.35%	59.42%	58.82%	55.63%	55.40%	56.20%	60.24%
	BiMark	31.58%	34.02%	34.07%	30.11%	33.34%	31.26%	34.88%
	MC^{2}MARK	**95.61%**	**95.28%**	**96.06%**	**94.84%**	**94.69%**	**95.65%**	**95.77%**

Table 9: Comparison of full-sequence extraction accuracy on Llama-3.2-3B-Instruct across datasets for different message lengths n. [**KEY**: Best]

n	Method	*book_report*	*story*	*fake_news*	*dolly_cw*	*longform*	*finance*	*c4_subset*
16	MPAC(1.0)	18.00%	20.00%	22.00%	4.00%	2.00%	3.00%	27.00%
	MPAC(1.5)	40.00%	60.00%	59.00%	11.00%	25.00%	13.00%	51.00%
	BiMark	36.00%	65.00%	61.00%	47.00%	63.00%	66.00%	75.00%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
32	MPAC(1.0)	1.00%	0.00%	2.00%	1.00%	0.00%	0.00%	3.00%
	MPAC(1.5)	10.00%	21.00%	18.00%	4.00%	2.00%	4.00%	30.00%
	BiMark	3.00%	7.00%	9.00%	3.00%	8.00%	11.00%	26.00%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**	**100.00%**
64	MPAC(1.0)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MPAC(1.5)	1.00%	0.00%	1.00%	0.00%	1.00%	1.00%	1.00%
	BiMark	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MC^{2}MARK	**99.00%**	**100.00%**	**98.00%**	**100.00%**	**100.00%**	**98.00%**	**98.00%**
128	MPAC(1.0)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MPAC(1.5)	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	BiMark	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	MC^{2}MARK	**76.00%**	**97.00%**	**89.00%**	**76.00%**	**79.00%**	**83.00%**	**92.00%**

Table 10: Comparison of bit-wise accuracy on Phi-3.5-mini-instruct across datasets for different message lengths n. [**KEY**: Best]

n	Method	*book_report*	*story*	*fake_news*	*dolly_cw*	*longform*	*finance*	*c4_subset*
16	MPAC(1.0)	78.88%	86.52%	82.63%	75.69%	83.69%	75.81%	85.50%
	MPAC(1.5)	92.56%	97.66%	93.31%	88.00%	94.16%	89.25%	95.19%
	BiMark	95.66%	98.82%	97.83%	96.00%	97.21%	95.90%	98.19%
	MC^{2}MARK	**100.00%**	**100.00%**	**100.00%**	**93.49%**	**99.88%**	**100.00%**	**100.00%**
32	MPAC(1.0)	68.75%	74.45%	70.66%	67.34%	68.44%	66.44%	74.31%
	MPAC(1.5)	80.50%	86.17%	81.84%	76.06%	81.11%	76.47%	84.34%
	BiMark	88.96%	94.24%	92.55%	90.15%	92.12%	88.41%	93.47%
	MC^{2}MARK	**100.00%**	**99.97%**	**100.00%**	**99.27%**	**98.91%**	**99.97%**	**100.00%**
64	MPAC(1.0)	66.78%	71.00%	68.20%	66.38%	67.00%	65.96%	70.00%
	MPAC(1.5)	71.33%	77.20%	75.28%	68.28%	72.66%	69.38%	76.44%
	BiMark	80.01%	85.66%	83.43%	79.89%	82.10%	78.78%	86.19%
	MC^{2}MARK	**98.98%**	**97.95%**	**99.89%**	**98.57%**	**99.33%**	**99.59%**	**99.03%**
128	MPAC(1.0)	57.38%	61.29%	58.67%	58.13%	59.57%	58.46%	60.82%
	MPAC(1.5)	66.30%	71.17%	68.70%	64.21%	67.35%	65.08%	70.74%
	BiMark	68.45%	73.92%	69.41%	67.65%	69.62%	66.99%	73.28%
	MC^{2}MARK	**97.57%**	**99.58%**	**95.77%**	**94.78%**	**98.36%**	**98.54%**	**98.80%**
256	MPAC(1.0)	57.29%	58.75%	57.64%	55.75%	57.66%	56.95%	58.34%
	MPAC(1.5)	61.27%	63.59%	60.68%	58.70%	61.89%	60.34%	63.16%
	BiMark	52.33%	56.82%	54.16%	51.40%	52.87%	50.77%	55.28%
	MC^{2}MARK	**92.57%**	**94.98%**	**92.59%**	**91.20%**	**93.20%**	**94.08%**	**93.95%**
512	MPAC(1.0)	54.21%	56.55%	56.13%	54.83%	55.42%	54.90%	56.44%
	MPAC(1.5)	57.49%	59.33%	56.66%	56.55%	56.98%	57.10%	58.42%
	BiMark	34.22%	37.44%	35.51%	33.89%	34.45%	33.07%	36.15%
	MC^{2}MARK	**86.69%**	**88.95%**	**86.20%**	**86.67%**	**87.37%**	**86.84%**	**90.16%**

Table 11: Bit-wise detection accuracy with standard deviation across different datasets and message lengths n. The results are from four repeated runs with different random seeds.

n	Method	*book_report*	*story*	*c4_subset*
32	MPAC(1.0)	57.36%±3.06%	61.18%±1.64%	59.41%±2.48%
	MPAC(1.5)	62.21%±3.49%	64.45%±2.85%	60.53%±3.05%
	Qu et al. [21]	95.56%±0.29%	99.74%±0.37%	99.65%±0.17%
	StealthInk	96.23%±0.43%	97.89%±0.21%	93.96%±0.62%
	BiMark	70.51%±1.28%	74.83%±1.16%	83.23%±0.78%
	MC^{2}MARK	**99.97%±0.03%**	**99.62%±0.34%**	**97.62%±3.10%**
64	MPAC(1.0)	55.30%±1.63%	55.57%±2.06%	54.21%±1.73%
	MPAC(1.5)	59.15%±0.99%	60.78%±1.62%	58.21%±2.01%
	Qu et al. [21]	91.69%±2.08%	97.27%±0.79%	92.69%±1.32%
	StealthInk	91.08%±0.51%	92.15%±0.43%	86.98%±0.64%
	BiMark	63.11%±0.49%	65.45%±0.87%	73.49%±0.99%
	MC^{2}MARK	**99.46%±0.39%**	**99.93%±0.11%**	**99.69%±0.18%**
128	MPAC(1.0)	52.86%±0.74%	53.21%±0.79%	52.60%±0.59%
	MPAC(1.5)	55.45%±1.40%	57.08%±0.69%	54.65%±0.77%
	Qu et al. [21]	82.29%±0.98%	90.66%±0.28%	94.56%±0.77%
	StealthInk	82.73%±0.13%	84.38%±0.34%	79.75%±0.79%
	BiMark	54.80%±0.55%	56.18%±0.62%	61.13%±0.94%
	MC^{2}MARK	**98.78%±0.53%**	**98.97%±0.27%**	**97.99%±1.03%**
256	MPAC(1.0)	52.84%±0.79%	53.42%±0.64%	52.79%±0.75%
	MPAC(1.5)	52.90%±1.38%	54.08%±1.24%	52.91%±0.72%
	Qu et al. [21]	67.48%±0.98%	74.34%±1.83%	79.82%±0.49%
	StealthInk	74.06%±0.10%	75.47%±0.29%	71.80%±0.50%
	BiMark	41.76%±0.57%	43.26%±0.15%	45.58%±0.26%
	MC^{2}MARK	**96.58%±0.77%**	**97.52%±0.39%**	**95.77%±0.75%**

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We emphasize the contributions in the abstract. The contributions are also listed clearly in the last paragraph of the introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to the Limitation section in Appendix F.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: We include the assumptions when we formulate the problem in Section 3.1 and Section 3.2. A detailed proof of the distortion-freeness of our solution is included in Appendix B.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We include all necessary information to reproduce the main experimental results in Section 4. We also plan to release our code upon official publication.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [No]

Justification: While the code is currently not included in the paper, we plan to release it on GitHub at a later date.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Experiments details are included in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We quantify the statistical significance in Table 11 of Appendix C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computational resources are described in Section 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work is conducted in accordance with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The potential impacts are discussed in Section 1 and Section 5.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The creators or original owners of assets (e.g., code, data, models), used in the paper, are properly credited, and the license and terms of use are explicitly mentioned and are properly respected.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.

- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets introduced in the paper are well documented alongside the assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.

- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.