Optimizing Multi-bit LLM Watermark Detectability under Controlled Text Quality Degradation

Chenxi Gu
Monash University
chenxi.gu@monash.edu

Si Chen
Monash University
sche0499@student.monash.edu

John Grundy
Monash University
john.grundy@monash.edu

Xiaoning Du
Monash University
xiaoning.du@monash.edu

Abstract

Watermarking has emerged as an important mechanism for tracing the provenance of large language model (LLM) outputs. Multi-bit watermarks are particularly desirable because they can encode rich metadata, but embedding a more reliable watermark generally requires stronger modifications to the model's next-token distribution, potentially degrading text quality. This is a fundamental trade-off between distributional distortion and watermark detectability. Usually, a tolerable distortion is often treated as a constraint in practical deployment; a central question arises: *under a constrained level of distributional modification, how can detectability be optimized?* Existing multi-bit watermarking methods do not explicitly formulate watermark embedding under such a distributional constraint, nor do they provide a principled mechanism for optimizing detectability. A key challenge is that common detectability metrics, such as bit accuracy (BA), only become available after generation and decoding, so they cannot be optimized directly during token generation. We thus introduce a computable *detectability proxy* that is directionally aligned with BA and can be evaluated at each generation step. However, directly maximizing this proxy is computationally prohibitive under the hash-based designs commonly used in watermarking. We instead derive a tractable lower bound on the proxy and propose **BoundMark** to maximize this lower bound under a constrained distributional shift. Empirical evaluations on two LLMs and four tasks show that BoundMark achieves performance competitive with state-of-the-art multi-bit watermarks. The code is available at https://github.com/Leileileisa/Multibit_BoundMark.

1 Introduction

Large language models (LLMs) can generate high-quality, human-like text, making them powerful engines for a wide range of applications. However, these capabilities also introduce substantial risks: LLM outputs can be misused to infringe proprietary intellectual property, circumvent usage policies, or facilitate the large-scale generation of convincing misinformation Liu et al. [2024]. For example, prominent providers such as OpenAI, Anthropic, and Google Gemini explicitly prohibit the use of their outputs to train competing models Houman [2025], Google [2025], Anthropic [2024]. Yet, proving such violations is difficult when generated text cannot be reliably traced to its source. As regulatory bodies increasingly call for mechanisms that support content provenance and accountability Union [2024], reliable LLM watermarking has become an important research problem.

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

A major line of work studies zero-bit watermarking, which embeds an invisible statistical signal into generated text and subsequently tests whether the signal is present Kirchenbauer et al. [2024], Park et al. [2026], Lee et al. [2024], Lu et al. [2024]. While such methods can determine whether a piece of text is likely watermarked, their binary nature limits the amount of information that can be conveyed. In many provenance applications, it is desirable to encode richer metadata, such as a user identifier, model identifier, or generation context. This has motivated increasing interest in multi-bit watermarking Yoo et al. [2024], Fernandez et al. [2023], Qu et al. [2025], Li et al. [2024], Jiang et al. [2025], which embeds a recoverable message directly during generation.

Most existing multi-bit watermarking methods encode information by modifying the model's next-token distribution Yoo et al. [2024], Xu et al. [2026]. Stronger modifications generally make the embedded message more reliably recoverable, but also move the watermarked distribution further away from the model's original distribution, potentially degrading text quality. Multi-bit watermarking therefore faces a fundamental trade-off between *watermark detectability* and *distributional distortion*.

In practice, a tolerable distortion is usually treated as a constraint for a watermarking approach: given a prescribed budget on how much the original next-token distribution may be modified, an effective watermarking mechanism should successfully recognize the existence of embedded watermarks. The goal to design the most effective watermarking mechanism motivates the following question:

> *Under a constrained level of distributional modification, how can the detectability of a multi-bit watermark be optimzied?*

To the best of our knowledge, this problem has been rarely studied for multi-bit watermarking methodsYoo et al. [2024], Xu et al. [2026], Jovanović et al. [2024], Li et al. [2024]; either they fail to rigorously respect such constraints, or they have not attempted to optimize the detectability.

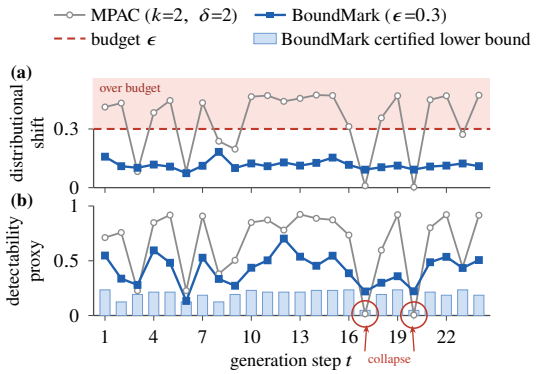

Figure 1: Per-step distributional shift and detectability proxy of MPAC Yoo et al. [2024] and BoundMark (Ours). (a) MPAC (k=2, δ=2), a logit-based multi-bit watermark, exceeds the distributional-shift budget ϵ on most steps, since nothing in its design constrains the shift, while BoundMark always stays within it. (b) MPAC's detectability proxy collapses on low-entropy steps (circled), whereas BoundMark always respects its certified lower bound.

A central issue is therefore how to quantify and constrain the distortion introduced by watermarking. Prior methods like Yoo et al. [2024], Xu et al. [2026] commonly use a watermark-strength parameter such as the logit bias δ. However, δ controls how strongly the logits of original distribution are perturbed rather than how much the distribution actually changes. For the same value of δ, the induced distributional shift can vary substantially across generation steps depending on the shape of the original next-token distribution. Consequently, constraining δ alone does not provide a consistent bound on the actual distortion introduced by watermarking. We instead quantify distributional modification using the Kullback–Leibler (KL) divergence between the original distribution p_t and the watermarked distribution q_t. By directly measuring the change induced in the next-token distribution, KL divergence allows us to impose a consistent per-step distortion budget ϵ. On this basis, we demonstrate how the distributional shift becomes more controllable with the KL-based distortion budget ($\epsilon = 0.3$) in Figure 1(a).

A second challenge in solving this constrained optimization problem is that common detectability metrics, such as bit accuracy (BA), only become accessible after the generated sequence is decoded, meaning that they are unavailable as direct optimization objectives during token generation. To bridge this gap, we introduce a computable *detectability proxy* D_t at generation step t. D_t can be evaluated at each generation step and is directionally aligned with the final BA achieved by message decoding (proofs provided in Appendix D).

Further, directly maximizing this proxy is computationally prohibitive under the hash-based mechanisms commonly adopted in multi-bit watermarking. We instead derive a tractable lower bound on the

detectability proxy and offer a way to maximize this lower bound under a prescribed distributional-shift constraint, which we call **BoundMark**. A comparison of the detection performance between the BoundMark method and MPAC Yoo et al. [2024], is depicted in Figure 1(b). Additionally, we provide the performance lower bound certified by BoundMark, which shows that MPAC does not always adhere to this lower bound and fails during certain steps.

Our main contributions are as follows:

- We formulate multi-bit LLM watermarking as a per-step constrained optimization problem that seeks to maximize watermark detectability under an explicit distributional-shift budget, providing a principled formulation of the detectability–distortion trade-off.
- We introduce a computable per-step detectability proxy that is directionally aligned with BA, and show that its direct maximization is computationally prohibitive under commonly used hash-based watermarking mechanisms. We therefore derive a tractable lower bound that can be optimized during generation.
- We derive a KL-minimal probability-bias allocation for a given total probability shift and obtain a closed-form bound on the allowable shift under a prescribed KL budget. This provides an explicit per-step guarantee on distributional distortion and enables a formal lower bound on the detectability proxy.
- We develop BoundMark, and evaluate it against seven multi-bit watermarking baselines on two LLMs and four tasks. BoundMark achieves performance competitive with state-of-the-art multi-bit watermarks while additionally providing explicit guarantees on both distributional shift and the detectability proxy.

2 Preliminaries

In this section, we formalize the standard auto-regressive text generation process and define the foundational components of multi-bit Large Language Model (LLM) watermarking, which has been adopted in a series of related worksYoo et al. [2024], Jiang et al. [2025], Xu et al. [2026], Qu et al. [2025], including the message routing mechanism, vocabulary partitioning, and detection pipeline.

2.1 LLM Token Generation

Given a vocabulary $\mathcal{V}$ of size $N = |\mathcal{V}|$, an auto-regressive language model generates text by predicting the next token based on a given context. Let $x_{<t} = (x_1, x_2, \ldots, x_{t-1})$ denote the sequence of previously generated tokens. At generation step t, the LLM computes a logit vector $\mathbf{z}^{(t)} \in \mathbb{R}^N$.

The logits are passed through a softmax function to obtain the natural probability distribution over the vocabulary, denoted as $\mathbf{p}^{(t)} \in \Delta^{N-1}$:

$$p_i^{(t)} = \frac{\exp(z_i^{(t)}/\tau)}{\sum_{j=1}^{N} \exp(z_j^{(t)}/\tau)}, \quad \forall i \in \mathcal{V} \tag{1}$$

where $\tau > 0$ is the temperature parameter. The next token x_t is then sampled from this distribution: $x_t \sim \mathbf{p}^{(t)}$.

2.2 Watermarked Text Generation

The objective of a multi-bit watermark is to embed a global m-bit secret message $M \in \{0,1\}^m$ into the generated text without significantly degrading text quality. To achieve this, the global message M is dynamically routed into a sequence of local k-bit payloads, where k represents the number of bits embedded at a specific generation step t.

Let $w_t \in \{0, 1, \ldots, 2^k - 1\}$ denote the target payload value to be embedded at step t. This payload is determined by a cryptographic hash function H_{msg} that takes the secret key K, the context history $x_{<t}$ (often a fixed-size window of previous tokens), and the global message M as inputs:

$$w_t = H_{\text{msg}}(K, x_{<t}, M) \pmod{2^k} \tag{2}$$

Concurrently, a second pseudo-random function H_{seed} is generated to partition the vocabulary $\mathcal{V}$ into 2^k disjoint sub-token sets, denoted as $V_0, V_1, \ldots, V_{2^k-1}$. The *chosen sub-token set* (or Green Set) corresponding to the desired payload w_t is defined as:

$$S_G^{(t)} = V_{w_t} \tag{3}$$

All other tokens form the Red Set, $S_R^{(t)} = \mathcal{V} \setminus S_G^{(t)}$. Let P_G denote the natural probability mass of the green set. It is defined using an indicator function $\mathbb{I}[\cdot]$ as:

$$P_G^{(t)} = \sum_{i \in \mathcal{V}_{w_t}} p_i^{(t)} \tag{4}$$

To embed the payload w_t, the watermark introduces a probability shift $\boldsymbol{\delta}^{(t)} \in \mathbb{R}^N$ to the original distribution, yielding the watermarked distribution $\mathbf{q}^{(t)}$:

$$q_i^{(t)} = p_i^{(t)} + \delta_i^{(t)} \tag{5}$$

To ensure $\mathbf{q}^{(t)}$ remains a valid probability distribution, the shifts must satisfy $\sum_{i \in \mathcal{V}} \delta_i^{(t)} = 0$ and $q_i^{(t)} \geq 0$. Typically, positive shifts ($\delta_i^{(t)} > 0$) are applied to tokens $i \in S_G^{(t)}$, effectively shifting a total probability mass Δ from the red Set into the green Set. Formally, this total shift is defined as the sum of the positive perturbations:

$$\Delta = \sum_{i \in S_G^{(t)}} \delta_i^{(t)} \tag{6}$$

This reallocation of probability mass increases the likelihood that the sampled token x_t will carry the desired payload w_t.

2.3 Watermark Detection and Extraction

Given a generated text sequence $X = (x_1, \ldots, x_T)$ and the secret key K, the detection process seeks to extract the embedded m-bit global message M.

For each step t, the detector reconstructs the context $x_{<t}$ and uses the secret key K to re-solve the maximin payload optimization, yielding the dynamic payload k_t. Steps with $k_t = 0$ carry no payload and are excluded from decoding. It then recomputes the 2^{k_t} vocabulary partitions $V_0, \ldots, V_{2^{k_t}-1}$ via the pseudo-random permutation. By observing which partition the actual generated token x_t falls into, the detector extracts the localized payload:

$$\hat{w}_t = j \quad \text{such that} \quad x_t \in V_j \tag{7}$$

Then the detector maps this localized payload back to a local candidate for the global message, denoted as $\hat{m}_t = \hat{w}_t$. The final extracted m-bit message $\hat{M}$ is reconstructed by performing a bit-wise majority vote across all valid local message candidates (i.e., from steps with $k_t > 0$) extracted from the sequence:

$$\hat{M}[i] = \text{Mode}\Big(\hat{m}_1[i], \hat{m}_2[i], \ldots, \hat{m}_T[i]\Big) \quad \text{for } i \in \{1, \ldots, m\} \tag{8}$$

Following prior multi-bit watermarking works, bit accuracy is measured as the fraction of message bits correctly recovered from the extracted message $\hat{M}$:

$$\text{BA}(M, \hat{M}) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{I}\Big[\hat{M}[i] = M[i]\Big], \tag{9}$$

3 Methodology

In this section, we construct our theoretical framework for multi-bit watermarking. We first define the formulations in Section 3.1, and introduce our algorithm in Section 3.2.

3.1 Formulation

3.1.1 Distributional Distortion

Definition 1 (Distributional Distortion). *At any generation step, let* $\mathbf{p}$ *denote the original next-token probability distribution produced by the LLM over a vocabulary* V*, where* p_x *is the probability of a token* x*. To embed a multi-bit watermark, a token-specific probability bias* δ_x *is applied to create a manipulated distribution* $\mathbf{q}$*, such that* $q_x = p_x + \delta_x$*. We define the local distributional distortion introduced by this probability shift as the Kullback-Leibler (KL) divergence from the original distribution* $\mathbf{p}$ *to the watermarked distribution* $\mathbf{q}$*:*

$$D_{\mathrm{KL}}(\mathbf{q} \parallel \mathbf{p}) = D_{\mathrm{KL}}(\mathbf{p} + \Delta \parallel \mathbf{p}) = \sum_{x \in V} (p_x + \delta_x) \ln \left(1 + \frac{\delta_x}{p_x}\right). \tag{10}$$

3.1.2 Multi-bit LLM Watermark Detectability Proxy

Although BA is intuitive for quantifying detectability, it cannot be computed during watermark embedding and therefore cannot be directly optimized. Therefore, we introduce the *Detectability Proxy*, a computable surrogate for watermark detectability.

Definition 2 (Detectability Proxy). *At any generation step* t*, let* $k \geq 0$ *denote the payload. We define the detectability proxy* D *of the embedded watermark signal as:*

$$D = k \cdot \Delta \tag{11}$$

Intuitively, an unwatermarked token carries no information about the message, while the shifted mass Δ is the extra probability with which the sampled token falls into the green set and thus signals its k payload bits. The product $k \cdot \Delta$ is therefore a simple per-step proxy for the amount of message information injected. In Appendix D, we show via a normal approximation of majority-vote decoding that, for a fixed sequence length and message length, the expected BA increases monotonically in both k and Δ, so increasing D is directionally consistent with improving BA.

3.1.3 Multi-bit LLM Watermark as a Constrained Optimization Problem

Definition 3 (Multi-Bit LLM Watermarking via Constrained Optimization). *We formally define the multi-bit watermarking process as a per-step constrained optimization problem. At any generation step with preceding context and a distortion budget* $\epsilon > 0$*, the framework seeks to maximize the detectability proxy* D*, subject to an explicit upper bound on the local distributional distortion and to the requirement that the watermarked distribution remains a valid probability distribution. Formally, the watermarking policy is the solution to:*

$$\max_{k \geq 0,\, \Delta \geq 0} \; k \cdot \Delta \quad \text{s.t.} \quad D_{\mathrm{KL}}(\mathbf{q} \parallel \mathbf{p}) \leq \epsilon, \quad \delta_i \geq 0 \;\; \forall i \in S_G. \tag{12}$$

3.2 BoundMark

Since k determines the green-token set S_G, the feasible range of Δ depends on both k and the distortion budget ϵ. Therefore, the constrained optimization problem in (12) is a nested optimization problem, we therefore decomposes the joint optimization over k and Δ into two levels. We first solve the inner optimization problem of determining the largest feasible Δ for any given k, and then solve the outer optimization problem of selecting k to maximize the detectability proxy.

Specifically, First, in Section 3.2.1, we investigate when Δ and k is given, how to allocate δ_i to achieve minimal D_{KL}. Moreover, the constraint about δ_i in (12) could be satisfied simultaneously. Based on this, in Section 3.2.2, we derive the maximal Δ that satisfies the $D_{KL} \leq \epsilon$ constraint when k is given. At last, in Section 3.2.3, we first explain why directly maximize $k \cdot \Delta$ is computationally intractable, and how we instead maximizing the lower bound of D.

3.2.1 Distortion-Minimal Bias Allocation

In prior multi-bit watermarking studies, the individual token bias is typically applied uniformly across the green setYoo et al. [2024], Xu et al. [2026]. However, uniform allocation is not optimal with

respect to the induced distributional shift. Instead, for any given total probability shift Δ, there exists a bias allocation that minimizes the KL divergence.

Theorem 1 *Given k and green token set S_G and a fixed valid probability shift Δ, the local distributional distortion $D_{\mathrm{KL}}(\mathbf{q} \parallel \mathbf{p})$ is strictly minimized if and only if the individual token bias $\delta_i = q_i - p_i$ is assigned proportionally to the original token probability p_i. Formally, the optimal closed-form solution for the token-level bias is:*

$$\delta_i = \begin{cases} p_i \left(\frac{\Delta}{P_G} \right), & \text{if } i \in S_G \\ -p_i \left(\frac{\Delta}{1-P_G} \right), & \text{if } i \in S_R \end{cases} \tag{13}$$

where $P_G = \sum_{j \in S_G} p_j$ represents the natural probability mass of the green subset, and S_R represents the red subset.

The formal proof of this optimization, solved via Lagrange multipliers, is detailed in Appendix B.

3.2.2 Maximal Probability Shift

With the optimal bias allocation established, we now determine the maximum allowable probability shift Δ that satisfies the distortion budget $\epsilon > 0$.

Theorem 2 (Maximal Probability Shift). *Given a distortion budget ϵ and a target vocabulary partition with a natural probability mass $P_G \in (0, 1)$, the proportional allocation of Theorem 1 with total shift Δ yields a valid distribution satisfying the local distortion constraint $D_{\mathrm{KL}}(\mathbf{q} \parallel \mathbf{p}) \leq \epsilon$ whenever*

$$0 \leq \Delta \leq \Delta_{\max} := \min \left\{ \sqrt{\epsilon P_G (1 - P_G)},\ 1 - P_G \right\}. \tag{14}$$

For simplicity, we denote $\min \left\{ \sqrt{\epsilon P_G (1 - P_G)},\ 1 - P_G \right\}$ as $g(P_G)$. The first term enforces the KL budget and the second enforces non-negativity of the red-set probabilities. The formal proof is detailed in Appendix C.

3.2.3 Optimization of Detectability Proxy

With the distributional distortion bounded by Theorem 2, we can now solve the objective of our constrained optimization problem (Definition 3): maximizing the detectability proxy.

By substituting the maximal admissible shift $\Delta_{\max}$ (Eq. (14)) into the detectability proxy $D = k \cdot \Delta$, the maximum detectability for a given payload k becomes:

$$D = k \cdot g(P_G) \tag{15}$$

Formally, we aim to solve:

$$\max_{k \geq 0} \Big(k \cdot g(P_G) \Big) \tag{16}$$

However, solving Eq. (16) directly during auto-regressive generation is computationally prohibitive. For any given payload k, P_G is determined by a pseudo-random hash assignment distributed over the entire vocabulary of size $|V|$. Evaluating the exact P_G to find the optimal k requires computing hash values and summing probabilities for the entire vocabulary at every single generation step. This heavy computational overhead introduces unacceptable inference latency, rendering direct exact maximization impractical for real-world LLM deployment.

To overcome this, our framework shifts its operational goal from direct maximization to a surrogate optimization. Specifically, we aim to find a payload k that maximizes k multiplied by a lower bound of $g(P_G)$ over all possible hash assignments. This surrogate objective is advantageous for two reasons:

1. **Computational Efficiency:** The lower bound of $g(P_G)$ can be deterministically calculated by sorting and evaluating only the top 2^k tokens in the natural distribution. This completely bypasses the need to compute hash assignments for the long tail of the vocabulary, reducing inference latency to a manageable constant.

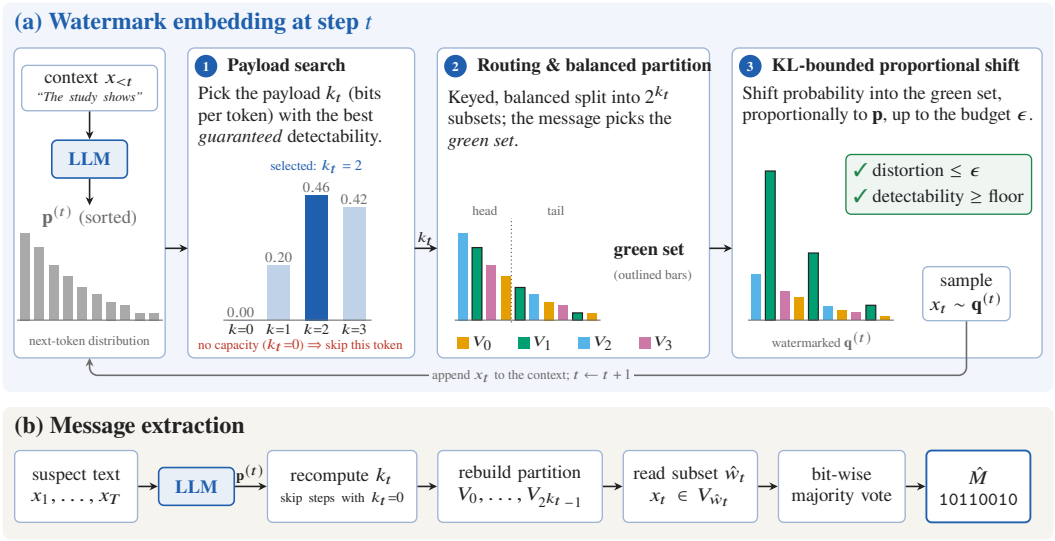

Figure 2: Overview of BoundMark. (a) At each generation step, BoundMark (1) selects the payload k_t that maximizes the guaranteed detectability $D_{\min}(k)$ under a balanced partition, abstaining if $k_t = 0$; (2) routes the message to a target subset and builds the balanced partition; and (3) shifts $\Delta_t = g(P_G)$ of probability mass to the green set proportionally to the original probabilities, which guarantees $D_{\mathrm{KL}}(\mathbf{q}\|\mathbf{p}) \le \epsilon$ and $D_t \ge D_{\min}(k_t)$. Numbers come from a toy distribution with $\epsilon = 0.5$. (b) Extraction recomputes k_t and the partition from the text and the key, reads the subset of each token, and recovers $\hat{M}$ by a bit-wise majority vote.

2. **Worst-Case Reliability:** By finding a k that maximizes the lower bound, we guarantee that the detectability proxy will always be greater than or equal to this bound, regardless of the actual pseudo-random hash assignment. This prevents the framework from selecting a fragile k that could result in ineffective watermark embedding.

Since $g(P_G)$ is concave on $[0, 1]$, its lower bound over any interval is attained at one of the endpoints. Therefore, given $P_G \in \left[\underline{P_G}, \overline{P_G}\right]$, we have

$$g(P_G) \ge \min\left\{g(\underline{P_G}), g(\overline{P_G})\right\}. \tag{17}$$

Thus, the lower bound of $g(P_G)$ is determined by the range of P_G.

However, Park et al. [2026], Gu et al. [2026] argued that the range of P_G is controlled by vocabulary partition algorithm, as depicted in (4). With unconstrained random partitioning, we have $\underline{P_G} = 0$ and $\overline{P_G} = 1$, since the green set may contain arbitrarily none or all of the probability mass. It therefore yields a trivial lower bound on $g(P_G)$, since $g(0) = g(1) = 0$. To alter the range of P_G, we adopt the balanced vocabulary partitioning strategy used by Park et al. [2026], Gu et al. [2026]. This strategy assigns exactly one of the top 2^k highest-probability tokens to each of the 2^k subsets, while assigning the remaining tokens through hashing.

Let $p_{(1)} \ge \cdots \ge p_{(2^k)}$ denote the top 2^k token probabilities, and let $P_{\mathrm{tail}}^{(k)} = 1 - \sum_{i=1}^{2^k} p_{(i)}$ denote the remaining probability mass. Because the green token set contains exactly one of these top-ranked tokens, its probability mass satisfies

$$\underbrace{p_{(2^k)}}_{\underline{P_G}} \le P_G \le \underbrace{p_{(1)} + P_{\mathrm{tail}}}_{\overline{P_G}}. \tag{18}$$

The $\underline{P_G}$ corresponds to assigning the smallest of the top-ranked tokens and no tail tokens to the target subset; the $\overline{P_G}$ corresponds to assigning the largest token and the entire tail. For $k \ge 1$, these bounds lie strictly inside $(0, 1)$ whenever the top 2^k token probabilities are positive. Consequently, for $\epsilon > 0$, this yields a positive lower bound on $g(P_G)$.

Multiplying this bound by k gives a guaranteed lower bound on the detectability proxy:

$$D_{\min}(k) := k \cdot \min\left\{g(\underline{P_G}), g(\overline{P_G})\right\} \leq k \cdot g(P_G). \tag{19}$$

BoundMark therefore selects

$$k_t \in \arg\max_{k \in \{0,\ldots,k_{\max}\}} D_{\min}(k), \tag{20}$$

where $D_{\min}(0) = 0$ represents embedding no payload and k_{max} is a hyper-parameter to limit the computational cost of payload selection.

3.2.4 Whole Pipeline of BoundMark

Figure 2 summarizes the whole pipeline of BoundMark. And Algorithms 1 and 2 in Appendix E give the full generation and detection procedures.

4 Experiments

4.1 Experimental Setup

Configurations and Baselines. To evaluate BoundMark, we simulate a multi-user environment consisting of 10 distinct users, each assigned a unique, randomly generated message for encoding. Each user provides two prompts, and the model generates a sequence of $T/2$ tokens per prompt. Consequently, a total of T tokens are produced per user for the decoding process, with the default sequence length established at $T = 150$.

We benchmark our approach against six state-of-the-art methods: DepthW Li et al. [2024], CycleShift Fernandez et al. [2023], MPAC Yoo et al. [2024], RSBH Qu et al. [2025], StealthInk Jiang et al. [2025], and XMark, LoSo Xu et al. [2026]. In the extended evaluation (Section 4.3) and the robustness study (Section 4.5), for a total of seven baselines. Due to the exponential growth in decoding latency associated with DepthW and CycleShift, their evaluation is restricted to a message length of $b = 8$ to remain computationally feasible. For XMark, we utilize default hyperparameters of $d = 2$ and $k = 2$. Across all applicable baselines (excluding StealthInk, which operates without a fixed bias), the watermark strength is set at $\delta = 2$. For BoundMark, we use $\epsilon = 2$ and $k_{max} = 5$ as default.

Datasets and Model Architectures. Our main analysis spans two representative downstream NLP tasks: *text completion* utilizing the C4 news dataset Raffel et al. [2023], and *text summarization* on the CNN/DailyMail corpus Hermann et al. [2015], with LLaMA-2-7B Touvron et al. [2023] as the generation model. To broaden the empirical coverage, we further evaluate BoundMark with Qwen2.5-7B Qwen et al. [2025] and *mathematical reasoning* on GSM8K Cobbe et al. [2021] and *code generation* on HumanEval Austin et al. [2021].

All experiments are conducted on Intel(R) Xeon(R) Gold 6330 CPU with NVIDIA A100 Graphic Card.

4.2 Evaluation Metrics

To assess the quality of the watermarked output, we measure **Perplexity (PPL)** using the larger LLaMA-2-13B model as the oracle. In the extended evaluation, we additionally report **BERTScore F1** Zhang et al. [2020] and **ROUGE-L** Lin [2004] against the reference outputs for summarization and text completion, **answer accuracy** for GSM8K, and **Pass@1** for HumanEval, each compared with the corresponding unwatermarked (Non-WM) generation.

Regarding watermark detectability, we adopt **Bit Accuracy (BA)** as the primary performance metric, consistent with established literature Fernandez et al. [2023], Yoo et al. [2024], Xu et al. [2026], Jiang et al. [2025]. A superior watermarking scheme is expected to maintain minimal PPL while maximizing BA.

4.3 Main Results

Overall Performance. As shown in Table 1, BoundMark$_{\epsilon=2}$ achieves the highest average bit accuracy on both tasks (99.38% on text completion, and 98.44% on summarization, tied with RSBH),

Table 1: Comparison of methods on two downstream tasks with $b = 8$ under different token budgets $T \in \{150, 200, 250, 300\}$. Best results in every column are denoted in bold.

Method	$T = 150$		$T = 200$		$T = 250$		$T = 300$		**Avg.**	
	BA↑	**PPL**↓	**BA**↑	**PPL**↓	**BA**↑	**PPL**↓	**BA**↑	**PPL**↓	**BA**↑	**PPL**↓
Text Completion										
CycleShift	97.50	4.98	95.00	5.16	**100.0**	4.90	95.00	4.55	96.88	4.90
DepthW	88.75	4.62	91.25	4.43	96.25	4.76	96.25	4.54	93.12	4.59
StealthInk	78.75	**4.54**	82.50	**3.46**	93.75	**4.16**	96.25	**4.02**	87.81	**4.04**
MPAC	97.5	5.12	98.75	5.03	96.25	4.84	98.75	4.88	97.81	4.97
RSBH	**100.0**	5.37	97.50	4.51	97.50	4.50	**100.0**	4.71	98.75	4.77
XMark	96.25	4.60	**100.0**	4.32	**100.0**	4.49	**100.0**	4.26	99.06	4.42
BoundMark$_{\epsilon=2}$	97.50	6.21	**100.0**	5.58	**100.0**	5.72	**100.0**	5.47	**99.38**	5.75
Text Summarization										
CycleShift	95.00	6.31	95.00	5.42	96.25	5.54	95.00	4.72	95.31	5.50
DepthW	91.25	6.07	87.50	4.57	83.75	4.55	92.50	5.25	88.75	5.11
StealthInk	75.00	5.14	88.75	**3.90**	95.00	**4.35**	90.00	**4.02**	87.18	**4.35**
MPAC	86.25	6.21	97.50	5.07	95.00	5.22	95.00	4.68	93.43	5.30
RSBH	**96.25**	5.92	97.50	5.33	**100.0**	5.08	**100.0**	5.11	**98.44**	5.36
XMark	82.50	5.75	**98.75**	4.95	96.25	4.64	**100.0**	5.08	94.37	5.11
BoundMark$_{\epsilon=2}$	**96.25**	**4.19**	**98.75**	5.19	**100.0**	5.44	98.75	5.21	**98.44**	5.01

and is the best or tied for the best in most individual columns. This indicates that the adaptive payload and maximin optimization provide competitive detectability. At the same time, BoundMark does not attain the lowest perplexity: the distortion budget $\epsilon = 2$ permits a relatively large shift, and methods such as StealthInk and XMark achieve lower PPL at slightly lower or comparable BA.

The empirical trade-off between text quality (PPL) and decoding reliability (BA) across C4 and CNN datasets is illustrated in Figure 3. Because these two metrics generally trade off against each other, no evaluated method consistently dominates all others on both metrics. BoundMark achieves a trade-off competitive with XMark, the state-of-the-art multi-bit method, while, unlike the baselines, providing an explicit bound on the per-step distributional shift and a lower bound on detectability proxy. Moreover, unlike XMark and DepthW, which show non-monotonic trends under varying δ, BoundMark follows a stable trajectory as ϵ varies, since ϵ directly controls the admissible shift.

Table 2 reports results with Qwen2.5-7B on CNN/DailyMail, C4, GSM8K, and HumanEval, together with an additional LLaMA-2-7B run on CNN/DailyMail with extended quality metrics (full results in Appendix F). BoundMark achieves the highest BA on Qwen2.5-7B CNN/DailyMail (97.50%) and GSM8K (87.50%), and remains close to the best method elsewhere (e.g., 98.75% vs. 100.00% on C4 and 87.50% vs. 90.78% on HumanEval). On the two low-entropy reasoning tasks, BoundMark preserves the task performance of the unwatermarked model in our runs (GSM8K accuracy 40.00%, HumanEval Pass@1 33.54%), whereas some high-BA baselines degrade it (e.g., CycleShift: 0.00% on GSM8K, 26.83% on HumanEval).

4.4 Impact of Message Length

Figure 4 illustrates the scaling behavior of BA as the message length m increases from 8 to 32 bits. Across both tasks, all methods exhibit a monotonic decline in decoding reliability, which is attributed to the increased information density per token. While baselines such as XMark and MPAC experience sharp performance degradation—with BA dropping toward 70% at $m = 32$—BoundMark degrades more gracefully, sustaining approximately 85% BA at $m = 32$ in our runs. In contrast to StealthInk and MPAC, which show high variance and wider confidence intervals as the payload grows, BoundMark exhibits comparatively stable behavior. This suggests that the payload-adaptive payload allocation helps accommodate the increased payload requirements of longer messages.

4.5 Robustness to Text-Modification Attacks

Although BoundMark is not designed to maximize adversarial robustness, we evaluate it under random insertion/deletion, synonym substitution, and DIPPER paraphrasing Krishna et al. [2023]

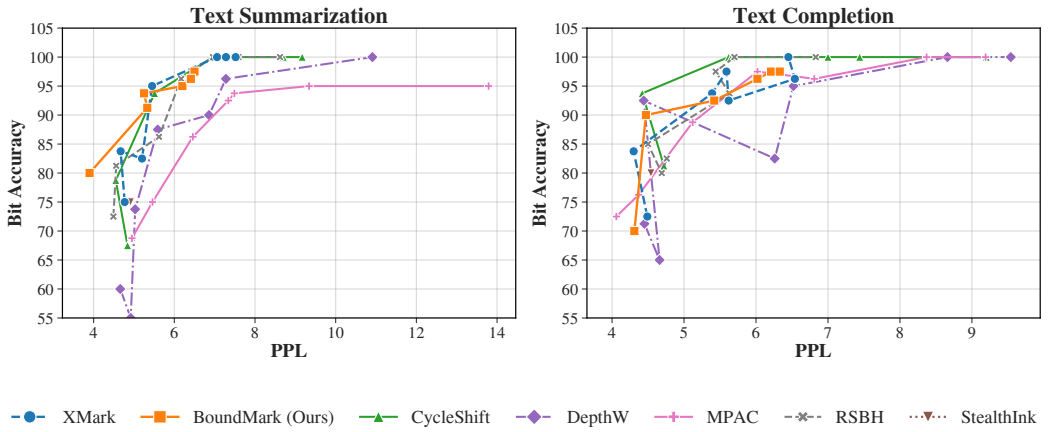

Figure 3: Results of watermarking methods on two downstream tasks. For BoundMark, we vary ϵ; for other methods, we vary δ.

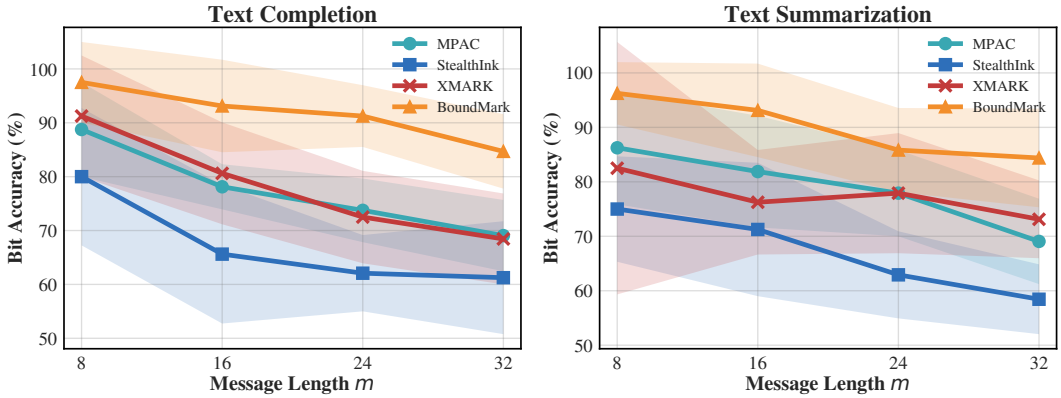

Figure 4: Results of watermarking methods on two downstream tasks when the message length varies.

(averaged results in Table 8 and per-setting results in Table 9, Appendix G). BoundMark has the highest clean BA (95.62%) and retains meaningful accuracy under attack (74.38%, 76.25%, and 55.00%, respectively), but it does not outperform the strongest baselines (e.g., MPAC, CycleShift, LoSo) under insertion/deletion and especially DIPPER, and its relative drop is larger than that of most baselines. We hypothesize that this is related to the adaptive payload: the detector recomputes k_t from the attacked context, so edits can also perturb the recovered payload schedule. Hence, the evidence does not support a claim that BoundMark improves adversarial robustness, consistent with its design goal of bounding distributional shift rather than resisting removal attacks; "detectability" in this paper thus refers to reliable message recovery under clean generation. Combining bounded distortion with strong robustness is left for future work.

5 Conclusion

We proposed BoundMark, a multi-bit LLM watermarking algorithm with explicitly controlled and theoretically bounded distributional shifts. Via per-step constrained optimization, BoundMark keeps the KL divergence of the watermarked next-token distribution within a budget, adapts payload and strength to the local distribution, and abstains when payload is insufficient. On two LLMs and four tasks, BoundMark achieves message recovery competitive with state-of-the-art multi-bit watermarks under clean generation while additionally providing a lower bound on the detectability proxy.

Acknowledgments and Disclosure of Funding

This research / project is supported by the Australian Research Council under its Discovery Early Career Researcher Award (DECRA) funding scheme (DE260100192) and Monash Faculty of Information Technology Research Stipend Scholarship (living allowance) and a Faculty of Information Technology International Postgraduate Research Scholarship.

References

Anthropic. Usage policy, June 2024. URL https://www.anthropic.com/legal/aup. Blog post.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL https://arxiv.org/abs/2108.07732.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models, 2023. URL https://arxiv.org/abs/2308.00113.

Google. Google apis terms of service, May 2025. URL https://ai.google.dev/gemini-api/terms#use-restrictions. Blog post.

Chenxi Gu, Xiaoning Du, and John Grundy. Ssg: Logit-balanced vocabulary partitioning for llm watermarking, 2026. URL https://arxiv.org/abs/2604.22438.

Karl Moritz Hermann, Tomáš Kočiský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701, 2015.

Houman. The openai vs deepseek knowledge distillation dispute: Technical and legal implications, January 2025. URL https://houman-asefi.medium.com. Blog post.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. Stealthink: A multi-bit and stealthy watermark for large language models, 2025. URL https://arxiv.org/abs/2506.05502.

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, 2024. URL https://arxiv.org/abs/2301.10226.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems*, 2023.

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. Who wrote this code? watermarking for code generation, 2024. URL https://arxiv.org/abs/2305.15060.

Liying Li, Yihan Bai, and Minhao Cheng. Where am I from? identifying origin of LLM-generated content. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12218–12229, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.681. URL https://aclanthology.org/2024.emnlp-main.681/.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81. Association for Computational Linguistics, 2004.

Aiwei Liu, Leyi Pan, Yijian Lu, Jingjing Li, Xuming Hu, Xi Zhang, Lijie Wen, Irwin King, Hui Xiong, and Philip S. Yu. A survey of text watermarking in the era of large language models, 2024. URL https://arxiv.org/abs/2312.07913.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method, 2024. URL https://arxiv.org/abs/2403.13485.

Shinwoo Park, Hyejin Park, Hyeseon Ahn, and Yo-Sub Han. Watermod: Modular token-rank partitioning for probability-balanced llm watermarking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(39):32683–32691, Mar. 2026. doi: 10.1609/aaai.v40i39.40546. URL https://ojs.aaai.org/index.php/AAAI/article/view/40546.

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, 2025. URL https://arxiv.org/abs/2401.16820.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

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, 2023. URL https://arxiv.org/abs/1910.10683.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

European Union. Regulation (eu) 2024/1689: Artificial intelligence act., June 2024. URL https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32024R1689. Blog post.

Jiahao Xu, Rui Hu, Olivera Kotevska, and Zikai Zhang. Xmark: Reliable multi-bit watermarking for llm-generated texts, 2026. URL https://arxiv.org/abs/2604.05242.

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4031–4055, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.224. URL https://aclanthology.org/2024.naacl-long.224/.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*, 2020.

A Limitations

Distributional rather than semantic guarantees. BoundMark bounds the per-step KL divergence between the watermarked and original next-token distributions. This is a proxy for distributional distortion and does not by itself guarantee semantic preservation; we also provide no sequence-level guarantee. Moreover, the objective $D = k \cdot \Delta$ is a heuristic proxy that is only directionally consistent with bit accuracy (Appendix D).

Robustness and security. BoundMark is not designed for adversarial robustness and does not consistently outperform the strongest baselines under insertion/deletion and paraphrasing attacks. It relies on a secret key, but we provide no formal unforgeability proof, and query-based watermark-stealing or spoofing attacks Jovanović et al. [2024], Sadasivan et al. [2023] are outside our scope.

Evaluated settings and conservatism. Our evaluation covers two LLMs and four tasks but is restricted to sequences of at most $T = 300$ tokens and payloads of at most $m = 32$ bits; we make no claims beyond these settings. Finally, $\Delta_{\max}$ relies on the inequality $\ln(1 + x) \leq x$ (a χ^2 upper bound on the KL divergence), which overestimates distortion in some contexts; a tighter bound would permit stronger signals within the same budget.

B Optimal Probability Shift Allocation

In this section, we prove that to minimize the distributional distortion $D_{KL}(\mathbf{q}||\mathbf{p})$ for a fixed total probability shift Δ, the individual token shifts δ_i must be allocated proportionally to their original probabilities p_i.

The Kullback-Leibler divergence from the original distribution $\mathbf{p}$ to the watermarked distribution $\mathbf{q}$ is defined as:

$$D_{KL}(\mathbf{q}||\mathbf{p}) = \sum_{x \in \mathcal{V}} q_x \ln\left(\frac{q_x}{p_x}\right) \tag{21}$$

The watermarking algorithm partitions the vocabulary into a Green Set S_G and a Red Set S_R. We require shifting a total mass of Δ into S_G. Thus, we have the following constraints for the green Set:

$$\sum_{i \in S_G} q_i = P_G + \Delta \tag{22}$$

where $P_G = \sum_{i \in S_G} p_i$. To find the optimal distribution $\mathbf{q}^*$ that minimizes the distortion solely within the green Set, we formulate the Lagrangian:

$$\mathcal{L}(\mathbf{q}, \lambda) = \sum_{i \in S_G} q_i \ln\left(\frac{q_i}{p_i}\right) - \lambda\left(\sum_{i \in S_G} q_i - (P_G + \Delta)\right) \tag{23}$$

Taking the partial derivative with respect to an individual token probability q_i and setting it to zero:

$$\frac{\partial \mathcal{L}}{\partial q_i} = \ln\left(\frac{q_i}{p_i}\right) + q_i\left(\frac{p_i}{q_i}\right)\left(\frac{1}{p_i}\right) - \lambda = 0 \tag{24}$$

$$\ln\left(\frac{q_i}{p_i}\right) + 1 - \lambda = 0 \implies q_i = p_i \exp(\lambda - 1) \tag{25}$$

This reveals that the optimal watermarked probability q_i is strictly proportional to p_i. To solve for the constant multiplier $\exp(\lambda - 1)$, we sum over all $i \in S_G$:

$$\sum_{i \in S_G} q_i = \exp(\lambda - 1) \sum_{i \in S_G} p_i \tag{26}$$

Substituting our constraint $\sum_{i \in S_G} q_i = P_G + \Delta$ and $P_G = \sum_{i \in S_G} p_i$:

$$P_G + \Delta = \exp(\lambda - 1) P_G \implies \exp(\lambda - 1) = \frac{P_G + \Delta}{P_G} = 1 + \frac{\Delta}{P_G} \tag{27}$$

Therefore, the optimal watermarked probability for target tokens is:

$$q_i = p_i \left(1 + \frac{\Delta}{P_G}\right) \implies \delta_i = p_i \left(\frac{\Delta}{P_G}\right) \tag{28}$$

Applying identical logic to the Red Set S_R, which must lose a total mass of Δ, yields:

$$q_j = p_j \left(1 - \frac{\Delta}{1 - P_G}\right) \implies \delta_j = -p_j \left(\frac{\Delta}{1 - P_G}\right), \quad \forall j \in S_R \tag{29}$$

This confirms that the optimal shift strategy to minimize D_{KL} is strictly proportional allocation.

C Proof of Estimation on Bound of Δ

To evaluate the distributional distortion analytically, we apply the first-order logarithmic inequality $\ln(1 + x) \leq x$ to our exact definition of D_{KL} (Definition 1). Because the sum of all probability biases across the vocabulary must equal zero ($\sum_{x \in V} \delta_x = 0$), the bound simplifies into:

$$D_{\mathrm{KL}}(\mathbf{q} \parallel \mathbf{p}) = \sum_{x \in V} (p_x + \delta_x) \ln \left(1 + \frac{\delta_x}{p_x}\right) \leq \sum_{x \in V} (p_x + \delta_x) \left(\frac{\delta_x}{p_x}\right) = \sum_{x \in V} \frac{\delta_x^2}{p_x} \tag{30}$$

Substituting the optimal proportional bias allocation (δ_x) defined in Theorem 1 into this bound, we separate the sum into the target subset S_G and the remaining subset S_R:

$$D_{\mathrm{KL}} \leq \sum_{x \in S_G} \frac{1}{p_x} \left(p_x \frac{\Delta}{P_G}\right)^2 + \sum_{x \in S_R} \frac{1}{p_x} \left(-p_x \frac{\Delta}{1 - P_G}\right)^2 = \frac{\Delta^2}{P_G} + \frac{\Delta^2}{1 - P_G} = \frac{\Delta^2}{P_G(1 - P_G)} \tag{31}$$

To enforce the distortion constraint at the current generation step, it suffices that this distribution-dependent upper bound does not exceed the budget ϵ:

$$\frac{\Delta^2}{P_G(1 - P_G)} \leq \epsilon \iff \Delta \leq \sqrt{\epsilon P_G(1 - P_G)}. \tag{32}$$

Feasibility. Under the proportional allocation, the red-set probabilities are $q_j = p_j\big(1 - \frac{\Delta}{1 - P_G}\big)$ for $j \in S_R$, which are non-negative if and only if $\Delta \leq 1 - P_G$. The green-set probabilities $q_i = p_i\big(1 + \frac{\Delta}{P_G}\big)$ are positive and sum to $P_G + \Delta \leq 1$, and $\sum_i q_i = 1$ holds by construction. At the boundary $\Delta = 1 - P_G$, the red-set probabilities become zero; with the standard convention $0 \ln 0 = 0$, the corresponding KL terms vanish and the inequality $q \ln(q/p) \leq q(q - p)/p$ used above still holds. Combining both conditions yields the maximal admissible shift $\Delta_{\max} = \min\{\sqrt{\epsilon P_G(1 - P_G)},\, 1 - P_G\}$, which establishes Theorem 2.

D Relation between the Detectability Proxy and Bit Accuracy

Consider a simplified model in which each watermarked token casts an independent vote for each of its k payload bits, and a vote is correct with probability $q = \frac{1}{2} + \frac{\Delta}{2}$. With T generated tokens and an m-bit message, each message bit receives approximately $N \approx Tk/m$ votes. By the normal approximation to the binomial majority vote,

$$\mathrm{BA} \approx \Phi\left(\frac{(q - \frac{1}{2})\sqrt{N}}{\sqrt{q(1 - q)}}\right) \approx \Phi\Big(\Delta\sqrt{N}\Big) = \Phi\left(\Delta\sqrt{\tfrac{Tk}{m}}\right), \tag{33}$$

where Φ is the standard normal CDF and the second approximation uses $q(1 - q) \approx \frac{1}{4}$ for small Δ. For fixed T and m, the expected bit accuracy is therefore monotonically increasing in both k and Δ. Although Eq. (33) and $D = k \cdot \Delta$ are not identical functions, increasing D (by increasing k, Δ, or both) is directionally consistent with improving BA, which motivates our use of D as the optimization objective.

Algorithm 1 The BoundMark Generation Pipeline (at step t)

Require: Context $x_{<t}$, Secret Key K, Global Message M, Distortion Budget ϵ, Maximum Payload $k_{\max}$

Ensure: Watermarked token x_t

 # LLM Forward Pass: Obtain original distribution $\mathbf{p}^{(t)}$ over vocabulary $\mathcal{V}$.
Sort $\mathbf{p}^{(t)}$ in descending order to obtain order statistics: $p_{(1)} \geq p_{(2)} \cdots \geq p_{(N)}$.
Define $g(P) := \min\{\sqrt{\epsilon P(1-P)},\, 1-P\}$ {KL budget and feasibility, Eq. (14)}
$D_{\min}(0) \leftarrow 0$
for candidate $k = 1, 2, 3, \ldots, k_{\max}$ **do**
    Calculate bounds: $P^{(k)}_{G,\min} \leftarrow p_{(2^k)}$ and $P^{(k)}_{G,\max} \leftarrow 1 - \sum_{i=2}^{2^k} p_{(i)}$
    Calculate guaranteed detectability: $D_{\min}(k) \leftarrow k \cdot \min\left(g(P^{(k)}_{G,\min}),\, g(P^{(k)}_{G,\max})\right)$
end for
Select payload: $k_t \leftarrow \arg\max_{0 \leq k \leq k_{\max}} D_{\min}(k)$.
if $k_t = 0$ **then**
    # Abstention: no admissible shift; sample $x_t \sim \mathbf{p}^{(t)}$ and **return** x_t.
end if
Compute target payload: $w_t \leftarrow H_{\text{msg}}(K, x_{<t}, M) \pmod{2^{k_t}}$.
Generate pseudo-random permutation π of $\{0, 1, \ldots, 2^{k_t} - 1\}$ seeded by $H_{\text{seed}}(K, x_{<t}, k_t)$.
for $j = 1$ to 2^{k_t} **do**
    Assign the j-th largest token $p_{(j)}$ to subset $V_{\pi(j-1)}$.
end for
for each remaining token v in $\{p_{(2^{k_t}+1)}, \ldots, p_{(N)}\}$ **do**
    Compute hash: $m \leftarrow H_{\text{seed}}(K, x_{<t}, v) \pmod{2^{k_t}}$.
    Assign token v to subset V_m.
end for
Set Green Set: $S^{(t)}_G \leftarrow V_{w_t}$ and Red Set: $S^{(t)}_R \leftarrow \mathcal{V} \setminus S^{(t)}_G$.
Calculate Green Set Probability: $P_G \leftarrow \sum_{i \in S^{(t)}_G} p^{(t)}_i$.
Calculate probability shift: $\Delta_t \leftarrow g(P_G) = \min\{\sqrt{\epsilon \cdot P_G(1-P_G)},\, 1-P_G\}$.
for each token $i \in \mathcal{V}$ **do**
    if $i \in S^{(t)}_G$ **then**
        $q^{(t)}_i \leftarrow p^{(t)}_i \left(1 + \frac{\Delta_t}{P_G}\right)$
    else
        $q^{(t)}_i \leftarrow p^{(t)}_i \left(1 - \frac{\Delta_t}{1-P_G}\right)$
    end if
end for
 # Sampling: Sample next token $x_t \sim \mathbf{q}^{(t)}$.
return x_t

E Generation and Detection Algorithms of BoundMark

Algorithm 1 details the generation procedure of BoundMark at a single step, and Algorithm 2 describes the detection procedure.

F Full Results of the Extended Evaluation

Tables 3–7 report the complete results of the extended evaluation, including the standard deviation of bit accuracy across users, BERTScore F1, ROUGE-L, and the perplexity of watermarked (WM) and unwatermarked (Non-WM) generations.

G Full Robustness Results

Table 8 reports bit accuracy under the three attacks considered in Section 4.5, averaged over four model–dataset settings, and Table 9 reports the per-setting results.

Algorithm 2 The BoundMark Detection Pipeline

Require: Suspect Document $\mathbf{x} = \{x_1, x_2, \ldots, x_T\}$, Secret Key K, Distortion Budget ϵ, Maximum Payload $k_{\max}$

Ensure: Extracted Global Message $\hat{M}$

Initialize an empty list of message candidates $C \leftarrow [\,]$.
for $t = 1$ to T **do**
  # LLM Forward Pass: Obtain original distribution $\mathbf{p}^{(t)}$ using context $x_{<t}$.
  Sort $\mathbf{p}^{(t)}$ in descending order: $p_{(1)} \geq p_{(2)} \cdots \geq p_{(N)}$.
  $D_{\min}(0) \leftarrow 0$
  for candidate $k = 1, 2, 3, \ldots, k_{\max}$ **do**
    Calculate bounds: $P^{(k)}_{G,\min} \leftarrow p_{(2^k)}$ and $P^{(k)}_{G,\max} \leftarrow 1 - \sum_{i=2}^{2^k} p_{(i)}$
    Calculate guaranteed detectability: $D_{\min}(k) \leftarrow k \cdot \min\left(g(P^{(k)}_{G,\min}),\, g(P^{(k)}_{G,\max})\right)$ {same g and ϵ as generation}
  end for
  Recover payload: $k_t \leftarrow \arg\max_{0 \leq k \leq k_{\max}} D_{\min}(k)$.
  if $k_t = 0$ **then**
    continue {abstained step carries no payload}
  end if
  Generate pseudo-random permutation π of $\{0, 1, \ldots, 2^{k_t} - 1\}$ seeded by $H_{\text{seed}}(K, x_{<t}, k_t)$.
  for $j = 1$ to 2^{k_t} **do**
    Assign the j-th largest token $p_{(j)}$ to subset $V_{\pi(j-1)}$.
  end for
  for each remaining token v in $\{p_{(2^{k_t}+1)}, \ldots, p_{(N)}\}$ **do**
    Compute hash: $m \leftarrow H_{\text{seed}}(K, x_{<t}, v) \pmod{2^{k_t}}$.
    Assign token v to subset V_m.
  end for
  Identify the subset index $\hat{w}_t$ such that the observed token $x_t \in V_{\hat{w}_t}$.
  Append $\hat{m}_t = \hat{w}_t$ to C.
end for
Calculate the mode of all extracted message candidates:
$\hat{M} \leftarrow \text{Mode}(C)$
return $\hat{M}$

Table 2: Extended evaluation on additional models and tasks. We report bit accuracy (BA, %) for all settings, and the downstream task performance of the watermarked output for GSM8K (answer accuracy, %) and HumanEval (Pass@1, %); the corresponding unwatermarked references are 40.00% and 33.54%, respectively. Best results per column are in bold. Standard deviations and additional quality metrics are provided in Appendix F.

	Qwen2.5-7B						**LLaMA-2-7B**
	CNN/DM	**C4**	**GSM8K**		**HumanEval**		**CNN/DM**
Method	BA↑	BA↑	BA↑	Acc.↑	BA↑	Pass@1↑	BA↑
XMark	87.50	93.75	83.75	30.00	**90.78**	**33.54**	88.75
LoSo	82.50	80.00	70.00	20.00	80.11	31.71	81.25
MPAC	92.50	96.25	86.25	30.00	86.13	31.10	90.00
RSBH	88.75	88.75	65.00	30.00	79.50	**33.54**	86.25
CycleShift	86.25	**100.00**	77.50	0.00	**90.78**	26.83	**91.25**
DepthW	70.00	76.25	52.50	**40.00**	63.57	**33.54**	67.50
StealthInk	48.75	47.50	50.00	**40.00**	50.76	**33.54**	71.25
BoundMark	**97.50**	98.75	**87.50**	**40.00**	87.50	**33.54**	90.00

Table 3: Extended results: Qwen2.5-7B on CNN/DailyMail.

Method	BA (%)↑	BERTScore↑	WM PPL↓	Non-WM PPL↓	ROUGE-L↑
XMark	87.50±14.79	0.8550	6.20	4.84	0.2000
LoSo	82.50±12.75	0.8533	5.22	4.84	0.1907
MPAC	92.50±12.75	0.8563	5.33	4.84	0.2030
RSBH	88.75±18.07	0.8555	7.13	4.84	0.1827
CycleShift	86.25±27.64	0.8582	5.50	4.84	0.2092
DepthW	70.00±16.96	0.8482	5.57	4.84	0.1800
StealthInk	48.75±19.72	0.8569	5.41	4.84	0.2420
BoundMark	97.50±5.00	0.8536	5.93	4.84	0.2017

Table 4: Extended results: Qwen2.5-7B on C4.

Method	BA (%)↑	BERTScore↑	WM PPL↓	Non-WM PPL↓	ROUGE-L↑
XMark	93.75±8.39	0.8695	7.92	6.20	0.2272
LoSo	80.00±13.92	0.8670	5.74	6.20	0.2230
MPAC	96.25±5.73	0.8551	7.84	6.20	0.1989
RSBH	88.75±14.20	0.8585	7.79	6.20	0.2094
CycleShift	100.00±0.00	0.8565	9.33	6.20	0.2131
DepthW	76.25±28.20	0.8541	6.43	6.20	0.1965
StealthInk	47.50±15.61	0.8544	5.40	6.20	0.2161
BoundMark	98.75±3.75	0.8564	8.22	6.20	0.1861

Table 5: Extended results: Qwen2.5-7B on GSM8K.

Method	BA (%)↑	WM Acc.↑	Non-WM Acc.	BERTScore↑	WM PPL↓	Non-WM PPL↓
XMark	83.75±14.84	30.00	40.00	0.8548	2.28	2.43
LoSo	70.00±10.00	20.00	40.00	0.8525	2.50	2.43
MPAC	86.25±10.38	30.00	40.00	0.8572	2.98	2.43
RSBH	65.00±27.84	30.00	40.00	0.8510	2.72	2.43
CycleShift	77.50±27.84	0.00	40.00	0.8569	2.69	2.43
DepthW	52.50±14.58	40.00	40.00	0.8564	2.00	2.43
StealthInk	50.00±15.81	40.00	40.00	0.8607	1.93	2.43
BoundMark	87.50±9.68	40.00	40.00	0.8566	2.67	2.43

Table 6: Extended results: Qwen2.5-7B on HumanEval.

Method	BA (%)↑	WM Pass@1↑	Non-WM Pass@1
XMark	90.78±11.62	33.54	33.54
LoSo	80.11±14.79	31.71	33.54
MPAC	86.13±13.46	31.10	33.54
RSBH	79.50±23.13	33.54	33.54
CycleShift	90.78±20.42	26.83	33.54
DepthW	63.57±24.90	33.54	33.54
StealthInk	50.76±17.33	33.54	33.54
BoundMark	87.50±12.88	33.54	33.54

Table 7: Extended results: LLaMA-2-7B on CNN/DailyMail.

Method	BA (%)↑	BERTScore↑	WM PPL↓	Non-WM PPL↓	ROUGE-L↑
XMark	88.75±13.05	0.8258	4.59	3.98	0.1530
LoSo	81.25±12.81	0.8278	4.27	3.98	0.1182
MPAC	90.00±9.35	0.8333	4.57	3.98	0.1705
RSBH	86.25±21.97	0.8301	4.17	3.98	0.1676
CycleShift	91.25±26.25	0.8391	4.81	3.98	0.1604
DepthW	67.50±27.50	0.8366	4.16	3.98	0.1502
StealthInk	71.25±17.72	0.8395	3.59	3.98	0.2007
BoundMark	90.00±19.20	0.8237	4.19	3.98	0.1379

Table 8: Bit accuracy (%) under text-modification attacks, averaged over the four model–dataset settings ({Qwen2.5-7B, LLaMA-2-7B} $\times$ {CNN/DailyMail, C4}). "Attacked avg." is the mean over the three attacks. Best results per column are in bold. Per-setting results are given in Table 9.

Method	**Clean**	**Insert/Delete**	**Synonym**	**DIPPER**	**Attacked avg.**
XMark	90.94	80.31	74.06	54.69	69.69
LoSo	80.94	76.56	75.62	**63.75**	71.98
MPAC	92.50	**87.50**	**80.31**	62.81	**76.88**
RSBH	86.25	74.38	69.06	57.81	67.08
CycleShift	94.38	85.31	80.00	61.25	75.52
DepthW	75.31	64.38	60.31	53.75	59.48
StealthInk	61.88	57.50	52.19	50.94	53.54
BoundMark	**95.62**	74.38	76.25	55.00	68.54

Table 9: Per-setting bit accuracy (%) under text-modification attacks. C: clean; I/D: insertion/deletion; Syn: synonym substitution; DIP: DIPPER paraphrasing.

	Qwen2.5-7B / CNN/DM				**Qwen2.5-7B / C4**				**LLaMA-2-7B / CNN/DM**				**LLaMA-2-7B / C4**			
Method	C	I/D	Syn	DIP	C	I/D	Syn	DIP	C	I/D	Syn	DIP	C	I/D	Syn	DIP
XMark	87.50	88.75	72.50	52.50	93.75	82.50	82.50	56.25	88.75	75.00	60.00	56.25	93.75	75.00	81.25	53.75
LoSo	82.50	76.25	71.25	60.00	80.00	72.50	75.00	65.00	81.25	78.75	76.25	67.50	80.00	78.75	80.00	62.50
MPAC	92.50	87.50	83.75	66.25	96.25	92.50	88.75	60.00	90.00	86.25	68.75	63.75	91.25	83.75	80.00	61.25
RSBH	88.75	75.00	66.25	47.50	88.75	82.50	78.75	55.00	86.25	70.00	68.75	65.00	81.25	70.00	62.50	63.75
CycleShift	86.25	71.25	75.00	52.50	100.00	88.75	86.25	62.50	91.25	92.50	75.00	57.50	100.00	88.75	83.75	72.50
DepthW	70.00	67.50	65.00	51.25	76.25	67.50	61.25	53.75	67.50	48.75	51.25	48.75	87.50	73.75	63.75	61.25
StealthInk	48.75	45.00	52.50	43.75	47.50	46.25	42.50	50.00	71.25	70.00	51.25	48.75	80.00	68.75	62.50	61.25
BoundMark	97.50	67.50	70.00	60.00	97.50	77.50	81.25	55.00	90.00	75.00	73.75	57.50	97.50	77.50	80.00	47.50

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: Our main contribution is a multi-bit LLM watermarking algorithm with explicitly controlled and theoretically bounded per-step distributional shift (KL divergence), with an adaptive payload. The abstract and introduction state this claim, clarify that the KL bound is not a semantic guarantee, and state that empirical performance is competitive rather than uniformly superior.

   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: We discuss the limitations in Section A, including the distributional (not semantic) nature of the bound, the heuristic detectability proxy, limited adversarial robustness, the evaluated sequence lengths and payload sizes, and security against forgery.

   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: The proofs of theoretical result are in the Appendix.

   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 use the code from Xu et al. [2026] for reproducibility.

   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: [Yes]

   Justification: We only use open-soure datasets and models in experiments. And we release our code.

   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: We describe all settings 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 report average performance over multiple group of experiments with std.

   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: We have given the details of compute resources 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: After checking the NeurIPS code of Ethics, we did not find any evidence to show our research violate it.

   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: We discuss it in abstract and introduction part.

    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: We do not have any high risk model or code for misuse.

    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: We only use open-source models and datasets.

    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: We release our code.

    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: We did not have experiments 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: We does not involve any 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: We use LLMs only for improving writing.

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.