RAMOMARK: MULTI-BIT WATERMARKING VIA RANDOMNESS-MODULATED SPECULATIVE SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Speculative sampling accelerates LLM inference through a draft-then-verify paradigm, where proposed tokens are accepted or rejected by comparing the probability ratio of target and draft models against a random threshold. In this paper, we identify this threshold as an intrinsic and controllable source of randomness, and show that multi-bit watermarks can be embedded into the generated text by modulating it—hence unifying efficient inference and output provenance within a single sampling framework. Specifically, we introduce a control variable that modulates the threshold, and construct a communication channel by treating this variable as the input and the generated token as the output. This construction naturally casts multi-bit watermarking as a channel coding problem. Drawing on coding theory, we propose **ra**ndomness-**mo**dulated speculative sampling with water**mark** (RamoMark), a keyless and robust watermarking algorithm that provably preserves the output distribution. As the codebook is constructed offline, RamoMark scales naturally to long-bitstream watermarking. Empirical evaluations confirm reliable watermark detection with negligible degradation in generation quality and superior performance over baseline methods.

1 INTRODUCTION

Sampling lies at the heart of large language models (LLMs) and increasingly serves purposes beyond controlling generation diversity. Recent studies have developed sampling-based methods to improve inference efficiency and establish output provenance. For example, speculative sampling (Chen et al., 2023; Leviathan et al., 2023; Sun et al., 2023) accelerates inference by using a lightweight draft model to propose multiple tokens, which are then verified in parallel by the larger target model; a carefully designed acceptance-rejection mechanism provably preserves the target model's output distribution. Meanwhile, many watermarking methods embed provenance information into generated text by controlling the token-sampling process (Hu et al., 2024; Kuditipudi et al., 2024; Dathathri et al., 2024; Boroujeny et al., 2024). For instance, Kirchenbauer et al. (2023) dynamically partitions the vocabulary into green and red lists at each decoding step and embeds a statistical watermark by increasing the sampling probabilities of tokens on the green list.

These developments naturally raises the question of whether a single sampling algorithm can simultaneously enable efficient inference and reliable watermarking. In fact, recent work has demonstrated that watermarking can be integrated with speculative sampling (Dathathri et al., 2024; Hu & Huang, 2024; He et al., 2026b; Liu et al., 2026). However, these approaches focus primarily on *zero-bit* detection, which determines only whether a watermark is present. Extending this integration to *multi-bit* watermarking remains nontrivial. Many multi-bit watermarking methods encode payload information by mapping the message bits or symbols to message-dependent token partitions or probability regions, and modifying token sampling accordingly (Wang et al., 2024; Yoo et al., 2024; Park et al., 2026b; Zhu et al., 2026; Xu et al., 2026). A naive combination with speculative sampling breaks down at the rejection step: when a draft token selected to carry a watermark signal is rejected, the corrective resample draws from the residual distribution, which is message-independent, discarding the embedded signal. Conversely, forcing the correction step to respect the partition would alter the target model's output distribution. Either way, naive integration fails to simultaneously preserve the message and the generation quality.

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

In this paper, we develop a framework that effectively integrates multi-bit watermarking with speculative sampling by modulating the random acceptance threshold, rather than directly controlling token selection. The key observation is that the output token—whether an accepted proposal or a resample from the residual distribution—has a distribution that depends on this threshold, giving us an indirect but effective handle on the output. Specifically, we introduce a binary variable A and a constant $\theta \in (0, 1)$ to control the sampling of the threshold u. If $A = 0$, we sample $u \sim \mathrm{Unif}[0, \theta]$; if $A = 1$, we sample $u \sim \mathrm{Unif}[\theta, 1]$. As a result, the distribution of the output token depends on the value of A. Meanwhile, the marginal distribution of u remains $\mathrm{Unif}[0, 1]$ if $A \sim \mathrm{Bernoulli}(1 - \theta)$; in this case the output distribution of speculative sampling is preserved.

At the core of our framework is a communication-theoretical perspective: we view the system as a communication channel, with A as the channel input and the output token as the channel output. This allows us to cast multi-bit watermarking as a channel coding problem—mapping a multi-bit message to a sequence of A's and recovering it from the resulting sequence of output tokens. Drawing on coding theory (Polyanskiy & Wu, 2025), we construct a constant-composition code (Graham & Sloane, 1980) that maps each message to an input sequence consisting of a fraction $1 - \theta$ of 1s and a fraction θ of 0s, ensuring that the empirical distribution of A follows $\mathrm{Bernoulli}(1 - \theta)$. We call this method randomness-modulated speculative sampling (RamoMark). Unlike existing methods, RamoMark embeds the multi-bit message using an offline-constructed code, which offers several merits. First, it not only preserves generation quality — since it preserves the output distribution — but also preserves generation efficiency, as it introduces negligible run-time overhead. Second, it is naturally scalable to long bitstream messages. Third, many existing watermarking methods rely on a private key to detect the presence of a watermark, whereas RamoMark requires no such key.

The contributions of this paper are summarized as follows. **(i)** We introduce the idea of embedding multi-bit information into speculative sampling by modulating its random acceptance threshold, and formulate the resulting problem as a channel coding problem. **(ii)** We develop RamoMark, which embeds multi-bit watermarks via an offline-constructed constant-composition code. **(iii)** We demonstrate through experiments that RamoMark achieves reliable multi-bit watermark detection while introducing only minimal degradation in generation quality.

2 Related Work

Speculative Sampling for LLMs. Speculative sampling (Sun et al., 2023; Chen et al., 2023; Leviathan et al., 2023) accelerates autoregressive generation without changing the target distribution. Instead of sampling every token directly from a large target model, a smaller draft model proposes candidate continuations, which are verified by the target model in parallel and then either accepted or corrected. Recent works further extend this paradigm through multi-draft decoding (Rowan et al., 2025; Khisti et al., 2025), online learning (Liu et al., 2023; Qian et al., 2026) and self-speculative decoding (Zhang et al., 2024; Xia et al., 2025).

LLM Watermarking. As LLM-generated text becomes increasingly fluent, it is becoming more difficult to distinguish from human-written content, making output watermarking important and potentially necessary (Anthropic, 2026). Early methods typically modify token probabilities, for example, by biasing sampling toward key-dependent token subsets (Kirchenbauer et al., 2023). In contrast, unbiased or distortion-free watermarking schemes use pseudorandomness (Christ & Gunn, 2024) to correlate the generated tokens with a watermark while preserving the original target distribution (Aaronson & Kirchner, 2023; Kuditipudi et al., 2024); however, these methods rely on the availability of a secret key, which is arguably less practical. SynthID (Dathathri et al., 2024) developed a scalable generative watermark and an algorithm that can be combined with speculative sampling. Recently, Hu & Huang (2024) identified a fundamental tradeoff between speculative sampling efficiency and watermark detectability, which was further improved by He et al. (2026b); Liu et al. (2026).

Most watermarking methods focus on *zero-bit* detection, whereas *multi-bit* watermarking encodes a payload that must be recovered from the generated text (Wang et al., 2024; Yoo et al., 2024). Recent schemes improve the robustness, stealthiness, or decoding efficiency of multi-bit watermarking (Jiang et al., 2025; Qu et al., 2025; Feng et al., 2025; Xu et al., 2026; Chao et al., 2026; Zhu et al., 2026; Huang et al., 2026; Wang et al., 2026).

Information and Coding Theory. Watermarking has a long history in information theory, where it is viewed as communicating hidden information under distortion and adversarial constraints (Cohen & Lapidoth, 2002; Moulin & O'Sullivan, 2003; Somekh-Baruch & Merhav, 2004; Wang & Moulin, 2008). More recently, information- and coding-theoretic perspectives have been developed for LLM watermarking, including practical algorithms (Tsur et al., 2025; Park et al., 2026a), capacity-oriented constructions (Gilani et al., 2026), and fundamental limits (He et al., 2026a; Guo et al., 2026). In this paper, we utilize constant-composition codes (Graham & Sloane, 1980; Agrell et al., 2000; Chee et al., 2010) to embed information. Multi-bit watermarking and steganography have also been studied through polar codes (Li et al., 2020) and minimum entropy coupling (Schroeder de Witt et al., 2023).

3 BACKGROUND

Speculative Sampling. Speculative sampling (Chen et al., 2023; Leviathan et al., 2023; Sun et al., 2023) is a distribution-preserving inference acceleration technique built on the principle of rejection sampling (MacKay et al., 2003). Rejection sampling produces a sample following the target distribution by drawing samples from a more accessible proposal distribution, and by selectively accepting or rejecting each sample. Speculative sampling instantiates this idea in the autoregressive LLM decoding setting, using a small draft model $\mathcal{M}_s$ as the proposal and a large target model $\mathcal{M}_l$ as the target.

Specifically, given a context $x_{:t} := (x_1, x_2, \ldots, x_t)$, we want to sample the next token from the target model, which follows the conditional distribution $\mathcal{M}_l(\cdot|x_{:t})$. Instead of querying $\mathcal{M}_l$ directly at each step, speculative sampling employs a smaller draft model in a four-stage procedure:

1. **Draft.** The draft model $\mathcal{M}_s$ autoregressively samples L candidate tokens $\tilde{x}_{t+1}, \ldots, \tilde{x}_{t+L}$, recording the draft conditional distribution $\mathcal{M}_s(\cdot\,|\,x_{:t}, \tilde{x}_{t+1:t+i-1})$ at positions $i = 1, \ldots, L$.
2. **Query.** In one parallel forward pass, the target model $\mathcal{M}_l$ evaluates the distributions
$$\mathcal{M}_l(\cdot\,|\,x_{:t}), \mathcal{M}_l(\cdot\,|\,x_{:t}, \tilde{x}_{t+1}), \ldots, \mathcal{M}_l(\cdot\,|\,x_{:t}, \tilde{x}_{t+1:t+L}).$$
3. **Verification.** The candidate tokens are evaluated sequentially. For each i, a uniform random variable $u_i \sim \mathrm{Unif}(0, 1)$ is sampled, and the corresponding candidate token is accepted if
$$u_i \leq \min\left\{1, \frac{\mathcal{M}_l(\cdot\,|\,x_{:t}, \tilde{x}_{t+1:t+i})}{\mathcal{M}_s(\cdot\,|\,x_{:t}, \tilde{x}_{t+1:t+i})}\right\}. \tag{1}$$
We write $x_{t+i} = \tilde{x}_{t+i}$ to denote an accepted candidate.
4. **Correction.** Upon the first rejection position k, a correction token x_{t+k} is sampled from the residual distribution defined as
$$P_{\mathrm{res}}(x) \triangleq \frac{\mathcal{M}_l(x|\,x_{:t+k-1}) - \min\{\mathcal{M}_l(x|\,x_{:t+k-1}), \mathcal{M}_s(x|\,x_{:t+k-1})\}}{1 - \sum_y \min\{\mathcal{M}_l(y|x_{:t+k-1}), \mathcal{M}_s(y|x_{:t+k-1})\}}. \tag{2}$$
The remaining candidate tokens are discarded and decoding proceeds to the next iteration.

The crucial property of this verification-correction mechanism is that the accepted and correction tokens follow *exactly the same distribution* as direct autoregressive sampling from the target model. In other words, we accelerate inference without changing the target model's output distribution.

In speculative sampling, each output token is drawn either from the draft model $\mathcal{M}_s$ or the residual distribution P_{res}, governed by the probability ratio of the two models and the random threshold u_i. As we will show, the randomness in u_i is a natural medium for embedding information into the generated tokens. Before introducing our method, we first briefly review the basics of channel coding.

Channel Coding. In information theory, a *channel with a state* is characterized by a tuple $(\mathcal{A}, \mathcal{Y}, \mathcal{S}, P_{Y,S'|S,A})$, where $\mathcal{A}$ is the input alphabet, $\mathcal{Y}$ is the output alphabet, $\mathcal{S}$ is the channel state alphabet, and $P_{Y,S'|S,A}$ is the channel law governing the joint distribution of the output Y and the next state S' given the current state S and input A.

Channel coding addresses the problem of reliably transmitting information through a given channel. Specifically, let M denote a message of B bits that is sampled uniformly from the message set

$\mathcal{B} := \{0, 1, \ldots, 2^B - 1\}$. To transmit message M over N channel uses, we need a *code* comprising an encoder and a decoder. The encoder is a sequence of mappings $\mathcal{E} \triangleq \{\mathcal{E}_t : 1 \le t \le N\}$, where

$$\mathcal{E}_t : \mathcal{B} \times \mathcal{S}^t \times \mathcal{A}^{t-1} \to \mathcal{A}$$

determines the channel input at time step t based on the message, the observed channel states, and the previous inputs. After collecting all N channel outputs, the message is reconstructed by the decoder $\mathcal{D} : \mathcal{Y}^N \to \mathcal{B}$. The integer N is referred to as the *blocklength* of the code. The decoding error probability is defined as

$$P_e^{(N)} \triangleq P\left(M \neq \mathcal{D}(Y^N)\right). \tag{3}$$

The multi-bit watermarking problem is naturally cast as a channel coding problem: we wish to recover a B-bit message from a sequence of tokens generated by the language model with minimal error probability. We formalize this connection in the next section.

4 Multi-Bit Watermark via Randomness Modulation

In this section, we first construct a communication channel within the speculative sampling framework and formulate multi-bit watermarking as a channel coding problem. Building on this formulation, we propose RamoMark, an efficient watermarking algorithm that draws on channel coding techniques. We also provide a theoretical guarantee that RamoMark preserves the target distribution over long generated sequences.

4.1 Channel Construction

Although speculative sampling drafts and verifies a sequence of candidate tokens in each iteration, the acceptance decision at each position follows the same token-level coupling mechanism, as illustrated in Figure 1a. It therefore suffices to focus on a single token: we describe the draft-and-verify procedure at one position, and show how to embed a watermark within it.

Algorithm 1: Token-level coupling with control bit

Input: Target and draft distributions P_h, Q_h; draft token $X \sim Q_h$; control bit $A \in \{0, 1\}$.
if $A = 0$ **:**
  Sample $U \sim \mathrm{Unif}[0, \theta]$
else
  Sample $U \sim \mathrm{Unif}[\theta, 1]$
if $U \le \alpha_h(X)$ **:**
  return $Y = X$ `// Accept`
return $Y \sim R_h$ `// Correction`

To simplify notation, conditioned on a fixed context h, let P_h and Q_h denote the target and draft next-token distributions, respectively. For a token x from vocabulary Ω, define $\alpha_h(x) := \min\left\{1, \frac{P_h(x)}{Q_h(x)}\right\}$ if $Q_h(x) > 0$, and 0 otherwise. Let $\rho_h := \sum_{x \in \Omega} Q_h(x)\alpha_h(x) < 1$ and the residual distribution is

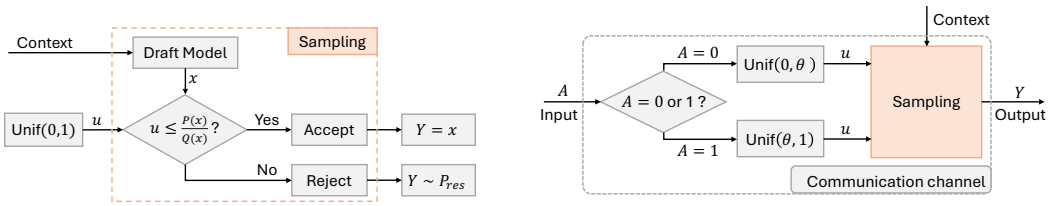
$$R_h(y) := \big(P_h(y) - \min\{P_h(y), Q_h(y)\}\big)/(1 - \rho_h) \qquad y \in \Omega. \tag{4}$$

The vanilla token-level speculative sampling draws a candidate $X \sim Q_h$ and an independent $U \sim \mathrm{Unif}[0, 1]$. The candidate is accepted if $U \le \alpha_h(X)$, in which case $Y = X$; otherwise, a correction token $Y \sim R_h$ is sampled. The key observation is that the distribution of the output Y depends on the value of U, which allows us to indirectly control the output distribution by controlling the sampling of U.

Specifically, we introduce a binary variable $A \in \{0, 1\}$. For a design parameter $\theta \in (0, 1)$, if $A = 0$, we sample $U \sim \mathrm{Unif}[0, \theta]$, whereas if $A = 1$, we sample $U \sim \mathrm{Unif}[\theta, 1]$. The choice of θ affects both sampling efficiency and watermarking performance, as we discuss later. For any fixed θ, the choice of A results in different distributions of Y, hence the output token Y carries information about A. This motivates defining a communication channel with input A, output Y, channel state h, and channel law $P_{Y,h'|A,h}$, as illustrated in Figure 1b.

For a context h, parameter $\theta \in (0, 1)$, and $a \in \{0, 1\}$, define

$$\beta_{h,\theta,a}(x) := \Pr\left(U \le \alpha_h(x) \mid A = a, h\right). \tag{5}$$



a Token-level coupling b Implicit communication channel

Figure 1: Illustrations of the speculative sampling and the constructed communication channel.

Then we can find

$$\beta_{h,\theta,0}(x) = \min\left\{1, \frac{\alpha_h(x)}{\theta}\right\}, \qquad \beta_{h,\theta,1}(x) = \frac{(\alpha_h(x) - \theta)_+}{1-\theta}. \tag{6}$$

Consequently, the induced token channel from A to Y is

$$K_{h,\theta}(y \mid a) = Q_h(y)\beta_{h,\theta,a}(y) + \Big(1 - \sum_{x\in\Omega} Q_h(x)\beta_{h,\theta,a}(x)\Big) R_h(y), \qquad y \in \Omega. \tag{7}$$

With this channel construction, a B-bit message can be mapped by an encoder to a sequence of binary inputs $A_1, A_2, \ldots, A_n$, which drives the sampling of successive tokens $Y_1, Y_2, \ldots, Y_n$; a decoder could recover the message from the output sequence. The decoding error probability $P_e^{(n)}$ is the error probability of watermark detection. When the input A follows a proper Bernoulli distribution, the marginal distribution of u at each step remains $\mathrm{Unif}[0,1]$, making the stochastic property of speculative sampling unchanged. This is formally stated in the following theorem.

Theorem 4.1 (Unbiasedness). *Fix a context h, a parameter $\theta \in (0,1)$, and target and draft distributions P_h and Q_h. Let $K_{h,\theta}(\cdot \mid a)$ be defined by* (7)*. If the binary bit A is sampled from* $\mathrm{Bernoulli}(1-\theta)$*, then for every $y \in \Omega$,*

$$\theta K_{h,\theta}(y \mid 0) + (1-\theta) K_{h,\theta}(y \mid 1) = P_h(y),$$

and hence the next-token marginal is exactly the target distribution.

The proof is in Appendix A. Theorem 4.1 motivates our use of a constant-composition codebook to communicate over this channel. Specifically, we choose each codeword so that approximately a fraction θ of its control bits are 0 and a fraction $1-\theta$ are 1. Thus, over a long sequence, the empirical input mixture matches the unbiased mixture $(\theta, 1-\theta)$. Intuitively, the effects of the two conditional samplers therefore average out, making the overall sampling "approximately" unbiased. We hence decide to utilize the constant-composition code, which is elaborated as follows.

Remark 4.2. *Parameter θ controls both the strength of the watermark and the speculative efficiency. Increasing θ makes the conditional acceptance thresholds more stringent, especially when $A = 1$, and can therefore reduce the empirical acceptance rate. This motivates our choice of $\theta = 0.75$ over more extreme values such as $\theta = 0.95$. On the other hand, moving θ below $1/2$ makes the codeword composition more imbalanced and reduces the achievable communication rate, since $I_{h,\theta}(A;Y) \le H_2(\theta)$ where $H_2(\cdot)$ is the binary entropy function. Thus θ balances speculative efficiency, and coding rate; and hence, the overall performance.*

4.2 Coding Scheme for Watermark Embedding

We now propose a coding scheme for the channel constructed previously. The induced channel is state-dependent, since the channel law depends on the context h. Nevertheless, we deliberately separate encoding and decoding: the encoder uses a context-independent constant-composition codebook, while the decoder exploits the realized context-dependent channel law through maximum-likelihood decoding. We leave the design of a context-dependent codebook for future work. We note that our detector does not need to know the original users' prompts for multi-bit watermark detection.

Encoder. Recall that B denotes the number of payload bits and n denotes the blocklength. We construct a codebook independently of the channel state. We map a message $M \in \mathcal{B}$ to a codeword

$$a(M) = \big(a_1(M), \ldots, a_n(M)\big) \in \mathcal{C} = \{a(0), \ldots, a(2^B - 1)\} \subseteq \{0,1\}^n. \tag{8}$$

Every codeword has the same composition,

$$\big|\{i : a_i(m) = 0\}\big| = w_0, \qquad \big|\{i : a_i(m) = 1\}\big| = w_1 = n - w_0. \tag{9}$$

where we choose $w_0/n = \theta$ whenever θn is an integer. Thus, each codeword contains exactly a fraction θ of zeros and a fraction $1-\theta$ of ones, enabling the unbiasedness in Theorem 4.1. This is usually referred to as the *constant-composition code* in classical coding theory (Graham & Sloane, 1980; Agrell et al., 2000; Chee et al., 2010).

We construct the nonlinear codebook offline using a greedy max-min Hamming-distance criterion, which suffices for the payload sizes considered in this work. Given the selected codeword $a(M)$, its i-th symbol $a_i(M)$ controls the threshold distribution at the i-th watermark-bearing token.

Decoder. For channels with state, knowledge of the channel state at the decoder is typically beneficial for decoding accuracy. In our setting, this requires the watermark detector to know the context h, which includes the original prompt x used to generate the output tokens. When this is available, we can directly apply a likelihood decoder to recover the multi-bit message as follows:

$$\widehat{m} = \arg\max_{m \in \mathcal{B}} \sum_{i=1}^{n} \log K_{h_i,\theta}\left(y_i \mid a_i(m)\right), \tag{10}$$

where y_i is the i-th watermark-bearing token and $h_i = (x, y_{<1}, y_{1:i-1})$ is the context at this step: the prompt x, the tokens $y_{<1}$ generated before this coding block, and the tokens $y_{1:i-1}$ generated earlier within this coding block.

In practice, however, the watermark detector usually does not know the original prompt x used to generate the text, and hence does not have full access to the context h_i at each time step. In this case, we instead use $\bar{h}_i = (y_{<1}, y_{1:i-1})$ as an approximate context and apply the maximum-likelihood decoding based on $\bar{h}_i$ in place of h_i. This introduces a model mismatch: the true channel law is $K_{h_i,\theta}$, whereas the decoder relies on an estimated channel model $K_{\bar{h}_i,\theta}$. Specifically, given a message m and a codeword $a(m)$, the distribution of the output token y_i is

$$K_{h_i,\theta}(y_i|a_i(m)) = P_m(y_i|x, y_{<1}, y_{1:i-1}).$$

Without knowledge of x, we instead use the estimated model

$$K_{\bar{h}_i,\theta}(y_i|a_i(m)) = P_m(y_i|y_{<1}, y_{1:i-1}).$$

This approximation is well justified in practice: for LLMs, the influence of the prompt x on the next-token distribution is largely absorbed into the already-generated context, hene we usually have $K_{\bar{h}_i,\theta}(y_i|a_i(m)) \approx K_{h_i,\theta}(y_i|a_i(m))$ when $y_{<1}$ is reasonably long. In practice, this length requirement can be satisfied in either of two ways: by deliberately generating an unwatermarked prefix of sufficient length before embedding the first coding block, or, when decoding the second and subsequent coding blocks, by using the preceding coding block itself as the context $y_{<1}$.

5 Experimental Evaluation

In this section, we evaluate our scheme from two perspectives. First, we assess its multi-bit watermark embedding capacity, comparing it with the baselines (Yoo et al., 2024; Xu et al., 2026) to demonstrate it achieves state-of-the-art performance. Second, we reduce our scheme to the single-bit (i.e., zero-bit) watermarking setting, which allows us to compare with — and potentially improve upon — the sampling–watermark tradeoff recently identified in this setting (Hu & Huang, 2024; He et al., 2026b).

5.1 Multi-Bit Watermarking

Comparison with multi-bit baselines. We first evaluate the multi-bit watermarking performance of RamoMark in two target–draft model pairs, Qwen2.5-7B–Qwen2.5-0.5B and Vicuna-7B–Vicuna-68M, and embed messages of four and eight bits. The message recovery ratios of four-bit watermarks in different dataset are reported in Figure 2. RamoMark achieves high detection accuracy

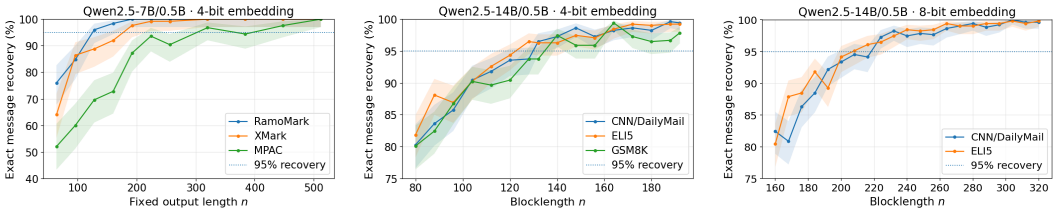


Figure 2: 4-bit and 8-bit recovery with $N = 512$ prompts per point, on Qwen-7B model and Qwen-14B model. Shading gives pointwise 95% Wilson intervals.

We then compare RamoMark with MPAC (Yoo et al., 2024) and XMark (Xu et al., 2026) on CNN/DailyMail. For both baselines we use their autoregressive variants, denoted MPAC and XMark. The comparison uses fixed payload lengths and a common target–draft vocabulary convention; Figure 3 reports exact whole-message recovery as a function of the watermark blocklength.

Across all four model–payload settings, RamoMark achieves substantially higher recovery than the two baselines at representative high-recovery operating points. For Qwen with a four-bit payload, at $n = 128$ RamoMark achieves 98.84% exact recovery, compared with 88.80% for XMark and 69.60% for MPAC. For eight bits, at $n = 192$ RamoMark achieves 96.10%, compared with 63.93% and 34.43%, respectively. The gap is also pronounced for the more challenging Vicuna pair: at $n = 224$, four-bit recovery is 96.43% for RamoMark, compared with 78.57% for XMark and 43.75% for MPAC; at $n = 512$, eight-bit recovery is 95.89%, compared with 66.00% and 29.00%, respectively.

Using the first tested empirical crossing as a descriptive summary, RamoMark reaches the 95% recovery thresholds at $n = 128$ for Qwen 4-bit, 192 for Qwen 8-bit, 224 for Vicuna 4-bit, and 512 for Vicuna 8-bit. For Qwen, the corresponding XMark crossings are 192 and 384, while MPAC requires 320 for 4 bits and does not reach 90% within the 8-bit range through $n = 512$. For Vicuna, XMark and MPAC do not reach 95% through; for eight bits, neither baseline reaches 95% through $n = 768$.

These first-crossing values are empirical scan statistics rather than estimates of a minimum required blocklength or coding capacity. In particular, different blocklengths use independently constructed codebooks, so recovery is not constrained to be monotone in n. A more conservative picture is obtained from the pointwise Wilson lower bounds. See Figure 2 and Figure 3 for details.

Moreover, we use Table 1 to show that, at representative high-recovery operating points, RamoMark achieves approximately 95–97% exact message recovery while maintaining sampling efficiency and target-model likelihood close to those of matched vanilla speculative sampling. The mean AATPS ratios are all close to 100%, and their pointwise paired 95% confidence intervals all contain 100%. Likewise, the observed raw-NLL differences are small, with all corresponding confidence intervals containing zero. Thus, at these operating points, we do not observe a statistically resolved degradation in either AATPS or raw target likelihood.

Summary. The experiments show that using approximate context retains the main multi-bit advantage of RamoMark while removing the detector's dependence on the original prompt. In the tested settings, RamoMark achieves approximately 95% or higher whole-message recovery at blocklengths where XMark and MPAC remain substantially below that level, with particularly large gaps for eight-bit payloads and for the Vicuna model pair.

5.2 Sampling-Watermarking Tradeoff

Recently, a fundamental tradeoff between speculative sampling efficiency and watermark strength has been discovered by (Hu & Huang, 2024). It has been shown that one cannot simultaneously achieve the best possible sampling efficiency and watermark strength within a single scheme. This tradeoff is for "zero-bit" watermarking, i.e., one always attempts to watermark the generated text and the detector aims to successfully detect the watermark, while the tradeoff frontier has been improved and characterized in some cases (Hu & Huang, 2024; He et al., 2026b; Liu et al., 2026).

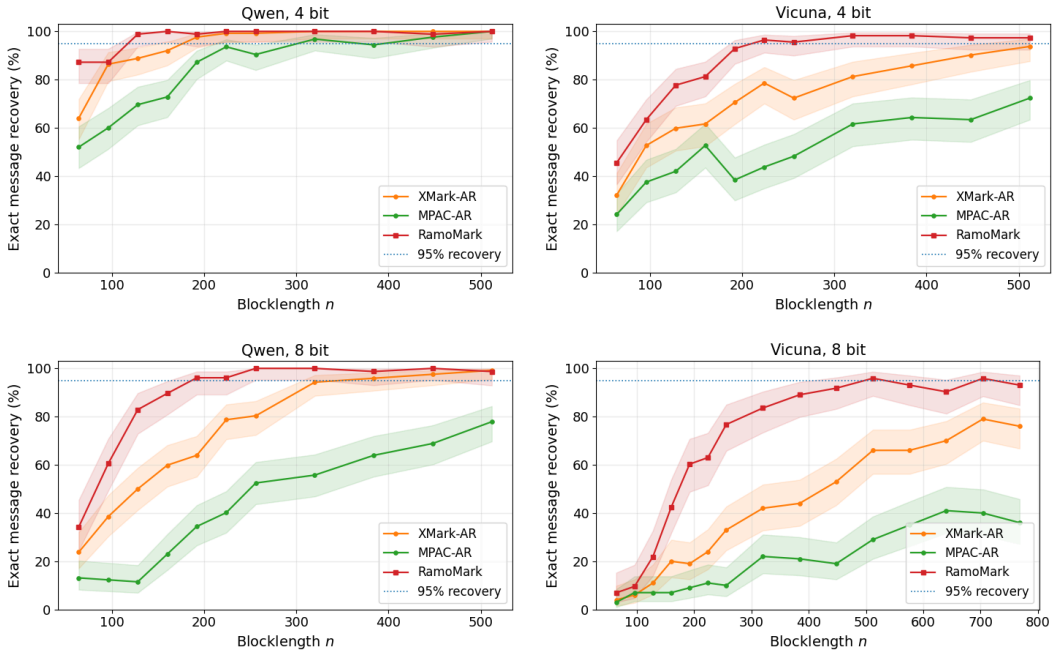


Figure 3: Message recovery for 4-bit (top) and 8-bit (bottom) watermarks on CNN/DailyMail. RamoMark does not require the original user prompt. The current RamoMark cohorts and XMark/MPAC baseline cohorts contain $N = 100$ prmpts per tested blocklength. Shading denotes pointwise 95% Wilson intervals.

Table 1: RamoMark-versus-vanilla diagnostics at representative high-recovery operating points. AATPS is the per-prompt RamoMark/vanilla ratio, and Δ raw NLL is RamoMark minus vanilla in nats/token, so positive values indicate lower target likelihood. Brackets are pointwise paired 95% percentile-bootstrap intervals. All AATPS intervals contain 100%, and all raw-NLL intervals contain zero.

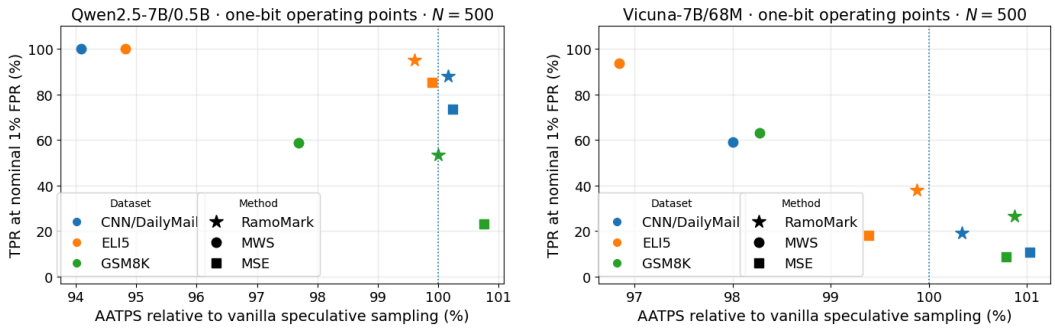
Cell	Exact rec.	AATPS / vanilla	Δ raw NLL
Qwen 4-bit	96.00%	101.61% [99.50,103.72]	+0.0065 [-0.0252,+0.0380]
Qwen 8-bit	95.08%	100.86% [99.19,102.59]	-0.0234 [-0.0497,+0.0030]
Vicuna 4-bit	97.32%	100.20% [98.51,101.86]	+0.0249 [-0.0039,+0.0541]
Vicuna 8-bit	97.00%	98.73% [97.04,100.48]	-0.0036 [-0.0349,+0.0279]

In this part, we show that our scheme can be used to further improve the tradeoff frontier. Specifically, although our scheme is designed for multi-bit watermarking in the first place, it can be reduced to zero-bit watermarking by embedding only one bit, with our detector aiming to decode this message.

For this part, each data point uses 500 prompts, with the matched vanilla speculative-sampling outputs serving as null samples. We compare against two methods proposed by (Hu & Huang, 2024): Maintain Sampling Efficiency (MSE) and Maintain Watermark Strength (MWS), both of which use their original U-score detector, while we continue to use our likelihood-based detector. We report the true positive rate (TPR) at a desired false positive rate (FPR) of 1%. We note that the main objective of this subsection is to investigate our scheme on the tradeoff frontier. Hence, we use Average Accepted Tokens Per Step (AATPS) as the measure of sampling efficiency, consistent with (Hu & Huang, 2024; Liu et al., 2026), rather than the wall-clock time.

5.3 Output Quality and Uncertainty

We use processed log-perplexity in natural-log units, $\mathrm{LPPL} = -\frac{1}{T}\sum_{i=1}^{T} \ln \widetilde{P}_{h_i}(Y_i)$, where $\widetilde{P}$ is the processed target distribution used by the evaluator. Lower LPPL indicates higher target likelihood,



(a) Qwen2.5-7B/Qwen2.5-0.5B. (b) Vicuna-7B/Vicuna-68M.

Figure 4: Fixed-one-bit watermark–sampling frontiers at $n = 128$ and $N = 500$. The horizontal coordinate is AATPS relative to matched vanilla speculative sampling and the vertical coordinate is TPR at 1% FPR. Each dataset curve connects MWS, Ours, and MSE.

Table 2: Fixed-one-bit output-quality comparison ($n = 128$ with $N = 500$ prompts). Each ΔLPPL is measured relative to vanilla speculative sampling in the same model–dataset cell. Negative values should be interpreted as sampling variability rather than as a claimed quality improvement.

Model pair	Data	Vanilla LPPL	ΔMSE	ΔMWS	ΔOurs	Raw PPL (van./ours)
Qwen2.5-7B/0.5B	CNN	0.8377	+0.0095	+0.0120	-0.0019	2.351/2.346
Qwen2.5-7B/0.5B	ELI5	0.8644	-0.0097	-0.0231	-0.0110	2.415/2.389
Qwen2.5-7B/0.5B	GSM8K	0.3386	+0.0016	+0.0069	+0.0048	1.412/1.418
Vicuna-7B/68M	CNN	0.4165	+0.0022	-0.0152	+0.0006	1.538/1.540
Vicuna-7B/68M	ELI5	0.6488	+0.0212	+0.0263	+0.0209	1.948/1.994
Vicuna-7B/68M	GSM8K	0.3960	+0.0103	+0.0084	+0.0017	1.502/1.503

but does not by itself imply improved correctness or semantic quality. The raw-PPL column reports the average of per-sequence raw-target perplexities and should not be reconstructed from the mean processed LPPL.

Across the six model–dataset cells, RamoMark's ΔLPPL relative to matched vanilla speculative sampling ranges from -0.0110 to $+0.0209$. Using 4,000 paired prompt-bootstrap replicates, the largest increase occurs for Vicuna/ELI5, with $\Delta\text{LPPL} = +0.0209$ and a pointwise 95% interval of approximately $[+0.0005, +0.0409]$ nats/token; the other five intervals contain zero. These exploratory intervals are not multiplicity-adjusted and should not be interpreted as equivalence guarantees. At the same time, the raw-PPL, repetition, distinct-2, and top-1 diagnostics remain close to their matched vanilla values, and negative ΔLPPL values are not interpreted as genuine quality improvements. The complete quality results are provided in Appendix B, with the changes in processed LPPL shown in Figure 5.

6 Conclusion

This paper studied multi-bit watermarking for LLM generation via speculative sampling. We identified the random acceptance threshold u in speculative sampling as an effective handle for embedding information into the generated tokens. Building on this idea, we introduced a binary variable A to control the sampling of u, thereby constructing a communication channel with A as channel input and the output token as channel output. This construction naturally casts multi-bit watermarking as a channel coding problem. Leveraging techniques from coding theory, we proposed RamoMark, which embeds multi-bit information into the generated tokens using an offline-constructed channel code. We showed that RamoMark preserves the output distribution of speculative sampling and introduces negligible run-time overhead, so it can watermark LLM generation without degrading generation quality or efficiency. Extensive experiments confirmed that RamoMark achieves reliable multi-bit watermark detection with only minimal impact on generation quality.

DISCLOSURE OF USING AI

The authors acknowledge the use of ChatGPT and Claud to assist with specific aspects of this research, as detailed below.

- **Research Design / Formulation (e.g., Proofs / Synthetic Data):** No AI tools were used for the core mathematical formulation or synthetic data generation.
- **Methodology and Experiments:** No AI tools were used to develop the methodology or design the experiments.
- **Writing, Editing, and Coding Assistance:** The authors drafted the manuscript independently and subsequently used ChatGPT to improve the clarity and presentation of the text. In addition, Claude was used to assist with implementing the Python code for the experiments. The authors reviewed and verified the correctness of the implementation.

The authors have reviewed and verified all AI-assisted outputs and take full responsibility for the integrity and accuracy of this work.

REFERENCES

Scott Aaronson and H Kirchner. Watermarking of large language models. In *Large Language Models and Transformers Workshop at Simons Institute for the Theory of Computing*, 2023.

Erik Agrell, Alexander Vardy, and Kenneth Zeger. Upper bounds for constant-weight codes. *IEEE Transactions on Information Theory*, 46(7):2373–2395, 2000.

Anthropic. How claude's text watermark works. `https://www.anthropic.com/news/claude-text-watermark`, August 2026.

Massieh Kordi Boroujeny, Ya Jiang, Kai Zeng, and Brian Mark. Multi-bit distortion-free watermarking for large language models. *arXiv preprint arXiv:2402.16578*, 2024.

Patrick Chao, Yan Sun, Edgar Dobriban, and Hamed Hassani. Watermarking language models with error correcting codes. *Transactions on Machine Learning Research*, 2026.

Yeow Meng Chee, Son Hoang Dau, Alan C. H. Ling, and San Ling. Linear size optimal q-ary constant-weight codes and constant-composition codes. *IEEE Transactions on Information Theory*, 56(1):140–151, 2010.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

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

Aaron S Cohen and Amos Lapidoth. The gaussian watermarking game. *IEEE transactions on Information Theory*, 48(6):1639–1667, 2002.

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

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. BiMark: Unbiased multilayer watermarking for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 17049–17067, 2025.

Atefeh Gilani, Sajani Vithana, Carol Xuan Long, Oliver Kosut, Lalitha Sankar, and Flavio P Calmon. Arcmark: Distortion-free multi-byte LLM watermark via optimal transport. *arXiv preprint arXiv:2602.07235*, 2026.

Ronald L. Graham and Neil J. A. Sloane. Lower bounds for constant weight codes. *IEEE Transactions on Information Theory*, 26(1):37–43, 1980.

Sidong Guo, Tyler Kann, Teodora Baluta, and Matthieu R. Bloch. Covert multi-bit LLM watermarking: An information theory and coding approach. In *2026 IEEE Information Theory Workshop (ITW)*, pp. 1–6. IEEE, 2026.

Haiyun He, Yepeng Liu, Zhuoer Shen, Ziqiao Wang, Yongyi Mao, and Yuheng Bu. Fundamental trade-offs in multi-bit watermarking of stochastic processes. *arXiv preprint arXiv:2605.08826*, 2026a.

Weiqing He, Xiang Li, Li Shen, Weijie Su, and Qi Long. Improving the trade-off between watermark strength and speculative sampling efficiency for language models. In *International Conference on Learning Representations (ICLR)*, 2026b.

Zhengmian Hu and Heng Huang. Inevitable trade-off between watermark strength and speculative sampling efficiency for language models. *Advances in Neural Information Processing Systems*, 37: 55370–55402, 2024.

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

Yu-Shin Huang, Chao Tian, and Krishna Narayanan. Optimal multi-bit generative watermarking schemes under worst-case false-alarm constraints. *arXiv preprint arXiv:2604.08759*, 2026.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. StealthInk: A multi-bit and stealthy watermark for large language models. In *International Conference on Machine Learning (ICML)*, volume 267, pp. 27685–27709, 2025.

Ashish Khisti, M Reza Ebrahimi, Hassan Dbouk, Arash Behboodi, Roland Memisevic, and Christos Louizos. Multi-draft speculative sampling: Canonical decomposition and theoretical limits. In *International Conference on Learning Representations (ICLR)*, 2025.

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

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning (ICML)*, pp. 19274–19286. PMLR, 2023.

Weixiang Li, Weiming Zhang, Li Li, Hang Zhou, and Nenghai Yu. Designing near-optimal steganographic codes in practice based on polar codes. *IEEE Transactions on Communications*, 68(7): 3948–3962, 2020.

Xiaoxuan Liu, Lanxiang Hu, Peter Bailis, Alvin Cheung, Zhijie Deng, Ion Stoica, and Hao Zhang. Online speculative decoding. *arXiv preprint arXiv:2310.07177*, 2023.

Yanxiao Liu, Sicheng Wan, Zhan Gao, and Deniz Gündüz. Watermarkable multi-draft speculative sampling via poisson processes. *arXiv preprint arXiv:2609.21858*, 2026.

David JC MacKay, David JC Mac Kay, et al. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

Pierre Moulin and Joseph A O'Sullivan. Information-theoretic analysis of information hiding. *IEEE Transactions on information theory*, 49(3):563–593, 2003.

Gang-Hyun Park, Ju-Hyeong Lee, Hee-Youl Kwak, and Dae-Young Yun. Weavemark: Robust and scalable multi-bit LLM watermarking via coded payload spreading. *arXiv preprint arXiv:2609.02177*, 2026a.

Shinwoo Park, Hyejin Park, Hyeseon Ahn, and Yo-Sub Han. WaterMod: Modular token-rank partitioning for probability-balanced LLM watermarking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32683–32691, 2026b.

Yury Polyanskiy and Yihong Wu. *Information theory: From coding to learning*. Cambridge university press, 2025.

Yu-Yang Qian, Hao-Cong Wu, Yichao Fu, Hao Zhang, and Peng Zhao. When drafts evolve: Speculative decoding meets online learning. *arXiv preprint arXiv:2603.12617*, 2026.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. Provably robust multi-bit watermarking for {AI-generated} text. In *34th USENIX Security Symposium (USENIX Security 25)*, pp. 201–220, 2025.

Joseph Rowan, Buu Phan, and Ashish Khisti. List-level distribution coupling with applications to speculative decoding and lossy compression. *Advances in Neural Information Processing Systems*, 38, 2025.

Christian Schroeder de Witt, Samuel Sokota, J. Zico Kolter, Jakob Nicolaus Foerster, and Martin Strohmeier. Perfectly secure steganography using minimum entropy coupling. In *International Conference on Learning Representations (ICLR)*, 2023.

Anelia Somekh-Baruch and Neri Merhav. On the capacity game of public watermarking systems. *IEEE Transactions on Information Theory*, 50(3):511–524, 2004.

Ziteng Sun, Ananda Theertha Suresh, Jae Hun Ro, Ahmad Beirami, Himanshu Jain, and Felix Yu. Spectr: Fast speculative decoding via optimal transport. *Advances in Neural Information Processing Systems*, 36:30222–30242, 2023.

Dor Tsur, Carol Xuan Long, Claudio Mayrink Verdun, Sajani Vithana, Hsiang Hsu, Richard Chen, Haim H. Permuter, and Flavio P. Calmon. Heavywater and simplexwater: Distortion-free LLM watermarks for low-entropy distributions. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38, 2025.

Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. In *International Conference on Learning Representations*, volume 2024, pp. 39375–39401, 2024.

Yidan Wang, Yubing Ren, Yanan Cao, and Li Guo. Worldcup sampling for multi-bit LLM watermarking. *arXiv preprint arXiv:2602.01752*, 2026.

Ying Wang and Pierre Moulin. Perfectly secure steganography: Capacity, error exponents, and code constructions. *IEEE Transactions on Information Theory*, 54(6):2706–2722, 2008.

Heming Xia, Yongqi Li, Jun Zhang, Cunxiao Du, and Wenjie Li. SWIFT: On-the-fly self-speculative decoding for LLM inference acceleration. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=EKJhH5D5wA`.

Jiahao Xu, Rui Hu, Olivera Kotevska, and Zikai Zhang. Xmark: Reliable multi-bit watermarking for llm-generated texts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14747–14763, 2026.

KiYoon Yoo, Wonhyuk Ahn, and Nojun 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)*, pp. 4031–4055, 2024.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. Draft & verify: Lossless large language model acceleration via self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11263–11282. Association for Computational Linguistics, 2024.

Junlin Zhu, Baizhou Huang, and Xiaojun Wan. Quantilemark: A message-symmetric multi-bit watermark for llms. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6790–6806, 2026.

A Proof of Theorem 4.1

Proof. For any $\alpha \in [0,1]$, the definition of $\beta_{h,t,a}(x)$ satisfies the pointwise identity

$$t \min\left\{1, \frac{\alpha}{t}\right\} + (1-t)\frac{(\alpha - t)_+}{1-t} = \alpha. \tag{11}$$

Indeed, if $\alpha < t$, the left-hand side equals $t(\alpha/t) = \alpha$; if $\alpha \geq t$, it equals $t + (\alpha - t) = \alpha$. Applying (11) with $\alpha = \alpha_h(x)$ gives

$$t\beta_{h,t,0}(x) + (1-t)\beta_{h,t,1}(x) = \alpha_h(x). \tag{12}$$

Therefore,

$$\begin{aligned} tc_{h,t,0} + (1-t)c_{h,t,1} &= \sum_{x\in\Omega} Q_h(x)\left[t\beta_{h,t,0}(x) + (1-t)\beta_{h,t,1}(x)\right] \\ &= \sum_{x\in\Omega} Q_h(x)\alpha_h(x) = \rho_h. \end{aligned} \tag{13}$$

Using (7) and (13),

$$\begin{aligned} &tK_{h,t}(y \mid 0) + (1-t)K_{h,t}(y \mid 1) \\ &\quad = Q_h(y)\left[t\beta_{h,t,0}(y) + (1-t)\beta_{h,t,1}(y)\right] + (1-\rho_h)R_h(y) \\ &\quad = Q_h(y)\alpha_h(y) + (1-\rho_h)R_h(y). \end{aligned} \tag{14}$$

By definition, $Q_h(y)\alpha_h(y) = \min\{P_h(y), Q_h(y)\}$. If $\rho_h < 1$, the definition of R_h in (4) yields

$$\begin{aligned} Q_h(y)\alpha_h(y) + (1-\rho_h)R_h(y) &= \min\{P_h(y), Q_h(y)\} + P_h(y) - \min\{P_h(y), Q_h(y)\} \\ &= P_h(y). \end{aligned} \tag{15}$$

If $\rho_h = 1$, then $P_h = Q_h$, $\alpha_h(X) = 1$ almost surely under Q_h, and the sampler accepts almost surely, so the same conclusion holds. □

B Experimental Results

The experiments use CNN/DailyMail, ELI5, and GSM8K. Unless otherwise stated, we use temperature 1, top-$k = 50$, top-$p = 1$, speculative lookahead $L = 4$, and threshold $t = 0.75$. The multi-bit experiments use the Qwen2.5-7B–Qwen2.5-0.5B and Vicuna-7B–Vicuna-68M target–draft pairs.

For RamoMark, the multi-bit experiments follow the self-prompt protocol described in the main text. The maximum-likelihood decoder scores the embedded message under the same reconstructed self-prompt context and does not use the original user prompt. In the fixed-length baseline comparison, the reported blocklength n refers to the watermark-bearing payload block; the self-prompt is additional visible context. A trial is counted as a successful recovery only when the exact message is decoded, the likelihood replay is valid, and the true-message likelihood is finite; invalid replays are conservatively counted as failures. Pointwise recovery bands are two-sided 95% Wilson intervals.

The fixed-one-bit experiments use $n = 128$ and $N = 500$ prompts per model–dataset cell. Matched vanilla speculative-sampling outputs are used as the null/control samples, and AATPS is reported relative to the matched vanilla run in the same cell.

To keep the appendix aligned with the current experimental protocol, we report only supplementary diagnostics that directly support the experiments in Section 5. In particular, we include the numerical values underlying the main multi-bit baseline-comparison curves, while omitting auxiliary decoder analyses and other legacy results that are not needed to interpret the current method.

B.1 Multi-Bit Baseline-Comparison Data

Tables 3 and 4 report the exact message-recovery values underlying Figure 3. The RamoMark entries use the self-prompt protocol described in the main text. XMark and MPAC use the corresponding fixed-length baseline cohorts.

Table 3: Exact message recovery (%) underlying the four-bit panels of Figure 3. For RamoMark, the number in parentheses is the number of available self-prompt trials at that blocklength. Each data point is from at least $N = 100$ trials.

	Qwen2.5-7B/0.5B			Vicuna-7B/68M		
n	RamoMark (N)	XMark	MPAC	RamoMark (N)	XMark	MPAC
64	87.21	64.00	52.00	45.54	32.14	24.11
96	87.21	86.40	60.00	63.39	52.68	37.50
128	98.84	88.80	69.60	77.68	59.82	41.96
160	100.00	92.00	72.80	81.25	61.61	52.68
192	98.84	97.60	87.20	92.86	70.54	38.39
224	100.00	99.20	93.60	96.43	78.57	43.75
256	100.00	99.20	90.40	95.54	72.32	48.21
320	100.00	100.00	96.80	98.21	81.25	61.61
384	100.00	100.00	94.40	98.21	85.71	64.29
448	98.82	100.00	97.60	97.32	90.18	63.39
512	100.00	100.00	100.00	97.32	93.75	72.32

Table 4: Exact message recovery (%) underlying the four-bit panels of Figure 3. For RamoMark, the number in parentheses is the number of available self-prompt trials at that blocklength. Each data point is from at least $N = 100$ trials

	Qwen2.5-7B/0.5B			Vicuna-7B/68M		
n	RamoMark (N)	XMark	MPAC	RamoMark (N)	XMark	MPAC
64	34.21	23.77	13.11	6.94	4.00	3.00
96	60.53	38.52	12.30	9.59	6.00	7.00
128	82.89	50.00	11.48	21.92	11.00	7.00
160	89.61	59.84	22.95	42.47	20.00	7.00
192	96.10	63.93	34.43	60.27	19.00	9.00
224	96.10	78.69	40.16	63.01	24.00	11.00
256	100.00	80.33	52.46	76.71	33.00	10.00
320	100.00	94.26	55.74	83.56	42.00	22.00
384	98.70	95.90	63.93	89.04	44.00	21.00
448	100.00	97.54	68.85	91.78	53.00	19.00
512	98.68	99.18	77.87	95.89	66.00	29.00
576	–	–	–	93.06	66.00	35.00
640	–	–	–	90.28	70.00	41.00
704	–	–	–	95.83	79.00	40.00
768	–	–	–	93.06	76.00	36.00

B.2 Fixed-One-Bit Supplementary Metrics

Table 5 gives the complete fixed-one-bit detection and algorithmic sampling-efficiency statistics underlying Figure 4. The main text focuses on TPR at 1% FPR and AATPS; here we additionally report AUROC, TPR at 5% FPR, and one-bit decoding accuracy for RamoMark.

For the RamoMark likelihood detector, the retained positive/null counts for Qwen/CNN, Qwen/ELI5, Qwen/GSM8K, Vicuna/CNN, Vicuna/ELI5, and Vicuna/GSM8K are respectively $500/500$, $500/499$, $500/500$, $500/498$, $500/499$, and $500/499$. The five excluded null replays are not treated as negative examples; AUROC and fixed-FPR TPR are computed on the retained evaluation sets. MSE and MWS retain all $500/500$ positive/null samples in each cell.

B.3 Output-Quality Details

Table 6 expands the compact quality summary in the main text with repetition, diversity, and top-1 diagnostics. Figure 5 shows the corresponding change in processed LPPL relative to matched vanilla speculative sampling. These likelihood-based quantities are diagnostics of target-model fit and should not be interpreted as direct measures of semantic quality.

Table 5: Complete fixed-one-bit detection and algorithmic sampling-efficiency results ($n = 128$, $N = 500$). AATPS is normalized to matched vanilla speculative sampling within each model–dataset cell. Bit accuracy is defined only for RamoMark.

Model pair	Data	Method	AUROC	TPR@1%	TPR@5%	AATPS	AATPS/van.	Bit acc.
Qwen2.5-7B/0.5B	CNN	Vanilla	–	–	–	2.5962	100.00%	–
		MSE	0.9745	0.7360	0.8920	2.6022	100.23%	–
		MWS	0.9999	1.0000	1.0000	2.4425	94.08%	–
		RamoMark	0.9969	0.8800	0.9980	2.6003	100.16%	0.9900
Qwen2.5-7B/0.5B	ELI5	Vanilla	–	–	–	2.5367	100.00%	–
		MSE	0.9851	0.8540	0.9120	2.5342	99.90%	–
		MWS	1.0000	1.0000	1.0000	2.4053	94.82%	–
		RamoMark	0.9982	0.9500	1.0000	2.5266	99.60%	0.9980
Qwen2.5-7B/0.5B	GSM8K	Vanilla	–	–	–	3.4834	100.00%	–
		MSE	0.8505	0.2320	0.4740	3.5096	100.75%	–
		MWS	0.9565	0.5880	0.8300	3.4029	97.69%	–
		RamoMark	0.9769	0.5320	0.8040	3.4833	100.00%	0.9940
Vicuna-7B/68M	CNN	Vanilla	–	–	–	2.0541	100.00%	–
		MSE	0.7626	0.1060	0.3320	2.0753	101.03%	–
		MWS	0.9603	0.5900	0.8420	2.0130	98.00%	–
		RamoMark	0.9119	0.1920	0.4260	2.0611	100.34%	0.9360
Vicuna-7B/68M	ELI5	Vanilla	–	–	–	1.8773	100.00%	–
		MSE	0.8275	0.1800	0.4560	1.8658	99.39%	–
		MWS	0.9953	0.9380	0.9760	1.8180	96.84%	–
		RamoMark	0.9562	0.3800	0.6560	1.8750	99.88%	0.9680
Vicuna-7B/68M	GSM8K	Vanilla	–	–	–	1.9925	100.00%	–
		MSE	0.7034	0.0860	0.2100	2.0082	100.79%	–
		MWS	0.9511	0.6320	0.7840	1.9582	98.28%	–
		RamoMark	0.9313	0.2640	0.5640	2.0098	100.87%	0.9540

Table 6: Fixed-one-bit output quality on long-form prompts ($n = 128$, $N = 500$). ΔLPPL is measured relative to vanilla speculative sampling within each model–dataset cell.

Model pair	Data	Method	LPPL	ΔLPPL	Raw PPL	Rep.-5	Dist.-2	Top-1
Qwen2.5-7B/0.5B	CNN	Vanilla	0.8377	+0.0000	2.351	0.0049	0.9427	0.7385
Qwen2.5-7B/0.5B	CNN	MSE	0.8472	+0.0095	2.373	0.0039	0.9450	0.7335
Qwen2.5-7B/0.5B	CNN	MWS	0.8497	+0.0120	2.382	0.0044	0.9434	0.7341
Qwen2.5-7B/0.5B	CNN	RamoMark	0.8359	-0.0019	2.346	0.0042	0.9454	0.7382
Qwen2.5-7B/0.5B	ELI5	Vanilla	0.8644	+0.0000	2.415	0.0020	0.9563	0.7238
Qwen2.5-7B/0.5B	ELI5	MSE	0.8547	-0.0097	2.395	0.0020	0.9555	0.7271
Qwen2.5-7B/0.5B	ELI5	MWS	0.8413	-0.0231	2.360	0.0025	0.9569	0.7293
Qwen2.5-7B/0.5B	ELI5	RamoMark	0.8534	-0.0110	2.389	0.0020	0.9574	0.7272
Qwen2.5-7B/0.5B	GSM8K	Vanilla	0.3386	+0.0000	1.412	0.0301	0.8101	0.8786
Qwen2.5-7B/0.5B	GSM8K	MSE	0.3402	+0.0016	1.413	0.0294	0.8072	0.8784
Qwen2.5-7B/0.5B	GSM8K	MWS	0.3456	+0.0069	1.422	0.0285	0.8087	0.8771
Qwen2.5-7B/0.5B	GSM8K	RamoMark	0.3434	+0.0048	1.418	0.0313	0.8063	0.8782
Vicuna-7B/68M	CNN	Vanilla	0.4165	+0.0000	1.538	0.0052	0.9302	0.8599
Vicuna-7B/68M	CNN	MSE	0.4187	+0.0022	1.560	0.0054	0.9309	0.8619
Vicuna-7B/68M	CNN	MWS	0.4013	-0.0152	1.512	0.0042	0.9341	0.8661
Vicuna-7B/68M	CNN	RamoMark	0.4171	+0.0006	1.540	0.0047	0.9313	0.8598
Vicuna-7B/68M	ELI5	Vanilla	0.6488	+0.0000	1.948	0.0087	0.9103	0.7887
Vicuna-7B/68M	ELI5	MSE	0.6700	+0.0212	2.007	0.0082	0.9158	0.7845
Vicuna-7B/68M	ELI5	MWS	0.6751	+0.0263	2.035	0.0089	0.9113	0.7838
Vicuna-7B/68M	ELI5	RamoMark	0.6697	+0.0209	1.994	0.0080	0.9139	0.7841
Vicuna-7B/68M	GSM8K	Vanilla	0.3960	+0.0000	1.502	0.0459	0.7659	0.8665
Vicuna-7B/68M	GSM8K	MSE	0.4063	+0.0103	1.518	0.0483	0.7630	0.8642
Vicuna-7B/68M	GSM8K	MWS	0.4043	+0.0084	1.517	0.0466	0.7640	0.8644
Vicuna-7B/68M	GSM8K	RamoMark	0.3977	+0.0017	1.503	0.0490	0.7595	0.8661

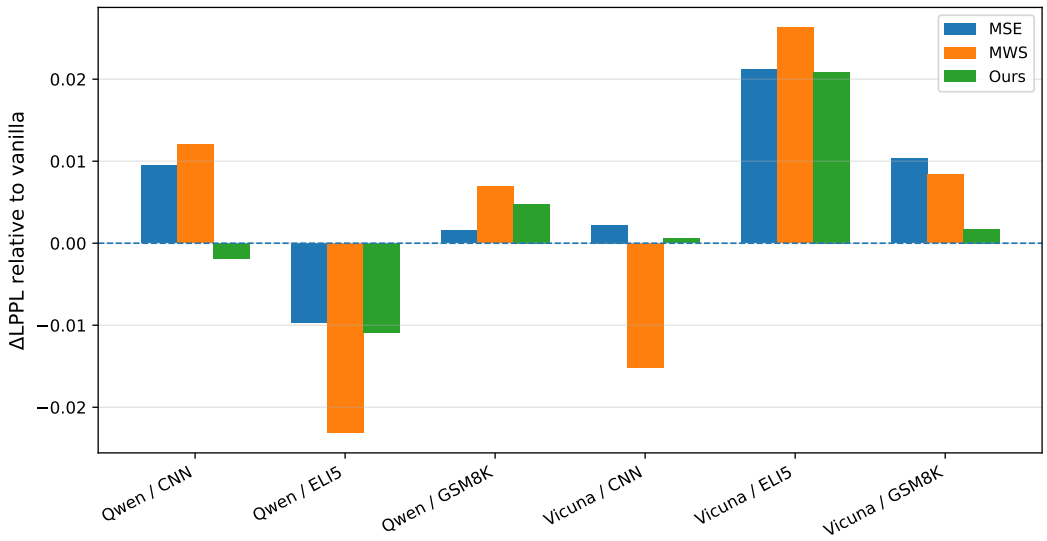


Figure 5: Change in processed LPPL relative to matched vanilla speculative sampling for the three fixed-one-bit watermarking methods.