WORLDCUP SAMPLING FOR MULTI-BIT LLM WATERMARKING

Anonymous authors
Paper under double-blind review

ABSTRACT

As large language models (LLMs) generate increasingly human-like text, watermarking has emerged as a promising solution for reliable attribution and traceability beyond mere detection. While multi-bit watermarking enables richer provenance encoding, existing approaches typically extend zero-bit schemes through static logit perturbations and hard counting-based decoding, which can degrade text quality and compromise decoding robustness as the payload increases. In this paper, we propose **WorldCup**, a multi-bit watermarking framework that extends tournament sampling and embeds message bits through hierarchical competition, in which complementary functions guide the selection of candidate tokens. Moreover, WorldCup incorporates entropy-aware modulation to adapt watermark strength to local uncertainty in the model distribution and preserve text quality, while confidence-aware decoding uses agreement across tournament layers to aggregate evidence for robust message recovery. Comprehensive experiments across multiple LLMs and diverse downstream tasks show that WorldCup offers a favorable trade-off among message capacity, detectability, robustness, text quality, and decoding efficiency, consistently outperforming prior baselines.

1 INTRODUCTION

Large language models (LLMs) (Guo et al., 2025; Comanici et al., 2025; Achiam et al., 2023) have shown remarkable performance across a wide range of real-world applications, including creative writing, math reasoning and AI agents (Luo et al., 2025; Chen et al., 2021; Cursor, 2026; Anthropic, 2026), rendering LLM-generated text progressively indistinguishable from human-written text (Mitchell et al., 2023). While these advances significantly boost productivity, they also amplify serious risks such as misinformation dissemination, academic plagiarism, and phishing attacks (Tang et al., 2024). As a consequence, LLM watermarking has emerged as a promising technique to mitigate these concerns by embedding imperceptible yet verifiable signals into generated text, thereby enabling reliable attribution, detection and traceability (Bengio et al., 2025).

Inference-time LLM watermarking can be broadly categorized into zero-bit and multi-bit schemes based on embedding capacity (Liu et al., 2024b; Wang et al., 2025a; Liang et al., 2026). While zero-bit watermarking focuses solely on detection, multi-bit watermarking further enables the extraction of rich metadata, such as model identity or generation timestamps. The prevailing multi-bit watermarking paradigm builds upon zero-bit schemes by implicitly embedding message into the stochastic generation process. Typically, a pseudo-random seed is derived by hashing a predefined secret key with the context and is associated with a target bit string to steer an underlying zero-bit watermarking mechanism, thereby establishing a verifiable link between the message and the generated text. Decoding reverses this process by aggregating token-level evidence to identify the message that best aligns with the expected watermark signal.

However, despite their effectiveness, existing multi-bit watermarking still inherits key limitations from their zero-bit foundations. In particular, most approaches rely on static logit perturbations to inject watermark signals, resulting in an implicit and weakly controlled interaction with the original model distribution. Such a design can interfere with the natural generation dynamics, especially at higher payloads, thereby degrading text quality and fluency. Meanwhile, these methods typically adopt counting-based decoding, where tokens are discretely mapped to binary outcomes and con-

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.

tribute equally. This creates a mismatch between soft probabilistic embedding and hard aggregation, leading to suboptimal trade-off between message capacity and robustness.

To overcome these limitations, we revisit multi-bit watermarking from the perspective of the sampling process. Rather than relying on static logit biasing, we directly incorporate the message into tournament sampling, allowing it to guide the competition among candidate tokens. This approach is inspired by Google's SynthID Text (Dathathri et al., 2024), which shows that tournament sampling can embed statistically verifiable watermark signals while preserving text quality. Building on this insight, we extend this mechanism to multi-bit watermarking and introduce a multi-round, complementary sampling strategy that integrates message embedding with the competitive selection of candidate tokens at each step of the autoregressive generation process.

Specifically, we propose **WorldCup**, a multi-bit LLM watermarking framework that leverages the inherent redundancy of autoregressive token generation to support robust and scalable information embedding. At the **encoding** stage, WorldCup conceptualizes watermarking as a structured competition process, in which multiple complementary signals jointly guide token selection and induce stable statistical separation, while preserving sufficient flexibility for high-quality natural language generation. To balance detectability and fluency, the framework further incorporates an entropy-aware modulation mechanism that adaptively adjusts watermark strength in response to local uncertainty in the model output distribution. At the **decoding** stage, WorldCup moves beyond simple counting-based detection and adopts a soft confidence-aware aggregation that uses agreement across tournament layers as token-level evidence, thereby mitigating the disproportionate influence of low-entropy tokens. These design choices enable fine-grained control over the divergence between clean and watermarked distributions, and become increasingly advantageous as the embedded payload grows, yielding improved bit accuracy at higher payloads while maintaining efficient decoding.

To this end, experiments across LLMs and downstream tasks show that WorldCup improves bit accuracy in most settings, with a strong balance among multi-bit capacity, watermark detectability, robustness, text quality, and decoding efficiency. In summary, our contributions are threefold:

- We depart from static logit perturbations and extend SynthID's tournament sampling from zero-bit detection to multi-bit watermarking, with message bits guiding the selection of candidate tokens.
- We propose WorldCup, a versatile multi-bit watermarking framework that integrates complementary functions, entropy-aware modulation, and confidence-aware decoding.
- We present theoretical analysis and extensive empirical evaluation across multiple models and datasets, including systematic studies of key components and hyperparameter choices.

2 PRELIMINARIES

Notations. Consider an autoregressive LLM Θ based on transformer (Vaswani et al., 2017), let $\mathcal{V}$ denote the vocabulary set of all tokens with size $|\mathcal{V}|$. Given a prompt p_0 and previously contextual tokens $\mathbf{x}_{<t} = (x_1, x_2, \ldots, x_{t-1})$, the LLM generates an imminent token $x_t \in \mathcal{V}$ sequentially from the conditional distribution $P_\Theta(x_t \mid p_0, \mathbf{x}_{<t}) \in \Delta_\mathcal{V}$. The process repeats until either a predefined maximum length is reached or an end-of-sequence token is produced.

2.1 SYNTHID-TEXT ZERO-BIT WATERMARKING

Watermark Embedding. At each generation step t, SynthID-Text (Dathathri et al., 2024) derives a random seed r_t by hashing the previously generated text together with a secret watermark key $\xi \in \Xi$. The seed is then provided to m independent pseudo-random functions (PRFs) $\mathbf{g} = (g_1, \ldots, g_m)$ (Goldreich et al., 1986), each of which assigns a binary value $g_\ell(x_i, r_t) \in \{0, 1\}$ to every token $x_i \in \mathcal{V}$, where $\ell \in \{1, \ldots, m\}$. Token selection proceeds via an m-layer tournament-based sampling scheme. Concretely, N^m candidate tokens are independently sampled from the model distribution $P_\Theta(\cdot \mid p_0, \mathbf{x}_{<t})$ to form the leaves of a complete N-ary tournament tree of depth m. At each layer ℓ, candidates are grouped in sets of N and compared using the corresponding PRF scores $g_\ell(\cdot, r_t)$. In practice, N is set to 2, which corresponds to pairwise comparisons. The token with the highest score in each group advances to the next layer, with ties broken at random. This elimination process continues until a single token remains, which is emitted as the output token x_t.

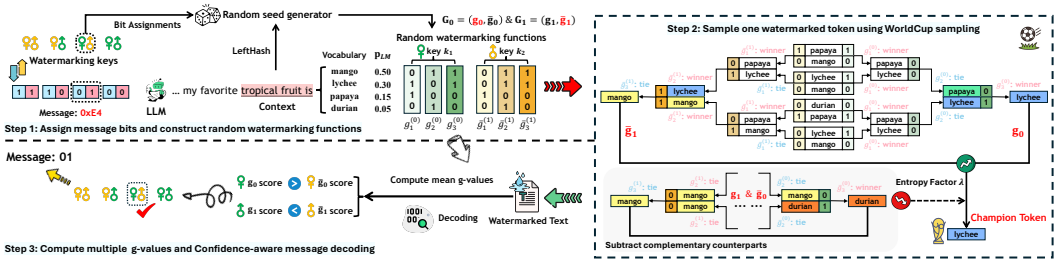

Figure 1: An overview of our multi-bit watermarking framework **WorldCup** for LLMs.

Watermark Detection. By design, tournament sampling biases the generation process toward tokens that attain higher scores under the watermarking functions $\mathbf{g}$. Given a generated text $\mathbf{y} = (y_1, y_2, \ldots, y_T)$, watermark detection measures how well the text $\mathbf{y}$ aligns with the functions $\mathbf{g}$. Formally, the detection statistic is defined as the average watermark score over the sequence:

$$s(\mathbf{y}; \mathbf{g}) = \frac{1}{mT} \sum_{t=1}^{T} \sum_{\ell=1}^{m} g_\ell(y_t, r_t). \tag{1}$$

Since watermarked tokens tend to yield higher g-values, watermarked texts attain higher scores $s(\mathbf{y}; \mathbf{g})$ than unwatermarked text, enabling reliable statistical zero-bit watermark detection.

2.2 MPAC Multi-bit Watermarking

Message Encoding. Let $\mathbf{m} \in \Sigma^b$ be the message over an s-ary alphabet $\Sigma = \{0, \ldots, s-1\}$ and b is its length. At each step t, MPAC uses a PRF with seed r_t to sample a position $p \in \{0, \ldots, b-1\}$ and retrieves $\mathbf{m}[p]$. The vocabulary is randomly shuffled and partitioned into s disjoint subsets $\mathcal{V}_t = [C_0, \ldots, C_{s-1}]$; the subset indexed by $\mathbf{m}[p]$ is treated as the green list and given a positive logits bias δ, increasing its sampling probability and thus encoding the message symbol.

Message Decoding. MPAC extracts the embedded multi-bit messages from text via a majority voting matrix $\mathbf{M} \in \mathbb{R}^{b \times s}$ initialized to zero. For each token y_t, the decoder reconstructs the message position p and the vocabulary partitions $\mathcal{V}_t$ using the same PRF as in the encoding stage. If $y_t \in C_j$, the corresponding entry $\mathbf{M}[p][j]$ is incremented by one. After processing the entire text, each message symbol is recovered via $\hat{\mathbf{m}}[p] = \arg\max_{j \in \{0,\ldots,s-1\}} (\mathbf{M}[p][j])$. Under the assumption that watermarked text contains more green-list tokens than non-green-list tokens, the aggregate majority count $\sum_{p=0}^{b-1} \max_j (\mathbf{M}[p][j])$ serves as the total green tokens for computing the z-score statistic, which subsequently determines whether the text is watermarked.

3 Methodology

Overview. As illustrated in Fig. 1, WorldCup framework comprises two central stages: message embedding (Section 3.1, Section 3.2 and Section 3.3) and message decoding (Section 3.4).

3.1 Binary WorldCup Watermarking

Definition 3.1. (random seed generator). Given a security parameter κ, define the random seed space as $\mathcal{R} = \{0,1\}^\kappa$. Let $h(\cdot)$ be a hash function and ξ a watermark key. At generation step t, the random seed is computed as

$$r_t = h(x_{t-c}, \ldots, x_{t-1}, \xi) \in \mathcal{R}, \tag{2}$$

where c denotes the sliding-window size. We assume that $r_t \sim \mathrm{Unif}(\mathcal{R})$ for any $\mathbf{x}_{<t}$ where $\mathrm{Unif}(\cdot)$ represents the uniform distribution.

Definition 3.2. (g-value). Given a token $x \in \mathcal{V}$, a random seed $r \in \mathcal{R}$, and a layer index $\ell \in \{1, \ldots, m\}$, a g-value function is a pseudo-random mapping $g_\ell : \mathcal{V} \times \mathcal{R} \to \mathbb{R}$. The g-value of token x at layer ℓ is the random variable $g_\ell(x, r)$.

Algorithm 1 Binary WorldCup Watermarking

Input: LLM distribution $P_\Theta(\cdot \mid p_0, \mathbf{x}_{<t})$, random seed r_t, layers number m, leaves number $N \geq 2$, message bit $\mathbf{m}[p]$, g-value function families $\mathbf{g}_0, \mathbf{g}_1$
Draw N^m i.i.d tokens $y_0^0, \ldots, y_{N^m-1}^0 \sim P_\Theta(\cdot \mid p_0, \mathbf{x}_{<t})$
Initialize $(g_1, \ldots, g_m) \leftarrow \mathbf{g}_0$ if $\mathbf{m}[p] = 0$ else $\mathbf{g}_1$
for $1 \leq \ell \leq m$ **do**
  for $0 \leq j \leq N^{m-\ell} - 1$ **do**
    $Y := [y_{Nj}^{\ell-1}, \ldots, y_{Nj+N-1}^{\ell-1}]$ // may contain repeats
    $Y^* := [y \in Y : g_\ell(y, r_t) = \max_{y' \in Y} g_\ell(y', r_t)]$ // may contain repeats
    Sample $y_j^\ell \sim \mathrm{Unif}(Y^*)$
  end for
end for
Return $x_t \leftarrow y_0^m$

To implement binary WorldCup watermarking, we introduce two independent families of g-value functions $\mathbf{g}_0 = (g_1^{(0)}, g_2^{(0)}, \ldots, g_m^{(0)})$ and $\mathbf{g}_1 = (g_1^{(1)}, g_2^{(1)}, \ldots, g_m^{(1)})$, to encode a binary message $\mathbf{m} \in \{0,1\}^b$. At each generation step t, we first sample N^m (typically 2^m) candidate tokens $\{y_0, y_1, \ldots, y_{N-1}\}$ with replacement from the model distribution $P_\Theta(\cdot \mid p_0, \mathbf{x}_{<t})$. Then a cryptographic hash function h is applied to randomly assign the current token x_t to a target message bit, determining which bit is to be embedded.

Conditioned on the selected bit value, tournament sampling is carried out as follows. If the target bit equals 0, candidates are scored using the function family $\mathbf{g}_0$: at each tournament layer ℓ, the score $g_\ell^{(0)}(\cdot, r_t)$ is evaluated and the token with the higher score advances to the next layer. This elimination process continues for m rounds, ultimately yielding a single winner x_t, which is emitted as the watermarked token. If the target bit equals 1, the same tournament structure is applied using the alternative family $\mathbf{g}_1$, as illustrated in Algorithm 1.

For efficiency, instead of explicitly running the tournament in Algorithm 1, we use an equivalent vectorized formulation to sample from the resulting watermarked distribution[1]:

Definition 3.3. (watermarked distribution). Given a probability distribution $P_\Theta \in \Delta_\mathcal{V}$, a random seed $r_t \in \mathcal{R}$, layers number $m \geq 1$, leaves number $N \geq 2$ and g-value functions $\mathbf{g}_0, \mathbf{g}_1$. For message $\mathbf{m}$, position p, the watermarked distribution of the winner in Algorithm 1 is defined as:

$$q(x_t) = \mathbb{P}[\text{Alg. } 1(P_\Theta, r_t, m, N, \mathbf{m}[p], \mathbf{g}_0, \mathbf{g}_1) \Rightarrow x_t]. \tag{3}$$

Details and equivalence proofs are given in Appendix G.2.

3.2 Complementary g-value Functions

Thus far, the two g-value families have been treated as arbitrary pseudo-random constructions. We now consider a principled design in which $\mathbf{g}_0$ and $\mathbf{g}_1$ are constructed as complementary pairs, and show that this choice is optimal in terms of statistical discriminability.

Definition 3.4. (complementary g-value). Given a g-value $g_\ell(x, r)$ as defined in Definition 3.2, its complementary g-value $\bar{g}_\ell(x, r)$ is defined as follows:

$$\bar{g}_\ell(x, r) \triangleq 1 - g_\ell(x, r). \tag{4}$$

In this paper, we focus on the case where the g-value follows a Bernoulli distribution $g_\ell(x, r) \sim \mathrm{Bernoulli}(0.5)$. Under this setting, both $g_\ell(x, r)$ and $\bar{g}_\ell(x, r)$ take values in $\{0, 1\}$.

Intuitively, this complementary construction induces perfect anti-correlation between $\mathbf{g}_0$ and $\mathbf{g}_1$: tokens favored under $\mathbf{g}_0$ are deterministically disfavored under $\mathbf{g}_1$. As a result, the two message hypotheses are pushed to opposite extremes of the g-value spectrum, yielding the maximum possible separation between their induced embedding distributions. From a decoding perspective, this symmetry directly maximizes the decision margin and improves robustness against noise.

[1] In general, we assume $2^m \gg |\mathcal{V}|$.

Figure 2 visualizes this effect. Compared to independently sampled (random) g-value functions, complementary g-values exhibit a strict linear relationship, making the two message states (0 and 1) substantially easier to distinguish. This observation motivates the following proposition, which formalizes the optimality of complementary g-value functions in the multi-bit watermarking setting.

Proposition 3.5. *Let $g_\ell(x, r)$ be a Bernoulli g-value and $\bar{g}_\ell(x, r)$ its complementary counterpart. Encoding bits 0 and 1 by selecting between g_ℓ and $\bar{g}_\ell$ yields two embedding distributions whose statistical discriminability is maximized.*

A formal proof based on the maximization of the expected squared difference $\mathbb{E}[(\mathbf{g}_1 - \mathbf{g}_0)^2]$ and an explicit correlation analysis are deferred to Appendix G.1.

3.3 Generalized WorldCup Watermarking

While binary WorldCup sampling enables one bit per token, the information allocation becomes increasingly imbalanced as the message length grows (Qu et al., 2025). Although the nominal capacity per token remains fixed, the global utilization of token-level degrees of freedom is suboptimal. To fully exploit the information capacity of each token, we generalize WorldCup by introducing k groups of g-value function families, allowing each token to simultaneously encode k bits.

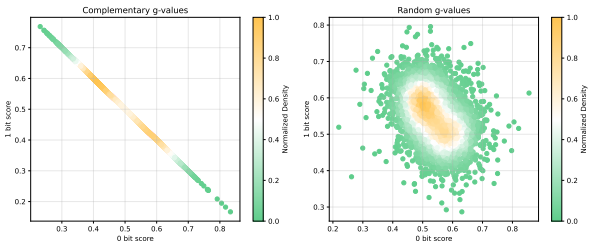

Figure 2: Complementary g-values vs. random g-values.

We consider the simplest but nontrivial setting where each token carries two bits (i.e. $k = 2$). To this end, we sample two independent families of g-value functions, denoted by $\mathbf{G}_0 = (\mathbf{g}_0, \bar{\mathbf{g}}_0)$ and $\mathbf{G}_1 = (\mathbf{g}_1, \bar{\mathbf{g}}_1)$. The first bit is encoded using $\mathbf{G}_0$, following the binary setting: when the bit equals 0, the watermark distribution q_0 (defined in Definition 3.3) is derived from $\mathbf{g}_0$; otherwise, its complementary function $\bar{\mathbf{g}}_\mathbf{0}$ is applied, yielding the distribution $\bar{q}_0$. The second bit is encoded analogously using $\mathbf{G}_1$, producing either q_1 or $\bar{q}_1$.

Conceptually, a token that correctly embeds both bits should score highly under both corresponding g-value functions. We thus aggregate their distributions additively. However, $\mathbf{G}_0$ and $\mathbf{G}_1$ are independent rather than mutually exclusive, so naive aggregation yields limited separability between bit patterns. To enhance discriminability, we subtract contributions from complementary distributions, thereby sharpening the contrast between competing message hypotheses. In terms of tournament sampling, this strategy favors tokens that rank highly under the intended g-value functions while ranking poorly under their complementary counterparts.

Although this more aggressive separation improves detectability, it can also distort the underlying language model distribution and degrade text quality, as demonstrated in Section 5.1. To strike a better balance between detectability and fluency, we introduce an entropy-aware dynamic adjustment factor. Concretely, for a two-bit message $\mathbf{m}'$, we define the updated watermarked scores as follows:

$$P_{\Theta,\mathbf{m}} = \begin{cases} (q_0 + q_1) - \lambda(\overline{q_0} + \overline{q_1}), & \mathbf{m}' = 00 \\ (q_0 + \overline{q_1}) - \lambda(\overline{q_0} + q_1), & \mathbf{m}' = 01 \\ (\overline{q_0} + q_1) - \lambda(q_0 + \overline{q_1}), & \mathbf{m}' = 10 \\ (\overline{q_0} + \overline{q_1}) - \lambda(q_0 + q_1), & \mathbf{m}' = 11 \end{cases}. \tag{5}$$

The scores are first made non-negative via clipping, then log-transformed for numerical stability and normalized via softmax, from which the **champion token** is ultimately sampled. Here, the coefficient λ is adaptively determined by the entropy of the base LLM distribution:

$$\lambda = \alpha \cdot \sigma(\sum -P_\Theta \log P_\Theta), \tag{6}$$

where α is a hyperparameter that controls the watermark strength, and σ denotes the activation function, with their specific choices reported in Appendix I.5. Under this formulation, high-entropy

(low-confidence) token positions permit stronger separation, while low-entropy tokens favor more conservative modulation to preserve generation quality.

This construction can be naturally generalized to $k > 2$ groups of g-value functions, allowing each token to embed k bits. Detailed formulations are provided in Appendix G.3.

3.4 Confidence-aware Message Decoding

Compared with counting-based decoding, our confidence-aware decoder aggregates token scores for each message position instead of binarizing individual tokens, as illustrated in Fig. 3.

Problem setup. Let $\mathbf{y}_{1:T} = (y_1, y_2, \ldots, y_T)$ denote a generated text sequence of length T. As in the encoding stage, we consider k independent groups of g-value function families, denoted by $\{\mathbf{G}_j = (\mathbf{g}_j, \bar{\mathbf{g}}_j)\}_{j=1}^{k}$. These k families jointly encode a k-bit message string at each token position. Accordingly, a binary message in $\{0,1\}^b$ is reparameterized as a sequence of 2^k-ary message $\mathbf{m} = (m_0, \ldots, m_{B-1})$, where $m_p \in \{0, \ldots, 2^k - 1\}$ for $p \in \{0, \ldots, B-1\}$, and $B = b/k$ denotes the total number of message symbols.

1) Recovering message positions. We first employ the same hash function and watermarking key used during message embedding stage to identify the message symbol position p associated with each token y_t. All tokens assigned to the same index p are then grouped together for joint decoding of the corresponding 2^k-ary symbol m_p.

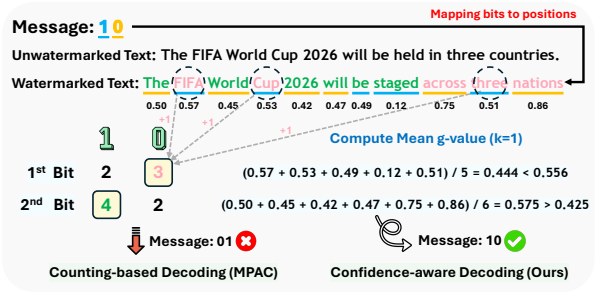

Figure 3: The comparison of multi-bit message decoding: counting-based decoding vs. confidence-aware decoding.

2) Computing confidence scores. For each message position p and group $j \in \{1, \ldots, k\}$, we average the g-values of its assigned tokens following Eq. 1, obtaining s_j^p and $\bar{s}_j^p$ for $\mathbf{g}_j$ and $\bar{\mathbf{g}}_j$, respectively. These quantities serve as confidence scores indicating whether the j-th bit of m_p favors 0 or 1.

3) Extracting message symbols. For each message position p, we independently infer the j-th bit of the embedded symbol by comparing s_j^p and $\bar{s}_j^p$. The recovered 2^k-ary message symbol $\hat{m}_p \in \{0, \ldots, 2^k - 1\}$ is then obtained as

$$\hat{m}_p = \sum_{j=1}^{k} 2^{k-j}\, \mathbb{I}\big(s_j^p < \bar{s}_j^p\big)\,. \tag{7}$$

Computing the watermark z-score. We cast watermark detection as a hypothesis test under the null hypothesis that the text is unwatermarked. Detection relies on a standardized z-score (see Appendix G.4 for its definition and derivation). Under the null hypothesis, unwatermarked texts yield low z-scores, whereas watermarked texts produce systematically higher values. Thus, a sufficiently large z-score leads to rejection of the null hypothesis, indicating the presence of a watermark.

4 Experiments

Overview. We conduct a comprehensive empirical evaluation of WorldCup along three fundamental dimensions: **message embedding capacity** (Section 4.2) including message decoding accuracy and watermark detectability, **text quality** (Section 4.4), and **robustness** (Section 4.4).

4.1 Experimental Setup

Baselines. We compare WorldCup with representative public, model-agnostic, training-free multi-bit watermarking methods, including BiMark (Feng et al., 2025), MPAC (Yoo et al., 2024), SegMark

Table 1: The comparison of different multi-bit watermark performance on LLaMA3 and Gemma2.

Bit Length	Watermark	LLAMA3-8B-BASE								GEMMA2-9B-BASE							
		MAX 128 TOKENS				MAX 256 TOKENS				MAX 128 TOKENS				MAX 256 TOKENS			
		AUC ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	AUC ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	AUC ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	AUC ↑	Bit Acc ↑	PPL ↓	Time (s) ↓
16 bits	MPAC	0.999	0.960	16.25	0.014	0.996	0.980	13.56	0.027	0.980	0.920	13.69	0.015	0.985	0.940	12.00	0.030
	SegMark	0.993	0.948	15.88	0.551	0.995	0.999	12.94	0.839	0.979	0.895	13.19	1.234	0.999	0.977	10.94	2.402
	StealthInk	0.992	0.937	14.69	0.016	0.997	0.972	13.28	0.038	0.951	0.873	11.72	0.014	0.978	0.918	9.500	0.031
	BiMark	1.000	0.977	14.34	0.026	1.000	0.987	11.25	0.044	1.000	0.929	11.06	0.031	0.999	0.955	9.625	0.039
	SynthID + MPAC	0.998	0.977	16.25	0.008	1.000	0.990	13.56	0.013	0.997	0.925	11.56	0.008	0.997	0.955	10.63	0.011
	WorldCup	0.999	0.982	12.94	0.009	1.000	0.990	11.25	0.015	0.998	0.931	10.84	0.008	0.998	0.956	8.906	0.012
24 bits	MPAC	0.996	0.916	16.75	0.012	0.997	0.955	14.00	0.016	0.972	0.875	13.69	0.017	0.959	0.905	12.00	0.035
	SegMark	0.947	0.839	17.13	0.721	0.992	0.992	13.38	0.804	0.849	0.734	13.19	1.670	0.974	0.922	10.13	2.853
	StealthInk	0.989	0.885	15.16	0.018	0.995	0.935	13.19	0.022	0.917	0.830	11.81	0.019	0.975	0.877	10.13	0.039
	BiMark	1.000	0.941	14.34	0.027	1.000	0.970	11.34	0.036	0.998	0.875	11.63	0.030	0.990	0.921	9.938	0.043
	SynthID + MPAC	1.000	0.935	15.19	0.008	1.000	0.971	13.19	0.011	0.995	0.893	13.56	0.008	0.995	0.917	9.810	0.011
	WorldCup	1.000	0.943	13.69	0.009	1.000	0.972	11.53	0.014	0.998	0.894	10.44	0.008	0.999	0.925	8.625	0.012
32 bits	MPAC	0.997	0.890	16.75	0.021	0.996	0.939	14.00	0.020	0.947	0.837	13.81	0.016	0.940	0.869	11.63	0.034
	SegMark	0.909	0.802	16.63	0.642	0.989	0.964	12.94	1.053	0.824	0.691	12.75	1.751	0.955	0.868	10.25	3.362
	StealthInk	0.990	0.845	14.44	0.024	0.994	0.919	12.75	0.029	0.943	0.797	12.00	0.018	0.962	0.850	10.13	0.036
	BiMark	1.000	0.890	13.91	0.026	1.000	0.947	12.38	0.040	0.996	0.806	12.00	0.031	0.989	0.865	9.938	0.041
	SynthID + MPAC	1.000	0.894	15.38	0.008	1.000	0.938	13.81	0.011	0.995	0.826	13.81	0.010	0.990	0.901	12.00	0.011
	WorldCup	1.000	0.915	13.38	0.010	1.000	0.958	11.25	0.017	0.995	0.859	10.94	0.009	0.994	0.905	8.688	0.013
48 bits	MPAC	0.993	0.828	16.63	0.017	0.990	0.891	13.19	0.039	0.936	0.770	14.00	0.018	0.914	0.821	12.19	0.038
	SegMark	0.837	0.628	16.38	0.725	0.968	0.848	13.56	1.381	0.770	0.600	13.38	2.668	0.900	0.697	9.938	3.382
	StealthInk	0.977	0.804	14.69	0.020	0.983	0.870	12.94	0.041	0.904	0.738	11.81	0.017	0.932	0.797	9.938	0.039
	BiMark	1.000	0.783	14.25	0.027	0.999	0.880	11.63	0.037	0.985	0.688	11.53	0.029	0.975	0.767	10.03	0.040
	SynthID + MPAC	1.000	0.787	15.19	0.009	1.000	0.900	13.28	0.012	0.990	0.737	13.28	0.009	0.985	0.814	11.72	0.012
	WorldCup	1.000	0.862	13.69	0.011	1.000	0.916	11.63	0.017	0.993	0.800	10.34	0.010	0.989	0.851	9.063	0.015

(Qu et al., 2025), StealthInk (Jiang et al., 2025b). We additionally include SynthID Text (Dathathri et al., 2024) combined with MPAC as a baseline, with its g-functions sampled independently at random. Message lengths are set to 16, 24, 32, and 48 bits. Detailed settings are in Appendix B.

Datasets. Following prior work (Kirchenbauer et al., 2023; Zhao et al., 2024), we randomly sample 200 prompts from the C4 (Raffel et al., 2020) and OpenGen (Krishna et al., 2023) datasets for text generation. We further evaluate downstream performance on four representative tasks with varying input and output lengths, including machine translation, text summarization, question answering, and math reasoning. More details are in Appendix C.

Metrics. We primarily use Bit Accuracy to evaluate message decoding correctness. Detectability is measured using AUC, Best F1 score, and TPR@1% FPR, while text quality is assessed with Median Perplexity (PPL), ROUGE-L, BLEU, Pass@1, and the GPT-4 Score. Decoding efficiency is measured using decoding time (s). Robustness is evaluated using AUROC. TPR@1% FPR captures detection power at a stringent low FPR operating point. Metric definitions are in Appendix D.

Models. For detectability and robustness evaluation, we use LLaMA3-8B-Base Grattafiori et al. (2024) and Gemma2-9B-Base Team et al. (2024). For downstream tasks, we additionally evaluate their Instruct-tuned variants including LLaMA3.1-8B-Instruct, Gemma2-9B-Instruct and the latest Ministral-8B-Instruct (Liu et al., 2026). For fairness, we compute PPL using the larger Vicuna-13B-v1.5 model (Chiang et al., 2023). The details are in Appendix E.

Implementation Details. Our method is implemented in Python 3.10.0 with PyTorch 2.6.0. All experiments are conducted on a single NVIDIA A100 80 GB GPU. We use a default setup with g-value function number $k = 2$, temperature = 1.0, top-k = 50, top-p = 0.95, no_repeat_ngram_size = 4, layers number $m = 30$, and $\alpha = 1.2$, together with lefthash and window size $c = 2$.

4.2 MESSAGE EMBEDDING CAPACITY

Multi-bit watermarking scenario. We evaluate decoding performance with maximum sequence lengths of 128 and 256 tokens and a minimum length of 64. Table 1 shows that WorldCup achieves the highest bit accuracy in most evaluated settings. On LLaMA3-8B, it improves bit accuracy over StealthInk by **3.2%**, **4.8%**, **5.5%**, and **5.2%** for 16-bit, 24-bit, 32-bit, and 48-bit messages, respectively. WorldCup matches or exceeds SynthID+MPAC in bit accuracy, with larger gains at higher payloads. Overall, these results suggest that multiple g-functions improve information utilization per token and message recovery. We also observe that BiMark, MPAC, SegMark, and WorldCup exhibit lower average decoding accuracy on Gemma2-9B than LLaMA3-8B, with drops of **7.1%**, **5.3%**, **8.0%**, and **5.2%**, respectively. This gap is likely due to token-level uncertainly and details are in Appendix I.1. Increasing both lengths proportionally does not always preserve bit accuracy.

Table 2: The performance of various watermarking methods across four different downstream tasks.

Model	Task 1: Short Q, Short A *Machine Translation*			Task 2: Long Q, Short A *Text Summarization*			Task 3: Short Q, Long A *Long-form QA*			Task 4: Long Q, Long A *Math Reasoning*		
+ Watermark	Best F1 ↑	Bit Accuracy ↑	BLEU ↑	Best F1 ↑	Bit Accuracy ↑	ROUGE-L ↑	Best F1 ↑	Bit Accuracy ↑	GPT4 Score ↑	Best F1 ↑	Bit Accuracy ↑	Pass@1 ↑
MINISTRAL-8B-IT	-	-	0.200 ± 0.002	-	-	0.221 ± 0.001	-	-	5.250 ± 0.073	-	-	0.340 ± 0.010
+ MPAC	0.745 ± 0.005	0.886 ± 0.004	0.166 ± 0.001	0.951 ± 0.003	0.929 ± 0.002	0.215 ± 0.001	0.946 ± 0.002	0.879 ± 0.001	5.122 ± 0.029	0.785 ± 0.002	0.879 ± 0.001	0.265 ± 0.000
+ BiMark	0.776 ± 0.006	0.803 ± 0.030	0.193 ± 0.003	0.977 ± 0.005	0.943 ± 0.005	0.208 ± 0.002	0.992 ± 0.000	0.894 ± 0.002	4.998 ± 0.088	0.767 ± 0.006	0.877 ± 0.002	0.310 ± 0.050
+ StealthInk	0.723 ± 0.001	0.843 ± 0.010	0.181 ± 0.008	0.930 ± 0.003	0.923 ± 0.006	0.206 ± 0.005	0.876 ± 0.011	0.799 ± 0.006	5.213 ± 0.016	0.832 ± 0.003	0.919 ± 0.010	0.305 ± 0.010
+ SynthID + MPAC	0.748 ± 0.010	0.849 ± 0.001	0.194 ± 0.002	0.978 ± 0.004	0.927 ± 0.000	0.208 ± 0.003	0.995 ± 0.001	0.836 ± 0.001	5.150 ± 0.008	0.890 ± 0.007	0.861 ± 0.007	0.312 ± 0.018
+ **WorldCup**	0.752 ± 0.004	0.919 ± 0.009	0.202 ± 0.005	0.983 ± 0.001	0.955 ± 0.002	0.216 ± 0.004	0.996 ± 0.001	0.905 ± 0.002	5.207 ± 0.092	0.898 ± 0.000	0.953 ± 0.000	0.315 ± 0.000
LLAMA3.1-8B-IT	-	-	0.273 ± 0.004	-	-	0.249 ± 0.001	-	-	5.243 ± 0.069	-	-	0.500 ± 0.020
+ MPAC	0.700 ± 0.000	0.805 ± 0.002	0.224 ± 0.002	0.880 ± 0.010	0.869 ± 0.004	0.249 ± 0.002	0.940 ± 0.003	0.855 ± 0.003	5.110 ± 0.052	0.856 ± 0.006	0.927 ± 0.004	0.343 ± 0.007
+ BiMark	0.718 ± 0.002	0.768 ± 0.068	0.258 ± 0.005	0.960 ± 0.004	0.898 ± 0.004	0.242 ± 0.001	0.981 ± 0.000	0.872 ± 0.005	5.066 ± 0.110	0.868 ± 0.005	0.922 ± 0.006	0.420 ± 0.020
+ StealthInk	0.672 ± 0.001	0.766 ± 0.018	0.256 ± 0.005	0.877 ± 0.021	0.848 ± 0.003	0.243 ± 0.001	0.845 ± 0.005	0.779 ± 0.001	5.391 ± 0.050	0.775 ± 0.007	0.872 ± 0.008	0.350 ± 0.015
+ SynthID + MPAC	0.702 ± 0.006	0.760 ± 0.005	0.262 ± 0.002	0.954 ± 0.003	0.860 ± 0.009	0.244 ± 0.002	0.985 ± 0.001	0.814 ± 0.007	5.220 ± 0.129	0.889 ± 0.007	0.942 ± 0.003	0.438 ± 0.008
+ **WorldCup**	0.705 ± 0.003	0.850 ± 0.025	0.272 ± 0.001	0.963 ± 0.004	0.919 ± 0.006	0.247 ± 0.002	0.986 ± 0.000	0.882 ± 0.005	5.149 ± 0.024	0.892 ± 0.003	0.947 ± 0.001	0.445 ± 0.005
GEMMA2-9B-IT	-	-	0.407 ± 0.001	-	-	0.322 ± 0.005	-	-	6.327 ± 0.033	-	-	0.650 ± 0.010
+ MPAC	0.673 ± 0.000	0.686 ± 0.007	0.393 ± 0.002	0.671 ± 0.000	0.619 ± 0.001	0.309 ± 0.001	0.838 ± 0.000	0.770 ± 0.001	5.920 ± 0.003	0.670 ± 0.000	0.670 ± 0.000	0.625 ± 0.000
+ BiMark	0.675 ± 0.002	0.606 ± 0.027	0.398 ± 0.002	0.669 ± 0.000	0.598 ± 0.011	0.317 ± 0.003	0.883 ± 0.004	0.762 ± 0.001	5.946 ± 0.027	0.676 ± 0.002	0.706 ± 0.003	0.590 ± 0.000
+ StealthInk	0.667 ± 0.000	0.604 ± 0.001	0.379 ± 0.003	0.667 ± 0.003	0.593 ± 0.006	0.316 ± 0.002	0.712 ± 0.002	0.690 ± 0.004	6.233 ± 0.008	0.670 ± 0.001	0.663 ± 0.007	0.562 ± 0.008
+ SynthID + MPAC	0.685 ± 0.001	0.643 ± 0.002	0.365 ± 0.001	0.666 ± 0.000	0.557 ± 0.001	0.301 ± 0.001	0.857 ± 0.001	0.716 ± 0.002	6.161 ± 0.011	0.676 ± 0.003	0.687 ± 0.004	0.640 ± 0.030
+ **WorldCup**	0.673 ± 0.002	0.713 ± 0.002	0.401 ± 0.003	0.668 ± 0.001	0.664 ± 0.001	0.312 ± 0.001	0.886 ± 0.001	0.800 ± 0.002	6.136 ± 0.017	0.678 ± 0.005	0.755 ± 0.010	0.645 ± 0.020

Comparing 16 bits in 128 tokens with 32 bits in 256 tokens shows that tokens per bit alone do not determine recovery difficulty, possibly due to uneven position assignment and varying token entropy.

Zero-bit watermarking scenario. Using the max-based statistic described in Appendix G.4 as an empirical detection score, WorldCup achieves an average AUC of approximately **99.7%** across all scenarios. At a fixed 1% FPR, WorldCup further achieves an average TPR of **99.3%** across all settings (Table 4). Moreover, 256-token sequences generally provide stronger detection signals than 128-token sequences, as longer contexts accumulate more watermark evidence. These results not only confirm that detectability remains stable under large payloads, but also demonstrate that WorldCup gracefully subsumes the zero-bit watermarking regime without additional mechanisms.

4.3 TEXT QUALITY PRESERVATION

Perplexity (PPL). As shown in Table 1 and Fig. 7, WorldCup generally achieves lower perplexity than the evaluated baselines, suggesting competitive text fluency. We also observe that longer generated sequences tend to yield lower PPL. For instance, compared to 128-token outputs, 256-token sequences generated by WorldCup reduce PPL on average by **2.01** and **1.82** on LLaMA3-8B and Gemma2-9B, respectively.

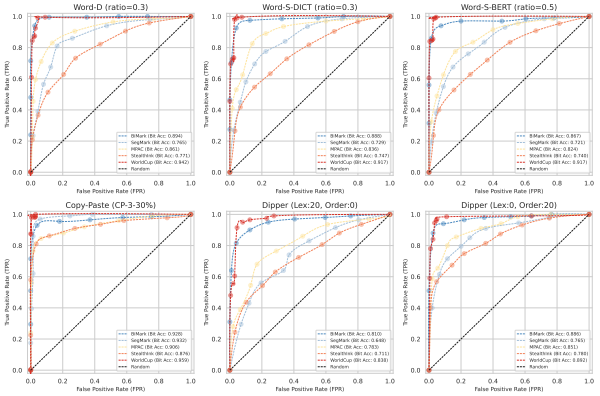

Figure 4: The AUROC under various attacks on LLaMA3-8B-Base with 16-bit messages and 256 tokens.

Downstream Task Performance. As summarized in Table 2, WorldCup balances message recovery and task performance across tasks. Text quality remains competitive with other watermarking methods, although some tasks incur a loss relative to unwatermarked generation. Meanwhile, WorldCup attains the highest multi-bit decoding accuracy and outperforms baselines on several tasks, especially those involving longer outputs. For example, in the long-form QA task, it surpasses MPAC by an average of **2.6%**, **2.7%**, and **3.0%** on Ministral, LLaMA3.1, and Gemma2, respectively. These results support a favorable balance between bit accuracy and text quality across the evaluated downstream tasks.

4.4 ROBUSTNESS TO VARIOUS ATTACKS

Attack Settings. We assess robustness under three common attacks (Kirchenbauer et al., 2024): editing (word deletion and synonym substitution), copy-paste, and paraphrasing using Dipper (Krishna et al., 2023) using 256-token watermarked texts. Full settings are provided in Appendix F.

As shown in Fig. 4, Fig. 8, and Fig. 9, WorldCup generally improves AUC and bit accuracy under word-level editing and copy-paste attacks. In Fig. 4, it reaches an average AUC of **98.9%** and

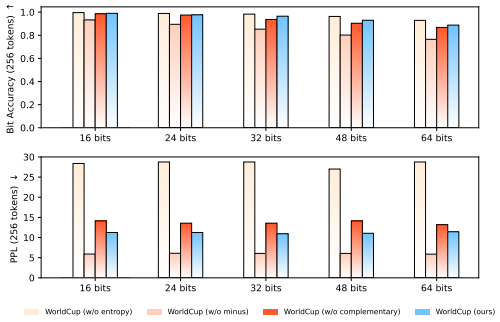

Figure 5: The ablation study of WorldCup.

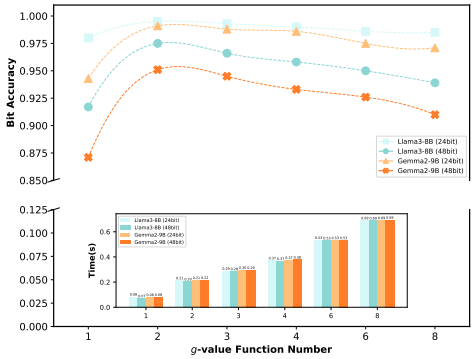

Figure 6: Effect of varying g-value function numbers k on accuracy and encoding time.

decoding accuracy of **91.1%**, outperforming BiMark, MPAC, StealthInk, and SegMark by **3.2%**, **6.7%**, **14%**, and **15.1%**, respectively. However, back-translation and strong paraphrasing disrupt bit assignment from context and reduce detectability and bit recovery. Since higher payloads leave fewer tokens per bit, reliable extraction generally requires longer sequences. Appendix I.6 further shows that larger hash windows reduce robustness by weakening local signal aggregation.

5 Further Analysis

5.1 Ablation Study

Settings. To assess the contribution of each design in WorldCup, i.e., $q - \lambda\bar{q}$, where q denotes the distribution induced by k groups of g-value functions, we perform a component ablation on LLaMA3-8B-Base. We evaluate three variants: (i) removing the entropy-aware factor λ (**WorldCup w/o entropy**, i.e., $q - \bar{q}$); (ii) disabling the subtraction of complementary g-value distributions (**WorldCup w/o minus**, i.e., q); and (iii) replacing complementary g-value functions with randomly sampled ones (**WorldCup w/o complementary**, i.e., $q - \lambda q'$).

As shown in Fig. 5, the entropy-aware factor is key to preserving text quality: it slightly reduces decoding accuracy but substantially lowers PPL. Sampling from the original g-values without subtracting complements also gives low PPL, but weakens discriminability and hurts decoding accuracy. Complementary g-value functions consistently outperform random ones for both bit 0 and bit 1 in decoding accuracy and text quality. This empirical comparison complements the score-level analysis in Appendix G. Overall, the results confirm the importance and synergy of each component.

5.2 Computational Cost Analysis

For **encoding**, we measure time and bit accuracy when embedding k bits *per token* with k groups of g-value functions (Fig. 6). Encoding time grows roughly linearly with k: generating one token takes **0.08**s at $k = 1$ and **0.69**s at $k = 8$ for 256-token sequences. However, accuracy does not improve monotonically because each token has limited capacity and $k = 2$ provides the best empirical trade-off among the evaluated settings. For **decoding**, we report time per sample, including z-score computation and message recovery (Table 1). WorldCup requires **0.01**$\sim$**0.02**s per sample on average through parallel aggregation of token scores and message positions. Decoding is generally slower than SynthID+MPAC but faster than the other evaluated baselines across both backbones.

6 Conclusion

We propose WorldCup, a multi-bit watermarking framework that incorporates message encoding directly into the tournament sampling process during text generation. By combining tournament-style sampling with entropy-aware modulation and confidence-aware decoding, it achieves high-capacity, robust watermarking without sacrificing text quality, outperforming prior methods in accuracy, efficiency, and robustness. We believe that WorldCup provides a principled foundation for scalable multi-bit watermarking and supports practical deployment for reliable LLM content attribution.

AI USE STATEMENT

In preparing this manuscript, we used generative AI tools only for language editing, including grammar correction, spelling correction, word choice, and sentence-level polishing. We did not use generative AI tools to formulate the research idea, design the proposed method, make technical or experimental decisions, derive theoretical results, implement code, analyze data, generate experimental results, or produce figures and tables. All ideas, methodological choices, theoretical analyses, experiments, and conclusions were developed and verified by the authors. All AI-assisted text was carefully reviewed, edited, and approved by the authors. We take full responsibility for the final content of this work, including all claims, analyses, results, and artifacts.

ETHICS STATEMENT

This paper presents WorldCup, a multi-bit watermarking framework for large language models that enables reliable content attribution and integrity verification. By supporting robust message embedding while preserving text quality, WorldCup promotes accountable and transparent LLM deployment. At the same time, we acknowledge potential risks such as privacy concerns or misuse for surveillance and encourage responsible research and governance to ensure watermarking serves transparency and user rights.

REPRODUCIBILITY STATEMENT

We fully disclose the information necessary to reproduce the main experimental results. The experimental setup, evaluation protocol, implementation details, model configurations, and metrics are described in Section 4.1. Detailed baseline configurations are provided in Appendix B, dataset descriptions and processing details are provided in Appendix C, and attack settings are provided in Appendix F. For the theoretical results, we state the assumptions and provide complete proofs in Appendix G. We also include the data, code, and detailed reproduction instructions in the supplementary material to facilitate independent reproduction of the main experimental results.

REFERENCES

Scott Aaronson. Watermarking of large language models. In *Large Language Models and Transformers Workshop at Simons Institute for the Theory of Computing, 2023.*, 2023. URL https://www.youtube.com/watch?v=2Kx9jbSMZqA.

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

Anthropic. Claude code overview. https://code.claude.com/docs, 2026. Official Claude Code documentation. Accessed April 12, 2026.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. Findings of the 2019 conference on machine translation (WMT19). In Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, André Martins, Christof Monz, Matteo Negri, Aurélie Névéol, Mariana Neves, Matt Post, Marco Turchi, and Karin Verspoor (eds.), *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pp. 1–61, Florence, Italy, August 2019. Association for Computational Linguistics. doi: 10.18653/v1/W19-5301. URL https://aclanthology.org/W19-5301/.

Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, et al. International ai safety report. *arXiv preprint arXiv:2501.17805*, 2025.

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. *arXiv preprint arXiv:2406.10281*, 2024.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ruibo Chen, Yihan Wu, Junfeng Guo, and Heng Huang. Improved unbiased watermark for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 20587–20601, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.1005. URL https://aclanthology.org/2025.acl-long.1005/.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL https://lmsys.org/blog/2023-03-30-vicuna/.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Aloni Cohen, Alexander Hoover, and Gabe Schoenbach. Watermarking language models for many adaptive users. In *2025 IEEE Symposium on Security and Privacy (SP)*, pp. 2583–2601. IEEE, 2025.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, et al. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*, 2025.

Cursor. Cursor: The best way to code with AI. https://cursor.com/, 2026. Official product website. Accessed April 12, 2026.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/.

Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. Publicly-detectable watermarking for language models. *arXiv preprint arXiv:2310.18491*, 2023.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. ELI5: Long form question answering. In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3558–3567, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1346. URL https://aclanthology.org/P19-1346/.

Xiaoyan Feng, He Zhang, Yanjun Zhang, Leo Yu Zhang, and Shirui Pan. Bimark: Unbiased multilayer watermarking for large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=Zvyb3WAg03.

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

Jiayi Fu, Xuandong Zhao, Ruihan Yang, Yuansen Zhang, Jiangjie Chen, and Yanghua Xiao. GumbelSoft: Diversified language model watermarking via the GumbelMax-trick. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 5791–5808, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.315. URL https://aclanthology.org/2024.acl-long.315/.

Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Richard W Hamming. Error detecting and error correcting codes. *The Bell system technical journal*, 29(2):147–160, 1950.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28, 2015.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4067–4082, Mexico City, Mexico, June 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.226. URL https://aclanthology.org/2024.naacl-long.226/.

Abe Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-SemStamp: A clustering-based semantic watermark for detection of machine-generated text. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1706–1715, Bangkok, Thailand, August 2024b. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.98. URL https://aclanthology.org/2024.findings-acl.98/.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*, 2023.

Haoyu Jiang, Xuhong Wang, Ping Yi, Shanzhe Lei, and Yilun Lin. Credid: Credible multi-bit watermark for large language models identification, 2025a. URL https://arxiv.org/abs/2412.03107.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. StealthInk: A multi-bit and stealthy watermark for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 27685–27709. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/jiang25j.html.

Ya Jiang, Massieh Kordi Boroujeny, Surender Suresh Kumar, and Kai Zeng. Mirrormark: A distortion-free multi-bit watermark for large language models. 2026.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=DEJIDCmWOz.

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

Gregory Kang Ruey Lau, Xinyuan Niu, Hieu Dao, Jiangwei Chen, Chuan-Sheng Foo, and Bryan Kian Hsiang Low. Waterfall: Scalable framework for robust text watermarking and provenance for LLMs. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 20432–20466, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.1138. URL https://aclanthology.org/2024.emnlp-main.1138/.

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. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 4890–4911, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.268. URL https://aclanthology.org/2024.acl-long.268/.

Boquan Li, Zirui Fu, Mengdi Zhang, Peixin Zhang, Jun Sun, and Xingmei Wang. Efficient and universal watermarking for llm-generated code detection, 2025. URL https://arxiv.org/abs/2402.07518.

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

Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pp. 74–81, 2004.

Aiwei Liu, Leyi Pan, Xuming Hu, Shiao Meng, and Lijie Wen. A semantic invariant robust watermark for large language models. In *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=6p8lpe4MNf.

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

Alexander H Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026.

Yijian Lu, Aiwei Liu, Dianzhi Yu, Jingjing Li, and Irwin King. An entropy-based text watermarking detection method. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11724–11735, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.630. URL https://aclanthology.org/2024.acl-long.630/.

Junyu Luo, Weizhi Zhang, Ye Yuan, Yusheng Zhao, Junwei Yang, Yiyang Gu, Bohan Wu, Binqi Chen, Ziyue Qiao, Qingqing Long, et al. Large language model agent: A survey on methodology, applications and challenges. *arXiv preprint arXiv:2503.21460*, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11): 39–41, 1995.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *International conference on machine learning*, pp. 24950–24962. PMLR, 2023.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

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.

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

Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

Yubing Ren, Ping Guo, Yanan Cao, and Wei Ma. Subtle signatures, strong shields: Advancing robust and imperceptible watermarking in large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5508–5519, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.327. URL https://aclanthology.org/2024.findings-acl.327/.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. The science of detecting llm-generated text. *Communications of the ACM*, 67(4):50–59, 2024.

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

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. WaterBench: Towards holistic evaluation of watermarks for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1517–1542, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.83. URL https://aclanthology.org/2024.acl-long.83/.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Apurv Verma, Hai Phan, and Shubhendu Trivedi. Watermarking degrades alignment in language models: Analysis and mitigation. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=w2ATKQcfWx.

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 *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=JYu5Flqm9D.

Xuhong Wang, Haoyu Jiang, Yi Yu, Jingru Yu, Yilun Lin, Ping Yi, Yingchun Wang, Yu Qiao, Li Li, and Fei-Yue Wang. Building intelligence identification system via large language model watermarking: a survey and beyond. *Artificial Intelligence Review*, 58(8):249, 2025a.

Yidan Wang, Yubing Ren, Yanan Cao, and Binxing Fang. From trade-off to synergy: A versatile symbiotic watermarking framework for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10306–10322, Vienna, Austria, July 2025b. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.509. URL https://aclanthology.org/2025.acl-long.509/.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. Dipmark: A stealthy, efficient and resilient watermark for large language models. 2023.

Xiaojun Xu, Jinghan Jia, Yuanshun Yao, Yang Liu, and Hang Li. Robust multi-bit text watermark with LLM-based paraphrasers. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=DVjkling5x.

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 (eds.), *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, 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/.

Zhuohao Yu, Xingru Jiang, Weizheng Gu, Yidong Wang, Qingsong Wen, Shikun Zhang, and Wei Ye. SAEMark: Steering personalized multilingual LLM watermarks with sparse autoencoders. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=tXnyVPNOfa.

Or Zamir. Excuse me, sir? your language model is leaking (information). *arXiv preprint arXiv:2401.10360*, 2024.

Ruisi Zhang, Shehzeen Samarah Hussain, Paarth Neekhara, and Farinaz Koushanfar. Remark-llm: A robust and efficient watermarking framework for generative large language models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 1813–1830, 2024.

Xuandong Zhao, Prabhanjan Vijendra Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=SsmT8aO45L.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623, 2023.

A Related Work

Existing LLM watermarking methods can be broadly categorized into zero-bit watermarking and multi-bit watermarking approaches (Liu et al., 2024b), depending on whether message bits are explicitly embedded.

A.1 Zero-bit Watermarking

The pioneering KGW method (Kirchenbauer et al., 2023) first introduced token-level watermarking by partitioning the vocabulary into "green" and "red" token lists and modifying the logits distribution during inference to embed a watermark signal. To enhance robustness, Unigram (Zhao et al., 2024) adopted a globally fixed vocabulary partition, while subsequent works (Liu et al., 2024a; Ren et al., 2024; Hou et al., 2024b;a; Verma et al., 2026) leveraged semantic and frequency-based features to defend against editing attacks. To mitigate text quality degradation, several studies proposed entropy-based watermarking schemes (Lu et al., 2024; Lee et al., 2024; Wang et al., 2025b), while others explored unbiased reweighting strategies (Wu et al., 2023; Hu et al., 2023; Chen et al., 2025). Additionally, Dathathri et al. (2024); Aaronson (2023); Fu et al. (2024) designed alternative token sampling mechanisms that preserve the original logits distribution. Overall, zero-bit watermarking can only determine whether a text contains a watermark, limited in broader scenarios such as content provenance tracing.

A.2 Multi-bit Watermarking

Existing work (Fernandez et al., 2023; Jiang et al., 2025a) embedded multi-bit message by establishing a mapping between predefined watermark keys and message bits. Wang et al. (2024); Cohen et al. (2025) divide both the text and message into multiple independent blocks, sequentially encoding each message segment into a corresponding text block. Although these methods achieve moderate decoding accuracy, they require enumerating all candidate messages during decoding, resulting in high exponential computational complexity with respect to message length (i.e., $O(2^b)$ for b bits). To overcome this limitation, Zhang et al. (2024), Xu et al. (2025), Lau et al. (2024) explore training-based and post-hoc approaches, while Yu et al. (2025) propose a black-box method for generating sentence-level watermarks. In contrast, MPAC (Yoo et al., 2024) assigns distinct message bits to different tokens via hash functions and extends the KGW framework accordingly. Building upon this idea, BiMark (Feng et al., 2025), StealthInk (Jiang et al., 2025b), MirroMark (Jiang et al., 2026), Zamir (2024) and Boroujeny et al. (2024) developed distortion-free multi-bit watermark variants, while Qu et al. (2025) introduced error-correcting codes (Chao et al., 2024; Fairoze et al., 2023; Li et al., 2025) to further enhance decoding robustness. Despite these advances, current multi-bit watermarking techniques still struggle to jointly optimize capacity, decoding accuracy, efficiency, and text quality.

Table 3: The comparison between **WorldCup** and related work: yes (✔), partial (P), or no (✗).

Multi-bit Watermarking	Detectability	Text Quality	Robustness	Token Capacity	Accuracy	Efficiency
MPAC (Yoo et al., 2024)	✔	P	✗	✔	✔	✔
BiMark (Feng et al., 2025)	✔	✔	P	P	✔	✔
SegMark (Qu et al., 2025)	✔	P	✔	✔	✔	✗
StealthInk (Jiang et al., 2025b)	✔	✔	✗	✔	P	✔
WorldCup (Ours)	✔	✔	✔	✔	✔	✔

B Baselines

For each baseline, we follow the configurations in original papers. The key hyperparameter settings are as follows:

- **BiMark** (Feng et al., 2025): The base scaling factor $\tilde{\delta}$ is set to 1.0, and the number of layers d is 10. The proportion γ of green lists is 0.5 and the window size is 2. The values of c_key and bit_idx_key are 530773 and 283519, respectively.

- **MPAC** (Yoo et al., 2024): We adopt the lefthash scheme. The window size is 2, and the proportion γ of green lists is 0.5. A bias $\delta = 2.0$ is added to the logit scores of green tokens. The hash key is 15485863.
- **SegMark** (Qu et al., 2025): We use the RSBH scheme (balanced segment assignment with ECC). The window size is 2, the proportion γ of green lists is 0.5, and a bias $\delta = 2.0$ is added to the logit of green tokens. The salt key is 35317.
- **StealthInk** (Jiang et al., 2025b): We use "simple_3" seeding scheme with $H = 1$ message chunks. The window size is 2, and the hash key is 15485863.
- **SynthID+MPAC** (Dathathri et al., 2024; Yoo et al., 2024): We use MPAC's hash-based message-position assignment and SynthID tournament sampling. The two g-functions for each bit are sampled independently rather than constructed as a complementary pair. Mean g-values are aggregated within each assigned message position for decoding.

C DATASETS

We follow previous work (Tu et al., 2024) to evaluate our multi-bit watermark method on the following datasets:

- **C4** (Raffel et al., 2020) dataset is a large-scale, high-quality English pretraining corpus constructed by Google from Common Crawl. After extensive cleaning and filtering to remove non-linguistic, low-quality, and duplicate content, it yields roughly 750 GB of clean English text. We use the processed version in https://huggingface.co/datasets/allenai/c4.
- **OpenGen** (Krishna et al., 2023) dataset contains 3,000 blocks of two sentences drawn from the validation split of WikiText-103 (Merity et al., 2016), with the subsequent 300 tokens written by human. We sample 200 instances from this dataset for our experiments. The dataset is in https://github.com/XuandongZhao/Unigram-Watermark.
- **WMT** (Barrault et al., 2019) dataset is a widely recognized benchmark in machine translation, containing parallel corpora from diverse sources and covering multiple language pairs. For our downstream evaluation, we primarily collect 200 samples from the WMT'19 De-En subset, with decoding parameters set to max_new_tokens = 64 and min_new_tokens = 16 (**short input, short output**). We embed **2-bit** message into each generated sample. It can be found at the following link: https://huggingface.co/datasets/wmt/wmt19/viewer/de-en/validation.
- **CNN_DailyMail** (Hermann et al., 2015) dataset is a large-scale English news corpus containing over 300,000 unique articles written by journalists from CNN and the Daily Mail. The current release supports both extractive and abstractive summarization. We collect 200 samples and prompt the model to produce a one-sentence summary, with max_new_tokens = 64 and min_new_tokens = 32 (**long input, short output**), and embed a **16-bit** message into each generated summary. Details of the dataset can be found at: https://huggingface.co/datasets/abisee/cnn_dailymail.
- **ELI5** (Fan et al., 2019) dataset is a long-form QA dataset sourced from the Reddit community "Explain Like I'm Five." It contains 270k diverse questions that require multi-sentence, explanatory answers supported by web evidence. We get 200 data points and set max_new_tokens = 256 and min_new_tokens = 64 (**short input, long output**), and embed a **32-bit** message into each generated answer. We use the processed subset available at https://github.com/THU-KEG/WaterBench/blob/main/data/WaterBench/2-1_longform_qa.jsonl.
- **GSM8K** (Cobbe et al., 2021) dataset consists of 8,500 high-quality grade school math word problems requiring 2-8 steps of reasoning, with answers expressed in natural language. For downstream evaluation, we sample 200 instances and use an 8-shot setting with max_new_tokens = 256 and min_new_tokens = 64 (**long input, long output**), and embed a **4-bit** message into each generated solution. The dataset is available at: https://huggingface.co/datasets/openai/gsm8k.

D METRICS

All evaluation metrics used in our experiments are described in detail below:

- **Bit Accuracy (Bit Acc.)** measures the proportion of correctly extracted bits across all samples. Let $\mathbf{m}^{(j)} = (m_1^{(j)}, \ldots, m_L^{(j)})$ denote the embedded message of length L for the j-th sample, and $\hat{\mathbf{m}}^{(j)} = (\hat{m}_1^{(j)}, \ldots, \hat{m}_L^{(j)})$ be the corresponding extracted message. The Bit Accuracy over n samples is defined as

$$\text{Bit Acc.} = \frac{1}{nL} \sum_{j=1}^{n} \sum_{i=1}^{L} \mathbb{I}\big[\hat{m}_i^{(j)} = m_i^{(j)}\big] \tag{8}$$

  where $\mathbb{I}[\cdot]$ denotes the indicator function. It reflects fine-grained bit-level decoding performance, as shown in Table 1.

- **Message Extracted Rate (ME Rate)** quantifies the probability of perfectly recovering the entire embedded message. A message is considered successfully extracted if and only if all its bits are correctly recovered. Formally, the Message Extracted Rate over n samples is defined as:

$$\text{ME Rate} = \frac{1}{n} \sum_{j=1}^{n} \mathbb{I}\big[\hat{\mathbf{m}}^{(j)} = \mathbf{m}^{(j)}\big]. \tag{9}$$

  It is a strict metric that penalizes any bit error and reflects end-to-end message recovery reliability, as shown in Table 4.

- **F1 Score** is the harmonic mean of precision and recall:

$$\text{F1} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{10}$$

  Best F1 Score denotes the maximum F1 score obtained over a threshold sweep, commonly used for evaluating binary classifiers without fixing a specific decision threshold.

- **AUROC Curve** (Receiver Operating Characteristic) plots **TPR** (True Positive Rate) against **FPR** (False Positive Rate) under varying decision thresholds. The area under this curve (**AUC**) summarizes the watermark detector's ranking ability, with values closer to 1 indicating stronger discriminative performance.

- **Perplexity (PPL)** measures how well a language model predicts a given text. For a sequence $x_{1:T}$:

$$\text{PPL} = \exp\left(-\frac{1}{T} \sum_{t=1}^{T} \log p(x_t \mid x_{<t})\right) \tag{11}$$

  Lower perplexity implies more confident and accurate language modeling. We report median PPL rather than mean PPL, as it provides more stable estimates and is less sensitive to extreme values (Jiang et al., 2025b).

- **BLEU** (Papineni et al., 2002) (Bilingual Evaluation Understudy) is a standard automatic metric that quantifies lexical similarity by computing n-gram precision between machine-generated translations and human reference texts, with a brevity penalty to discourage overly short outputs.

- **Pass@K** (Chen et al., 2021) measures the probability that at least one of the k generated solutions is correct. In this work, we follow the standard setting and report the results of GSM8K dataset using **Pass@1**.

- **Cosine similarity** computes the cosine of the angle between two vectors to measure their semantic similarity. We use Sentence-BERT (Reimers & Gurevych, 2019) to obtain sentence embeddings and apply cosine similarity to quantify the semantic closeness of natural texts and AI-generated texts.

- **ROUGE Score** (Lin, 2004) measures overlap between generated and reference text. Among its variants, we use ROUGE-L, which computes the longest common subsequence (LCS) between the candidate and reference, capturing sentence-level structural similarity beyond contiguous n-gram overlap.

- **Log Diversity** quantifies textual diversity by measuring n-gram uniqueness. For each $n \in \{2, 3, 4\}$, let unique_n and $\text{total}_n > 0$ denote the numbers of distinct and total n-grams in text split on whitespace, respectively. We aggregate their uniqueness ratios as

$$\text{LogDiversity} = -\log\left(\max\left(1 - \prod_{n\in\{2,3,4\}} \frac{\text{unique}_n}{\text{total}_n},\, e^{-20}\right)\right). \tag{12}$$

Each uniqueness ratio lies in $[0, 1]$. The floor e^{-20} prevents $\log 0$ and caps the score at 20; higher scores indicate more diverse, less repetitive text.

- **GPT4 Score** (Zheng et al., 2023) leverages GPT-4 (Achiam et al., 2023) as an evaluator. The model is prompted to rate the quality, correctness, or faithfulness of generated text relative to a reference or specification. This human-aligned evaluation correlates closely with expert judgments. The scoring template we use is as follows:

GPT-4 Judge Template

You are an strict text quality evaluator. Your task is to compare a candidate text against a reference text (which serves as the ground truth) and produce a single final quality score.
Evaluation Criteria:
1. Fluency (Naturalness): How natural, grammatical, and readable the candidate text is.
2. Adequacy: How well the candidate preserves the meaning of the reference.
3. Coherence: How logically consistent and well-structured the candidate text is.
4. Relevance: How well the content matches the intent and key information of the reference.
5. Style Consistency: How closely the candidate matches the tone and style of the reference.
Scoring:
- For each criterion, assign a score from 1 to 10.
- Compute the final score as the average of all criterion scores.
- Output ONLY the final numerical score (e.g., 3.8). Do not explain, justify, or output intermediate scores.
Reference Text: xxx Candidate Text: xxx Score:

E Backbone Models

We primarily employ the following backbone models in our experiments:

- **LLaMA3** family (Grattafiori et al., 2024) is developed by Meta, built upon an optimized Transformer architecture, which includes both pre-trained and instruction-tuned generative text models with sizes of 8B and 70B parameters. Both the 8B and 70B variants adopt Grouped Query Attention (GQA) to improve inference scalability. In this paper, we use the LLaMA3-8B Base and LLaMA3.1-8B-Instruct version, more details can be found in https://huggingface.co/collections/meta-llama/meta-llama-3.
- **Gemma2** family (Team et al., 2024) is a series of lightweight open-source models released by Google, developed using the same research foundations and technologies as the Gemini models. These models are text-to-text, decoder-only LLMs that currently support English, and are suitable for a wide range of text generation tasks. In this paper, we mainly use the Gemma2-9B-Base and Gemma2-9B-Instruct version, more details can be found in https://huggingface.co/google/gemma-2-9b.
- **Ministral** family (Liu et al., 2026) belongs to Mistral AI's latest third-generation models released in 2025, which includes three state-of-the-art small, dense models (3B, 8B, and 14B). These models support applications that understand text, images, and logic across 40+ languages, and can be used for coding, collaboration, or document analysis. In this paper, we utilize the Ministral-8B-Instruct, more details can be found in https://huggingface.co/mistralai/Ministral-8B-Instruct-2410.

F Attack Settings

To evaluate the robustness of the watermark, we design the following various attack scenarios:

- **Word-D (ratio = ρ)**: Randomly deletes a proportion ρ of words from the watermarked text.
- **Word-S-Dict (ratio = ρ)**: Randomly replaces a proportion ρ of words with their synonyms based on the WordNet (Miller, 1995) lexical dictionary.
- **Word-S-BERT (ratio = ρ)**: Randomly substitutes a proportion ρ of words with context-aware synonyms generated by a BERT-based (Devlin et al., 2019) model.
- **Copy–Paste ($n - \rho$)**: Randomly splits the non-watermarked text into n segments and inserts them into watermarked text, such that the inserted non-watermarked content accounts for a total proportion ρ.
- **Translation (en–zh)**: Translates the watermarked text from English to Chinese and then back to English using a fine-tuned T5 translation model: https://huggingface.co/utrobinmv/t5_translate_en_ru_zh_small_1024.
- **Rephrase (GPT-4o)**: Rewrites the watermarked text using the GPT-4o API with the $\tau = 0.7$.
- **Rephrase (DIPPER-1)**: Rephrases the watermarked text using the DIPPER model with setting one (lex_diversity=ρ_1 order_diversity=ρ_2, max_new_tokens=256, sent_interval=1, top_p=0.75).
- **Rephrase (DIPPER-2)**: Rephrases the watermarked text using the DIPPER model with setting two (lex_diversity=ρ_1' order_diversity=ρ_2', max_new_tokens=256, sent_interval=1, top_p=0.75).

G Theoretical Results

G.1 Proof of Proposition 3.5

Proof. For a fixed token x and layer ℓ, let $G_0 = g_\ell^{(0)}(x, R)$ and $G_1 = g_\ell^{(1)}(x, R)$ denote the scores for bits 0 and 1, where R is the random seed. Both variables have Bernoulli(0.5) marginals; their joint distribution may vary. Expectations below are over R.

Step 1: Discriminability at the scoring level. We first quantify the separation between the two encoding hypotheses at the scoring level by the expected squared difference:

$$D \triangleq \mathbb{E}\big[(G_1 - G_0)^2\big]. \tag{13}$$

Expanding this expression yields:

$$D = \mathbb{E}(G_1^2) + \mathbb{E}(G_0^2) - 2\mathbb{E}[G_1 G_0] \tag{14}$$

Let $\mu_i = \mathbb{E}[G_i]$ and $\sigma_i^2 = \mathrm{Var}(G_i)$ for $i \in \{0, 1\}$. Rewriting Eq. 14 in mean–variance form gives

$$D = \sigma_1^2 + \sigma_0^2 - 2\,\mathrm{Cov}(\mathrm{G}_1, \mathrm{G}_0) + (\mu_1 - \mu_0)^2 \tag{15}$$

With identical marginals ($\mu_0 = \mu_1 = \mu$, $\sigma_0^2 = \sigma_1^2 = \sigma^2$), Eq. 15 simplifies to

$$D = 2\sigma^2 - 2\,\mathrm{Cov}(G_1, G_0) \tag{16}$$

For fixed marginals, D is maximized by minimizing the covariance between G_0 and G_1. By the Cauchy-Schwarz inequality:

$$\mathrm{Cov}(G_1, G_0) \geq -\sqrt{\mathrm{Var}(G_1)\,\mathrm{Var}(G_0)} = -\sigma^2 \tag{17}$$

with equality if and only if G_1 and G_0 are perfectly anti-correlated. Substituting this bound into Eq. 16 gives the upper bound

$$D \leq 2\sigma^2 - 2(-\sigma^2) = 4\sigma^2. \tag{18}$$

Step 2: Consistency with equal-mean constraint. Perfect anti-correlation implies $G_1 = a - G_0$ almost surely. Enforcing $\mathbb{E}[G_1] = \mathbb{E}[G_0] = \mu$ gives $a = 2\mu$, and hence:

$$G_1 = 2\mu - G_0. \tag{19}$$

Under the symmetric Bernoulli setting used in our experiments ($\mu = 0.5$), this reduces to the complementary construction $G_1 = 1 - G_0$.

Step 3: Attainment and scope. For Bernoulli(0.5) marginals, $\sigma^2 = 1/4$, so $D \leq 1$. The complementary construction attains $D = 1$, since $(G_1 - G_0)^2 = 1$ almost surely. Conversely, for binary scores, $D = \mathbb{P}(G_0 \neq G_1)$; hence $D = 1$ requires $G_1 = 1 - G_0$ almost surely. This proves the claimed optimality among couplings of the two score variables.

The result concerns score-level separation. The distinguishability of the induced token distributions also depends on the base distribution and tournament parameters, and requires a separate analysis for any chosen statistical distance.

G.2 Vectorized WorldCup Sampling

In the multi-bit WorldCup watermark, once a message bit is assigned to each token, the corresponding g-value function is uniquely determined. Consequently, conditioning on the assigned message bits, the generation of each token is distributionally equivalent to that of a zero-bit watermark. This observation allows us to directly adopt the analytical framework of SynthID (Dathathri et al., 2024).

Let $p(\cdot)$ denote the base language model distribution over the vocabulary V. For any token x_t, random seed r, and g-value function $g_\ell(\cdot, r)$ at layer ℓ, we define:

$$\begin{aligned} p\left(\mathcal{V}^{=g_\ell(x_t,r)}\right) &:= \sum_{x\in\mathcal{V}:\, g_\ell(x,r)=g_\ell(x_t,r)} p(x), \\ p\left(\mathcal{V}^{<g_\ell(x_t,r)}\right) &:= \sum_{x\in\mathcal{V}:\, g_\ell(x,r)<g_\ell(x_t,r)} p(x), \\ p\left(\mathcal{V}^{\leq g_\ell(x_t,r)}\right) &:= \sum_{x\in\mathcal{V}:\, g_\ell(x,r)\leq g_\ell(x_t,r)} p(x). \end{aligned} \tag{20}$$

Theorem G.1. *(Vectorized form, single-layer WorldCup sampling). Given a probability distribution P_Θ over $\mathcal{V}$, random seed $r \in \mathcal{R}$, g-value distribution $\mathbf{g}_0$ and $\mathbf{g}_1$, and the number of leaves $N \geq 2$, message $\mathbf{m}$, the watermarked distribution $q(\cdot \mid P_\Theta, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1)$ for $m = 1$ is given by:*

$$q(\cdot \mid P_\Theta, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1) = \begin{cases} p(x_t)\left(\frac{p\left(V^{\leq g(x_t,r)}\right)^N - p\left(V^{<g(x_t,r)}\right)^N}{p(V=g(x_t,r))}\right) & \text{if } p(x_t) \neq 0 \\ 0 & \text{if } p(x_t) = 0 \end{cases} \tag{21}$$

where $g = \mathbf{g}_0 = g_1^{(0)}$ if $\mathbf{m}[p] = 0$ else $g = \mathbf{g}_1 = g_1^{(1)}$ when $m = 1$

Proof. We first note that if $p(x_t) = 0$, then $\mathbb{P}(\text{Alg. 1 returns } x_t) = 0$. Hence, we assume $p(x_t) > 0$ in the following derivation. In a single-layer tournament, N samples participate in the selection. Let $|Y^*| = j$ denote the number of tokens in the winning set, and suppose that x_t appears i times among these j tokens (pairwise comparison is not required). The probability that outputs x_t is:

$$\begin{aligned} \mathbb{P}(\text{Alg. 1 returns } x_t) &= \sum_{j=1}^{N}\sum_{i=1}^{j} \mathbb{P}(|Y^*| = j,\ x_t \text{ appears } i \text{ times in } Y^*,\ \text{Alg. 1 returns } x_t) \\ &= \sum_{j=1}^{N}\sum_{i=1}^{j} \binom{N}{j} p\left(V^{<g(x_t,r)}\right)^{N-j} \binom{j}{i} p(x_t)^i p\left(V^{=g(x_t,r)} \setminus x_t\right)^{j-i} \frac{i}{j}. \end{aligned} \tag{22}$$

Rearranging the summations yields:

$$\mathbb{P}(\text{Alg. 1 returns } x_t) = \sum_{j=1}^{N} \binom{N}{j} p\Big(V^{<g(x_t,r)}\Big)^{N-j} \sum_{i=1}^{j} \binom{j}{i} \frac{i}{j} p(x_t)^i p\Big(V^{=g(x_t,r)} \setminus x_t\Big)^{j-i}. \quad (23)$$

Lemma G.2.

$$\sum_{i=1}^{j} \binom{j}{i} \frac{i}{j} a^i b^{j-i} = a\,(a+b)^{j-1} \quad (24)$$

Using the identity G.2, and letting $a = p(x_t)$ and $b = p\big(V^{=g(x_t,r)} \setminus x_t\big)$, we obtain:

$$\sum_{i=1}^{j} \binom{j}{i} \frac{i}{j} p(x_t)^i p\Big(V^{=g(x_t,r)} \setminus x_t\Big)^{j-i} = p(x_t)\, p\Big(V^{=g(x_t,r)}\Big)^{j-1}. \quad (25)$$

Substituting back gives:

$$\begin{aligned} \mathbb{P}(\text{Alg. 1 returns } x_t) &= \sum_{j=1}^{N} \binom{N}{j} p\Big(V^{<g(x_t,r)}\Big)^{N-j} p(x_t)\, p\Big(V^{=g(x_t,r)}\Big)^{j-1} \\ &= \frac{p(x_t)}{p\big(V^{=g(x_t,r)}\big)} \sum_{j=1}^{N} \binom{N}{j} p\Big(V^{<g(x_t,r)}\Big)^{N-j} p\Big(V^{=g(x_t,r)}\Big)^{j}. \end{aligned} \quad (26)$$

Applying the binomial theorem finally yields:

$$\mathbb{P}(\text{Alg. 1 returns } x_t) = \frac{p(x_t)}{p\big(V^{=g(x_t,r)}\big)} \Big(p\Big(V^{\le g(x_t,r)}\Big)^{N} - p\Big(V^{<g(x_t,r)}\Big)^{N} \Big). \quad (27)$$

In particular if $N = 2$, then:

$$\mathbb{P}(\text{Alg. 1 returns } x_t) = p(x_t) \Big[p\Big(V^{=g(x_t,r)}\Big) + 2p(V^{<g(x_t,r)}) \Big] \quad (28)$$

When the g-value distribution $\mathbf{g}$ is binary (i.e., $g \in \{0,1\}$), the watermark distribution induced by a single-layer tournament with N candidates is given by

$$q(\cdot \mid P_\Theta, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1) = \begin{cases} p(x_t)\, p\big(V^{g=0}\big)^{N-1}, & \text{if } g(x_t,r) = 0, \\ p(x_t)\, \dfrac{1 - p\big(V^{g=0}\big)^{N}}{p(V^{g=1})}, & \text{if } g(x_t,r) = 1, \end{cases} \quad (29)$$

where $p\big(V^{g=0}\big) := \sum_{x \in V:\, g(x,r)=0} p(x)$ and $p\big(V^{g=1}\big) := \sum_{x \in V:\, g(x,r)=1} p(x)$.

By a straightforward induction on the number of layers, the above result extends directly to the m-layer WorldCup sampling:

Theorem G.3. *(Vectorized form, multi-layer WorldCup sampling). Given a single-layer WorldCup sampling distribution $q(P_\Theta, g(\cdot), N)$:*

$$\begin{aligned} q^{(1)}(\cdot) &:= q(\cdot \mid P_\Theta, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1) \\ q^{(2)}(\cdot) &:= q(\cdot \mid q^{(1)}, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1) \\ &\cdots \\ q^{(m)}(\cdot) &:= q(\cdot \mid q^{(m-1)}, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1) \end{aligned} \quad (30)$$

It follows that $q^{(m)}(\cdot)$ is equal to the m-layer WorldCup watermarked distribution $q(\cdot \mid Alg.\ 1$ return $x_t)$

Proof. The above theorem follows straightforwardly by induction on m. The case $m = 1$ is ensured by Theorem G.1. Assume that the statement holds for $m-1$. For an m-layer tournament, we may equivalently first execute N-many $(m-1)$-layer tournaments and then apply a single-layer tournament to the resulting winners via $g_m(\cdot, r)$. By the induction assumption, the N winners are drawn from $q^{(m-1)}(\cdot)$ as defined in Eq. 30, and by Theorem G.1 the winner of the single-layer tournament is given by $q(\cdot \mid q^{(m-1)}, r_t, m, N, \mathbf{m}, p, \mathbf{g}_0, \mathbf{g}_1)$.

G.3 k-BIT WORLDCUP WATERMARKING

We can naturally extend Equation 5 to $k > 2$ bits. Let $\mathbf{m}' = (b_0, \ldots, b_{k-1}) \in \{0,1\}^k$. For each $i \in \{0, \ldots, k-1\}$, let q_i and $\bar{q}_i$ be the token distributions induced by the g-value families $\mathbf{g}_i$ and $\bar{\mathbf{g}}_i$, respectively. The unnormalized watermark scores are

$$P_{\Theta,\mathbf{m}} = \sum_{i=0}^{k-1} \big(b_i \bar{q}_i + (1-b_i) q_i\big) - \lambda \sum_{i=0}^{k-1} \big(b_i q_i + (1-b_i)\bar{q}_i\big) \tag{31}$$

Thus, $b_i = 0$ contributes $q_i - \lambda \bar{q}_i$, whereas $b_i = 1$ contributes $\bar{q}_i - \lambda q_i$, consistently with Equation 5.

G.4 WATERMARK DETECTION SCORE

Let $\mathbf{y} = (y_1, \ldots, y_T)$ be a generated text. The shared hash and key assign each evaluated token to a message symbol position $p_t \in \{0, \ldots, B-1\}$, where $B = b/k$. Let $\mathcal{T}_p = \{t : p_t = p\}$, and let $a_t \in \{0,1\}$ be the detector mask for valid tokens. Following the implementation, set $n_p = \max(\sum_{t \in \mathcal{T}_p} a_t, 10^{-8})$ to avoid division by zero. For each group $j \in \{1, \ldots, k\}$, we first aggregate complementary binary scores over the assigned tokens and m tournament layers:

$$\begin{aligned} s_j^p &= \frac{1}{m n_p} \sum_{t \in \mathcal{T}_p} a_t \sum_{\ell=1}^{m} w_\ell g_{\ell,j}(y_t, r_t), \\ \bar{s}_j^p &= \frac{1}{m n_p} \sum_{t \in \mathcal{T}_p} a_t \sum_{\ell=1}^{m} w_\ell \big(1 - g_{\ell,j}(y_t, r_t)\big). \end{aligned} \tag{32}$$

The nonnegative layer weights satisfy $\sum_{\ell=1}^{m} w_\ell = m$. The mean detector uses $w_\ell = 1$; the weighted-mean detector uses the normalized decreasing weights in Appendix I.3. We then select $c_j^p = \max(s_j^p, \bar{s}_j^p)$. Thus, the maximum is taken after aggregation within a message position and group. For a nonempty valid group, $\bar{s}_j^p = 1 - s_j^p$, so $c_j^p = \frac{1}{2} + |s_j^p - \frac{1}{2}|$ varies with the text's agreement across tokens and layers.

Let $p_1, \ldots, p_J$ denote the observed symbol positions in order of their first occurrence in the text, matching the detector's accumulation order. After processing the first h positions, the accumulated selected score and normalization count are

$$A_h = \sum_{u=1}^{h} \frac{m n_{p_u}}{k} \sum_{j=1}^{k} c_j^{p_u}, \qquad N_h = \sum_{u=1}^{h} m n_{p_u}. \tag{33}$$

The implementation reports the largest normalized score over these prefixes:

$$z_h = \frac{A_h - N_h/2}{\sqrt{N_h/4}}, \qquad s(\mathbf{y}) = \max_{1 \le h \le J} z_h. \tag{34}$$

Selecting the larger complementary aggregate introduces a positive bias under H_0, and the maximum over prefixes further changes the null distribution. Layer weighting and dependence between token scores also affect its variance. We therefore use $s(\mathbf{y})$ as a detection score with empirical threshold calibration; the normalization above does not establish a standard Gaussian null distribution.

For every setting, we choose $\tau_{0.01}$ from matched unwatermarked scores so that the empirical FPR under the decision rule $s(\mathbf{y}) \ge \tau_{0.01}$ is no greater than 1%. We report $\mathrm{T@1\%F} = |\mathcal{W}|^{-1} \sum_{\mathbf{y} \in \mathcal{W}} \mathbb{I}[s(\mathbf{y}) \ge \tau_{0.01}]$ in Table 4. For each attack in Figs. 8 and 9, the threshold is estimated only from the corresponding clean unwatermarked outputs and then applied unchanged to the attacked watermarked outputs without further calibration.

H More Experimental Results

Table 4: The Comparison of different multi-bit watermark performance with ME Rate. T@1%F denotes the true positive rate at a fixed 1% false positive rate.

Bit Length	Watermark	LLAMA3-8B-BASE								GEMMA2-9B-BASE							
		MAX 128 TOKENS				MAX 256 TOKENS				MAX 128 TOKENS				MAX 256 TOKENS			
		AUC ↑	ME Rate ↑	PPL ↓	T@1%F ↑	AUC ↑	ME Rate ↑	PPL ↓	T@1%F ↑	AUC ↑	ME Rate ↑	PPL ↓	T@1%F ↑	AUC ↑	ME Rate ↑	PPL ↓	T@1%F ↑
16 bits	MPAC	0.999	0.520	16.25	0.940	0.996	0.785	13.56	0.975	0.980	0.305	13.69	0.875	0.985	0.500	12.00	0.835
	SegMark	0.993	0.815	15.88	0.950	0.995	**0.995**	12.94	0.995	0.979	0.510	13.19	0.710	0.999	**0.905**	10.94	0.985
	BiMark	1.000	0.750	14.34	**1.000**	1.000	0.860	11.25	1.000	1.000	0.410	11.06	**1.000**	0.999	0.620	9.625	**0.990**
	Ours ($k = 1$)	1.000	**0.915**	**7.938**	**1.000**	1.000	0.970	**6.563**	**1.000**	0.998	**0.710**	**7.250**	0.980	0.994	0.860	**6.344**	0.975
	Ours ($k = 2$)	0.998	**0.940**	**12.94**	0.990	1.000	**0.990**	**10.94**	**1.000**	1.000	**0.850**	**10.94**	**1.000**	1.000	**0.965**	**9.563**	**1.000**
24 bits	MPAC	0.996	0.175	16.75	0.930	0.997	0.445	14.00	0.965	0.972	0.085	13.69	0.805	0.959	0.190	12.00	0.785
	SegMark	0.947	0.450	17.13	0.180	0.992	**0.975**	13.38	0.940	0.849	0.190	13.19	0.080	0.974	**0.695**	10.13	0.550
	BiMark	1.000	0.280	14.34	**0.995**	1.000	0.700	**11.34**	**1.000**	0.998	0.085	11.63	0.965	0.990	0.415	9.938	**0.970**
	Ours ($k = 1$)	0.999	**0.505**	**7.938**	0.980	0.996	0.820	**6.625**	0.995	0.998	**0.280**	**7.375**	**0.970**	0.993	0.610	**5.844**	0.955
	Ours ($k = 2$)	1.000	**0.710**	**12.84**	**0.995**	1.000	**0.885**	**10.75**	**1.000**	0.999	**0.500**	**10.44**	**0.980**	1.000	**0.815**	**9.125**	**1.000**
32 bits	MPAC	0.997	0.035	16.75	0.915	0.996	0.255	14.00	0.955	0.947	0.005	13.81	0.640	0.940	0.025	11.63	0.745
	SegMark	0.909	**0.240**	16.63	0.120	0.989	**0.840**	12.94	0.710	0.824	0.025	12.75	0.065	0.955	**0.465**	10.25	0.385
	BiMark	1.000	0.055	13.91	**1.000**	1.000	0.425	12.38	0.995	0.996	0.005	12.00	0.965	0.989	0.115	9.938	0.965
	Ours ($k = 1$)	0.996	0.100	**8.250**	0.990	1.000	0.565	**6.547**	**0.995**	0.996	**0.060**	**7.125**	**0.970**	0.998	0.355	**6.250**	**0.965**
	Ours ($k = 2$)	0.998	**0.390**	**13.28**	**0.995**	1.000	**0.800**	**10.56**	**1.000**	1.000	**0.105**	**10.66**	**1.000**	1.000	**0.565**	**9.500**	**0.995**
48 bits	MPAC	0.993	0.000	16.63	0.935	0.990	0.010	13.19	0.870	0.936	0.000	14.00	0.685	0.914	0.000	12.19	0.735
	SegMark	0.837	0.000	16.38	0.030	0.968	**0.340**	13.56	0.400	0.770	0.000	13.38	0.090	0.900	**0.030**	9.938	0.125
	BiMark	1.000	0.000	14.25	**1.000**	0.999	0.060	11.63	0.985	0.985	0.000	11.53	0.940	0.975	0.000	10.03	0.935
	Ours ($k = 1$)	0.998	0.000	**7.750**	0.985	1.000	0.140	**6.688**	**1.000**	0.983	0.000	**7.000**	**0.950**	0.985	0.010	**5.938**	**0.945**
	Ours ($k = 2$)	0.999	**0.025**	**12.84**	**0.990**	1.000	**0.310**	**11.44**	**1.000**	0.996	**0.015**	**10.50**	**0.955**	0.998	**0.130**	**9.313**	**0.985**

To evaluate end-to-end message recovery, we report ME Rate after Hamming decoding (Hamming, 1950). In Table 4, Bit Length denotes the encoded length n; $r = \lceil \log_2(n+1) \rceil$ positions are reserved for parity and the remaining $n - r$ bits form the information payload. Thus, encoded lengths of 16, 24, 32, and 48 bits correspond to 11, 19, 26, and 42 information bits, respectively. A message is counted as correct only when all information bits are recovered. As reported in Table 4, WorldCup consistently outperforms existing baselines across most settings. Fig.7 shows that our method achieves more reliable detection while preserving better text quality. Additional robustness results (Fig.8 and Fig. 9) indicate that sentence-level attacks (e.g., back-translation and paraphrasing) are more harmful to multi-bit watermarking than word-level attacks (e.g., word deletion and synonym substitution). This is inherent to bit-allocation-based schemes, as sentence-level transformations disrupt the contextual alignment required to recover token-level bit information. Improving robustness to such attacks remains an open challenge.

The numerical results corresponding to Fig. 5 and Fig. 6 are reported in Table 5 and Table 6, respectively. Based on these experimental results, we select the hyperparameter $\alpha = 1.2$ and set the number of g-value functions to 2, as this configuration provides a favorable trade-off among bit accuracy, watermark detectability, and text quality.

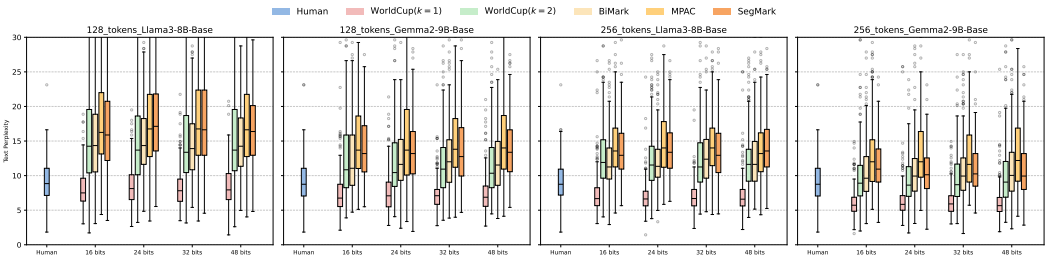

Figure 7: The PPL of different multi-bit watermarking methods on LLaMA3-8B and Gemma2-9B.

I Additional Analysis

I.1 Entropy Analysis

From the results in Table 1, we observe that existing multi-bit watermarking methods achieve substantially lower bit decoding accuracy on Gemma2-9B-Base than on LLaMA3-8B-Base. To better understand this phenomenon, we conduct an entropy-based analysis. Specifically, we visualize the average token-level entropy for each sample during text generation, as illustrated in Fig. 10.

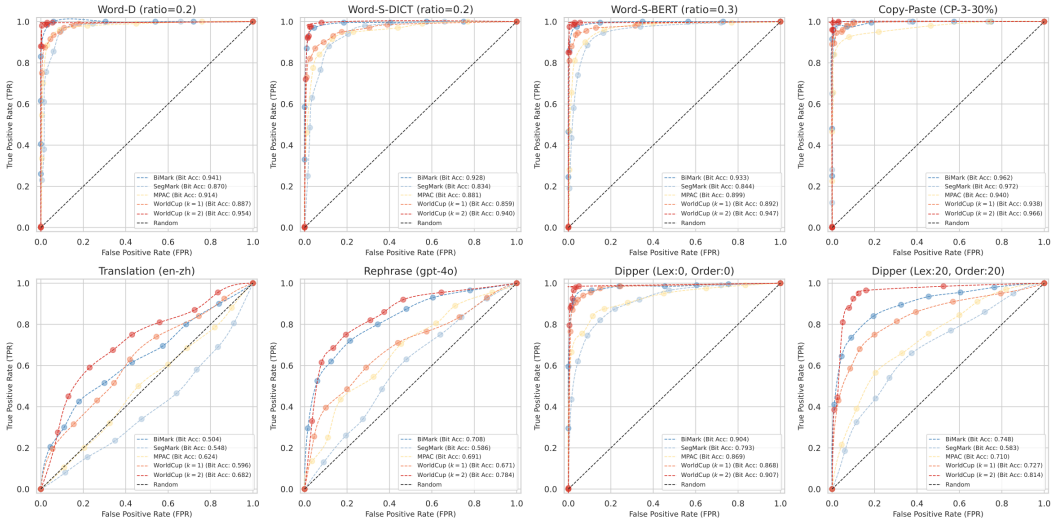

Figure 8: The AUROC curves under different attacks on LLaMA3-8B-Base model (16 bits).

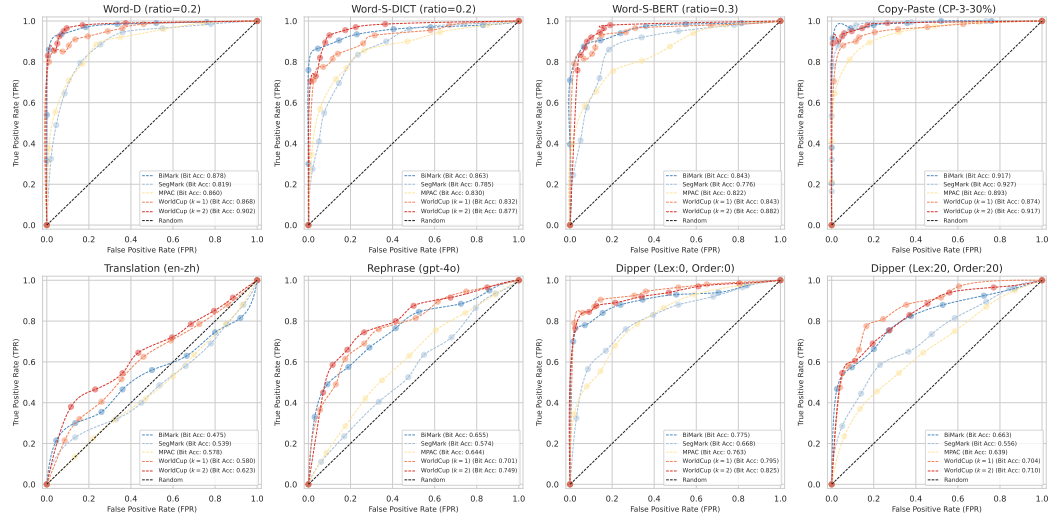

Figure 9: The AUROC curves under different attacks on Gemma2-9B-Base model (16 bits).

Table 5: The numerical results of the ablation study on different components of WorldCup.

Model	C4 Dataset														
	16 bits			24 bits			32 bits			48 bits			64 bits		
+ Watermark	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓
LLAMA3-8B-BASE															
+ WorldCup ($k = 1$, ours)	0.997	0.986	6.688	0.997	0.958	6.688	0.992	0.930	6.844	0.998	0.880	6.563	0.997	0.807	6.594
+ WorldCup ($k = 1$, w/o comp.)	0.995	0.949	6.813	1.000	0.907	6.656	1.000	0.874	7.031	1.000	0.797	6.563	0.998	0.754	6.750
+ WorldCup ($k = 2$, w/o entropy)	1.000	0.996	28.38	1.000	0.989	28.75	1.000	0.983	28.75	1.000	0.963	27.00	0.998	0.929	28.75
+ WorldCup ($k = 2$, w/o minus)	0.940	0.933	5.906	0.946	0.895	6.094	0.910	0.853	6.063	0.884	0.802	6.063	0.870	0.765	5.875
+ WorldCup ($k = 2$, w/o comp.)	0.995	0.986	14.13	0.995	0.975	13.56	0.995	0.937	13.56	0.992	0.904	14.13	0.987	0.868	13.19
+ WorldCup ($k = 2$, ours, λ)	0.995	0.988	9.813	0.980	0.974	10.13	0.985	0.956	9.313	0.953	0.923	9.813	0.954	0.873	9.406
+ WorldCup ($k = 2$, ours, 1.2λ)	0.990	0.990	11.25	0.985	0.977	11.25	0.988	0.965	10.94	0.970	0.930	11.06	0.963	0.888	11.44
+ WorldCup ($k = 2$, ours, 1.5λ)	0.998	0.995	14.81	0.992	0.981	14.81	0.993	0.973	15.38	0.985	0.936	15.38	0.975	0.909	14.44
+ WorldCup ($k = 2$, ours, 2.0λ)	1.000	0.998	49.00	1.000	0.995	49.75	1.000	0.989	48.63	1.000	0.963	49.38	1.000	0.938	49.38

Table 6: The effect of g-value function number on different LLMs.

Model	g_function_num=1		g_function_num=2		g_function_num=3		g_function_num=4		g_function_num=6		g_function_num=8	
	Bit Acc ↑	Enc Time ↓	Bit Acc ↑	Enc Time ↓	Bit Acc ↑	Enc Time ↓	Bit Acc ↑	Enc Time ↓	Bit Acc ↑	Enc Time ↓	Bit Acc ↑	Enc Time ↓
LLAMA3-8B-BASE	-	-	-	-	-	-	-	-	-	-	-	-
WorldCup (24 bit)	0.980	0.083	0.995	0.213	0.993	0.287	0.990	0.371	0.986	0.533	0.985	0.690
WorldCup (48 bit)	0.917	0.069	0.975	0.203	0.966	0.285	0.958	0.366	0.950	0.530	0.939	0.689
GEMMA2-9B-BASE	-	-	-	-	-	-	-	-	-	-	-	-
WorldCup (24 bit)	0.943	0.079	0.991	0.214	0.988	0.296	0.986	0.374	0.975	0.532	0.971	0.691
WorldCup (48 bit)	0.871	0.080	0.951	0.215	0.945	0.294	0.933	0.377	0.926	0.531	0.910	0.693

The results clearly indicate that Gemma2-9B-Base produces tokens with consistently lower entropy, aligning with prior findings that low-entropy text is inherently more challenging for watermark embedding. A plausible explanation is that Gemma2-9B-Base, owing to its larger parameter scale compared to LLaMA3-8B-Base, generates tokens with higher confidence, thereby reducing entropy and limiting the effective embedding capacity for multi-bit watermarks.

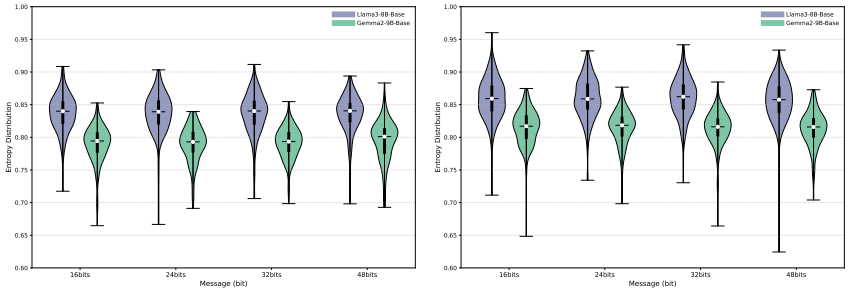

Figure 10: The comparison of spike entropy between LLaMA3-8B-Base and Gemma2-9B-Base.

I.2 DISTORTIONARY AND NON-DISTORTIONARY g-VALUE FUNCTIONS

According to SynthID (Dathathri et al., 2024), we define distortionary and non-distortionary g-value functions as follows:

Definition I.1. A sampling algorithm $\mathcal{S} : \Delta\mathcal{V} \times \mathcal{R} \to \mathcal{V}$ is (single-token) non-distortionary if for any probability distribution $p \in \Delta\mathcal{V}$ and token $x \in \mathcal{V}$:

$$\mathbb{E}_{r\sim\text{Unif}(\mathcal{R})}[\mathbb{P}(\mathcal{S}(p, r) = x)] = p(x) \tag{35}$$

If $\mathcal{S}$ is not non-distortionary, we call it distortionary.

Therefore, following the design of non-distortionary and distortionary g-value functions in SynthID, we evaluate their performance in the multi-bit watermarking setting. The results are summarized in Fig. 11 and Table 7. Specifically, WorldCup-v1, v2, v3, and v4 in the figure correspond to the configurations ($k = 1$, non-distortionary), ($k = 1$, distortionary), ($k = 2$, non-distortionary), and ($k = 2$, distortionary) in the Table 7, respectively. We observe that the non-distortionary setting yields higher cosine similarity between the generated watermarked text and the ground truth compared to the distortionary setting, at the cost of reduced bit accuracy. Consequently, in practical scenarios, the choice between these two settings can be made based on application requirements. Unless otherwise stated, our experiments primarily adopt the distortionary configuration.

I.3 DIFFERENT DETECTORS

Following SynthID Text (Dathathri et al., 2024), we consider two watermark detectors: mean detector (D1) and weighted-mean detector (D2).

For one message position and one g-value family, let x collect its T valid tokens. The mean detector computes the average across these tokens and all layers as

$$\text{MeanScore}(x) := \frac{1}{mT}\sum_{t=1}^{T}\sum_{\ell=1}^{m} g_{t,\ell}, \tag{36}$$

Table 7: The comparison of distortionary g-value function and non-distortionary g-value function.

Model	C4 Dataset														
	50 tokens			100 tokens			150 tokens			200 tokens			250 tokens		
+ Watermark	Best F1 ↑	Bit Acc ↑	Cos Sim ↑	Best F1 ↑	Bit Acc ↑	Cos Sim ↑	Best F1 ↑	Bit Acc ↑	Cos Sim ↑	Best F1 ↑	Bit Acc ↑	Cos Sim ↑	Best F1 ↑	Bit Acc ↑	Cos Sim ↑
LLAMA3-8B-BASE															
+ BiMark	0.972	0.513	0.422	0.998	0.743	0.457	1.000	0.851	0.487	1.000	0.911	0.510	1.000	0.943	0.515
+ WorldCup ($k = 1$, non-distortionary)	0.883	0.463	0.425	0.916	0.647	0.452	0.883	0.719	0.468	0.881	0.754	0.496	0.869	0.767	0.497
+ WorldCup ($k = 1$, distortionary)	0.935	0.494	0.425	0.967	0.705	0.449	0.980	0.799	0.471	0.992	0.858	0.490	0.990	0.889	0.488
+ WorldCup ($k = 2$, non-distortionary)	0.934	0.673	0.395	0.941	0.812	0.433	0.929	0.853	0.454	0.915	0.871	0.479	0.912	0.879	0.478
+ WorldCup ($k = 2$, distortionary)	0.947	0.689	0.397	0.995	0.853	0.426	0.997	0.914	0.451	1.000	0.949	0.478	1.000	0.968	0.481
GEMMA2-9B-BASE															
+ BiMark	0.949	0.480	0.430	0.988	0.706	0.466	0.995	0.816	0.490	1.000	0.879	0.505	0.998	0.909	0.504
+ WorldCup ($k = 1$, non-distortionary)	0.807	0.439	0.439	0.864	0.620	0.493	0.888	0.699	0.514	0.909	0.741	0.532	0.879	0.760	0.536
+ WorldCup ($k = 1$, distortionary)	0.865	0.455	0.438	0.935	0.653	0.478	0.948	0.747	0.506	0.972	0.803	0.532	0.975	0.838	0.533
+ WorldCup ($k = 2$, non-distortionary)	0.870	0.630	0.415	0.950	0.780	0.454	0.958	0.834	0.479	0.962	0.861	0.504	0.959	0.876	0.508
+ WorldCup ($k = 2$, distortionary)	0.915	0.665	0.412	0.992	0.825	0.459	0.990	0.888	0.477	0.993	0.920	0.503	0.997	0.942	0.505

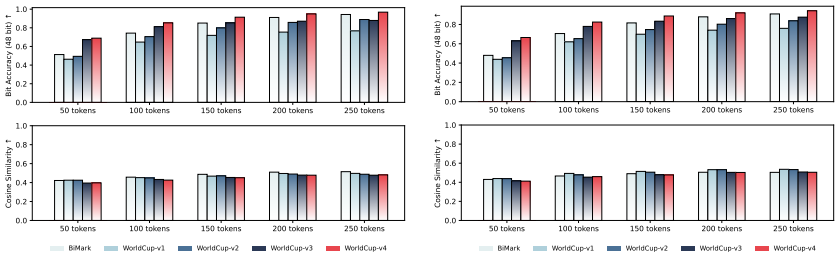

Figure 11: The comparison of distortionary and non-distortionary g-value functions

where T denotes the number of tokens and m the number of layers.

The weighted-mean detector assigns non-increasing weights $\alpha_1 \geq \cdots \geq \alpha_m \geq 0$ to different layers, with $\sum_{\ell=1}^{m} \alpha_\ell = m$:

$$\text{WeightedMeanScore}(x, \boldsymbol{\alpha}) := \frac{1}{mT} \sum_{t=1}^{T} \sum_{\ell=1}^{m} \alpha_\ell g_{t,\ell}. \tag{37}$$

Specifically, we set $\alpha_1 = \tau, \quad \alpha_2 = \tau - \frac{\tau-\mu}{m-1}, \quad \alpha_3 = \tau - 2\frac{\tau-\mu}{m-1}, \ \ldots, \ \alpha_m = \mu$, with $\tau = 10$ and $\mu = 1$, and then renormalize the weights such that $\sum_{\ell=1}^{m} \alpha_\ell = m$.

For multi-bit detection, we apply these averages to each message position and complementary g-value pair, then select and accumulate the larger aggregates as specified in Appendix G.4.

We evaluate both detectors under $k = 1$ and $k = 2$ on the C4 and OpenGen datasets across varying token lengths. Figure 12 shows the corresponding detection curves, and the quantitative results are summarized in Table 8. Our results indicate that for $k = 1$, the weighted-mean detector consistently outperforms the mean detector, in agreement with the findings of SynthID. This is because the contribution of watermarking evidence from each layer diminishes as depth increases, making layer-wise weighting beneficial. However, for $k = 2$, this trend reverses, and the mean detector achieves superior performance. A plausible explanation is that the use of multiple g-functions alleviates the attenuation of watermark signals across layers, thereby diminishing the advantage of layer weighting.

I.4 WORLDCUP SAMPLING LAYERS

To investigate the impact of the number of layers in WorldCup, we conducted experiments with different layer counts $m = 5, 10, 20, 30, 50$ on LLaMA3-8B-Base and Gemma2-9B-Base, with results summarized in Table 9. Empirically, a layer count of $m = 30$ yields relatively optimal performance, consistent with findings from SynthID-Text (Dathathri et al., 2024). Lower layer counts generally lead to decreased decoding accuracy, while increasing the number of layers exhibits a diminishing returns effect. The number of layers, however, has no significant impact on perplexity (PPL). Therefore, in this work, we primarily adopt $m = 30$ for our experiments.

Table 8: The comparison of mean and weighted mean watermark detector on WorldCup.

Dataset	LLAMA3-8B-BASE									
	50 tokens		100 tokens		150 tokens		200 tokens		250 tokens	
+ Watermark	Best F1 ↑	Bit Acc ↑	Best F1 ↑	Bit Acc ↑	Best F1 ↑	Bit Acc ↑	Best F1 ↑	Bit Acc ↑	Best F1 ↑	Bit Acc ↑
C4 DATASET	-	-	-	-	-	-	-	-	-	-
+ BiMark	1.000 ± 0.000	0.530 ± 0.001	1.000 ± 0.000	0.758 ± 0.007	1.000 ± 0.000	0.866 ±0.007	1.000 ± 0.000	0.919 ± 0.006	1.000 ± 0.000	0.949 ± 0.008
+ WorldCup ($k = 1$, D1)	0.931 ± 0.039	0.509 ± 0.009	0.980 ± 0.010	0.736 ±0.010	0.985 ± 0.015	0.832 ± 0.002	0.995 ± 0.005	0.879 ± 0.002	0.995 ± 0.005	0.906 ± 0.001
+ WorldCup ($k = 1$, D2)	0.953 ± 0.005	0.520 ± 0.007	0.985 ± 0.005	0.744 ± 0.005	0.995 ± 0.005	0.846 ± 0.001	0.990 ± 0.010	0.895 ± 0.004	0.995 ± 0.005	0.918 ± 0.000
+ WorldCup ($k = 2$, D1)	0.985 ± 0.015	0.697 ± 0.004	0.995 ± 0.005	0.869 ± 0.001	1.000 ± 0.000	0.930 ± 0.002	1.000 ± 0.000	0.957 ± 0.003	1.000 ± 0.000	0.974 ± 0.001
+ WorldCup ($k = 2$, D2)	0.975 ± 0.015	0.683 ± 0.003	1.000 ± 0.000	0.857 ± 0.006	1.000 ± 0.000	0.914 ± 0.001	1.000 ± 0.000	0.948 ± 0.003	1.000 ± 0.000	0.966 ± 0.002
OPENGEN DATASET										
+ BiMark	0.995 ± 0.005	0.538 ± 0.008	1.000 ± 0.000	0.765 ± 0.004	1.000 ± 0.000	0.774 ± 0.006	1.000 ± 0.000	0.927 ± 0.006	1.000 ± 0.000	0.955 ± 0.003
+ WorldCup ($k = 1$, D1)	0.934 ± 0.006	0.517 ± 0.004	0.975 ± 0.015	0.749 ± 0.006	1.000 ± 0.000	0.848 ± 0.006	1.000 ± 0.000	0.896 ± 0.004	1.000 ± 0.000	0.927 ± 0.004
+ WorldCup ($k = 1$, D2)	0.960 ± 0.020	0.526 ± 0.011	0.995 ± 0.005	0.768 ± 0.006	1.000 ± 0.000	0.868 ± 0.001	1.000 ± 0.000	0.915 ± 0.002	1.000 ± 0.000	0.943 ± 0.000
+ WorldCup ($k = 2$, D1)	0.975 ± 0.005	0.702 ± 0.005	1.000 ± 0.000	0.871 ± 0.005	1.000 ± 0.000	0.930 ± 0.004	1.000 ± 0.000	0.959 ± 0.001	1.000 ± 0.000	0.972 ± 0.006
+ WorldCup ($k = 2$, D2)	0.985 ± 0.005	0.684 ± 0.004	1.000 ± 0.000	0.850 ± 0.003	1.000 ± 0.000	0.909 ± 0.002	1.000 ± 0.000	0.949 ± 0.003	1.000 ± 0.000	0.966 ± 0.000

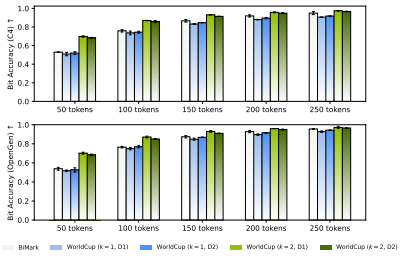

Figure 12: The comparison of mean detector and weighted mean detector.

In addition, we examine the effect of the number of leaves N, as reported in Table 10. The results indicate that setting $N = 2$ provides the best trade-off between message decoding accuracy and text quality.

I.5 DIFFERENT ACTIVATION FUNCTION

We evaluate three activation functions in Eq. 6, namely ReLU, Sigmoid, and Tanh. The results are reported in Table 11. Overall, the choice of activation function leads to comparable performance trends, and all variants exhibit a consistent trade-off between bit-level detection accuracy and text quality measured by perplexity. Specifically, ReLU exhibits relatively stronger detection performance due to its linear behavior for positive inputs; however, this gain is accompanied by a noticeable degradation in text quality. In contrast, Tanh and Sigmoid introduce nonlinear saturation, resulting in more conservative modulation and better preservation of generation quality. Since entropy values are non-negative, Tanh effectively normalizes them into the interval $[0, 1]$, whereas Sigmoid maps them into a narrower range of $[0.5, 1]$, which limits the dynamic range of modulation. Based on these considerations, we adopt the Tanh activation function in our method. Combined with a fixed scaling factor $\alpha = 1.2$, this choice provides a balanced trade-off between detection accuracy and text quality, while allowing stable and interpretable control of watermark strength in practice.

I.6 KEY GENERATION HYPERPARAMETERS

During watermark generation, both the *no_repeat_ngram_size* and the window size significantly affect multi-bit watermark performance, since the random seed for embedding is derived from hashing tokens within a context window. If text diversity is too low (e.g., *no_repeat_ngram_size* $=$ 0), identical seeds are repeatedly used, leading to uneven bit allocation across tokens and poor decoding accuracy (Table 12). However, increasing diversity does not always help: setting *no_repeat_ngram_size* $= 1$ severely degrades fluency, as reflected by high perplexity. Balancing decoding accuracy and text quality, we set *no_repeat_ngram_size* $= 4$. Moreover, window size also influences robustness by determining the extent of context used in hashing. Empirically, larger windows reduce robustness under attacks (Table 13 and Table 14). We therefore use a window size of 2 to achieve a favorable trade-off between robustness and fluency.

Table 9: The result of Worldcup sampling across different number of layers m.

Model	C4 Dataset																			
	layers $m=5$				layers $m=10$				layers $m=20$				layers $m=30$				layers $m=50$			
+ Watermark	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓
LLAMA3-8B-BASE																				
+ WorldCup ($k=1$)	0.954	0.931	5.688	0.011	0.998	0.972	6.188	0.011	0.995	0.985	6.625	0.011	1.000	**0.987**	6.563	0.011	0.998	0.984	7.063	0.011
+ WorldCup ($k=2$)	0.995	0.990	9.250	0.011	1.000	0.994	9.250	0.012	0.997	0.991	10.13	0.012	1.000	**0.995**	10.94	0.012	0.997	0.985	16.00	0.012
GEMMA2-9B-BASE																				
+ WorldCup ($k=1$)	0.932	0.883	5.719	0.011	0.975	0.933	5.938	0.011	0.993	0.957	6.078	0.012	0.997	**0.972**	6.125	0.013	0.980	0.963	6.031	0.012
+ WorldCup ($k=2$)	0.997	0.978	8.250	0.011	0.993	0.978	8.500	0.011	0.993	0.983	8.375	0.012	0.997	**0.984**	9.500	0.012	1.000	0.977	10.94	0.012

Table 10: The result of Worldcup sampling across different number of leaves N.

Model	C4 Dataset															
	leaves $N=1$				leaves $N=2$				leaves $N=3$				leaves $N=4$			
+ Watermark	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Time (s) ↓
LLAMA3-8B-BASE																
+ WorldCup (16 bit)	0.670	0.495	5.500	0.011	0.998	0.985	6.719	0.011	1.000	0.998	15.38	0.011	1.000	1.000	53.75	0.012
+ WorldCup (24 bit)	0.672	0.492	5.500	0.012	0.997	0.958	6.781	0.012	1.000	0.988	15.88	0.012	1.000	0.994	60.00	0.012
+ WorldCup (32 bit)	0.670	0.493	5.500	0.012	0.995	0.925	6.875	0.013	1.000	0.972	15.63	0.013	1.000	0.991	61.75	0.013
+ WorldCup (48 bit)	0.668	0.469	5.469	0.014	0.990	0.875	6.547	0.013	1.000	0.933	16.25	0.014	1.000	0.968	58.00	0.014

Table 13: The effect of window size c across our framework WorldCup on LLaMA3-8B-Base.

Watermark	LLAMA3-8B-BASE															
	window_size $c=1$				window_size $c=2$				window_size $c=3$				window_size $c=4$			
+ Attack	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑
WORLDCUP ($k=1$)	0.995	0.005	0.995	0.951	1.000	0.000	1.000	0.983	0.995	0.000	0.997	0.984	0.990	0.000	0.995	0.987
+ Word-D (ratio=0.2)	0.985	0.020	0.983	0.907	0.960	0.000	0.980	0.933	0.930	0.005	0.961	0.911	0.895	0.020	0.935	0.873
+ Word-S-DICT (ratio=0.2)	0.965	0.025	0.970	0.895	0.930	0.005	0.961	0.908	0.960	0.065	0.948	0.868	0.845	0.070	0.883	0.842
+ Word-S-BERT (ratio=0.3)	0.970	0.025	0.972	0.893	0.950	0.000	0.974	0.924	0.930	0.030	0.949	0.906	0.900	0.020	0.938	0.871
+ Copy-Paste (CP-3-30%)	0.880	0.025	0.924	0.886	0.910	0.000	0.953	0.926	0.925	0.015	0.954	0.931	0.885	0.015	0.932	0.913
+ Translation (en-zh)	0.990	0.975	0.668	0.605	0.990	0.960	0.671	0.603	0.985	0.955	0.670	0.563	0.980	0.965	0.666	0.538
+ Rephrase (GPT-4o)	0.870	0.250	0.821	0.742	0.735	0.325	0.714	0.729	0.885	0.630	0.704	0.657	0.805	0.565	0.679	0.634
+ Dipper-1 (lex=0, order=0)	0.945	0.020	0.962	0.881	0.940	0.035	0.952	0.904	0.910	0.045	0.931	0.887	0.885	0.020	0.929	0.873
+ Dipper-2 (lex=20, order=20)	0.895	0.110	0.893	0.793	0.865	0.180	0.846	0.772	0.765	0.130	0.807	0.738	0.770	0.295	0.746	0.698
WORLDCUP ($k=2$)	0.995	0.020	0.988	0.974	1.000	0.000	1.000	0.991	1.000	0.000	1.000	0.993	1.000	0.000	1.000	0.994
+ Word-D (ratio=0.2)	0.995	0.055	0.971	0.941	0.975	0.000	0.987	0.949	0.975	0.010	0.982	0.933	0.980	0.035	0.973	0.905
+ Word-S-DICT (ratio=0.2)	0.980	0.050	0.966	0.920	0.975	0.020	0.977	0.942	0.940	0.050	0.945	0.909	0.925	0.035	0.944	0.888
+ Word-S-BERT (ratio=0.3)	0.995	0.055	0.971	0.923	0.980	0.035	0.973	0.940	0.945	0.045	0.950	0.922	0.975	0.060	0.958	0.897
+ Copy-Paste (CP-3-30%)	0.920	0.055	0.932	0.909	0.925	0.045	0.939	0.940	0.950	0.010	0.969	0.955	0.900	0.010	0.942	0.936
+ Translation (en-zh)	0.850	0.625	0.687	0.676	0.990	0.960	0.671	0.636	0.960	0.905	0.670	0.589	0.995	0.975	0.670	0.582
+ Rephrase (GPT-4o)	0.725	0.080	0.803	0.783	0.850	0.330	0.780	0.735	0.830	0.425	0.736	0.693	0.810	0.570	0.681	0.666
+ Dipper-1 (lex=0, order=0)	0.970	0.060	0.956	0.890	0.970	0.035	0.968	0.911	0.945	0.055	0.945	0.903	0.860	0.035	0.908	0.883
+ Dipper-2 (lex=20, order=20)	0.860	0.080	0.887	0.798	0.825	0.165	0.829	0.790	0.845	0.390	0.756	0.719	0.765	0.330	0.730	0.693

I.7 TEXT DIVERSITY ANALYSIS

We further compute Log Diversity to assess textual diversity. The results show that WorldCup ($k = 1$) exhibits lower diversity than the unwatermarked baseline and other methods, indicating a higher degree of repetition (Table 15). This observation helps explain its lower PPL, as the metric can be influenced by repeated high-probability n-grams. However, we emphasize that the generated text under WorldCup ($k = 1$) remains generally fluent and coherent, rather than degenerating into meaningless repetition. Therefore, the improvement in PPL is not merely an artifact, but also partially reflects a shift toward safer, high-probability generations.

Table 11: The comparison of different activation functions on C4 dataset.

Model	C4 Dataset															
	16 bits			24 bits			32 bits			48 bits			64 bits			
+ Watermark	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	
LLAMA3-8B-BASE																
+ WorldCup (ReLU, $\alpha=0.8$)	1.000	0.989	10.75	1.000	0.974	10.94	1.000	0.959	10.56	0.997	0.921	10.50	0.995	0.885	11.06	
+ WorldCup (ReLU, $\alpha=1$)	1.000	0.991	11.44	1.000	0.987	11.72	1.000	0.971	12.00	1.000	0.936	12.19	0.997	0.899	12.38	
+ WorldCup (Sigmoid, $\alpha=1$)	0.997	0.988	9.719	1.000	0.970	9.813	1.000	0.956	9.500	0.990	0.913	9.719	0.997	0.870	9.500	
+ WorldCup (Sigmoid, $\alpha=1.2$)	1.000	0.990	11.06	0.997	0.979	11.25	1.000	0.963	11.44	1.000	0.925	11.63	1.000	0.891	11.44	
+ WorldCup (Tanh, $\alpha=1$)	1.000	0.984	9.500	0.995	0.966	9.813	0.995	0.951	9.625	0.992	0.914	9.875	0.997	0.871	9.625	
+ WorldCup (Tanh, $\alpha=1.2$)	0.997	0.991	10.84	0.995	0.976	10.75	0.995	0.956	10.25	0.997	0.918	10.94	0.995	0.889	10.66	
GEMMA2-9B-BASE																
+ WorldCup (ReLU, $\alpha=0.8$)	0.997	0.960	8.375	0.993	0.930	8.500	0.990	0.911	8.188	0.985	0.871	8.375	0.983	0.822	8.000	
+ WorldCup (ReLU, $\alpha=1$)	0.987	0.967	9.250	0.992	0.955	9.063	0.990	0.924	9.125	0.992	0.886	9.406	0.982	0.841	9.188	
+ WorldCup (Sigmoid, $\alpha=1$)	0.990	0.968	8.375	0.992	0.940	7.750	0.992	0.914	7.938	0.977	0.867	8.375	0.980	0.827	8.250	
+ WorldCup (Sigmoid, $\alpha=1.2$)	1.000	0.968	9.406	0.993	0.957	9.313	0.992	0.924	9.313	0.990	0.879	9.813	0.987	0.835	9.625	
+ WorldCup (Tanh, $\alpha=1$)	0.995	0.962	7.750	0.995	0.934	7.688	0.992	0.901	7.875	0.977	0.853	7.750	0.972	0.813	7.875	
+ WorldCup (Tanh, $\alpha=1.2$)	0.993	0.970	8.844	0.990	0.941	8.500	0.982	0.916	8.688	0.992	0.869	8.500	0.988	0.826	8.500	

Table 12: The effect of different no_repeat_ngram_size on multi-bit watermarking methods.

Watermark	LLAMA3-8B-BASE														
	no_repeat_ngram_size=0			no_repeat_ngram_size=1			no_repeat_ngram_size=2			no_repeat_ngram_size=3			no_repeat_ngram_size=4		
	Diversity ↑	Bit Acc ↑	Perplexity ↓	Diversity ↑	Bit Acc ↑	Perplexity ↓	Diversity ↑	Bit Acc ↑	Perplexity ↓	Diversity ↑	Bit Acc ↑	Perplexity ↓	Diversity ↑	Bit Acc ↑	Perplexity ↓
BiMark	5.356	0.947	6.375	20.00	1.000	45.25	18.03	0.993	15.63	8.890	0.994	12.56	7.804	0.992	11.81
WorldCup ($k = 1$)	5.036	0.885	3.070	20.00	1.000	34.25	17.88	0.987	9.500	7.715	0.981	7.344	7.202	0.978	6.625
WorldCup ($k = 2$)	6.110	0.947	7.750	20.00	1.000	62.75	18.92	0.996	18.88	8.794	0.996	15.38	7.929	0.994	14.25

Table 14: The effect of window size c across our framework WorldCup on Gemma2-9B-Base.

Watermark	GEMMA2-9B-BASE															
	window_size $c = 1$				window_size $c = 2$				window_size $c = 3$				window_size $c = 4$			
+ Attack	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑	TPR ↑	FPR ↓	F1 ↑	Bit Acc ↑
WORLDCUP ($k = 1$)	0.995	0.015	0.990	0.909	0.995	0.010	0.993	0.943	0.995	0.000	0.997	0.973	1.000	0.005	0.998	0.974
+ Word-D (ratio=0.2)	0.955	0.030	0.962	0.872	0.875	0.085	0.893	0.868	0.860	0.050	0.901	0.848	0.830	0.105	0.858	0.813
+ Word-S-DICT (ratio=0.2)	0.945	0.075	0.936	0.853	0.890	0.165	0.866	0.832	0.945	0.235	0.867	0.827	0.810	0.170	0.818	0.784
+ Word-S-BERT (ratio=0.3)	0.930	0.025	0.951	0.851	0.925	0.140	0.896	0.843	0.830	0.055	0.881	0.847	0.815	0.150	0.830	0.828
+ Copy-Paste (CP-3-30%)	0.875	0.030	0.919	0.836	0.880	0.030	0.921	0.874	0.885	0.025	0.927	0.879	0.885	0.040	0.919	0.884
+ Translation (en-zh)	0.990	0.970	0.669	0.590	1.000	0.990	0.669	0.580	0.995	0.990	0.667	0.552	0.995	1.000	0.664	0.516
+ Rephrase (GPT-4o)	0.790	0.280	0.763	0.712	0.760	0.335	0.726	0.701	0.800	0.460	0.708	0.681	0.920	0.750	0.689	0.647
+ Dipper-1 (lex=0, order=0)	0.925	0.070	0.927	0.781	0.825	0.030	0.889	0.795	0.819	0.055	0.874	0.817	0.760	0.070	0.831	0.798
+ Dipper-2 (lex=20, Dipper=20)	0.840	0.140	0.848	0.716	0.790	0.175	0.804	0.704	0.845	0.480	0.727	0.688	0.880	0.640	0.698	0.655
WORLDCUP ($k = 2$)	0.970	0.020	0.975	0.947	0.995	0.000	0.997	0.968	0.995	0.000	0.997	0.983	0.995	0.000	0.997	0.984
+ Word-D (ratio=0.2)	0.950	0.085	0.934	0.909	0.935	0.045	0.944	0.905	0.915	0.060	0.927	0.887	0.905	0.065	0.919	0.858
+ Word-S-DICT (ratio=0.2)	0.905	0.060	0.921	0.877	0.845	0.045	0.894	0.882	0.860	0.060	0.896	0.869	0.855	0.110	0.870	0.824
+ Word-S-BERT (ratio=0.3)	0.910	0.060	0.924	0.896	0.850	0.045	0.897	0.878	0.865	0.060	0.899	0.867	0.815	0.110	0.847	0.830
+ Copy-Paste (CP-3-30%)	0.910	0.050	0.929	0.886	0.895	0.010	0.939	0.900	0.895	0.015	0.937	0.909	0.910	0.010	0.948	0.905
+ Translation (en-zh)	0.995	1.000	0.664	0.641	0.995	0.995	0.666	0.602	0.990	0.970	0.669	0.588	0.975	0.945	0.668	0.537
+ Rephrase (GPT-4o)	0.795	0.255	0.776	0.756	0.825	0.575	0.688	0.709	0.935	0.795	0.685	0.690	0.965	0.875	0.680	0.634
+ Dipper-1 (lex=0, order=0)	0.905	0.165	0.874	0.805	0.825	0.045	0.882	0.825	0.795	0.065	0.855	0.803	0.785	0.095	0.835	0.775
+ Dipper-2 (lex=20, Dipper=20)	0.890	0.290	0.817	0.715	0.865	0.400	0.764	0.710	0.840	0.545	0.704	0.649	0.875	0.645	0.694	0.626

Table 15: The diversity analysis of unwatermarked texts and different watermarked texts.

Model	C4 DATASET															
	16 bits				24 bits				32 bits				48 bits			
+ Watermark	Bit Acc ↑	PPL ↓	STS ↑	Diversity ↑	Bit Acc ↑	PPL ↓	STS ↑	Diversity ↑	Bit Acc ↑	PPL ↓	STS ↑	Diversity ↑	Bit Acc ↑	PPL ↓	STS ↑	Diversity ↑
LLAMA3-8B-BASE																
+ Human	-	9.188	1.000	8.499	-	9.188	1.000	8.499	-	9.188	1.000	8.499	-	9.188	1.000	8.499
+ Unwatermarked	-	11.25	0.443	7.729	-	11.06	0.443	7.736	-	11.25	0.442	7.653	-	11.34	0.437	7.730
+ BiMark	0.983	13.38	0.433	7.533	0.962	13.38	0.439	7.676	0.952	13.81	0.441	7.572	0.890	13.81	0.424	7.626
+ StealthInk	0.963	13.19	0.435	8.047	0.931	12.94	0.436	8.009	0.910	13.19	0.437	7.900	0.866	13.19	0.436	7.913
+ WorldCup ($k = 1$)	0.980	7.500	0.441	6.783	0.952	7.625	0.438	6.833	0.918	7.625	0.442	6.918	0.860	7.625	0.446	6.810
+ WorldCup ($k = 2$)	0.988	12.56	0.437	7.647	0.971	12.19	0.441	7.796	0.959	12.38	0.442	7.629	0.925	12.37	0.439	7.736

I.8 COUNTING-BASED DECODING VS. CONFIDENCE-AWARE DECODING

Both decoders use the same WorldCup encoder. Counting binarizes each token's mean g-value before aggregation, whereas confidence-aware decoding retains it, so $|2u_{t,j} - 1|$ weights the token's bit preference by agreement across tournament layers. This weighting uses neither the base model entropy nor the SynthID layer weights compared in Appendix I.3.

Table 16: The comparison of counting-based decoding and confidence-aware decoding.

Model	C4 DATASET														
	16 bits			24 bits			32 bits			48 bits			64 bits		
+ Watermark	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓	Best F1 ↑	Bit Acc ↑	PPL ↓
LLAMA3-8B-BASE															
+ WorldCup ($k = 1$, counting-based)	0.982	0.971	6.688	0.985	0.933	6.625	0.966	0.897	6.969	0.965	0.844	6.750	0.925	0.771	6.594
+ WorldCup ($k = 1$, confidence-aware)	0.998	0.984	6.625	0.997	0.961	6.750	0.995	0.930	7.000	0.992	0.875	6.563	0.990	0.813	6.547
+ WorldCup ($k = 2$, counting-based)	0.931	0.978	11.00	0.926	0.959	11.16	0.908	0.925	11.06	0.883	0.886	11.44	0.889	0.843	11.44
+ WorldCup ($k = 2$, confidence-aware)	1.000	0.990	11.00	1.000	0.982	11.25	1.000	0.962	11.16	1.000	0.925	11.44	1.000	0.887	11.16

Table 17: The results on larger LLMs including Mixtral-8x7B-Instruct and LLaMA3-70B-Base.

Model	C4 Dataset											
	16 bits			24 bits			32 bits			48 bits		
+ Watermark	Best F1 ↑	Bit Acc ↑	STS ↑	Best F1 ↑	Bit Acc ↑	STS ↑	Best F1 ↑	Bit Acc ↑	STS ↑	Best F1 ↑	Bit Acc ↑	STS ↑
MIXTRAL-8X7B-IT-V0.1												
+ MPAC	0.992	0.956	0.577	0.969	0.905	0.564	0.972	0.885	0.577	0.956	0.840	0.581
+ WorldCup ($k = 1$)	0.990	0.957	0.599	0.987	0.933	0.593	0.985	0.891	0.583	0.985	0.846	0.590
+ WorldCup ($k = 2$)	0.990	0.968	0.585	0.992	0.943	0.579	0.992	0.921	0.579	0.985	0.868	0.575
LLAMA3-70B-BASE												
+ MPAC	0.980	0.973	0.531	0.982	0.948	0.546	0.956	0.916	0.538	0.940	0.861	0.553
+ WorldCup ($k = 1$)	0.987	0.971	0.554	0.987	0.944	0.574	0.982	0.910	0.566	0.977	0.866	0.572
+ WorldCup ($k = 2$)	0.990	0.977	0.551	0.987	0.956	0.540	0.983	0.936	0.551	0.977	0.891	0.528

I.9 LARGER MODELS

We add Semantic Textual Similarity (STS) as an additional semantic quality metric, computed using sentence embeddings between the generated text and its corresponding reference text. We also conduct experiments on larger models (e.g., Mixtral-8x7B-Instruct and LLaMA3-70B-Base). The additional results (Table 17) remain consistent with our main findings: WorldCup maintains strong overall performance, while STS shows trends highly consistent with perplexity, further confirming good semantic preservation. Moreover, we note that entropy is not inherently tied to model size or capability. Experiments on other large models exhibit stable behavior across different entropy settings, demonstrating the scalability and robustness of our framework.

I.10 ALGORITHM

Algorithm 2 WorldCup Message Decoding (Confidence-Aware)

Input: generated text sequence $y_{1:T}$, hash function h, watermarking key ξ, total message length b, 2^k-ary and g-value function families $\{\mathbf{g}_j, \bar{\mathbf{g}}_j\}_{j=1}^{k}$
Compute number of message symbols $B \leftarrow b/k$
Initialize decoded message symbols $\hat{\mathbf{m}} \in \{0, \ldots, 2^k - 1\}^B$
Using preceding context and hash function h to compute random seed r_t
Use (r_t, ξ) to map each token y_t to a message position $p_t \in \{0, \ldots, B-1\}$
Group tokens by message position: $\mathcal{S}_p = \{y_t \mid p_t = p\}$
for $p = 0$ to $B - 1$ **do**
  for $j = 1$ to k **do**
    Compute $u_{t,j} \leftarrow \frac{1}{m} \sum_{\ell=1}^{m} g_\ell^{(j)}(y_t, r_t)$ for $y_t \in \mathcal{S}_p$
    Aggregate $d_j^p \leftarrow \frac{1}{|\mathcal{S}_p|} \sum_{y_t \in \mathcal{S}_p} (2u_{t,j} - 1)$
  end for
  Decode the 2^k-ary message symbol: $\hat{m}_p \leftarrow \sum_{j=1}^{k} 2^{k-j} \mathbb{I}(d_j^p < 0)$
end for
Assemble decoded symbols into message sequence: $\hat{\mathbf{m}} \leftarrow (\hat{m}_0, \hat{m}_1, \ldots, \hat{m}_{B-1})$
Return decoded message symbols $\hat{\mathbf{m}}$

J LIMITATIONS

While WorldCup achieves strong empirical performance by leveraging token redundancy, several aspects remain to be further explored. First, although multi-bit embedding per token is shown to be effective, we do not provide a formal analysis of its optimality. A theoretical understanding of token-level information capacity could further strengthen the framework. Second, we leave a comprehensive study of unforgeability in the multi-bit setting to future work. Compared to zero-bit schemes, multi-bit watermarking is generally harder to forge, since an attacker must recover both the bit assignment and the decoding rule. Developing a more systematic analysis of such attacks would provide additional insights into robustness.

K CASE STUDY

K.1 LLAMA3-8B-BASE EXAMPLES

Prompt (C4 Dataset)

Applications for entrance in the Fall of 2019 are now open. We will begin to evaluate applications on January 15, 2019. If your application is not

Natural Text (C4 Dataset)

received by then, it may not be evaluated. Please note that we do not typically admit for Spring or Summer admission. Offers are usually made between February and April 15. All applicants are considered for department support via research assistantships, teachings assistantships, and merit-based scholarships. Virtually all graduate students are supported for the academic year and in many cases during the summer term as well. Once the Graduate Admissions Committee has made a final decision on your application they will notify you via the email address that is on your application. There is no minimum GRE or GPA requirement. However, to be competitive for assistantships and scholarships, students need a GPA of at least 3.0, a combined verbal and quantitative score on the GRE of at least 300, and an analytical writing score on the GRE of at least 3.5. The average GRE score for students that were accepted into our program for Fall 2018 was 316.

Unwatermarked Text (Llama3-8B-Base)

complete by that date we will not be able to guarantee you an interview. Click here to begin an application. Please visit the requirements tab above for details about how to complete an application. The deadline for submitting applications for Fall, 2019 is March 31, 2019. A maximum of two interviews a week will be held every Tuesday at 1:00 p.m. and 3:00 p.m., and Wednesday at 9:00 a.m. and 1:00p.m. You may not request interview dates. There is no interview in the Spring. Interviewers are assigned by the Office of Admission. You will not be interviewed by anyone that you request for an interview.

Watermarked Text (Message: 1101000010110000 Bit Accuracy: 1.0 k=2)

complete or received before January 15, you will be placed in the second round of applications. Please be aware there is limited or no amount of spots for second round applications. Please be aware that all our students for the U.S. and Canada are in class by April 1, for non-residents this is 2 weeks later. We do not keep our spots open for the full year. If you cannot get into Canada by April 1 you cannot join our Program. We will not make arrangements for you to join the Program at a later date. For spots that become available, even if we have already started the Program, only students who have the ability to be in Canada by April 13 will be admitted.

Prompt (C4 Dataset)

Independent Bank Corp. (nasdaq global select market:INDB), parent of Rockland Trust Company, today announced 2018 fourth quarter net income of

Natural Text (C4 Dataset)

$29.9 million, or $1.07 per diluted share, compared to net income of $33.0 million, or $1.20 per diluted share, reported in the prior quarter of 2018. Excluding merger and acquisition expenses incurred in the fourth and third quarters of 2018 related to the MNB Bancorp (MNB) merger which closed in November 2018 and the Blue Hills Bancorp (BHB) merger announced on September 20, 2018, operating net income was $35.9 million, or $1.29 per diluted share during the fourth quarter of 2018 compared to $34.9 million, or $1.27 per diluted share during the third quarter of 2018. Full year net income was $121.6 million, or $4.40 on a diluted earnings per share basis, an increase of $34.4 million, or 39.5%, as compared to the prior year. In addition, full year operating net income was $129.8.

Unwatermarked Text (Llama3-8B-Base)

$35.8 million, or $0.78 per diluted share, an increase of 2.7% and 3.3% from the $34.9 million, or $1.02 per diluted share, reported for the 2017 fourth quarter. For the 2018 full year, net income increased 20.1% to $137.4 million, or $3.03 per diluted share, compared to $114.3 million, or $2.88 per diluted share, for the 2018 year ended December 31, 2017. The 2018 fourth quarters and full year included $5.6 million in total credit loss expense, compared to the corresponding prior year amounts of $4.0 million and $16.6 million. The 2017 fourth quarters and full years included $3.6 million and $6.3 million, respectively, in additional expense for the tax effect of deferred tax liabilities associated with the impact of the change in U.S. federal corporate tax rates from 35% to 21% as a result of the Tax Cuts and Jobs Act. “During the quarter we completed a successful acquisition of the 17 branches and other deposits from Eastern Bank.

Watermarked Text (Message: 001111111010010000101010 Bit Accuracy: 0.75 k=2)

$26.6 million, or $0.79 per share, compared to $23.5 million, or $1.40 per share, in the prior year’s fourth quarter. For the full year 2018, net income was $101.7 million, or $3.06 per share, compared with $85.2 million, or $5.29 per share, in 2017. The Company’s 2018 performance was impacted by the non-cash charge of $10.7 million, which was added back to net income in the calculation of earnings per share, to establish a valuation allowance against its deferred tax assets. This charge was partially offset by a $6.1 million, or $10.1 million on an after-tax basis, increase in net interest income due to lower interest expense. Pre-tax, pre-provision net revenue increased 15.1%, driven by growth in loans and deposits, higher non-interest income and lower non-interest expense.

K.2 GEMMA2-9B-BASE EXAMPLES

Prompt (C4 Dataset)

Our no. 1 sneaker of 2014 just got a serious upgrade in the form of the adidas Pure Boost ZG Prime. We first

Natural Text (C4 Dataset)

previewed this sneaker in January, and now the next-gen of Pure Boost has arrived. This model mixes a full-length Boost midsole with a layer of EVA on the outsole to produce what adidas calls a ̈floating sensation,̈hence the Zero Gravity's; (ZG) part of its name. That's not the only key feature here, as this version of the Pure Boost ZG also features a Primeknit upper for a cozy and adaptive sock-like fit. It's then finished off with leather tabs at the tongue and heel, adding a premium touch to an otherwise sporty silhouette. If for some reason this Primeknit variation isn't quite your cup of tea—or out of your budget—there's also a less expensive mesh-based version dropping this summer. The adidas Pure Boost ZG Prime is now available in black.

Unwatermarked Text (Gemma2-9B-Base)

laid eyes on the Adidas PureBoost ZG Prime at the recent 2015 PGA Show where the model made its debut on the feet of former World No. 1 and 3-time Major champion, Ernie Els. For the 2016 season, Ernie Els has teamed up with Adidas Golf and will don the PureBoost ZG on his feet during the Major season. It's no secret that adidas Pure Boost technology is the best when it comes to running. The adidas PureBoost ZG and PureBoost ZG Primes boast a flexible woven upper that provides a sock-like fit with minimal movement. adidas Pureboost ZG Prime are made with Primeknit upper, featuring woven technology for flexible fit. adidas Pure Boost ZG. Adidas Pure Boost ZG/Primes are engineered with Primeknit upper, featuring flexible material and delivering comfortable fit. adidas has now released the PureBoost Z and ZG Primes. Adidas PureBoost ZG will also launch this winter in six color options. adidas just released the PureBoost Prime, and now the ZG model is set to make its debut in the near future.

Watermarked Text (Message: 1101101011101010 Bit Accuracy: 1.0 k=2)

saw the Pure Boost ZG back in July, and one of our editors was lucky enough to try them out during the 2015 World Championships in Beijing, China. The adidas Pure Boost Zg Prime is an update to the most popular running shoe of 2004, the Pure Boost Zg. The Zg Prime has a more streamlined upper and a wider base for a more comfortable fit. The new upper is made of 2 layers of a breathable material, with a 3rd layer of support at the medial side of the shoe. The adidas pure boost zg prime is the latest in a long line of running shoes from adidas. It features a Boost midsole and a Primeknit upper. The shoe also features an EVA midsole for cushioning and a Torsion System for stability. The adidas Zg Prime is a running shoe that features an Air Max unit, a Boost midsole, and a Primeknt upper. The shoe is available in two colorways: Black/White/Grey and White/Black/Grey. The black version features a suede upper with a leather overlay, while the white version features a mesh upper with a leather liner. Both shoes feature a Boost midsole with an Air Max unit.

Prompt (C4 Dataset)

A child in Florida has become the first person to die of the flu this season, according to state health officials. State epidemiologists say the

Natural Text (C4 Dataset)

child had not been vaccinated and was otherwise healthy before getting sick with the flu. The child, who tested positive for influenza B, died sometime during the week of Sept. 30, although privacy concerns prevent officials from saying exactly where, CBS affiliate WTSP-TV reports. Last flu season, 183 children in the U.S. died from flu or flu-related causes. That's the most since the Centers for Disease Control and Prevention (CDC) began keeping these records in 2004. Overall, an estimated 80,000 Americans died from flu last season. CBS News medical contributor Dr. Tara Narula says this latest news should be a wake-up call to parents to get their children vaccinated. "What this is a strong clear message to parents about the importance of vaccination," she told "CBS This Morning." "This vaccine is safe. It is the most effective tool we have. And we know of the pediatric deaths last year, 80 percent were in kids who were unvaccinated.

Unwatermarked Text (Gemma2-9B-Base)

child died from influenza B. That's different from the type of flu A that caused death and illness in children last year. The CDC says 17 children died of the flu in the 2009-10 flu season, 11 of them from the H1N1 strain, or swine flu. Last year 18 children in 10 states died of swine flu, according to the AAP. That's in addition to eight children who died from the flu last year.

Watermarked Text (Message: 011001110110011011100110 Bit Accuracy: 0.958 k=2)

child died of flu complications, but did not specify the age of the child or where in the state they lived. It's unclear if the child had been vaccinated against the flu. A spokesperson for the Florida Department of Health said the state does not release information about the individual flu cases. The U.S. Centers for Disease Control and Prevention currently reports that the flu has spread to 46 states, and has killed at least 11 people so far this season. According to the CDC, the flu has hit people hardest 50 to 64 years old. The CDC also reports that in the last week, 45 people out of every 100,000 have sought medical treatment for flu-like symptoms. That's up from 26 the week before. Health officials say this year's flu strain is particularly. The CDC says 75 percent of people who die from the flu each year are 65 or older. Some experts say this year'll be flu season. But they say there's still time to get vaccinated against the flu, and they urge people to do so.