CLEF: Closed-Loop Error Feedback for Multi-Bit Watermarking in Short LLM-Generated Text

Jing Wang[1], Kejun Qian[1]*, Pengyu Xiong[1], Qiankun Zhang[2], Yi Yang[2], Jing Fan[3]
[1]Huazhong University of Science and Technology
[2]Naval University of Engineering
[3]Dongfeng Motor Corporation Research and
Development Institute, Dongfeng Motor Group Company Limited
{cswjing, qkjun, xpyu, qiankun}@hust.edu.cn, yangyi.ip@whu.edu.cn, fanjing@dfmc.com.cn

Abstract

Multi-bit watermarking for large language models (LLMs) enables richer provenance information, but existing schemes remain unreliable for short texts because they rely on local or pre-specified allocation strategies. To address this limitation, we propose a Closed-Loop Error Feedback (CLEF) Multi-Bit Watermarking framework. By treating the LLM as a noisy channel, CLEF tracks the global feature state of generated tokens and dynamically measures its deviation from an ideal multidimensional target trajectory. The resulting dynamic error vector is then used to modulate the logits of candidate tokens via an inner-product scoring mechanism, biasing generation toward tokens that compensate for underrepresented payload dimensions. This closed-loop control enables robust multi-bit embedding with a simple, fully linear decoding process. Experiments on short-text generation show that CLEF improves bit extraction accuracy and complete message recovery rate over the evaluated multi-bit watermarking baselines. Furthermore, CLEF achieves higher bit extraction accuracy under different text perturbations while maintaining acceptable generation quality across downstream tasks such as text summarization and machine translation.

1 Introduction

Large language models have rapidly become a foundational infrastructure for content generation, enabling applications ranging from writing assistance and dialogue systems to summarization and translation (OpenAI, 2023). As LLM-generated text is increasingly deployed in open and high-impact settings, concerns about provenance, attribution, misuse, and accountability have become correspondingly urgent (Liu et al., 2024; Liang et al., 2025; Piet et al., 2025). Watermarking has emerged as a widely studied approach for addressing these concerns by embedding imperceptible signals into generated text, thereby enabling post hoc verification of model-originated content. While early work (Kirchenbauer et al., 2023; Hu et al., 2024; Kuditipudi et al., 2024; Dathathri et al., 2024) has primarily focused on zero-bit watermarking for binary detection, many realistic governance scenarios require richer functionality, such as embedding user identifiers, generation sessions, timestamps, or model-version information. This makes multi-bit watermarking a more suitable paradigm for fine-grained traceability (Yoo et al., 2024; Feng et al., 2025; Qu et al., 2025; Xu et al., 2026; Jiang et al., 2025; Park et al., 2026).

Despite this promise, reliable multi-bit watermarking for short-text generation remains challenging. Existing methods typically allocate watermark capacity in a local or open-loop manner, where each generation step encodes information independently based only on the current context (Yoo et al., 2024; Feng et al., 2025; Xu et al., 2026; Park et al., 2026). A representative design is time-division multiplexing, which partitions the token sequence into disjoint subsets and assigns each subset to a particular bit. Such schemes can be effective for long texts, but become brittle when only tens of tokens are available, since each bit receives few observations and extraction becomes sensitive to sampling noise. Moreover, open-loop designs cannot monitor accumulated deviations or reallocate later tokens to compensate for underrepresented dimensions.

Thus, the central difficulty is not merely how to decode a noisy watermark, but how to embed it adaptively during generation. In this paper, we propose Closed-Loop Error Feedback (CLEF), a multi-bit watermarking framework that introduces a feedback-control perspective into LLM text generation. Instead of treating watermark injection as a sequence of independent local decisions, CLEF models the LLM as a noisy communication channel and maintains a global watermark state that accu-

*Corresponding author.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 4244–4260
November 6-10, 2026 ©2026 Association for Computational Linguistics

mulates the feature contributions of previously generated tokens. At each step, this state is compared against an ideal multidimensional target trajectory associated with the desired payload, producing a real-time error vector. The error is then used to modulate the logits of candidate tokens through an inner-product scoring rule, encouraging the model to select tokens whose feature vectors compensate for currently underrepresented dimensions. In this way, CLEF converts watermark injection from a static allocation mechanism into a dynamic closed-loop correction process. Notably, this generation-side control leads to a simple decoding stage: the decoder only needs to linearly integrate token features and recover each bit from the sign of the final accumulated signal.

The primary contributions of this work are summarized as follows:

- We formulate multi-bit LLM watermarking as a closed-loop trajectory tracking problem, where the accumulated token feature state is continuously compared with a payload-dependent target trajectory.
- We propose an error-driven logit modulation rule that corrects deviations from the target trajectory during generation, thereby compensating for underrepresented payload dimensions.
- We introduce a lightweight linear decoding rule and conduct systematic evaluations under short-text, high-payload, perturbation, utility, and length-generalization settings, demonstrating consistent reliability gains over existing multi-bit baselines.

2 Related Work

2.1 Zero-Bit Watermarking for Large Language Models

Text watermarking has been widely studied for identifying machine-generated content. In LLMs, early methods mainly focus on zero-bit watermarking, where the goal is to determine whether a text was generated by a watermarked model (Kirchenbauer et al., 2023, 2024). A representative approach partitions the vocabulary into preferred and non-preferred token sets according to a secret key and preceding context (Kirchenbauer et al., 2023), then detects the watermark by testing whether the generated text shows a significant bias toward the preferred set.

Subsequent studies improve this framework by reducing distributional distortion, providing unbiased or distortion-free guarantees (Hu et al., 2024; Kuditipudi et al., 2024), and enhancing robustness to paraphrasing or editing attacks (Kirchenbauer et al., 2024; Christ et al., 2024; Liang et al., 2025; Piet et al., 2025; Guo et al., 2024; Liu and Bu, 2024; Hou et al., 2024). However, zero-bit watermarking only supports binary verification and cannot directly encode provenance information such as user identity, session index, timestamp, or model version. This motivates multi-bit watermarking, where the embedded signal is both detectable and recoverable.

2.2 Multi-Bit Watermarking for Large Language Models

To increase watermark capacity, recent studies have moved from zero-bit detection to multi-bit watermarking, where the generator embeds a payload vector rather than a single signal (Yoo et al., 2024, 2023; Qu et al., 2025; Park et al., 2026). A representative method in this line is MPAC (Yoo et al., 2024), which follows a time-division multiplexing strategy: different positions or token subsets are assigned to different payload bits, and the final payload is recovered by aggregating evidence separately for each bit stream. This design is intuitive and effective when the generated text is sufficiently long, because each bit can accumulate enough token-level evidence over its assigned positions.

Recent methods refine this local allocation paradigm from different perspectives. BiMark (Feng et al., 2025) improves multi-bit embedding while preserving naturalness. XMark (Xu et al., 2026) partitions the vocabulary multiple times and uses the intersection of green lists to provide multiple observations per token. StealthInk (Jiang et al., 2025) emphasizes imperceptibility through constrained token selection, while other methods introduce error-correcting codes to improve decoding reliability under noise and adversarial transformations (Qu et al., 2025). Despite these improvements, most methods still make embedding decisions locally and allocate token budget in a largely pre-specified manner.

2.3 From Open-Loop Allocation to Feedback Control

The main limitation of existing multi-bit schemes (Yoo et al., 2024; Feng et al., 2025; Jiang et al.,

2025; Xu et al., 2026) is not only that each bit receives limited observations in short texts, but also that the generator cannot react to the realized embedding state. Once stochastic sampling causes some dimensions to deviate from the intended trajectory, open-loop methods continue to follow the same allocation rule and cannot redirect later tokens toward underrepresented bits. Decoding side techniques such as error-correcting codes (Qu et al., 2025) improve robustness after generation, but do not address this generation-time mismatch.

Feedback mechanisms offer a natural way to reduce accumulated error under noise. Instead of following a fixed trajectory, a feedback system observes its current state, compares it with a target, and adjusts future actions accordingly. Therefore, our work formulates multi-bit watermarking as a closed-loop generation problem. CLEF maintains a global watermark state, compares it with a payload-dependent target trajectory, and uses the resulting error vector to modulate candidate-token logits. This differs from prior methods that mainly improve coding, assignment, or decoding; CLEF explicitly controls the evolution of the accumulated watermark state during generation, which is particularly useful when the available token budget is small.

3 Method

3.1 Problem Formulation

In this section, we detail the proposed Closed-Loop Error Feedback multi-bit watermarking framework. The core philosophy of CLEF is to treat the LLM as a noisy communication channel and introduce a negative feedback mechanism from control theory to ensure that the cumulative generated signal tracks toward a multi-bit target payload, even within short texts. Here, noise refers to uncertainty from probabilistic sampling, linguistic-quality constraints, and the mismatch between desired trajectories and discrete token features.

We consider the problem of embedding a multi-bit payload into the generation process of a large language model. Let $b \in \{0,1\}^L$ denote the original binary payload to be embedded, where L is the payload length. To enable vector-valued accumulation and inner-product-based feedback control, we convert the binary payload into a bipolar representation:

$$M = 2b - 1, \quad M \in \{-1, +1\}^L. \tag{1}$$

Given a language model with vocabulary V, at each generation step t, the model produces a logit distribution over candidate tokens conditioned on the prefix c_t:

$$\text{Logits}_t(v),\ v \in V. \tag{2}$$

The goal is to generate a token sequence $\{v_1, v_2, \ldots, v_N\}$ such that the embedded payload can be recovered from the generated text, while ensuring that the text quality is within an acceptable range.

3.2 Closed-Loop State Tracking and Trajectory Control

As illustrated in Fig. 1, to enable multi-bit parallel encoding, we associate each candidate token with a pseudo-random feature vector:

$$F_K(c_t, v_t) \in \{-1, +1\}^L, \tag{3}$$

generated by a keyed pseudo-random function (PRF) using the secret key and the current context, ensuring that the feature assignment is reproducible for the detector but unpredictable without the key. Intuitively, each token contributes a vector-valued signal across all L bit dimensions. Thus, the generation process can be viewed as accumulating signals in an L-dimensional space.

We maintain a global state vector that aggregates the contributions of previously generated tokens:

$$S_{t-1} = \sum_{k=1}^{t-1} F_K(c_k, v_k^*) \tag{4}$$

where v_k^* denotes the sampled token at step k. To guide the embedding process, we define a target trajectory:

$$T_t = \alpha \cdot t \cdot M, \tag{5}$$

where α in (0,1] denotes the trajectory slope, which specifies the desired per-step accumulation rate of the watermark signal along the payload direction M. The deviation between the current state and the target is measured by the error vector:

$$E_t = T_t - S_{t-1}. \tag{6}$$

This error captures how far the current generation is from encoding the desired payload along each dimension.

Instead of directly forcing alignment with the target payload, we adopt a feedback-driven scoring mechanism. For each candidate token v_t, we

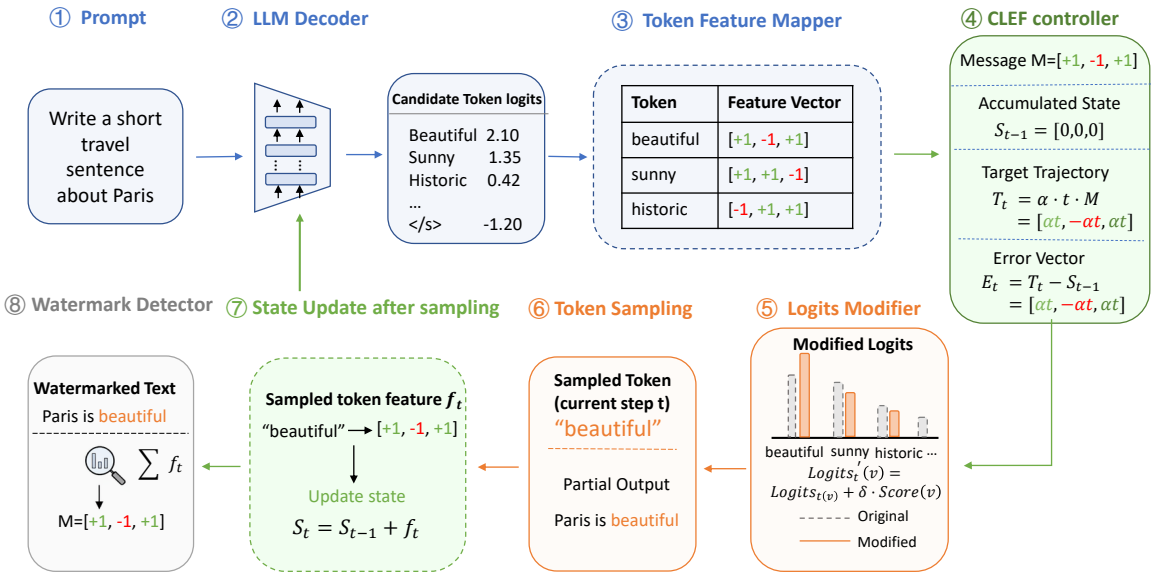


Figure 1: Workflow of Message Encoding and Recovery in CLEF for Multi-Bit Watermarking

compute:

$$\text{Score}(v) = \frac{1}{L} \sum_{i=1}^{L} F_K(c_t, v_t)_i \cdot (E_t)_i \quad (7)$$

The inner product formulation in Eq. 7 can be interpreted as a projection of the candidate token feature onto the error direction. Maximizing this score encourages the selection of tokens whose feature vectors align with the current error, thereby providing the most efficient correction in the high-dimensional embedding space.

Following the general logit-modulation paradigm used in prior LLM watermarking methods (Kirchenbauer et al., 2023; Guo et al., 2024; Xu et al., 2026), the final modulated logits $\text{Logits}'_t(v)$ are derived as follows:

$$\text{Logits}'_t(v) = \text{Logits}_t(v) + \delta_t \cdot \text{Score}(v) \quad (8)$$

where δ_t is the watermark strength parameter. In practice, we use a step-dependent watermark strength rather than a fixed value, which is detailed in Appendix C.3. This mechanism implements dynamic correction allocation: when the absolute value of E_t in the j-th dimension is large, the scoring function is dominated by that dimension. This forces the LLM to prioritize the selection of tokens that correct the lagging bits, ensuring effective self-correction throughout the generation process.

3.3 Linear Integration Decoding

Due to the adaptive control at the generation stage, the decoding process is lightweight and fully linear. For a given text of length N, the decoder does not require knowledge of the error adjustment history. It simply extracts the feature vector of each token and performs global integration:

$$V_{\text{decode}} = \sum_{t=1}^{N} F_K(c_t, v_t) \quad (9)$$

The extracted multi-bit message $\hat{M}$ is determined by the sign of the integrator vector in each dimension:

$$\hat{M}_i = \begin{cases} 1, & \text{if } (V_{\text{decode}})_i > 0 \\ -1, & \text{otherwise} \end{cases} \quad (10)$$

The sign of each dimension determines the recovered bipolar message $\hat{M}$. Finally, we convert $\hat{M}$ back to the binary payload domain by $\hat{b} = (\hat{M} + 1)/2$.

This design enables high bit extraction accuracy even if portions of the text are perturbed, as the global integration effectively filters out localized sampling noise.

3.4 Analytical Interpretation of the Feedback Advantage

Proposition 1 *Sample allocation advantage of CLEF over time-division.*

CLEF allows each generated token to contribute evidence to all payload dimensions, increasing the nominal observation opportunities per bit from Nr/L to N. Here, r denotes the number of payload bits assigned to each token in the time-division baseline. This does not imply fully independent observations across dimensions, but it reduces the sampling sparsity inherent in disjoint time-division allocation. This distinction changes the capacity–length trade-off. Under short-text regimes, time-division multiplexing spends only a small fraction of the token budget on each bit, whereas CLEF reuses every token across all bit dimensions and then reallocates correction strength through the feedback error E_t.

Proposition 2 *Error feedback modulation increases the expected corrective projection.*

At generation step t, let $p_t(v) = \mathrm{Softmax}(\mathrm{Logits}_t(v))$ denote the base token distribution before watermark modulation. Given the current tracking error E_t, define the normalized projection of a candidate token feature onto the error direction as

$$x_t(v) = \frac{\langle F_K(c_t, v_t), E_t \rangle}{L}. \tag{11}$$

Under CLEF, the modulated logits are $\mathrm{Logits}'_t(v) = \mathrm{Logits}_t(v) + \delta_t \cdot x_t(v)$, which induces the following token distribution:

$$p_t^{\delta}(v) = \frac{p_t(v)\exp(\delta_t x_t(v))}{\sum_{u \in V} p_t(u)\exp(\delta_t x_t(u))} \tag{12}$$

Let

$$A_t(\delta_t) = \mathbb{E}_{v \sim p_t^{\delta}}[x_t(v)] \tag{13}$$

denote the expected projection of the sampled token feature onto the current tracking error. Then,

$$\frac{\partial A_t(\delta_t)}{\partial \delta_t} = \mathrm{Var}_{v \sim p_t^{\delta}}\big(x_t(v)\big) \geq 0. \tag{14}$$

The proof is provided in Appendix B. This proposition shows that the CLEF modulation term does not merely impose a fixed payload bias. Instead, it exponentially upweights candidate tokens whose feature vectors have larger projections onto the current tracking error E_t. This result provides a local explanation for the feedback advantage. Since $F_K(c_t, v) \in \{-1, +1\}^L$, we have $\|F_K(c_t, v)\|^2 = L$. Consider the one-step residual with respect to the current error before the target trajectory advances:

$$\widetilde{E}_{t+1} = E_t - F_K(c_t, v_t). \tag{15}$$

Then,

$$\mathbb{E}\|\widetilde{E}_{t+1}\|^2 = \|E_t\|^2 + L - 2\mathbb{E}\langle F_K(c_t, v_t), E_t \rangle. \tag{16}$$

Therefore, increasing the expected projection on $\mathbb{E}\langle F_K(c_t, v_t), E_t \rangle$ reduces the expected one-step residual with respect to the current accumulated error.

It is important to note that this does not imply monotonic convergence of the tracking error, because the target trajectory also advances by αM at each generation step:

$$E_{t+1} = E_t + \alpha M - F_K(c_t, v_t). \tag{17}$$

Rather, Proposition 2 shows that CLEF locally allocates the modulation budget toward correcting the currently accumulated discrepancy. In contrast, an open-loop variant that always modulates tokens along the fixed payload direction M cannot adapt its modulation direction when stochastic sampling causes certain bit dimensions to lag behind the target trajectory. This directional adaptation is the key mechanism by which CLEF mitigates accumulated embedding deviations in short-text generation.

4 Empirical Evaluations

Our main evaluation focuses on short-text multi-bit watermarking scenarios with generation lengths of 50 and 100 tokens and payload sizes from 8 to 64 bits. The main experiments use Meta-Llama-3.1-8B (Llama Team, 2024) with prompts from C4 (Raffel et al., 2020), while downstream utility is evaluated on CNN/DailyMail (Nallapati et al., 2016) and WMT16 (Bojar et al., 2016). We compare against representative multi-bit watermarking baselines, including BiMark (Feng et al., 2025), MPAC (Yoo et al., 2024), WaterMod (Park et al., 2026), StealthInk (Jiang et al., 2025), and XMark (Xu et al., 2026), and report extraction reliability, decoding efficiency, robustness, and generation utility. Detailed experimental settings are deferred to Appendix C. Further experimental results are detailed in Appendix D.

4.1 Extraction Reliability in Short Texts

We first evaluate extraction reliability at target generation lengths of 50 and 100 tokens and payload sizes from 8 to 64 bits. We report both bit extraction accuracy and complete message recovery rate (CMR), defined as the percentage of examples for which all payload bits are correctly recovered. For

Table 1: Bit extraction accuracy (%) and generation quality (PPL) across payload capacities and text lengths.

Bits	Method	50 tokens		100 tokens	
		Acc. (%)	PPL	Acc. (%)	PPL
8	BiMark	95.26	6.36	97.62	6.04
	MPAC	90.63	7.47	95.83	6.87
	StealthInk	75.32	**6.16**	82.91	**6.02**
	WaterMod	83.93	7.56	86.21	7.18
	XMark($\delta = 2$)	87.26	7.18	94.95	7.13
	XMark($\delta = 4$)	97.04	9.28	99.31	9.09
	CLEF	**99.69**	8.09	**100.00**	8.56
16	BiMark	85.55	6.74	93.31	6.44
	MPAC	81.58	7.33	90.92	7.68
	StealthInk	72.87	**6.49**	74.04	**6.07**
	WaterMod	72.26	7.81	76.25	8.77
	XMark($\delta = 2$)	77.35	6.96	89.26	6.61
	XMark($\delta = 4$)	88.42	9.55	95.94	9.05
	CLEF	**95.71**	7.89	**99.14**	8.98
32	BiMark	66.35	6.81	82.69	6.23
	MPAC	71.42	7.65	79.97	7.66
	StealthInk	60.33	**6.26**	65.83	**6.17**
	WaterMod	69.92	7.80	72.61	8.72
	XMark($\delta = 2$)	67.98	6.86	78.64	6.98
	XMark($\delta = 4$)	75.46	8.98	89.64	9.37
	CLEF	**87.15**	8.25	**97.28**	8.58
64	BiMark	48.02	6.79	59.60	6.27
	MPAC	65.91	7.37	71.43	7.32
	StealthInk	54.21	**6.13**	57.54	**5.70**
	WaterMod	64.85	8.61	67.38	8.09
	XMark($\delta = 2$)	61.07	7.12	69.28	6.72
	XMark($\delta = 4$)	66.61	8.93	74.15	8.95
	CLEF	**72.84**	6.92	**85.27**	7.59

CMR, outputs shorter than half the target length are excluded within each run, and the rates computed on retained outputs are averaged over three different random seeds.

As shown in Table 1, CLEF achieves the highest bit extraction accuracy across all evaluated payload sizes and text lengths. Under the 32-bit/50-token and 64-bit/50-token settings, CLEF achieves 87.15% and 72.84% bit extraction accuracy, respectively, compared with 75.46% and 66.61% for XMark ($\delta = 4$). Table 2 further shows that these gains extend to exact message recovery. With a target length of 50 tokens and a 16-bit payload, CLEF achieves 82.14% CMR, compared with 33.33% for XMark ($\delta = 4$). At a target length of 100 tokens with a 32-bit payload, CLEF reaches 73.03%, exceeding XMark ($\delta = 4$) by 59.11 percentage points. CLEF achieves the highest CMR in all reported settings, showing that its extraction advantage also improves the fraction of messages recovered without any bit errors.

This result supports the motivation of CLEF: when the token budget is limited, open-loop schemes suffer from sparse observations for each bit, whereas closed-loop feedback allows the generator to adaptively compensate for underrepresented dimensions during generation. Results for longer outputs of up to 1024 tokens, additional datasets, and Qwen2.5-3B (Yang et al., 2024) are provided in Appendix D.2. Known-payload binary verification results at fixed false positive rates are reported separately in Appendix D.6.

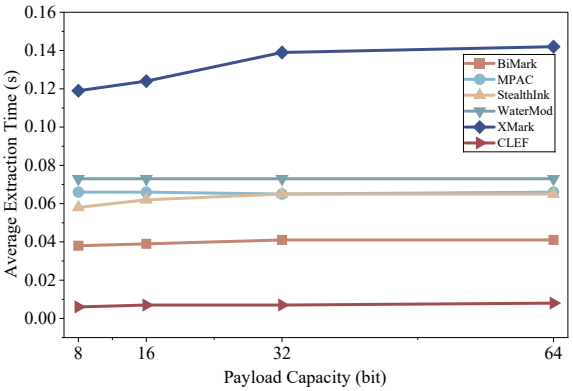


Figure 2: Average extraction time on 50-token generated texts across different payload capacities.

4.2 Extraction Efficiency

To further evaluate the practical efficiency of watermark extraction, we measure the average wall-clock time required to recover the embedded payload from generated texts. The measurement excludes the generation process and focuses only on the decoding stage. As shown in Fig. 2, these results demonstrate that the proposed closed-loop control does not introduce additional complexity during decoding. Instead, by concentrating adaptive correction on the generation side, CLEF enables a simple and efficient linear integration decoder. This supports the design motivation that CLEF improves extraction reliability while maintaining low decoding overhead.

4.3 Reliability–Quality Trade-off Curves

To examine whether the gain of CLEF is merely caused by stronger logit perturbation, we sweep the watermark strength of each method and visualize the resulting trade-off between watermark reliability and text quality in Fig. 3.

As shown in Fig. 3, CLEF consistently achieves a favorable trade-off across the practical operating range. In particular, the CLEF curve remains clearly above baselines, indicating that for a com-

Table 2: Complete message recovery rate (CMR, %) on C4 using Meta-Llama-3.1-8B across payload sizes and target generation lengths. A message is counted as recovered only when all payload bits are correct. Values are means over three different random seeds, computed after excluding outputs shorter than half the target length in each run. Bold values indicate the best result in each setting.

	50 tokens				**100 tokens**			
Method	8 bits	16 bits	32 bits	64 bits	8 bits	16 bits	32 bits	64 bits
BiMark	64.04	11.43	0.00	0.00	82.02	34.67	2.67	0.00
MPAC	48.82	12.72	1.96	0.00	79.77	32.18	2.41	0.00
StealthInk	28.08	0.00	0.00	0.00	41.86	18.60	0.00	0.00
WaterMod	19.54	0.00	0.00	0.00	38.37	4.49	0.00	0.00
XMark ($\delta = 4$)	87.35	33.33	3.70	0.00	95.60	70.65	13.92	0.00
CLEF	**97.59**	**82.14**	**16.00**	**2.24**	**100.00**	**96.47**	**73.03**	**3.45**

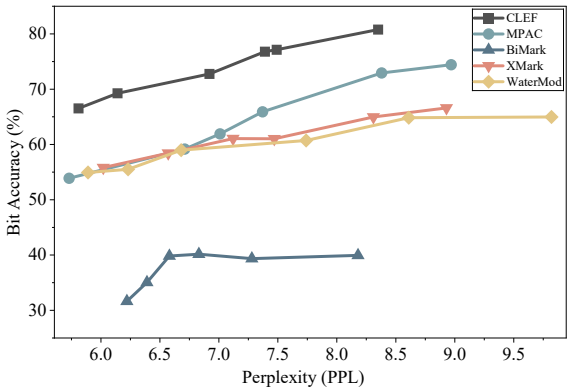


Figure 3: **Reliability–quality trade-off curves under the 50-token, 64-bit setting.** Each point corresponds to one watermark-strength configuration. The horizontal axis denotes perplexity (PPL), and the vertical axis denotes bit extraction accuracy. Curves closer to the upper-left region indicate a better trade-off between watermark reliability and text quality.

Table 3: Bit extraction accuracy (%) under DIPPER paraphrase attack (100 tokens). (lex_diversity, order_diversity) represents attack strength. The lex_diversity parameter controls word substitution intensity, while order_diversity refers to the extent of syntactic and word-order rearrangement in paraphrasing.

Bits	**Method**	**Attack Strength**			
		(0,0)	(20,20)	(40,20)	(40,40)
8	BiMark	97.62	79.62	67.86	65.62
	MPAC	95.83	80.58	71.74	67.12
	StealthInk	82.91	64.49	61.17	57.81
	WaterMod	86.21	64.46	62.03	58.43
	XMark($\delta = 2$)	94.95	77.74	70.12	68.90
	XMark($\delta = 4$)	99.31	80.48	73.31	69.59
	CLEF	**100.00**	**85.79**	**79.62**	**75.87**
16	BiMark	93.31	69.54	61.87	59.76
	MPAC	90.92	70.21	66.67	63.03
	StealthInk	74.04	61.92	57.06	56.10
	WaterMod	76.25	57.21	52.76	51.25
	XMark($\delta = 2$)	89.26	68.75	62.19	60.03
	XMark($\delta = 4$)	95.94	76.56	68.61	61.96
	CLEF	**99.14**	**81.10**	**72.16**	**69.58**

parable quality cost, CLEF yields higher bit extraction accuracy. These observations indicate that CLEF's gain does not merely come from stronger distortion, but from more efficient feedback-based allocation of watermark strength across lagging bit dimensions.

4.4 Robustness to Text Perturbations

Robustness against paraphrasing attacks. Table 3 evaluates robustness under DIPPER paraphrasing. CLEF retains higher extraction accuracy after paraphrasing, especially under stronger attacks. Under the strongest DIPPER setting with a 16-bit payload, CLEF preserves 69.58% accuracy, outperforming the strongest evaluated baselines. These results indicate that the global integration decoder can tolerate moderate perturbations, as the remaining token feature evidence still provides a recoverable signal.

Robustness against truncation, insertion, and deletion. We further evaluate robustness against truncation, random insertion, and random deletion. Truncation primarily reduces the amount of surviving watermark evidence, whereas insertion and deletion can additionally introduce synchronization errors because the token feature function depends on the generation context.

4.5 Generation Quality and Utility

Watermark injection inevitably introduces a trade-off with text quality. We measured the impact of our method on downstream natural language processing tasks.

Table 5 and Table 6 show that CLEF incurs moderate quality degradation from logit modulation,

Table 4: Bit extraction accuracy (%) under truncation, random insertion, and random deletion with perturbation ratio 0.2 in the 100-token, 32-bit setting.

Method	Truncation	Insertion	Deletion
BiMark	67.29	65.30	59.16
MPAC	78.79	71.76	71.35
StealthInk	64.01	60.83	59.85
WaterMod	67.82	57.76	55.89
XMark ($\delta = 4$)	85.34	73.51	72.95
CLEF	**97.11**	**83.02**	**82.11**

Table 5: Generation quality on the summarization task.

Method	BERTScore	ROUGE-1	PPL
No watermark	0.2609	0.3528	2.03
StealthInk	**0.2665**	**0.3606**	**2.29**
MPAC($\delta = 1.5$)	0.2514	0.3399	2.71
MPAC($\delta = 2.0$)	0.2533	0.3312	4.10
BiMark	0.2626	0.3485	2.57
WaterMod	0.2609	0.3410	3.51
XMark	0.2578	0.3507	3.31
CLEF	0.2582	0.3456	3.85

Table 6: Generation quality on the machine translation task.

Method	BERTScore	BLEU
No watermark	0.7323	0.3536
StealthInk	**0.7207**	**0.3507**
MPAC($\delta = 1.5$)	0.7038	0.3228
MPAC($\delta = 2.0$)	0.6902	0.3103
BiMark	0.7110	0.3248
WaterMod	0.6952	0.3007
XMark	0.7024	0.3190
CLEF	0.7040	0.3119

but remains comparable to existing multi-bit watermarking baselines on summarization and translation. Overall, CLEF improves extraction reliability while keeping downstream utility within the range of prior watermarking methods.

4.6 Ablation Study

Table 7 shows that the complete CLEF scheme achieves the best extraction performance across all settings. Removing the feedback mechanism causes the bit extraction accuracy to collapse to nearly random level behavior. This result confirms that dynamic error feedback is the primary source of CLEF's gain. Without conditioning token selection on the current tracking error, early stochastic deviations cannot be alleviated and thus accumulate quickly throughout generation.

By contrast, the w/o state variant remains between the full model and the w/o feedback baseline.

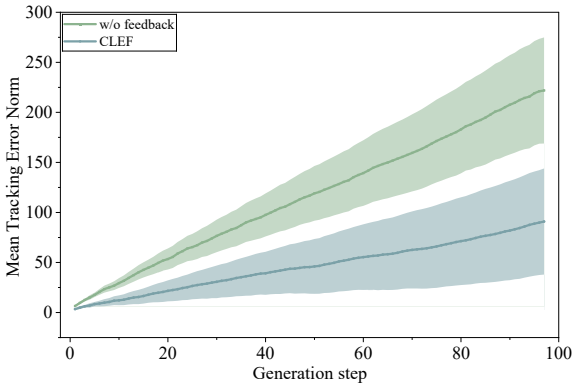


Figure 4: Evolution of the mean tracking error norm over generation steps.

Table 7: Ablation study results on different text lengths and payload capacities. In w/o state, the controller uses the target direction but does not accumulate the realized feature state S_{t-1}. In w/o feedback, the modulation direction is fixed to the payload vector M, the score is computed without using the real-time error E_t.

Ablation Study		32 bits		64 bits	
		Acc. (%)	PPL	Acc. (%)	PPL
50 tokens	CLEF	**87.15**	8.25	**72.84**	6.92
	w/o state	77.29	8.92	63.83	7.66
	w/o feedback	51.46	**5.61**	48.58	**5.57**
100 tokens	CLEF	**97.28**	8.58	**85.27**	7.59
	w/o state	94.68	12.71	79.06	8.67
	w/o feedback	51.20	**5.54**	50.57	**5.75**

This indicates that trajectory guidance alone still provides a weak directional prior, but without accurate state accumulation, the model cannot estimate how far the current generation has already deviated from the target, and therefore cannot perform precise online correction.

The PPL results show that w/o feedback causes weaker distributional distortion (Dathathri et al., 2020) but loses watermark reliability, whereas w/o state can produce unstable fluency due to overcorrection without accurate state tracking. Thus, feedback mainly determines reliability, while state accumulation stabilizes correction.

To better understand the behavior of the proposed closed-loop mechanism, we visualize the evolution of the error norm $||E_t||$ over generation steps. As shown in Fig. 4, the open-loop variant exhibits a rapid growth in error, indicating uncontrolled deviation from the target trajectory. CLEF does not force the tracking error to vanish; instead, it reduces the growth rate of the accumulated track-

ing error relative to the open-loop variant.

We attribute this performance gap to the different error dynamics between open-loop and closed-loop designs. In the open-loop variant, each token is selected based solely on the fixed target payload, without considering previously accumulated deviations. As a result, stochastic errors introduced at early steps cannot be alleviated, leading to error accumulation over time. In contrast, the proposed closed-loop mechanism continuously tracks the global embedding state and adjusts token selection based on the real-time error signal. This enables online correction of past mistakes, effectively alleviating error propagation and significantly improving decoding reliability.

We further analyze the sensitivity of CLEF to the trajectory slope α and initial watermark strength δ_0 in Appendix D.1. The results show that larger α and δ_0 generally improve extraction accuracy at the cost of higher perplexity, and our default setting is $\alpha = 0.4$, $\delta_0 = 0.8$.

5 Conclusion

In this paper, we studied reliable multi-bit watermarking for short-text generation in large language models. We identified the key limitation of existing open-loop schemes: they cannot correct accumulated embedding deviations under limited token budgets. To address this issue, we proposed CLEF, a closed-loop error feedback framework that tracks the global watermark state during generation and adaptively modulates candidate-token logits according to the real-time tracking error. This enables dynamic correction while keeping decoding linear.

Experiments show that CLEF improves both bit extraction accuracy and complete message recovery over the evaluated baselines under short generation budgets. Robustness, utility, and ablation results further confirm the effectiveness of feedback-based state tracking. These findings suggest that closed-loop generation control is a promising direction for scalable and reliable multi-bit watermarking of LLM-generated text.

Limitations

Despite the effectiveness of the proposed framework, several directions remain for further investigation.

First, our closed-loop formulation is designed to reduce the growth of trajectory-tracking error during generation, rather than to enforce exact convergence at every step. In practice, stochastic token sampling and the inherent mismatch between discrete token decisions and continuous target trajectories can introduce residual tracking deviations. Future work may explore stronger stability guarantees or adaptive correction mechanisms to further improve long-horizon tracking behavior.

Second, our evaluation focuses on controlled generation settings with fixed output lengths and payload sizes, which allows us to isolate and analyze the core behavior of the proposed method. Real-world deployments may involve more diverse generation-length distributions, multi-turn dialogue contexts, and domain-specific text distributions, which are not systematically evaluated in this work. Extending CLEF with adaptive payload allocation and redundancy strategies is a promising direction for improving robustness in these broader settings.

Ethical Considerations

This work aims to support provenance tracking and accountability for LLM-generated text through multi-bit watermarking. The proposed method does not require collecting personal data or identifying individuals. In practical deployments, payloads should avoid directly encoding identifiable information, and any mapping between payloads and metadata should be protected by proper access control. Watermark recovery is intended as a provenance signal rather than definitive evidence, and should be combined with other verification methods in high-stakes scenarios. Potential misuse, such as false attribution or unauthorized watermark manipulation, should be mitigated through responsible key management and deployment protocols.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62572208, 62402513), the Innovative Scientific and Technological Talents Program of Hubei Province, China (Project No. 2025DJA050), and the Natural Science Foundation of Hubei Province (Joint Fund Project), China (Grant No. 2025AFD748).

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie

Névéol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, and 2 others. 2016. Findings of the 2016 conference on machine translation. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Miranda Christ, Sam Gunn, Tal Malkin, and Mariana Raykova. 2024. Provably Robust Watermarks for Open-Source Language Models. *Preprint*, arXiv:2410.18861.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. 2020. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, Po-Sen Huang, Rob McAdam, Johannes Welbl, Vandana Bachani, Alex Kaskasoli, Robert Stanforth, Tatiana Matejovicova, Jamie Hayes, Nidhi Vyas, Majd Al Merey, Jonah Brown-Cohen, Rudy Bunel, Borja Balle, A. Taylan Cemgil, Zahra Ahmed, Kitty Stacpoole, and 5 others. 2024. Scalable watermarking for identifying large language model outputs. *Nature*, 634(8035):818–823.

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

Yuxuan Guo, Zhiliang Tian, Yiping Song, Tianlun Liu, Liang Ding, and Dongsheng Li. 2024. Context-aware watermark with semantic balanced green-red lists for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 22633–22646.

Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2024. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4067–4082.

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

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. 2025. StealthInk: A multi-bit and stealthy watermark for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 27685–27709. PMLR.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2024. On the reliability of watermarks for large language models. In *International Conference on Learning Representations*, volume 2024, pages 49660–49704.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. 2023. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. *Advances in neural information processing systems*, 36:27469–27500.

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

Jiacheng Liang, Zian Wang, Spencer Hong, Shouling Ji, and Ting Wang. 2025. Watermark under fire: A robustness evaluation of LLM watermarking. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 21050–21074, Suzhou, China. Association for Computational Linguistics.

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

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

Yepeng Liu and Yuheng Bu. 2024. Adaptive text watermark for large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 30718–30737. PMLR.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *Preprint*, arXiv:1907.11692.

Llama Team. 2024. The Llama 3 Herd of Models. *CoRR*, abs/2407.21783.

Ramesh Nallapati, Bowen Zhou, Cícero Nogueira dos Santos, Çaglar Gülçehre, and Bing Xiang. 2016. Abstractive Text Summarization using Sequence-to-sequence RNNs and Beyond. In *Conference on Computational Natural Language Learning*.

OpenAI. 2023. GPT-4 Technical Report. *CoRR*, abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Shinwoo Park, Hyejin Park, Hyeseon Ahn, and Yo-Sub Han. 2026. WaterMod: Modular Token-Rank Partitioning for Probability-Balanced LLM Watermarking. In *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence and Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence and Sixteenth Symposium on Educational Advances in Artificial Intelligence*. AAAI Press.

Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. 2025. Markmywords: Analyzing and evaluating language model watermarks. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pages 68–91. IEEE.

Wenjie Qu, Wengrui Zheng, Tianyang Tao, Dong Yin, Yanze Jiang, Zhihua Tian, Wei Zou, Jinyuan Jia, and Jiaheng Zhang. 2025. Provably Robust Multi-Bit Watermarking for AI-Generated Text. In *34th USENIX Security Symposium (USENIX Security 25)*, pages 201–220.

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

Jiahao Xu, Rui Hu, Olivera Kotevska, and Zikai Zhang. 2026. XMark: Reliable multi-bit watermarking for LLM-generated texts. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14747–14763, San Diego, California, United States. Association for Computational Linguistics.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *CoRR*, abs/2412.15115.

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

KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. 2024. Advancing beyond identification: Multi-bit watermark for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4031–4055.

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

Appendix

A Algorithm

We provide the pseudo-code of CLEF in Algorithms 1 and 2. Algorithm 1 summarizes the encoder, which performs closed-loop error feedback during token generation by comparing the accumulated token state with a payload-dependent target trajectory. Algorithm 2 summarizes the decoder, which aggregates token-level features from the generated sequence and recovers the embedded payload using a lightweight linear rule. These algorithms complement the main text by making the encoding, feedback, and recovery procedures explicit.

Algorithm 1: The Encoder of Closed-Loop Error Feedback (CLEF)

Input : Vocabulary $\mathcal{V}$, Initial context prefix c_1, Target message $M \in \{-1,1\}^L$, Generation length N, Trajectory slope $\alpha \in (0,1]$, Watermark strength δ_t, Context window size n.
Output : Watermarked token sequence Y.

$S_0 \leftarrow \mathbf{0} \in \mathbb{R}^L$;
$Y \leftarrow \emptyset$;
for $t \leftarrow 1$ **to** N **do**
  Compute base $\text{Logits}_t(v)$ for $v \in \mathcal{V}$ via LLM given c_t;
  $T_t \leftarrow \alpha \cdot t \cdot M$;
  $E_t \leftarrow T_t - S_{t-1}$;
  for $v \in \mathcal{V}_t$ **do**
    $F_K(c_t, v_t) \in \{-1,1\}^L$;
    $\text{Score}(v) = \frac{1}{L}\sum_{i=1}^{L} F_K(c_t, v_t)_i \cdot (E_t)_i$;
    $\text{Logits}'_t(v) \leftarrow \text{Logits}_t(v) + \delta_t \cdot \text{Score}(v)$;
  end
  Sample $v_t^* \sim \text{Softmax}(\text{Logits}'_t)$;
  Append v_t^* to Y;
  $S_t \leftarrow S_{t-1} + F_K(c_t, v_t^*)$;
  $c_{t+1} \leftarrow (v_{t-n+1}^*, ..., v_t^*)$;
end
return Y;

B Proof of Proposition 2

For a fixed generation step t, the prefix c_t, the tracking error E_t, and the token features $F_K(c_t, v)$ are fixed with respect to δ_t. Therefore, the projection

$$x_t(v) = \frac{\langle F_K(c_t, v), E_t \rangle}{L}$$

can be treated as a fixed scalar associated with each candidate token v.

Recall that the modulated token distribution is

$$p_t^\delta(v) = \frac{p_t(v)\exp(\delta_t x_t(v))}{\sum_{u\in V} p_t(u)\exp(\delta_t x_t(u))}.$$

Algorithm 2: The Decoder of Closed-Loop Error Feedback (CLEF)

Input : Received token sequence $Y = \{v_1^*, \ldots, v_N^*\}$, Initial context prefix c_1, Payload length L, Context window size n.
Output : Extracted message $\hat{M} \in \{-1,1\}^L$.

$V_{\text{decode}} \leftarrow \mathbf{0} \in \mathbb{R}^L$;
for $t \leftarrow 1$ **to** N **do**
  $F_K(c_t, v_t^*) \in \{-1,1\}^L$;
  $V_{\text{decode}} \leftarrow V_{\text{decode}} + F_K(c_t, v_t^*)$;
  $c_{t+1} \leftarrow (v_{t-n+1}^*, ..., v_t^*)$;
end
for $i \leftarrow 1$ **to** L **do**
  if $(V_{decode})_i > 0$ **then**
    $\hat{M}_i \leftarrow 1$;
  else
    $\hat{M}_i \leftarrow -1$;
  end
end
return $\hat{M}$;

Define the normalizing constant

$$Z(\delta_t) = \sum_{u\in V} p_t(u)\exp(\delta_t x_t(u)).$$

The expected corrective projection is

$$A_t(\delta_t) = \mathbb{E}_{v\sim p_t^\delta}[x_t(v)] = \sum_{v\in V} p_t^\delta(v) x_t(v).$$

Substituting the definition of $p_t^\delta(v)$ gives

$$A_t(\delta_t) = \frac{\sum_{v\in V} p_t(v)x_t(v)\exp(\delta_t x_t(v))}{Z(\delta_t)}.$$

Since

$$Z'(\delta_t) = \sum_{v\in V} p_t(v)x_t(v)\exp(\delta_t x_t(v)),$$

we have

$$A_t(\delta_t) = \frac{Z'(\delta_t)}{Z(\delta_t)}$$

Taking another derivative with respect to δ_t, we obtain

$$\frac{\partial A_t(\delta_t)}{\partial \delta_t} = \frac{Z''(\delta_t)Z(\delta_t) - [Z'(\delta_t)]^2}{[Z(\delta_t)]^2}.$$

By the definition of $p_t^\delta(v)$,

$$\frac{Z''(\delta_t)}{Z(\delta_t)} = \mathbb{E}_{v\sim p_t^\delta}[x_t(v)^2],$$

and

$$\left(\frac{Z'(\delta_t)}{Z(\delta_t)}\right)^2 = \left(\mathbb{E}_{v\sim p_t^\delta}[x_t(v)]\right)^2.$$

Therefore,

$$\frac{\partial A_t(\delta_t)}{\partial \delta_t} = \mathbb{E}_{v\sim p_t^\delta}[x_t(v)^2] - \left(\mathbb{E}_{v\sim p_t^\delta}[x_t(v)]\right)^2.$$

The right-hand side is exactly the variance of $x_t(v)$ under the modulated distribution p_t^δ, namely

$$\frac{\partial A_t(\delta_t)}{\partial \delta_t} = \mathrm{Var}_{v\sim p_t^\delta}\big(x_t(v)\big).$$

Since variance is always non-negative, we have

$$\frac{\partial A_t(\delta_t)}{\partial \delta_t} = \mathrm{Var}_{v\sim p_t^\delta}\big(x_t(v)\big) \geq 0.$$

This completes the proof.

C Experimental Settings

C.1 Models and Datasets

To evaluate the proposed Closed-Loop Error Feedback multi-bit watermarking scheme, we conduct our primary text generation experiments using Meta-Llama-3.1-8B (Llama Team, 2024) with temperature 1.0 and top-50 sampling. For the standard short-text capacity and robustness evaluations, we utilize prompt prefixes sampled from the C4 dataset (Raffel et al., 2020). To assess the impact of the watermark on specific downstream generation tasks, we employ standard benchmarks CNN/DailyMail (Nallapati et al., 2016) for text summarization and WMT16 (Bojar et al., 2016) for machine translation. These datasets are publicly available benchmarks. We do not create new datasets or collect demographic attributes. Unless otherwise specified, all experiments randomly select 200 prompts.

We use publicly available models and datasets in accordance with their respective licenses and terms of use. Meta-Llama-3.1-8B, C4, CNN/DailyMail, and WMT16 are used only for research evaluation. Since C4 is derived from web text and may contain personally identifying or offensive content, we use only prompt prefixes for controlled generation experiments and do not attempt to recover, store, or release any personal information. We do not use the datasets to profile individuals, and all reported results are aggregate statistics.

C.2 Baseline Methods

We benchmark our proposed method against five representative multi-bit LLM watermarking schemes. BiMark (Feng et al., 2025): A representative time-division multiplexing baseline that allocates specific subsets of the generated sequence to encode individual bits. MPAC (Yoo et al., 2024): An approach designed to embed multi-bit messages by dividing bit subsets and allocating positions. WaterMod (Park et al., 2026): A method sorts the vocabulary in descending order of model probability and partitions it based on the residue of the rank modulo k, subsequently applying a bias to embed the watermark. StealthInk (Jiang et al., 2025): An optimization-based watermarking baseline focused on maintaining generation quality. XMark (Xu et al., 2026): A robust, context-driven logit modulation scheme. For a comprehensive comparison, we evaluate XMark under two distinct watermark strength settings: standard ($\delta = 2$) and aggressive ($\delta = 4$).

To ensure a fair comparison, all methods are evaluated under the same base model, prompt set, decoding configuration, generation length, and payload length. For methods with tunable watermark strength, we evaluate their default configurations.

C.3 Implementation Details and Hyperparameters

The primary objective of our main experiments is to evaluate performance in short-text scenarios. Consequently, the main experiments use generation lengths of 50 and 100 tokens, with payload capacities of 8, 16, 32, and 64 bits. Additional evaluations on longer generation lengths are provided in Appendix D.2. For both encoding and decoding, we set the context window size to $n = 2$, $F_K(c_t, v)$ is conditioned on the two most recent tokens in the current context. The same setting is used throughout all experiments. For our proposed method, the trajectory slope α is empirically set to 0.4. The trajectory slope α controls the desired per-step accumulation margin along the payload direction. A smaller α produces a flatter target trajectory, which reduces logit distortion but also yields a smaller final decoding margin, making the sign-based decoder more sensitive to sampling noise and perturbations. A larger α creates a steeper target trajectory and increases the desired watermark margin, but it may become difficult for the discrete token-feature process to track, thereby requiring stronger logit modulation and causing larger quality degradation. In practice, we found it beneficial to use a step-dependent watermark strength instead of a fixed δ. Let t denote the number of valid generation steps. The watermark strength is scheduled

as

$$\delta_t = \begin{cases} \delta_0, & t < 40, \\ \max(\delta_{\min}, \delta_0 \gamma^{t-40}), & t \geq 40, \end{cases} \tag{18}$$

where $\delta_0 = 0.8$, $\gamma = 0.99$, and $\delta_{\min} = 0.1$. We adopt a step-decaying schedule for the watermark strength: early in generation, a relatively strong modulation helps the system quickly align the state trajectory with the target, avoiding large early deviations that become increasingly costly to correct; as generation proceeds, we gradually reduce the strength to mitigate excessive logit distortion, preserving the naturalness of the final text. This schedule is an empirical heuristic motivated by Proposition 2 and tuned on preliminary experiments, rather than the theoretically optimal control law. Further analysis of CLEF's sensitivity to trajectory slope α and initial watermark strength δ_0 is provided in Appendix D.1.

C.4 Evaluation Metrics

The performance is evaluated across four principal dimensions:

Extraction Reliability: We report bit extraction accuracy and complete message recovery rate (CMR). Bit extraction accuracy measures the fraction of correctly recovered payload bits, whereas CMR requires all bits in a message to be correct. For the CMR evaluation, each configuration is evaluated using three different random seeds. In each run, we exclude generated outputs shorter than half the target length.

Computational Efficiency: Measured by the average wall-clock time required to extract the embedded message payload from generated texts. Generation time is excluded, and all methods are evaluated under the same hardware, implementation environment, generated text length, and payload settings.

Text Quality: Quantified using generation Perplexity (PPL) to measure fluency. For downstream tasks, we utilize BERTScore (Zhang et al., 2020) and ROUGE-1 (Lin, 2004) for summarization, and BERTScore and BLEU (Papineni et al., 2002) for machine translation.

Robustness: We evaluate robustness using bit extraction accuracy after applying five types of text perturbations to the watermarked outputs. First, we consider semantic paraphrasing using DIPPER (Krishna et al., 2023) under varying lexical and word-order diversity settings, ranging from $(20, 20)$ to $(40, 40)$. Second, we evaluate truncation, random

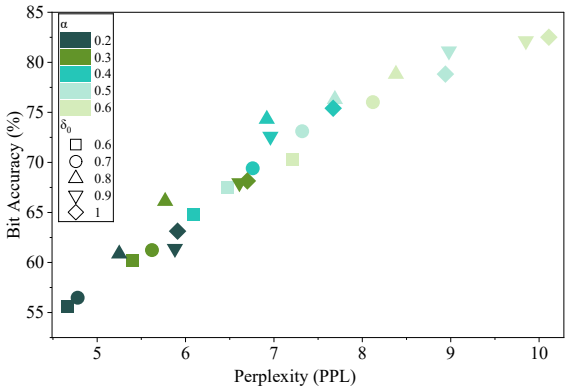


Figure 5: Impact of trajectory slope (α) and initial watermark strength (δ_0) on the trade-off between generation perplexity and bit extraction accuracy.

token insertion, and random token deletion, each with a perturbation ratio of 0.2. Finally, we consider random token replacement attacks with corruption ratios ranging from 0.1 to 0.3.

Downstream-task evaluation is implemented with the HuggingFace evaluate package. For summarization, ROUGE is computed after NLTK sentence segmentation, with stemming and score aggregation enabled. Summarization BERTScore uses the English setting with RoBERTa-large (Liu et al., 2019) and baseline rescaling. For machine translation, BLEU is computed with the default SacreBLEU configuration, and BERTScore uses the English setting with baseline rescaling. Mean BERTScore values are reported across examples.

Unless otherwise specified, all reported accuracy, perplexity, BERTScore, ROUGE, BLEU, and extraction-time results are averages over 200 randomly sampled prompts.

D Additional Experiments

D.1 Hyperparameter Analysis

To justify our configuration strategy, we analyze the sensitivity of the trajectory slope α and initial watermark strength δ_0 on the reliability-quality trade-off with 50 tokens and 64 bits, as illustrated in Fig. 5. The trajectory slope α dictates the desired per-step accumulation margin. Empirically, larger α values yield higher bit accuracy but significantly elevate perplexity. From a control-theoretic perspective, a steeper target trajectory amplifies the tracking error E_t, forcing the controller to heavily penalize naturally probable tokens lacking payload alignment.

Additionally, the initial strength δ_0 governs the

Table 8: Performance under longer generation lengths with a 64-bit payload. Bit Acc. denotes bit extraction accuracy, and CMR denotes complete message recovery rate, i.e., the percentage of samples for which all 64 payload bits are correctly recovered.

Method	200 tokens, 64 bits			512 tokens, 64 bits			1024 tokens, 64 bits		
	Bit Acc. (%)	CMR (%)	PPL	Bit Acc. (%)	CMR (%)	PPL	Bit Acc. (%)	CMR (%)	PPL
BiMark	68.75	0.00	5.91	81.14	0.00	5.33	90.65	10.53	5.35
MPAC	80.29	3.66	7.01	92.37	14.75	6.73	94.49	28.57	6.55
StealthInk	66.31	0.00	**5.84**	75.75	0.00	**5.05**	80.26	0.00	**5.47**
WaterMod	70.82	0.00	7.44	76.80	0.00	8.15	79.44	0.00	8.41
XMark ($\delta = 4$)	86.03	3.66	8.77	96.28	32.79	8.18	97.25	63.07	7.90
CLEF	**93.81**	**23.94**	7.53	**98.06**	**68.91**	7.03	**98.72**	**86.42**	7.42

magnitude of early-stage logit modulation. Amplifying δ_0 consistently shifts performance upward along the accuracy axis for any given α. This highlights the necessity of strong early-stage alignment: applying robust corrective feedback before stochastic sampling errors accumulate is crucial for short-text regimes. Ultimately, these dynamics map a distinct Pareto frontier. Our empirical default ($\alpha = 0.4, \delta_0 = 0.8$) is strategically positioned at the knee of this curve, delivering a balance between extraction reliability and linguistic naturalness.

D.2 Generalization Analysis

Length generalization. To evaluate whether CLEF remains effective beyond the short-text regime considered in the main experiments, we extend the generation length to 200, 512, and 1024 tokens under a 64-bit payload. In addition to bit extraction accuracy and perplexity, we report the complete message recovery rate (CMR), defined as the percentage of samples for which all 64 embedded bits are correctly recovered.

As shown in Table 8, CLEF consistently achieves the highest bit extraction accuracy and complete message recovery rate across all evaluated generation lengths. These results indicate that the benefit of closed-loop error feedback persists over substantially longer generation horizons and that the additional token budget substantially improves exact payload recovery.

Dataset generalization. To verify the cross-dataset generalization capability of the proposed CLEF watermarking scheme beyond the C4 dataset utilized in main experiments, we conduct identical evaluation protocols on the OpenGen dataset, with all experimental configurations fully consistent with the main experiments.

The quantitative results are summarized in Table 9. We observe consistent performance trends with the C4 dataset: CLEF achieves the highest bit extraction accuracy across all settings. Meanwhile, the perplexity of generated texts remains comparable with several baseline methods.

Table 9: Bit extraction accuracy (%) and generation quality on the OpenGen Dataset

Bits	Method	50 tokens		100 tokens	
		Acc. (%)	PPL	Acc. (%)	PPL
32	BiMark	58.27	5.25	73.63	5.17
	MPAC	75.35	6.07	83.18	5.96
	StealthInk	60.52	**5.06**	66.41	**4.80**
	WaterMod	63.60	7.11	67.04	7.33
	XMark($\delta = 2$)	67.93	5.63	79.19	5.52
	XMark($\delta = 4$)	79.81	6.89	89.90	7.61
	CLEF	**88.31**	6.82	**96.69**	7.52
64	BiMark	37.76	5.21	54.66	5.12
	MPAC	67.39	5.92	72.86	6.04
	StealthInk	55.96	**5.12**	59.56	**4.64**
	WaterMod	55.84	7.40	64.98	6.63
	XMark($\delta = 2$)	61.53	6.15	66.21	5.57
	XMark($\delta = 4$)	68.99	7.19	75.74	7.48
	CLEF	**70.04**	5.84	**84.02**	6.56

Backbone generalization. To examine whether the effectiveness of CLEF depends on a specific base language model, we further evaluate all methods on Qwen/Qwen2.5-3B under 32-bit and 64-bit payload settings. As shown in Fig. 6, CLEF consistently achieves the highest bit extraction accuracy across both payload capacities. This suggests that the advantage of closed-loop error feedback is not tied to the Llama-3.1-8B backbone used in the main experiments, but can transfer to another model from a different model family.

D.3 Bit Extraction Accuracy under Random Replacement Attack

We also tested the performance of CLEF against random replacement attack. As shown in Table 10, under different attack strengths, our method outperforms the evaluated baselines.

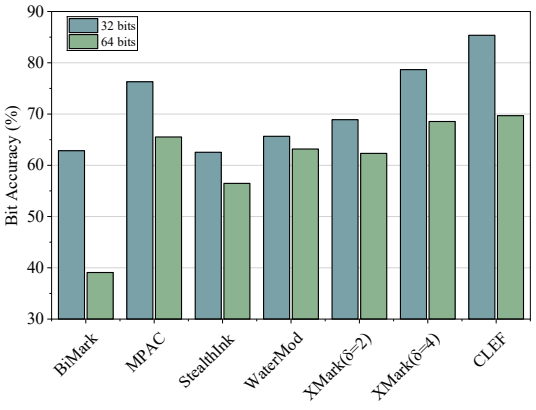


Figure 6: Bit extraction accuracy on Qwen2.5-3B under 32-bit and 64-bit payloads (50 tokens).

Table 10: Bit extraction accuracy (%) under random replacement attack (100 tokens). The replacement ratio indicates the fraction of tokens randomly replaced, with larger ratios corresponding to stronger perturbations.

Bits	Method	Replacement ratio			
		0	0.1	0.2	0.3
8	BiMark	97.62	91.02	81.71	70.37
	MPAC	95.83	84.75	76.50	68.25
	StealthInk	82.91	75.94	68.02	60.53
	WaterMod	86.21	71.32	67.37	62.35
	XMark($\delta = 2$)	94.95	84.63	70.31	61.70
	XMark($\delta = 4$)	99.31	91.22	86.76	70.74
	CLEF	**100.00**	**99.21**	**93.23**	**80.47**
16	BiMark	93.31	83.33	75.38	65.64
	MPAC	90.92	84.45	75.22	67.81
	StealthInk	74.04	67.44	61.47	58.69
	WaterMod	76.25	66.87	62.86	59.37
	XMark($\delta = 2$)	89.26	80.86	68.64	60.49
	XMark($\delta = 4$)	95.94	90.60	78.64	67.11
	CLEF	**99.14**	**96.61**	**87.88**	**78.09**

D.4 Generation-Side Efficiency

In addition to extraction latency reported in the main paper, we measure the generation-side computational cost under the 64-bit, 100-token setting. Table 11 reports the average generation time per sample and the corresponding generation throughput.

D.5 Dimension-Level Trajectory Visualization

To further illustrate the effect of feedback control, we visualize the cumulative token-feature signals for representative payload dimensions. As shown in Fig. 7, the open-loop variant exhibits unstable trajectories that often deviate from the target direction, whereas CLEF progressively steers individual dimensions toward the desired trajectory.

Table 11: Average generation time and generation throughput under the 64-bit, 100-token setting.

Method	Time (s/sample)	Tokens/s
BiMark	0.397	221.2
MPAC	2.589	34.9
StealthInk	2.533	37.4
WaterMod	**0.324**	**274.1**
XMark	0.948	88.5
CLEF	0.450	194.6

This provides additional evidence that the proposed feedback mechanism improves multidimensional coordination during generation.

D.6 Known-Payload Binary Watermark Verification

Table 12: Binary watermark detection performance under the 50-token, 32-bit setting. TPR is reported at fixed false positive rates calibrated on unwatermarked generations. This evaluation only detects watermark presence and does not recover the embedded payload bits.

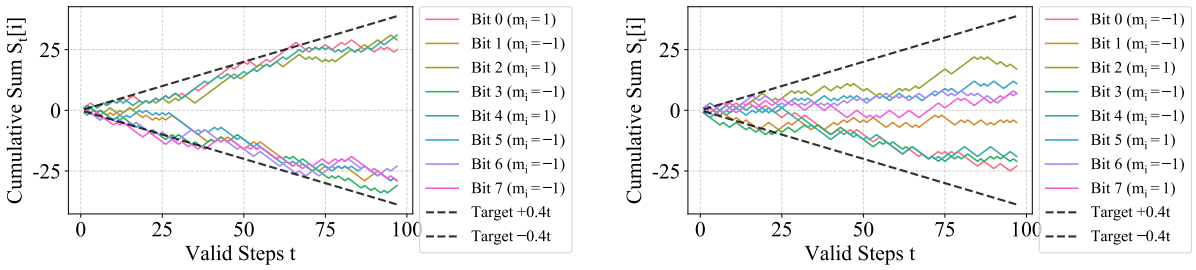
Method	TPR (%)		AUROC
	FPR≤1%	FPR≤0.1%	
BiMark	48.36	36.03	0.7956
MPAC	88.94	81.60	0.9868
StealthInk	65.25	44.16	0.8730
WaterMod	67.53	55.16	0.9084
XMark($\delta = 2$)	87.60	72.42	0.9617
CLEF	**96.95**	**94.91**	**0.9944**

Besides evaluating bit extraction accuracy, we further study whether a text can be identified as containing a specific target payload. This setting differs from payload-agnostic watermark detection: the verifier is given a candidate payload and only decides whether the text was generated with this payload.

For a generated text Y and a target payload $b \in \{0,1\}^L$, we first decode the payload $\hat{b}$ using the standard linear decoder. We then use the bit agreement ratio as the detection score:

$$s(Y, b) = \frac{1}{L} \sum_{i=1}^{L} \mathbf{1}[\hat{b}_i = b_i].$$

For each target false positive rate ρ, we calibrate the threshold τ_ρ on unwatermarked generations by taking the upper $(1-\rho)$ quantile of their scores. A text is classified as watermarked if $s(Y, b) \geq \tau_\rho$. We report the true positive rate on watermarked texts at fixed false positive rates, together with AUROC. The specific results are shown in Table 12.



(a) CLEF

(b) w/o feedback

Figure 7: Evolution of cumulative token signals for representative bit dimensions under CLEF and w/o feedback guidance.

This evaluation measures whether the watermark signal is distinguishable for a known payload, rather than whether the embedded payload can be fully recovered.