ARGH-Mark: Anchor-Synchronized Watermarking with Hamming Correction for Robust and Quality-Preserving LLM Attribution

He Li[123], Xiaojun Chen[123*], Jingcheng He[123], Zhendong Zhao[12], Shuguang Yuan[12], Xin Zhao[123], Yunfei Yang[123]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]State Key Laboratory of Cyberspace Security Defense, Beijing, China
[3]School of Cyber Security,University of Chinese Academy of Sciences, Beijing, China
{lihe2023, chenxiaojun, hejingcheng, zhaozhendong, yuanshuguang, zhaoxin, yangyunfei}@iie.ac.cn

Abstract

The proliferation of large language models has intensified demands for reliable content attribution, yet existing watermarking techniques face a fundamental trilemma: they cannot simultaneously optimize for robustness against attacks, minimal text quality degradation, and detection efficiency. To resolve this challenge, we propose **ARGH-Mark**, a novel watermarking framework that integrates three synergistic innovations: (1) **A**nchor-synchronized phase recovery for maintaining detection integrity under insertion/deletion attacks, (2) **RG**-balanced vocabulary modulation that dynamically partitions lexicons via contextual hashing to preserve generation quality, and (3) **H**amming-based error correction enabling single-bit error rectification through algebraic coding. Comprehensive evaluations across question answering (ELI5), summarization (CNN/DailyMail), and text generation (C4) demonstrate state-of-the-art performance: the proposed ARGH-Mark framework achieves near-perfect match rate and bit accuracy across diverse configurations, while preserving the quality of the generated text. It significantly reduces detection latency, enabling real-time extraction, and maintains high robustness against token tampering attacks through integrated Hamming error correction, ensuring reliable attribution in adversarial settings. ARGH-Mark achieves a new Pareto frontier in the watermarking design space and advances trustworthy deployment of generative AI in alignment-critical applications.

Code — https://github.com/lihe19980424/ARGH-Mark

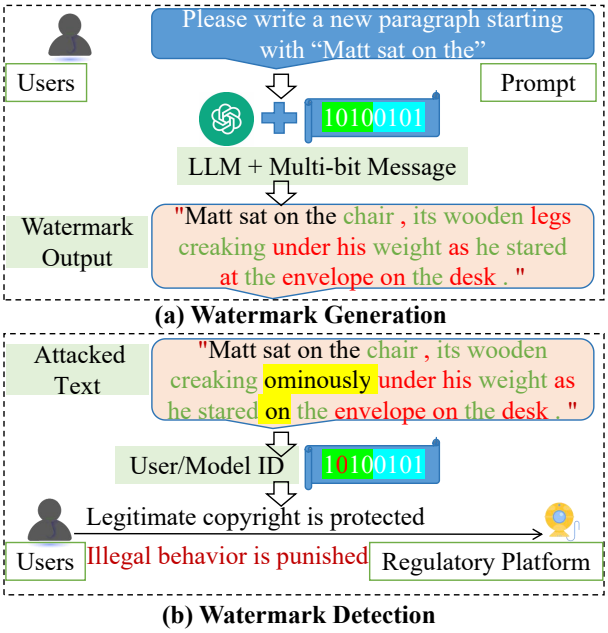

Figure 1: Application scenarios of the proposed watermarking framework. In the watermark generation phase, multi-bit messages are embedded into the generated text, whereas during the watermark detection phase, the embedded information is extracted to support copyright protection and combat illegal activities.

Introduction

The rapid proliferation of large language models (LLMs) (OpenAI 2023; Touvron et al. 2023; DeepSeek-AI 2025; Qwen3 2025) has precipitated an exponential growth in AI-generated content across specialized domains, posing unprecedented threats (Megías et al. 2021; Mirsky et al. 2023) to digital trust ecosystems and regulatory frameworks (Perez et al. 2022; Rillig et al. 2023). Multi-bit watermarking (Fernandez et al. 2023; Qu et al. 2024; Wang et al. 2024; Yoo, Ahn, and Kwak 2024; Cohen, Hoover, and Schoenbach 2025), which embeds traceable identifiers such as model fingerprints or user credentials, emerges as a critical mechanism for content attribution and compliance enforcement (Chen and Shu 2024; Bender et al. 2021; Shumailov et al. 2024). However, fundamental limitations in existing methodologies impede reliable deployment in adversarial environments.

Conventional approaches exhibit three critical deficiencies: vocabulary-partitioning methods suffer from *distributional skew*, reducing recovery accuracy due to imbalanced red/green list allocations; synchronization-free techniques demonstrate *phase fragility*, collapsing robustness under deletion attacks (Liu et al. 2024; Pang et al. 2024); and

*Corresponding author.

error-agnostic frameworks incur *bit error propagation*, limiting effective payload capacity (Qu et al. 2024). These deficiencies collectively undermine attribution reliability where precision traceability is paramount.

To resolve this trilemma, we propose *ARGH-Mark*, an integrated framework that synergizes three core innovations: anchor-synchronized phase recovery maintaining detection integrity through periodic binary sequences resistant to insertion/deletion attacks; RG-balanced vocabulary modulation partitioning lexicons via hashing to minimize quality degradation; and Hamming-based error correction enabling single-bit error rectification through algebraic coding theory. The triad establishes a new Pareto frontier in watermark design space by fundamentally reconciling the competing objectives of robustness, fidelity and efficiency.

Comprehensive evaluation across text generation, question answering and text summarization demonstrates SOTA performance: ARGH-Mark achieves a high match rate under a certain proportion of attack, while it delivers a significantly higher bit accuracy, which is superior to that of cyclic shift (Fernandez et al. 2023) and CTWL (Wang et al. 2024). The framework operates with low average detection latency and demonstrates substantial acceleration over method based on ECC (Qu et al. 2024), while it maintains negligible quality degradation. Crucially, watermarked text preserves a high level of baseline utility, establishing unprecedented practical viability for high-stakes deployment. As shown in Figure 1, ARGH-Mark enables regulatory-ready deployment across high-stakes applications including misinformation containment, and copyright enforcement.

This work advances AI alignment through three key contributions:

- We organically integrate green/red list and 0/1-bit, simplify the embedding and extraction steps, and uniformly distribute the watermark into each token.
- We proposed Anchor-synchronized phase recovery for maintaining detection integrity under insertion/deletion attacks.
- We adopt Hamming-based error correction to enable single-bit error rectification.

Related Work

We categorize existing multi-bit watermarking methods for LLM-generated text into three paradigms: message enumeration-based, bit assignment-based, and block-based, analyzing their principles, strengths, and limitations to highlight gaps addressed by ARGH-Mark.

Message Enumeration-based Methods

These methods embed multi-bit messages via specific signals and extract via exhaustive candidate enumeration.

CyclicShift (Fernandez et al. 2023): Uses cryptographic hashing and cyclic bit shifting to generate message-dependent green lists. During extraction, it enumerates all message candidates to find the one maximizing green token counts. While accurate for short messages, it has exponential complexity $O(2^B)$ (B = message bits), leading to prohibitive latency for 32-bit payloads.

CTWL (Wang et al. 2024): Formulates embedding as optimization, balancing text quality and message compatibility via a bias parameter. It uses a proxy LLM for balanced vocabulary partitioning but inherits exponential enumeration complexity, making 32-bit messages infeasible.

ECC (Qu et al. 2024): Integrates Reed-Solomon codes to encode message segments, pseudo-randomly assigning tokens to segments. Extraction enumerates segment candidates and decodes via ECC. It achieves strong robustness (e.g., 97.6% match rate for 20-bit/200-token texts) but incurs $O(k \cdot 2^{b/k})$ complexity (k = segments, b = bits) and relies on balanced token allocation.

Common limitations: Poor scalability with message length (exponential latency) and vulnerability to localized edits disrupting green list distribution.

Bit Assignment-based Methods

These methods distribute messages via pseudo-random bit position assignment, enabling independent embedding/extraction.

MPAC (Yoo, Ahn, and Kwak 2024): Decomposes messages into sub-units, allocating tokens to positions via previous-token hashing. It partitions vocabulary into "colorlists," biasing the list matching the allocated position's content. MPAC embeds 32-bit messages without extra latency but suffers from imbalanced token allocation (frequent tokens dominate specific positions) and degraded accuracy for long messages. It also faces a trade-off between multi-bit capacity and zero-bit detection robustness.

Block-based Methods

These partition text into blocks (entropy chunks, fixed segments) and embed bits into blocks to enhance resilience.

AEB (Cohen, Hoover, and Schoenbach 2025): Formalizes robustness via the AEB condition (detection if enough original blocks are retained). It combines L-bit schemes (from zero-bit via black-box reduction) with fingerprinting codes for multi-user tracing (O(log n) for non-colluders). However, it requires O(L) blocks (L = bits, = security parameter) for reliable extraction, limiting short-text applicability, and struggles with fine-grained perturbations (synonym substitution).

Limitations of Existing Work

Existing methods face a trilemma: balancing robustness, text quality, and efficiency. Enumeration methods lack scalability; bit assignment methods have imbalanced coverage; block methods need long texts. Few address insertion/deletion synchronization or lightweight error correction—gaps resolved by ARGH-Mark's integrated design.

Methodology

Design Philosophy: The proposed framework solves three challenges in LLM watermarking: (i) *maintaining detection integrity under insertion/deletion attacks* via anchor-synchronized phase recovery, (ii) *preserving generation*

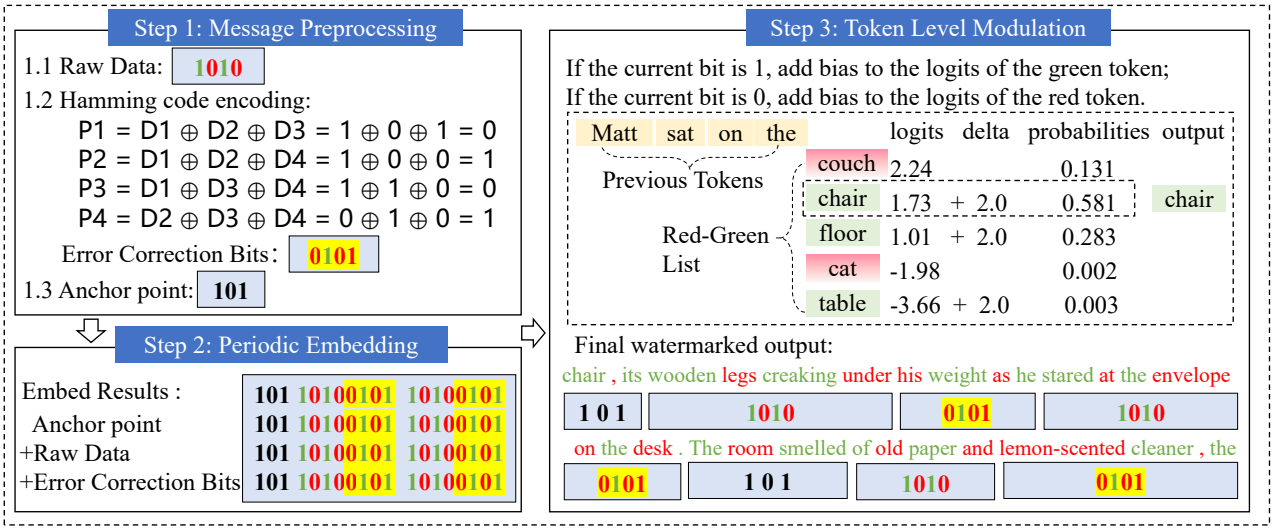

Figure 2: The watermark generation process is composed of three sequential stages: message preprocessing, periodic embedding, and token-level modulation.

quality through the use of red and green tokens (representing 0 and 1 bits respectively), (iii) *enabling single-bit error rectification* via Hamming-based error correction technique. This triadic optimization enables deployment in latency-sensitive production environments while resisting adversarial perturbations.

Watermark Generation Framework

As shown in the Figure 2, to address the trilemma of robustness, quality preservation, and detection efficiency, three synergistic mechanisms are proposed. First, the *Red-Green Dynamic Balancing* mechanism partitions the vocabulary into complementary subsets via hash-based RNG, enforcing strictly symmetric modulation. This equal-probability design preserves the original token distribution centroid, minimizing perplexity degradation during generation. Second, *Anchor Injection* inserts synchronization codes at deterministic intervals, creating fixed computational waypoints to enhance the recovery rate during detection. Third, *Hamming Redundancy Embedding* augments the payload with algebraic error-correcting bits to furnish error correction capabilities and enhance robustness against adversarial attacks.

Step 1: Message Preprocessing. The watermark generation process initiates with message preprocessing. Let $M = 1010$ denote the original message. This message is encoded using extended Hamming codes (n, k, d), where n denotes the total number of bits after Hamming encoding, k denotes the number of bits of original message, and d denotes the minimum Hamming distance. For a (8,4,4) code, parity bits are computed as:

$$\begin{aligned} P_1 &= D_1 \oplus D_2 \oplus D_3 \\ P_2 &= D_1 \oplus D_2 \oplus D_4 \\ P_3 &= D_1 \oplus D_3 \oplus D_4 \\ P_4 &= D_2 \oplus D_3 \oplus D_4 \end{aligned}, \tag{1}$$

yielding codeword

$$C = (D_1, D_2, D_3, D_4, P_1, P_2, P_3, P_4) \in \{0,1\}^n, \tag{2}$$

Synchronization anchors $A \in \{0,1\}^m$ (e.g., $A = 101$ for $m = 3$) are subsequently prepended to each instance of C.

Step 2: Periodic Embedding. Periodic embedding is then performed across the token sequence. Let T represent the message period where $T = |C|$. For every N message cycles (i.e., $N \cdot T$ tokens), m anchor bits are inserted at intervals τ. The combined sequence S is generated as:

$$S = \underbrace{A_1, \ldots, A_m}_{\text{anchor}}, \underbrace{C_1, \ldots, C_T}_{\text{message}}, \ldots, \underbrace{C_{(N-1)T+1}, \ldots, C_{NT}}_{\text{message}}. \tag{3}$$

Step 3: Token Level Modulation. Token-level modulation operates on each generation step. Given context window h and prompt $\mathbf{x}_{-P}^{-1} = (x_{-P}, \ldots, x_{-1})$, the language model produces logits $\ell = \text{LLM}(\mathbf{x}_{-P}^{-1})$. A context-dependent red-green list $\mathcal{R}$ is generated via:

$$\text{seed} = \text{Hash}(x_{-h}, \ldots, x_{-1}), \quad \mathcal{R} = \text{RNG}_{\text{seed}}(\mathcal{V}), \tag{4}$$

where $\mathcal{V}$ denotes the vocabulary and $|\mathcal{R}| = \frac{1}{2}|\mathcal{V}|$. For bit b_t in sequence S, modulated logits ℓ' are computed as:

$$\ell'_i = \begin{cases} \ell_i + \delta & \text{if } (b_t = 1 \wedge v_i \in \mathcal{R}) \vee (b_t = 0 \wedge v_i \notin \mathcal{R}) \\ \ell_i & \text{otherwise} \end{cases}, \tag{5}$$

Tokens are sampled from $P(w) = \text{softmax}(\ell')$.

Watermark Detection Protocol

As shown in the Figure 3, the detection pipeline optimizes the accuracy-efficiency tradeoff through algorithmic innovations. *Anchor-Guided Bucketization* first locates synchronization points via sliding-window correlation, reducing the

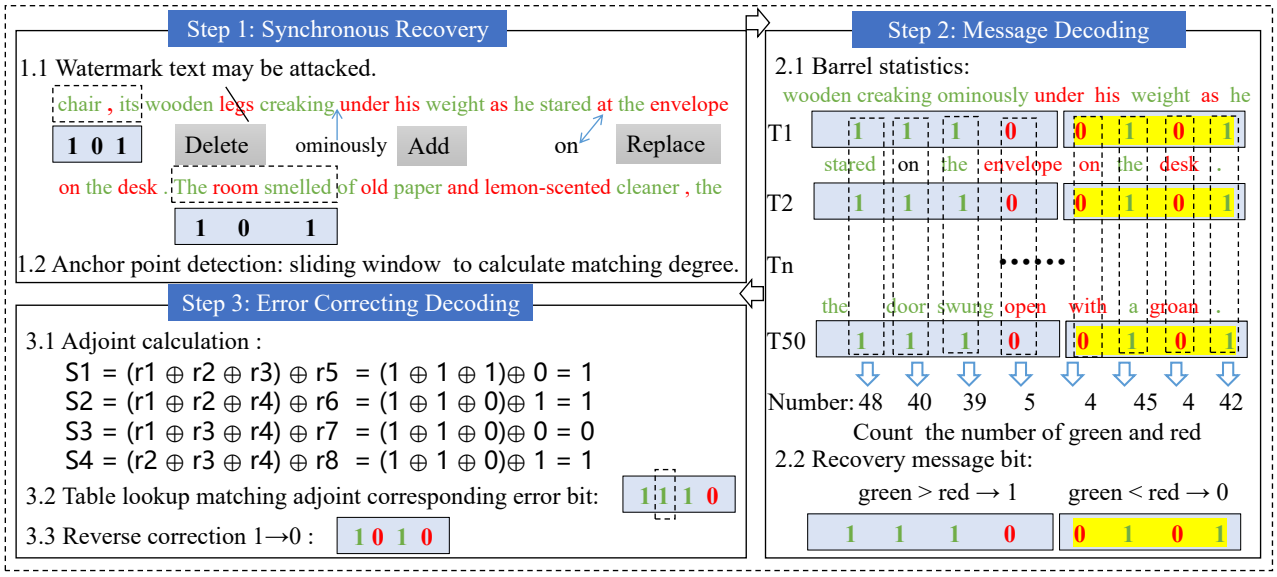

Figure 3: The watermark detection protocol is implemented through three sequential components: synchronous recovery, message decoding, and error correction decoding.

search space. A pair of anchors is deemed valid only if the interval between them matches the length (i.e., $N \cdot T$ tokens) of the message period. Recovered valid anchor pairs segment the text into parallel processing buckets. The *Majority-Voting Decoder* in each bucket converts fragile per-token decisions into robust bit estimations through statistical aggregation, achieving noise suppression via the law of large numbers. Messages between invalid anchor pairs are directly discarded and excluded from bucket-based statistics. Finally, *Syndrome-Accelerated Correction* exploits Hamming code linearity: syndrome computation requires only XOR operations, and error localization uses precomputed lookup tables. This layered architecture ensures that detection is completed in a short time, making it feasible for real-time deployment.

Step 1: Synchronous Recovery. Synchronization recovery first identifies anchor positions in potentially corrupted text $\mathbf{x}'$. For m-bit anchor A, a sliding window computes match scores:

$$\text{Score}(j) = \sum_{k=0}^{m-1} \mathbb{I}\left(x'_{j+k} \in \mathcal{G}_{j+k}^{(A_k)}\right), \tag{6}$$

where $\mathcal{G}_t^{(b)}$ denotes green lists for bit b at position t. Candidate positions $\{j : \text{Score}(j) \geq m-1\}$ are retained.

To prevent potential conflicts between anchor sequences and embedded message bits, a two-stage validation mechanism is employed. In the first stage, high-quality anchor candidates are identified. A sliding window approach is utilized to compute match scores, with positions achieving a higher match rate being retained as candidate anchor points. The second stage involves validating anchor pairs by verifying their spatial relationships. Anchor points are processed in pairs, where the interval between two consecutive anchors must approximate the expected message length within a tolerance margin. This dual-validation strategy ensures robust synchronization recovery while maintaining clear separation between anchor sequences and payload data, thereby mitigating interference during the detection process.

Step 2: Message Decoding. Message decoding partitions tokens into $T = |C|$ buckets $\{\mathcal{T}_i\}_{i=1}^{T}$. Each bucket's bit $\hat{b}_i$ is decoded:

$$\hat{b}_i = \begin{cases} 1 & \text{if } \sum_{x_t \in \mathcal{T}_i} \mathbb{I}(x_t \in \mathcal{G}_t^{(1)}) > \sum_{x_t \in \mathcal{T}_i} \mathbb{I}(x_t \in \mathcal{R}_t^{(1)}) \\ 0 & \text{if } \sum_{x_t \in \mathcal{T}_i} \mathbb{I}(x_t \in \mathcal{G}_t^{(1)}) < \sum_{x_t \in \mathcal{T}_i} \mathbb{I}(x_t \in \mathcal{R}_t^{(1)}) \end{cases}, \tag{7}$$

yielding estimated codeword $\hat{C} = (\hat{b}_1, \ldots, \hat{b}_T)$.

Step 3: Error Correcting Decoding. Error correction employs syndrome decoding. For (8,4,4) Hamming codes, syndromes $\mathbf{s} = (s_1, s_2, s_3, s_4)$ are computed as:

$$\begin{aligned} s_1 &= \hat{r}_1 \oplus \hat{r}_2 \oplus \hat{r}_3 \oplus \hat{r}_5 \\ s_2 &= \hat{r}_1 \oplus \hat{r}_2 \oplus \hat{r}_4 \oplus \hat{r}_6 \\ s_3 &= \hat{r}_1 \oplus \hat{r}_3 \oplus \hat{r}_4 \oplus \hat{r}_7 \\ s_4 &= \hat{r}_2 \oplus \hat{r}_3 \oplus \hat{r}_4 \oplus \hat{r}_8 \end{aligned}, \tag{8}$$

where $\hat{r}_i$ denotes received bits. The error pattern $\mathbf{e}$ is determined via syndrome mapping $\mathcal{M} : \{0,1\}^4 \to \{0,1\}^T$ such that $\hat{M} = \hat{C} \oplus \mathcal{M}(\mathbf{s})$ yields the corrected message.

Experimental Setup

Datasets and Models. Experimental validation was conducted on three distinct task categories: text generation, question answering, and text summarization. For the text generation task, the C4 dataset (Raffel et al. 2020) was employed with OPT-1.3B (Zhang et al. 2022) and Guanaco-7B-HF (Dettmers et al. 2023) models. The question answering evaluation utilized the ELI5 dataset (Fan et al. 2019) in conjunction with the Llama-3-8B-Instruct (meta 2024)

			8bits		16bits		32bits	
	Datasets/Models	**Methods**	**Match Rate↑**	**Bit Acc↑**	**Match Rate↑**	**Bit Acc↑**	**Match Rate↑**	**Bit Acc↑**
Tokens=384 Samples=100	C4/ OPT-1.3B	CyclicShift	97.00	99.00	97.00	96.00	99.00	99.60
		ECC	99.00	99.88	99.00	99.60	95.00	98.47
		CTWL	83.00	94.68	94.00	98.60	-	-
		ARGH-Mark(Our)	**99.90**	**99.90**	**99.90**	**99.90**	**99.90**	**99.90**
	ELI5/ Llama-3-8B	CyclicShift	99.00	99.00	97.00	97.00	97.00	99.30
		ECC	99.50	99.50	92.00	96.75	93.00	98.28
		CTWL	96.00	96.55	91.00	93.35	-	-
		ARGH-Mark(Our)	**99.90**	**99.90**	**99.90**	**99.90**	**99.90**	**99.90**
Tokens=200 Samples=200	C4/ Guanaco-7B-HF	CyclicShift	98.50	99.25	95.50	98.04	**96.10**	99.23
		ECC	98.50	99.75	98.50	99.60	90.50	97.90
		CTWL	98.00	99.61	**99.90**	99.34	-	-
		ARGH-Mark(Our)	**99.50**	**99.90**	**99.90**	**99.90**	96.00	**99.80**
	CNN-DailyMail/ Llama-3-8B	CyclicShift	99.00	99.30	95.00	97.90	**93.30**	98.57
		ECC	98.50	99.25	97.00	99.00	91.00	98.28
		CTWL	98.50	99.42	**99.90**	93.91	-	-
		ARGH-Mark(Our)	**99.90**	**99.90**	96.50	**99.60**	90.00	**99.50**

Table 1: Comparative evaluation of match rate and bit accuracy between the proposed ARGH-Mark framework and baseline methods across diverse text generation tasks and message lengths.

model. Additionally, text summarization performance was assessed using the CNN/DailyMail dataset (Hermann et al. 2015) with the Llama-3-8B-Instruct (meta 2024) model.

Baselines. Comparative analysis was conducted against three SOTA baselines: CyclicShift (Fernandez et al. 2023), ECC (Qu et al. 2024) and CTWL (Wang et al. 2024).

Evaluation Metrics. We mainly use matching rate and bit recovery accuracy to measure the proportion of generated text that can accurately extract embedded watermark information. We use decoding time to measure efficiency. Text quality degradation was evaluated through perplexity, computed using the GPT-2-XL model. Additionally, we employed three types of adversarial attacks—addition, deletion, and replacement-for the evaluation of robustness.

Implementation Details. For message embedding and anchor generation, biases of $\delta = 5.0$ and 3δ are used, respectively. A larger bias for anchor generation ensures more robust synchronization by strengthening the positional signals critical for message localization during detection. The extended Hamming code increases the minimum distance to 4 through parity check bits, which can not only correct one error, but also detect the existence of two errors. Three variants of extended Hamming codes—specifically $(8, 4, 4)$, $(16, 11, 4)$, and $(32, 26, 4)$—are tested to evaluate their error correction capabilities. All experiments are conducted under a temperature setting of $T = 1.0$ to maintain consistency in text generation. All experiments are conducted on a computing platform equipped with 2×NVIDIA A800 GPUs, and all reported results are derived from the average scores of three independent runs to ensure statistical reliability.

Experimental Results

Correctness Evaluation. The experimental results presented in Table 1 demonstrate that the proposed ARGH-Mark framework achieves consistently high bit recovery accuracy across diverse model-dataset configurations and bit-lengths, with match rates and bit accuracy exceeding 99.9% in most settings. Quantitative analysis reveals a notable dependency on token length L: for $L = 384$, the match rate remains uniformly at 99.9% across 8, 16, and 32-bit messages, whereas for $L = 200$, the match rate for 32-bit messages declines to 96.0% under certain conditions. This performance divergence stems from the enhanced reliability of anchor synchronization and statistical aggregation in longer sequences. Formally, the majority-voting decoder used in message decoding aggregates per-token decisions within each bucket $\mathcal{B}_i$, where the estimated bit $\hat{b}_i$ is determined by:

$$\hat{b}_i = \mathbb{I}\left(\frac{1}{K_i}\sum_{x_t \in \mathcal{B}_i} \mathbb{I}(x_t \in \mathcal{G}_t^{(1)}) > \theta\right), \tag{9}$$

with K_i denoting the number of tokens in bucket i. The variance of the bucket-wise statistic is bounded by $\text{Var}(\hat{p}_i) \leq \frac{1}{4K_i}$, which decreases as K_i grows with L. Consequently, longer texts provide more robust statistical estimates, particularly for high-capacity payloads (e.g., 32-bit), where the increased number of bits exacerbates susceptibility to local perturbations. These findings quantitatively validate the framework's capacity to maintain detection integrity across varying sequence lengths, underscoring its practical applicability in real-world attribution scenarios.

Robustness Analysis. The adversarial robustness of ARGH-Mark is systematically evaluated against three perturbation categories—token deletion, addition, and substitution—with quantitative results detailed in Figure 4. Under

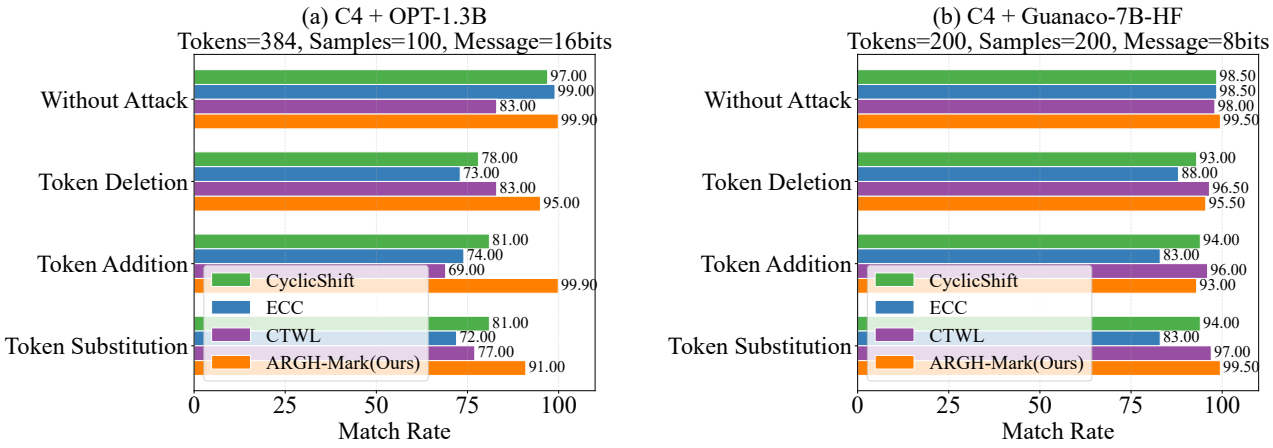

Figure 4: The robustness of the proposed framework is evaluated under three distinct adversarial attacks: token addition, deletion, and replacement, with performance compared against baseline methods.

token deletion attacks, ARGH-Mark maintains match rates of 95.0% for C4+OPT-1.3B and 95.5% for C4+Guanaco-7B-HF, significantly outperforming CyclicShift (78.0% and 93.0%) and ECC (73.0% and 88.0%). For token addition attacks, ARGH-Mark achieves near-perfect match rates of 99.9% and 93.0% for C4+OPT-1.3B, compared to CyclicShift (81.0%), ECC (74.0%), and CTWL (69.0%). This resilience is analytically attributed to the anchor synchronization mechanism, which mitigates positional drift through match score computation:

$$\text{Score}(j) = \sum_{k=0}^{m-1} \mathbb{I}\left(x'_{j+k} \in \mathcal{G}_{j+k}^{(A_k)}\right), \tag{10}$$

where j denotes the sliding window position, m is the anchor length, x' represents the adversarially modified text, and $\mathcal{G}_{j+k}^{(A_k)}$ indicates the green list corresponding to anchor bit A_k at offset $j+k$. Candidate positions satisfying $\text{Score}(j) \geq m-1$ are retained for phase recovery, enabling robust detection despite token removals. Under token substitution attacks, ARGH-Mark exhibits minimal degradation, sustaining match rates of 91.0% and 99.5% across the two configurations, while baselines suffer declines to 72.0%–81.0% and 83.0%–97.0%, respectively. The integrated synergy of anchor synchronization and algebraic error correction establishes a new Pareto frontier in adversarial robustness, reducing vulnerability to content manipulations compared to SOTA alternatives.

Text Quality. The perplexity distributions for non-watermarked and watermarked texts are compared in Figure 5, where a lower perplexity value indicates higher textual quality. Although the perplexity of watermark text has slightly increased, these values are still within an acceptable practical range. This preservation of text quality is attributed to the use of red/green tokens to represent the 0/1 bits, respectively, a design choice that minimizes distortion to the original token distribution during the generation process.

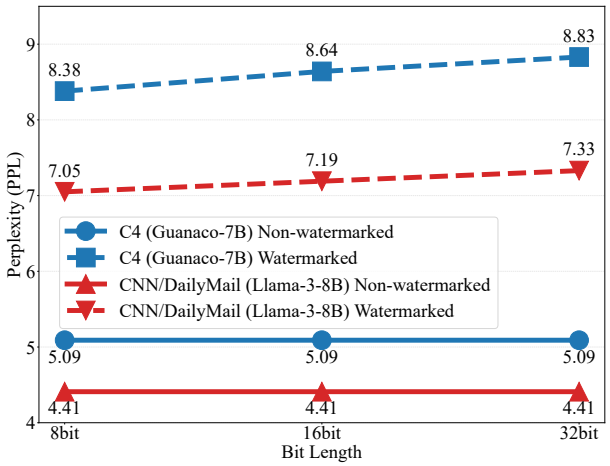

Figure 5: Text quality is assessed by comparing the perplexity distributions of watermarked and non-watermarked outputs from various models.

Decoding Time. The decoding efficiency of ARGH-Mark is rigorously evaluated against SOTA baselines, with comprehensive results presented in Figure 6. Quantitative analysis reveals that the proposed framework achieves consistently low decoding latencies between 0.06 and 0.15 seconds across diverse experimental configurations, demonstrating remarkable temporal stability independent of message length and model architecture. Specifically, ARGH-Mark maintains decoding times of 0.07 seconds for C4+Guanaco-7B-HF, and 0.06 seconds for CNN/DailyMail+Llama-3-8B across 8-bit to 32-bit payloads. This performance represents a substantial improvement over competing methods: ECC-based approaches exhibit significantly higher latencies ranging from 1.33 to 2.78 seconds, while CTWL fails to complete execution for 32-bit messages due to computational intractability. The superior efficiency is analytically attributed to the anchor-synchronized recovery mechanism, which re-

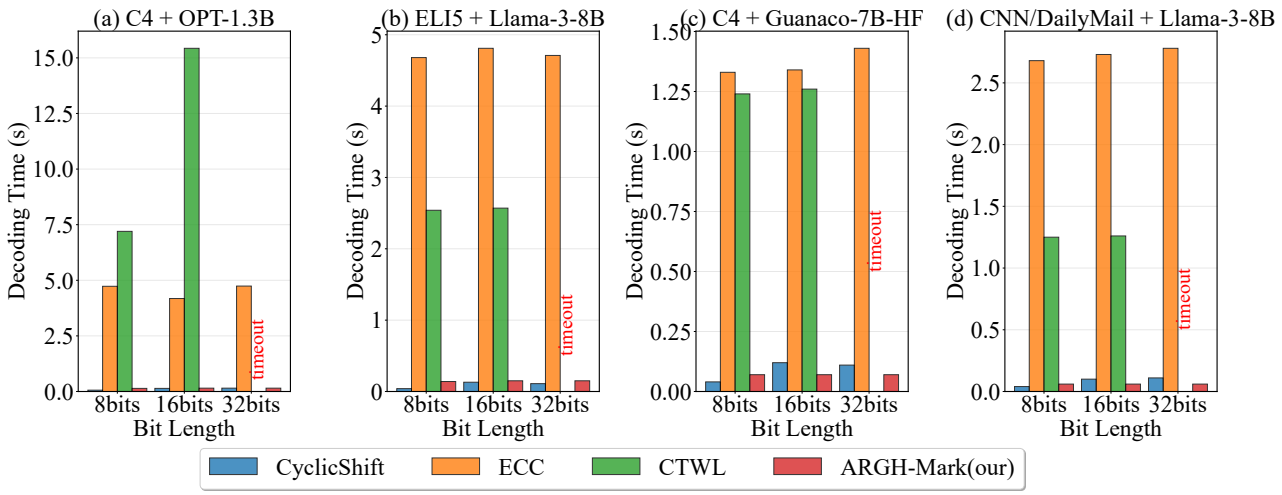

Figure 6: Substantial efficiency advantages of the proposed framework are demonstrated by comparing its decoding time with existing baseline methods.

duces detection complexity from the exponential $O(2^B)$ of enumeration-based methods to linear time. Formally, for text length L, anchor length m, and bucket count B, the detection time is bounded by:

$$T(L) = O(L \cdot m) + O(B) + O(1), \tag{11}$$

where L denotes the number of tokens, m represents the anchor length, and B is the number of decoding buckets. Although CyclicShift exhibits marginally reduced latency, this efficiency is accompanied by a substantial degradation in detection accuracy, with accuracy dropping notably for longer message payloads in contrast to the consistently high match rate maintained by ARGH-Mark. This trade-off underscores the framework's optimal balance between computational efficiency and attribution reliability, fulfilling critical requirements for real-time deployment in production environments where both latency and accuracy constraints are paramount.

Ablation Study. An ablation study quantifies each component's contribution through systematic removal from ARGH-Mark. As shown in Table 2, anchor removal causes a 4.50% match rate drop, highlighting its synchronization role. Eliminating RG-balance increases perplexity by 2.50, confirming its quality-preservation effect. Disabling Hamming correction reduces robustness by 3.50, demonstrating its error-resilience function. These quantified degradations establish the synergistic necessity of all three components.

Limitations and Future Directions. Despite achieving SOTA performance, ARGH-Mark exhibits three key limitations, with corresponding future improvements planned to address them. First, it incurs moderately elevated computational overhead during watermark embedding; future work will reduce this latency via precomputation of Hamming codes using lookup tables, streamlining the modulation of token logits. Second, its robustness degrades for payloads exceeding 32 bits; we will enhance resilience for longer bit sequences by integrating advanced error-correcting codes

Variant	Match Rate	Robustness	Perplexity
Full	0.00	0.00	0.00
-Anchor	-4.50	-1.50	+0.00
-RG-Balance	-0.50	-1.00	+2.50
-Hamming	-1.50	-3.50	+0.00

Table 2: Ablation study results are reported to quantify the contribution of each core component. Performance deviations are measured relative to the full model on the C4 dataset with a token length of 200.

(e.g., BCH codes) that offer stronger burst error correction. Third, the effectiveness of watermark detection fundamentally relies on successful identification of valid anchor pairs, and our current anchor design—though functional—remains susceptible to adversarial manipulations; future iterations will enforce stricter constraints to ensure anchor sequences are distinct from message bits, or explore more sophisticated anchor patterns to boost synchronization recovery under challenging conditions.

Conclusion

This paper presents *ARGH-Mark*, a robust watermarking framework integrating three synergistic innovations: anchor-synchronized phase recovery, RG-balanced vocabulary modulation, and Hamming-based error correction. Extensive evaluations across text generation, question answering and text summarization demonstrate SOTA performance. The framework establishes a new Pareto frontier in the watermarking trilemma: simultaneously optimizing for *robustness* against adversarial perturbations, *quality preservation* of generated content, and *linear-time detection* efficiency. By providing reliable attribution while maintaining semantic integrity, ARGH-Mark advances trustworthy deployment of large language models in alignment-critical applications.

Acknowledgments

We thank all the anonymous reviewers for their constructive feedback. This research is supported by Beijing Municipal Science & Technology Commission: New Generation of Information and Communication Technology Innovation - Research and Demonstration Application of Key Technologies for Privacy Protection of Massive Data for Large Model Training and Application (Z231100005923047).

References

Bender, E. M.; Gebru, T.; McMillan-Major, A.; and Shmitchell, S. 2021. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, 610–623. New York, NY, USA: Association for Computing Machinery. ISBN 9781450383097.

Chen, C.; and Shu, K. 2024. Can LLM-Generated Misinformation Be Detected? arXiv:2309.13788.

Cohen, A.; Hoover, A.; and Schoenbach, G. 2025. Watermarking Language Models for Many Adaptive Users. In *2025 IEEE Symposium on Security and Privacy (SP)*, 2583–2601.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.

Dettmers, T.; Pagnoni, A.; Holtzman, A.; and Zettlemoyer, L. 2023. QLoRA: Efficient Finetuning of Quantized LLMs. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems*, volume 36, 10088–10115. Curran Associates, Inc.

Fan, A.; Jernite, Y.; Perez, E.; Grangier, D.; Weston, J.; and Auli, M. 2019. ELI5: Long Form Question Answering. In Korhonen, A.; Traum, D.; and Màrquez, L., eds., *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 3558–3567. Florence, Italy: Association for Computational Linguistics.

Fernandez, P.; Chaffin, A.; Tit, K.; Chappelier, V.; and Furon, T. 2023. Three Bricks to Consolidate Watermarks for Large Language Models. In *2023 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–6.

Hermann, K. M.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; and Blunsom, P. 2015. Teaching Machines to Read and Comprehend. arXiv:1506.03340.

Liu, Z.; Cong, T.; He, X.; and Li, Q. 2024. On Evaluating The Performance of Watermarked Machine-Generated Texts Under Adversarial Attacks. arXiv:2407.04794.

Megías, D.; Kuribayashi, M.; Rosales, A.; and Mazurczyk, W. 2021. DISSIMILAR: Towards Fake News Detection Using Information Hiding. In *Signal Processing and Machine Learning. In The 16th International Conference on Availability, Reliability and Security (Vienna, Austria)(ARES 2021). Association for Computing Machinery, New York, NY, USA, Article*, volume 66.

meta. 2024. The Llama 3 Herd of Models. arXiv:2407.21783.

Mirsky, Y.; Demontis, A.; Kotak, J.; Shankar, R.; Gelei, D.; Yang, L.; Zhang, X.; Pintor, M.; Lee, W.; Elovici, Y.; and Biggio, B. 2023. The Threat of Offensive AI to Organizations. *Computers Security*, 124: 103006.

OpenAI. 2023. GPT-4 Technical Report.

Pang, Q.; Hu, S.; Zheng, W.; and Smith, V. 2024. Attacking llm watermarks by exploiting their strengths. In *ICLR 2024 Workshop on Secure and Trustworthy Large Language Models*.

Perez, E.; Huang, S.; Song, F.; Cai, T.; Ring, R.; Aslanides, J.; Glaese, A.; McAleese, N.; and Irving, G. 2022. Red Teaming Language Models with Language Models. In Goldberg, Y.; Kozareva, Z.; and Zhang, Y., eds., *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 3419–3448. Abu Dhabi, United Arab Emirates: Association for Computational Linguistics.

Qu, W.; Yin, D.; He, Z.; Zou, W.; Tao, T.; Jia, J.; and Zhang, J. 2024. Provably Robust Multi-bit Watermarking for AI-generated Text via Error Correction Code. *CoRR*, abs/2401.16820.

Qwen3. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Rillig, M. C.; Ågerstrand, M.; Bi, M.; Gould, K. A.; and Sauerland, U. 2023. Risks and Benefits of Large Language Models for the Environment. *Environmental Science & Technology*, 57(9): 3464–3466. PMID: 36821477.

Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; and Gal, Y. 2024. AI models collapse when trained on recursively generated data. *Nature*, 631: 755–759.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.-A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Wang, L.; Yang, W.; Chen, D.; Zhou, H.; Lin, Y.; Meng, F.; Zhou, J.; and Sun, X. 2024. Towards Codable Watermarking for Injecting Multi-Bits Information to LLMs. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yoo, K.; Ahn, W.; and Kwak, N. 2024. Advancing Beyond Identification: Multi-bit Watermark for Large Language Models. In Duh, K.; Gomez, H.; and Bethard, S., eds., *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 4031–4055. Mexico City, Mexico: Association for Computational Linguistics.

Zhang, S.; Roller, S.; Goyal, N.; Artetxe, M.; Chen, M.; Chen, S.; Dewan, C.; Diab, M. T.; Li, X.; Lin, X. V.; Mihaylov, T.; Ott, M.; Shleifer, S.; Shuster, K.; Simig, D.; Koura, P. S.; Sridhar, A.; Wang, T.; and Zettlemoyer, L. 2022. OPT: Open Pre-trained Transformer Language Models. *CoRR*, abs/2205.01068.