From TDMA to CDMA: A Multi-bit Watermark for Diffusion Language Models

Anonymous ACL submission

Abstract

While DLMs have emerged as an alternative to ARMs, robust content provenance mechanisms for this architecture remain unexplored. Existing multi-bit watermarking schemes, heavily reliant on the sequential generation of ARMs, cannot be directly applied to DLMs. In this paper, we reframe the multi-bit watermarking problem through a novel Digital Signal Processing (DSP) lens. We draw an analogy between prior works and TDMA (Time Division Multiple Access), revealing their inherent limitations. To overcome these limitations, we introduce **CDMArk**, the first multi-bit watermarking framework tailored for DLMs, orchestrating a paradigm shift from TDMA to CDMA (Code Division Multiple Access). CDMArk encodes the entire watermark message across all token positions holographically. We further provide rigorous statistical guarantees for its detection process. Extensive experiments demonstrate that CDMArk achieves a new state-of-the-art Pareto frontier between imperceptibility and effectiveness.

1 Introduction

The rapid rise of Large Language Models (LLMs) has revolutionized content creation, but it also raises serious concerns about copyright and the spread of fake information. To address these challenges, multi-bit watermarking has emerged as a mechanism for establishing content provenance and authenticity (Fernandez et al., 2023; Yoo et al., 2024; Jiang et al., 2025). Unlike zero-bit watermarking which functions solely as a binary verifier, multi-bit watermarking offers significantly greater utility by encoding rich messages, such as user IDs and model versions.

Recently, Diffusion Language Models (DLMs) have emerged as a compelling alternative to Autoregressive Language Models (ARMs), offering parallel generation capabilities and superior controllability via infilling and error correction (Nie et al., 2025; Ye et al., 2025; Zhu et al., 2025a) However, despite the rapid ascent of DLMs, existing multi-bit watermarking schemes are tailored for ARMs. This misalignment highlights a critical gap in safeguarding the provenance of content generated by this emerging class of models.

Directly applying existing ARM-specific multi-bit watermarking schemes to DLMs presents a fundamental architectural mismatch. Prevailing multi-bit methods typically adopt a divide-and-conquer paradigm (Yoo et al., 2024; Qu et al., 2024; Jiang et al., 2025), decomposing the m-bit watermark message into m independent chunks for watermarking. A critical component of this paradigm is the *positional allocation mapping* (Qu et al., 2024), which stochastically assigns token positions to specific message chunks using previous tokens as invariant seeds. This dependency relies strictly on the left-to-right generation process of ARMs. In contrast, DLMs operate via a non-causal, parallel denoising mechanism that lacks fixed context tokens for such seeding, rendering these methods nearly infeasible.

To fundamentally analyze the multi-bit watermarking problem, we introduce a perspective from Digital Signal Processing (DSP). We conceptualize the text generation process as a *Signal Transmission* task, where the multi-bit message is the target *Payload* and tokens act as *Carrier Signals* modulating the payload. Through this lens, existing multi-bit watermarking methods are analogous to **Time Division Multiple Access (TDMA)** schemes (Rappaport and Safari, 2001), where the communication channel (i.e. the watermarked text sequence) is segmented into distinct time slots (i.e. token positions), each exclusively responsible for transmitting the payload of a specific user (i.e. a message chunk).

Drawing inspiration from the evolution of modern telecommunications, we propose a paradigm shift from TDMA to **Code Division Multiple Access (CDMA)** (Rappaport and Safari, 2001) in the

field of multi-bit watermarking. Just as CDMA revolutionized signal communication by allowing multiple users to share the same communication channel, we propose a novel multi-bit watermarking scheme, **CDMArk**, which tries to **simultaneously encode every bit of the watermark message in each token position**.

In standard CDMA system, each user is assigned a unique, predefined vector known as *Spreading Code*, and the target composite signal to emit is derived as the linear combination of all spreading codes weighted by their corresponding user messages.

Our proposed method mirrors the procedure by treating each bit of the watermark as a virtual "user", each transmitting a single bit payload. We additionally assign each token in the vocabulary with a predefined *Signal Vector* and aim to select those tokens whose signal vectors better approximate the target composite signal. Consequently, the watermarking process is formulated as a constrained convex optimization problem, balancing alignment with the target signal against divergence from the original data distribution, which can be solved efficiently.

Our contributions are summarized as follows[1]:

- We introduce a novel Digital Signal Processing (DSP) perspective to multi-bit watermarking, rigorously formalizing existing methods as TDMA schemes and revealing their architectural limitations.
- We propose the **first** multi-bit watermarking scheme tailored for Diffusion Models, **CDMA**rk (pronounced as "CD-mark"). By shifting the paradigm from TDMA to CDMA, encodes the watermark holographically across all token positions, naturally aligning with the parallel nature of diffusion processes.
- Extensive experiments demonstrate that our method significantly outperforms other baselines, establishing a state-of-the-art Pareto frontier between Imperceptibility and Effectiveness.

2 Background and Preliminaries

2.1 Diffusion Language Models

We first introduce Diffusion Language Models (DLMs), focusing on the absorbing diffusion variant (Austin et al., 2023; Lou et al., 2024; Ou et al., 2025) which serves as the backbone of our method. Unlike auto-regressive models that generate tokens sequentially from left to right, DLMs model the joint distribution of the entire sequence through a forward diffusion process and a learnable reverse denoising process.

Forward Process and Training. The forward process $q(\boldsymbol{x}^t|\boldsymbol{x}^0)$ is defined as a corruption mechanism that gradually erases information from the clean data $\boldsymbol{x}^0 = [x_0^0, x_1^0, ...]$. From time $t \in [0, 1]$, tokens in the sequence are independently replaced by a mask token `[M]` according to a log-linear noise schedule, becomeing pure noise (i.e., fully masked) in time $t = 1$. To reverse this, a neural network $p_\theta(\boldsymbol{x}^0|\boldsymbol{x}^t)$ is trained to predict the original unmasked tokens given the corrupted context $\boldsymbol{x}^t$. The training objective is to minimizing the cross-entropy between the predicted distribution and the ground truth tokens at masked positions (Ou et al., 2025).

Inference via Tweedie Sampling. During the inference stage, the model generates text by iteratively reversing the corruption process, starting from a fully masked sequence $\boldsymbol{x}^T$. The Tweedie τ-leaping strategy (Lou et al., 2024) is commonly adopted for efficiency. Specifically, for a transition from timestep t to s (where $0 \le s < t \le 1$), the marginal distribution $P_{s|t}(\boldsymbol{x}^s|\boldsymbol{x}^t)$ factorizes over each token position i:

$$P_{s|t}(\boldsymbol{x}^s|\boldsymbol{x}^t) = \prod_i p_{s|t}(x_i^s|\boldsymbol{x}^t) \tag{1}$$

$$p_{s|t}(x_i^s|\boldsymbol{x}^t) = \begin{cases} \mathbb{I}(x_i^s = x_i^t), & \text{if } x_i^t \neq \texttt{[M]} \\ \frac{s}{t}, & \text{if } x_i^t = \texttt{[M]} \wedge x_i^s = \texttt{[M]} \\ \frac{t-s}{t}\, p_\theta(x_i^s|\boldsymbol{x}^t), & \text{if } x_i^t = \texttt{[M]} \wedge x_i^s \neq \texttt{[M]} \end{cases} \tag{2}$$

Inference in Practice. Practically, the reverse process is discretized into several diffusion timesteps $\{1, ..., T\}$. At each timestep t, the parametric model works as a mask predictor, takes in a masked token sequence $\boldsymbol{x}^t$ from the last timestep and predicts masked tokens of all positions i from $p_\theta(x_i^{t-1}|\boldsymbol{x}^t)$. Subsequently, $\frac{s}{t}$ of the predicted tokens will be remasked by a specific remasking strategy (i.e. random or semi-autoregressive remasking). The process is iteratively conducted for T steps until all tokens are unmasked.

[1]We will release the code to facilitate future research.

2.2 Watermarking in Auto-Regressive Models

Watermarking language models typically involves injecting a hidden pattern during generation process, which is typically achieved by biasing the generation distribution $p_\theta(x_i|\boldsymbol{x}_{<i})$ towards a specific watermarked distribution. The injected pattern can subsequently be detected via statistical hypothesis testing. A robust watermarking scheme must satisfy three primary objectives: (1) *Effectiveness*, which ensures the pattern can be detected with high statistical confidence; (2) *Imperceptibility*, which requires the watermarked text to remain indistinguishable from natural text from a statistical perspective (Christ et al., 2023; Hu et al., 2023); and (3) *Robustness*, which demands resilience against attacks such as paraphrasing or editing.

Zero-bit Watermarking. Zero-bit watermarking aims to *Detect* whether a text is generated by a specific model (Yoo et al., 2023; Christ et al., 2023; Lee et al., 2023; Kuditipudi et al., 2023; Zhao et al., 2023; Hu et al., 2023; Wu et al., 2023; Hou et al., 2023). The canonical paradigm for zero-bit watermarking is KGW (Kirchenbauer et al., 2023a,b). At each timestep i, KGW utilizes the context tokens $\boldsymbol{x}_{<i}$ as a seed to stochastically partition the vocabulary $\mathcal{V}$ into a "Green List" and a "Red List". The probability of green-list tokens is then raised. Consequently, the proportion of green-list tokens in the generated text becomes significantly higher than expected by chance, creating a statistical difference from natural text that enables effective detection.

Multi-bit Watermarking. Multi-bit watermarking extends the zero-bit framework to not only detect the existence of a watermark but also to *Decode* a specific m-bit message out of a watermarked text sequence. Prevailing methods typically employ a reductionist strategy, decomposing this task into multiple zero-bit watermarking sub-problems. Fernandez et al. (2023) formalize the problem into 2^m distinct zero-bit problems, traversing all 2^m possible message value for detection. The computational complexity of this method is exponential, rendering it prohibitive for larger message length. Conversely, more practical methods like MPAC (Yoo et al., 2024), Qu et al. (2025), and Stealthink (Jiang et al., 2025) adopt a divide-and-conquer approach These methods partition the m-bit message into several independent message chunks, assigning specific tokens to encode each chunk. Crucially, this assignment is governed by a pseudo-random *Positional Allocation Mapping* that is seeded by the context tokens $\boldsymbol{x}_{<i}$. Thus, the multi-bit problem is effectively reduced to a series of interleaved zero-bit problems: detecting single bits within partial subsequences, which are then solved via standard methods like KGW or DiPmark (Wu et al., 2023).

3 A Signal Processing Perspective on Watermarking

We propose to analyze the problem of multi-bit watermarking through the lens of Digital Signal Processing (DSP). In this section, we formalize watermarking text generation as a signal transmission task, a perspective that rigorously uncovers the inherent architectural limitations of existing schemes.

3.1 Multi-bit Watermarking as Signal Transmission

We model the generative process of language models as a discrete-time communication *Channel*. Within this framework, the injection of a multi-bit watermark is analogous to transmitting an m-bit *Payload* through this channel. Specifically, we establish the following correspondences:

- Each token position i represents a discrete *Time Slot*.
- Each generated token x_i functions as a chosen *Carrier Signal*.
- The model's vocabulary $\mathcal{V}$ serves as the predefined *Signal Constellation*.

Consequently, the standard sampling process of the language model corresponds to *Signal Modulation* that selecting a specific signal vector from the signal constellation. Crucially, to preserve generation quality, modern watermarking schemes avoid rudimentary "hard" selection (e.g., the Hard Green-Red List variant of KGW (Kirchenbauer et al., 2023a) which exclusively samples Green tokens). Instead, they employ probabilistic distribution shaping, softly biasing the sampling distribution in favor of tokens that encode the target payload (e.g. the Green tokens).

3.2 Limitations of Current Schemes in a TDMA View

Leveraging this DSP framework, we can re-evaluate the prevailing multi-bit watermarking schemes for auto-regressive models (Yoo et al., 2024; Qu et al., 2024; Jiang et al., 2025). As detailed in Section 2.2, these methods rely on a

stochastic mapping that assigns each token position $i \in \{1, \ldots, N\}$ to a specific message block.

This mechanism is structurally equivalent to the **Time Division Multiple Access (TDMA)** scheme (Rappaport and Safari, 2001). In DSP, the defining characteristic of TDMA is mutually exclusive access: at any given time slot i, the channel is exclusively dedicated to transmitting payloads from one user, rendering the channel unavailable to all others.

Existing multi-bit watermarking methods exhibit a similar characteristic that each token only encodes a specific message chunk. We illustrate this behavior using MPAC as a representative case in Figure 1 (left). While it offers conceptual simplicity, it imposes fundamental architectural bottlenecks when transplanted to DLMs.

Synchronization Overhead in DLMs. In standard TDMA protocols, a synchronization mechanism is necessary to ensure the receiver correctly maps each time slot to corresponding user. Analogously, the TDMA-based watermarking schemes also necessitate such "synchronization" to correctly associate each token x_i with its corresponding message block during detection. In ARMs, the strictly left-to-right generation process ($x_1 \to x_2 \to \ldots$) provides implicit synchronization at virtually zero cost, as the positional allocation mapping can be deterministically derived from the invariant context $\boldsymbol{x}_{<i}$. However, DLMs operate in a non-causal manner, lacking a fixed generation order to derive context tokens. Consequently, enforcing a TDMA structure in this setting incurs prohibitive synchronization overhead: the mapping is no longer "free" metadata but becomes part of the payload that needs to be explicitly transmitted, consuming valuable channel capacity.

Positional Allocation Imbalance. Even assuming perfect synchronization, TDMA-based schemes suffer from inherent positional allocation imbalance. Due to the non-uniform nature of language modeling distribution, the reliance on pseudo-randomized function to generate positional allocation mapping often yields a highly non-uniform allocation, where specific message chunks are stochastically favored while others are neglected. Crucially, the rigid mutual exclusivity of TDMA prevents those "starving" message chunks from borrowing “time slots" from others. Empirical studies (Yoo et al., 2024; Qu et al., 2025) show that as the message length m increases, this imbalance becomes more severe. The overall accuracy is bounded by the message chunk with the poorest allocation, regardless of how well others are transmitted.

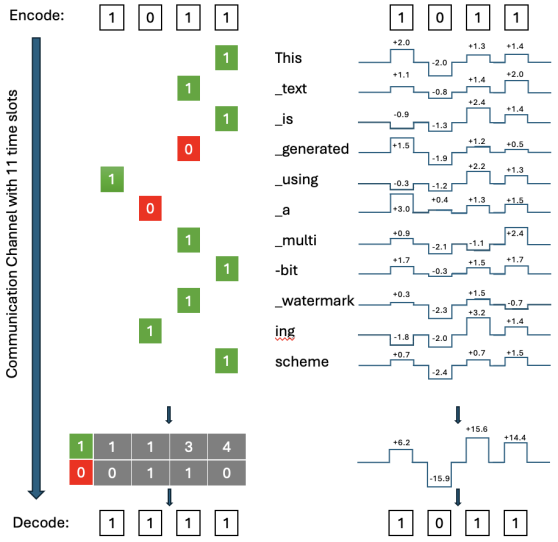

Figure 1: The different between paradigms of MPAC and our proposed **CDMArk** when encoding a watermarking message `1011` of length $m = 4$. *(Left)* MPAC adopts a TDMA scheme, where each token position is responsible to transmit only a specific message chunk (commonly with length 1). Due to the allocation imbalance, different chunks are assigned uneven number of tokens (e.g. the first bit is only assigned one token). *(Right)* **CDMArk** adopts a CDMA scheme, where each token corresponds to a 4-dimension "composite signal" vector, carrying the information of the entire message (we replace 0 as -1 as in common signal transmission practice). The mechanism makes the watermark more robustness to incorrect injection (e.g. "`_is`" transmits a false signal in the first bit) or attacks.

4 CDMArk: A CDMA-Based Watermarking Framework

In this section, we formally introduce **CDMArk**, a novel multi-bit DLM watermarking schemes designed to overcome the structural limitations of TDMA-based methods discussed in Section 3.2. Drawing inspiration from modern telecommunications, CDMArk orchestrates a paradigm shift by adopting Code Division Multiple Access (CDMA). We first introduce background of this mechanism and then derive an optimal injection strategy by formulating the watermarking process as a constrained convex optimization problem.

4.1 Background of CDMA Mechanism

To introduce CDMArk, we first revisit the core mechanism of classical CDMA. Unlike TDMA, which segments time, CDMA enables multiple users to transmit payloads simultaneously over the same communication channel. Consider a CDMA system with m users, each transmitting a 1-bit payload:

- *Code Space*: The system defines an m-dimensional space spanned by a set of m mutually orthogonal basis vectors, termed *Spreading Codes*. Each user is assigned one unique spreading codes.
- *Transmission*: Each user encodes their payload by multiplying it with its spreading codes. These distinct signals are then summed to create a unified composite signal.
- *Recovery*: Since the composite signal is essentially a linear combination of these spreading codes, it uniquely represents a point in the code space. Consequently, the receiver can isolate and recover any user's payload by simply projecting the received signal onto the corresponding spreading code.

4.2 CDMA in Multi-bit Watermarking

We transpose the paradigm of CDMA to the multi-bit watermarking of DLMs. Let $\mathbf{b} \in \{-1, +1\}^m$ denote the m-bit watermarking message. We treat each message bit as a user. Analogous to the process described in Section 4.1, we also construct a m-dimensional code space $\mathbb{R}^m$, with the j-th message bit assigned with the j-th standard basis vector $\mathbf{e}_j$ as its spreading code. Therefore, the composite signal $\mathbf{s} = \sum_j b_j \mathbf{e}_j$ can be derived as the linear combination of messages from different users.

A critical distinction exists in the generation process between classical communications and our scenario. In classical communications where the communication channel is continuous, the transmitter could simply emit the exact composite signal. However, a language model is constrained to sample from a discrete vocabulary $\mathcal{V}$, which functions as a fixed *Signal Constellation*. Each token $w \in \mathcal{V}$ corresponds to a predefined, immutable signal vector $\mathbf{v}_w \in \mathbb{R}^m$.

Therefore, we have to apply "quantization" in this process. At each time step, we aim to select a token w whose corresponding signal $\mathbf{v}_w$ provides good approximation of the target signal $\mathbf{s}$. We quantify this approximation using *Alignment Score* defined as the projection of the token's vector onto the target signal $u_w = \mathbf{v}_w^T \mathbf{s}$.

4.3 Optimal Watermark Injection Strategy

We now introduce our proposed multi-bit watermarking scheme **CDMArk**. As formulated in Section 4.2, the watermarking objective is to bias the generation distribution towards tokens whose signal vectors $\mathbf{v}_w$ exhibit high alignment scores with the target signal $\mathbf{s}$. Simultaneously, to ensure text quality, we must maintain imperceptibility, which is modeled as a constraint on the KL divergence from the original distribution.

Consider a DLM generating a sequence of length N. At each timestep t, the model takes in a text sequence $\boldsymbol{x}^t = [x_1^t, x_2^t, \ldots, x_N^t]$ and predicts the distribution $P_{t-1|t}(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t)$ of the next diffusion step. Standard DLMs assume a factorized distribution $P_{t-1|t}(\boldsymbol{x}^{t-1}|\boldsymbol{x}^t) = \prod_{i=1}^N p(x_i|\boldsymbol{x}^t)$ for the reverse transition following Eq. 2, enabling parallel sampling of different token position i.

Leveraging the factorized nature of the DLM, we formulate the watermarking injection as an independent optimization problem for each token position i. For simplicity, we omit the index i and timestep t, denoting the original distribution as $p(\cdot) = p(x_i|\boldsymbol{x}^t)$.

Primal Problem.

$$\begin{aligned} \underset{q}{\text{maximize}} \quad & \mathbb{E}_{w\sim q(\cdot)}\left[\mathbf{v}_w^T \mathbf{s}\right] \\ \text{subject to} \quad & D_{\text{KL}}(q\|p) \le \epsilon \\ & \sum_{w\in\mathcal{V}} q(w) = 1 \end{aligned} \tag{3}$$

Here, $q(\cdot)$ is the target watermarked distribution to be optimized and ϵ is the hyperparameter controlling the degree of allowed semantic deviation from the original distribution at each token position.

The optimization problem is strictly convex with linear constraints and a strictly feasible point (i.e., $q = p$), satisfying Slater's condition (Boyd and Vandenberghe, 2004). Thus, the Karush-Kuhn-Tucker (KKT) conditions are both necessary and sufficient for optimality. We therefore solve this problem using the method of Lagrange multipliers.

Proposition 4.1. *Given the primal problem in Eq. 3, the optimal watermarked distribution $q^*(\cdot)$ is given by the exponential tilting of the original distribution $p(\cdot)$. The optimal probability of sampling*

token $w \in \mathcal{V}$ is:

$$q^*(w) \propto p(w) \exp\left(\frac{\mathbf{v}_w^T \mathbf{s}}{\lambda}\right) \tag{4}$$

where $\lambda > 0$ is the Lagrange multiplier satisfying $D_{KL}(q^*||p) = \epsilon$.

In practice, rather than tuning the implicit constraint ϵ, we adopt λ-parametrization, directly setting $\delta = 1/\lambda$ as a hyper-parameter controlling the watermark strength. Remarkably, the exponential tilting format of optimal solution corresponds to adding a bias term to the logits, which is surprisingly similar to the watermarking injection procedure of KGW.

4.4 Statistical Detection and Message Decoding

Multi-bit Message Decoding Process The decoding process of a watermarked text sequence $\boldsymbol{x}$ also follows the demodulation of CDMA. We aggregate the signal vectors from all token positions $\sum_{x_i \in \boldsymbol{x}} \mathbf{v}_{x_i}$ and project the accumulated signal onto different spreading codes $\hat{b}_j = \text{sign}((\sum_{x_i \in \boldsymbol{x}} \mathbf{v}_{x_i})^T \mathbf{e}_j)$ as the decoded message. The overall process is presented in Figure 1 (right).

Zero-bit Binary Detection Process Multi-bit watermark is also required to detect whether a text is watermarked. To provide a rigorous theoretical guarantee for watermark detectability and to control the False Positive Rate (FPR) without empirical threshold tuning, we formulate the detection process as a statistical hypothesis testing problem (Kirchenbauer et al., 2023a), where the null hypothesis $\mathcal{H}_0$ is that the candidate text sequence is not drawn from the watermarked distribution.

For better statistical properties in our testing, we employ a specific construction strategy for the pre-defined signal sets $\{\mathbf{v}_w\}_{w \in \mathcal{V}}$. We define $\mathbf{V} = \text{concat}(\{\mathbf{v}_w^T\}_{w \in \mathcal{V}}) \in \mathbb{R}^{|\mathcal{V}| \times m}$ as follows:

1. **Initialization:** Sample a random matrix $\mathbf{G} \sim \mathcal{N}(0,1)^{|\mathcal{V}| \times m}$.
2. **Centering:** Subtract the column means to ensure zero-mean for each dimension: $\mathbf{G}' = \mathbf{G} - \bar{\mathbf{G}}$.
3. **Orthogonalization:** Perform QR decomposition $\mathbf{G}' = \mathbf{QR}$, where $\mathbf{Q} \in \mathbb{R}^{|\mathcal{V}| \times m}$ has orthonormal unit-length columns.
4. **Scaling:** Set $\mathbf{V} = \sqrt{|\mathcal{V}|} \cdot \mathbf{Q}$.

Due to the isotropy of normal distribution, each column vector of the constructed random matrix $\mathbf{V}$ is indeed uniformly distributed on the intersection of the sphere of radius $\sqrt{|\mathcal{V}|}$ and the zero-sum hyperplane (Mezzadri, 2007). This procedure guarantees critical properties for the subsequent statistical analysis: (1) *Zero Mean and Unit Sample Variance:* for all message bit position j, $\sum_{w \in \mathcal{V}} \mathbf{v}_w^T \mathbf{e}_j = 0$, $\frac{1}{|\mathcal{V}|} \sum_{w \in \mathcal{V}} (\mathbf{v}_w^T \mathbf{e}_j)^2 = 1$ and (2) *Column Orthogonality:* $\frac{1}{|\mathcal{V}|} \mathbf{V}^\top \mathbf{V} = \mathbf{I}$, which ensures that the signal from different dimensions are uncorrelated and therefore less interference.

We define the detection statistic as the magnitude of the accumulated signal vector over the sequence: $z = ||\sum_{i=1}^{N} \mathbf{v}_{x_i}||^2$. We then have the following proposition:

Proposition 4.2. *Under the null hypothesis H_0, as the length of sequence $N \to \infty$, the normalized detection statistic z/N converges to a chi-squared distribution with m degrees of freedom:*

$$\frac{z}{N} \sim \chi^2(m) \tag{5}$$

This proposition provides a theoretical guarantee for our watermark detection. We can now apply one-sided chi-square testing, enabling accurate P-value calculation and controlling Type-I error. We present both the watermark injection and detection algorithms in Appendix A.

5 Experiments

5.1 Experimental Settings

We conduct our experiments using LLaDA-MoE-Instruct (Zhu et al., 2025b). Following WaterBench (Tu et al., 2024) and Gloaguen et al. (2025), we employ two prompt datasets: ELI5 (Fan et al., 2019) and FinanceQA (Mateega et al., 2025). Unless otherwise specified, we generate sequences with a maximum length of $N = 512$ using a low-confidence semi-autoregressive remasking strategy.

We include several watermarking methods: (1) **CyclicShift** (Fernandez et al., 2023). To adapt to DLMs, we implement a global-hashed variant similar to Unigram (Zhao et al., 2023), where the Green/Red list partition is fixed globally. (2) **MPAC** (Yoo et al., 2024) is a representative TDMA-based multi-bit watermarking scheme. It involves two hyperparameters. Radix r implies the length of each message chunk, while δ controls the watermark strength. (3) **CDMArk** (Ours) involves a single hyperparameter δ that controls the watermark strength.

Table 1: Comparison between different multi-bit watermarking schemes. We only evaluate CyclicShift under message length $m = 6$, as its computational complexity scales exponentially (2^m).

	No Attack				Substitution (frac=0.1)		Substitution (frac=0.2)		Dipper (lex=20,order=20)	
Method	AUC-ROC	BitAcc	PPL	Oracle-PPL	AUC-ROC	BitAcc	AUC-ROC	BitAcc	AUC-ROC	BitAcc
No Watermark	–	–	3.00	2.83	–	–	–	–	–	–
Message Length=6										
CyclicShift(γ =0.25, δ =1.5)	66.86	69.71	**3.08**	**2.94**	47.53	70.83	32.69	69.04	58.42	60.58
CyclicShift(γ =0.25, δ =2)	79.69	84.21	3.09	2.94	64.10	85.38	49.50	84.46	66.49	68.08
CyclicShift(γ =0.5, δ =1.5)	66.77	73.79	3.18	3.05	51.23	71.33	35.70	67.38	58.88	61.00
CyclicShift(γ =0.5, δ =2)	77.85	81.79	3.21	3.09	70.26	77.12	64.91	75.25	66.20	64.33
MPAC(δ =1, r =1)	64.69	74.50	3.10	2.94	56.63	72.83	50.93	70.71	57.36	63.38
MPAC(δ =1, r =2)	62.32	71.04	3.12	2.96	53.65	70.42	46.63	66.54	55.58	60.46
MPAC(δ =2, r =1)	88.72	89.25	3.33	3.21	80.72	86.25	73.01	83.88	69.19	75.25
MPAC(δ =2, r =2)	86.35	90.71	3.31	3.20	78.54	87.46	69.93	84.29	67.83	72.75
CDMArk(δ =1.25)	**94.49**	**92.96**	3.12	2.99	**92.46**	**93.33**	**90.33**	**92.33**	**73.61**	**84.21**
Message Length=12										
MPAC(δ =1, r =1)	58.81	67.88	3.14	2.98	48.70	65.96	45.52	64.10	54.16	60.38
MPAC(δ =1, r =2)	54.72	63.81	3.12	2.95	42.65	62.67	37.36	61.19	50.31	56.31
MPAC(δ =2, r =1)	81.28	80.67	3.36	3.25	71.32	77.92	63.60	75.44	60.51	68.81
MPAC(δ =2, r =2)	78.98	79.48	3.39	3.25	68.91	77.77	59.13	73.58	57.32	65.27
CDMArk(δ =1.25)	**89.47**	**83.48**	**3.05**	**2.90**	**90.28**	**82.04**	**88.34**	**81.08**	**74.80**	**71.96**
Message Length=18										
MPAC(δ =1, r =1)	60.75	64.89	3.05	2.91	49.56	63.65	42.89	61.60	56.12	57.79
MPAC(δ =1, r =2)	55.04	62.26	3.10	2.94	44.27	59.99	39.26	57.86	49.83	56.07
MPAC(δ =2, r =1)	75.94	76.11	3.36	3.25	64.98	73.26	56.66	70.07	57.36	65.06
MPAC(δ =2, r =2)	75.83	77.21	3.41	3.30	64.12	74.79	55.78	71.82	55.77	63.92
CDMArk(δ =1.25)	**90.50**	**80.14**	**3.03**	**2.89**	**87.80**	**79.17**	**85.78**	**77.94**	**71.47**	**69.56**
Message Length=24										
MPAC(δ =1, r =1)	58.99	63.43	3.11	2.94	48.32	62.05	42.66	59.57	49.74	56.92
MPAC(δ =1, r =2)	53.33	59.74	3.10	2.93	43.74	59.66	37.92	58.24	49.75	54.66
MPAC(δ =2, r =1)	72.99	72.77	3.36	3.23	60.96	70.31	53.95	67.47	55.43	62.73
MPAC(δ =2, r =2)	73.99	72.42	3.37	3.24	62.62	70.14	53.89	68.52	57.04	61.69
CDMArk(δ =1.25)	**86.56**	**74.75**	**3.02**	**2.87**	**86.02**	**74.03**	**85.02**	**72.71**	**68.20**	**66.30**
Message Length=30										
MPAC(δ =1, r =1)	56.44	61.23	**3.04**	**2.86**	43.12	59.85	37.77	58.99	50.40	56.49
MPAC(δ =1, r =2)	52.85	58.99	3.08	2.95	40.32	58.42	34.35	56.88	47.68	54.89
MPAC(δ =2, r =1)	71.02	71.12	3.34	3.22	56.84	68.58	48.33	66.73	50.19	61.62
MPAC(δ =2, r =2)	68.93	70.28	3.34	3.23	54.85	69.06	46.03	66.10	53.46	60.42
CDMArk(δ =1.25)	**89.55**	**74.47**	3.28	2.95	**89.62**	**73.88**	**87.56**	**72.58**	**77.29**	**66.39**

We consider various metrics. For multi-bit watermarking, we report *Bit Accuracy (BitAcc)* following Yoo et al. (2024). For zero-bit detection, we report the *AUC-ROC* score. We measure the imperceptibility of the watermarked text using *PPL*, computed by the foundation model itself, to quantify the divergence from the original distribution. We also include *Oracle-PPL*, computed by a stronger external ARM, Qwen2.5-32B-Instruct, to assess the absolute text quality. To evaluate robustness, we consider two forms of attacks: (1) *Substitution*, where tokens are randomly replaced with synonyms; and (2) *Dipper*, using the Dipper paraphraser (Krishna et al., 2023) to rewrite the text.

5.2 Results

Main results. We present the main comparison results in Table 1. Existing methods tailored for ARMs generally struggle in DLMs. Specifically, MPAC exhibits a sharp performance degradation as the message length increases. While it maintains decent accuracy at $m = 6$, its BitAcc drops significantly at $m = 30$ (from $\sim 90\%$ to $\sim 70\%$). This trend demonstrates our analysis of the *Positional Allocation Imbalance* in Section 3.2. In contrast, **CDMArk** consistently establishes a superior Pareto frontier across all settings, achieving the highest BitAcc and AUC-ROC while maintaining good imperceptibility. It is worth highlighting the stability of CDMArk's detection capability (i.e. AUC-ROC) across varying settings of message length from $m = 6$ to $m = 30$, with only a marginal decline from its peak.

The advantages of our holographic injection mechanism become even more pronounced under post-editing attacks. In **Word Substitution** scenarios, CDMArk demonstrates remarkable resilience with small performance drops even when 20% of tokens are corrupted. Under the challenging **Dipper** paraphrase attack, CDMArk still outperforms all baselines, who almost equate to random guessing with AUC-ROC falling into about 50%.

Generalization studies. We analyze the Pareto frontiers of watermarking schemes between effectiveness (AUC-ROC) and imperceptibility (PPL)

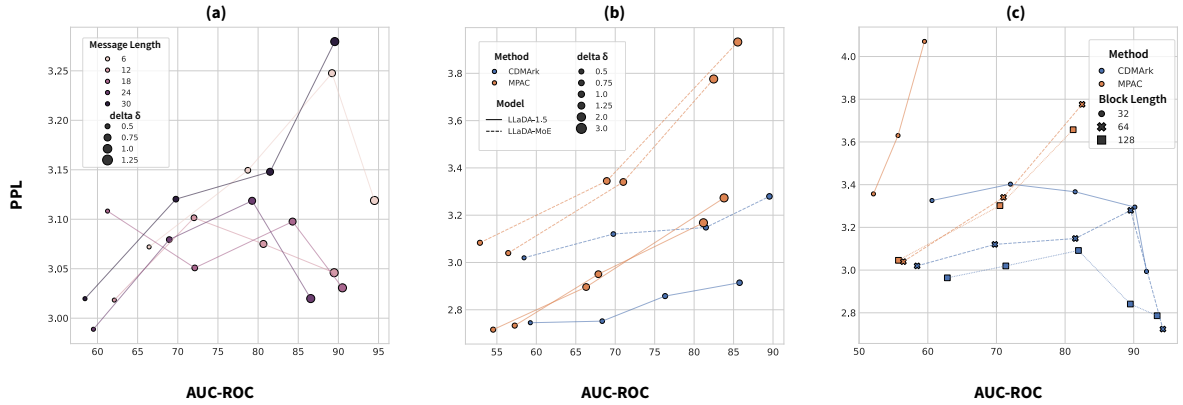

Figure 2: Pareto frontiers of different watermarking methods under different ablation settings. Note that a Pareto frontier closer to the bottom-right corner indicates a better trade-off *(a)* The performance of CDMArk under different message lengths and delta. *(b)* Comparison between CDMArk and MPAC (both $r = 1, 2$) with different DLMs backbones when encoding a message of length $m = 30$. *(c)* Comparison between CDMArk and MPAC (both $r = 1, 2$) under different remasking strategies when encoding a message of length $m = 30$.

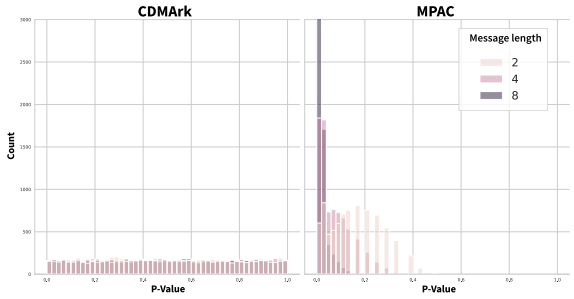

Figure 3: The P-value distribution of CDMArk and MPAC under null hypothesis.

under different settings by sweeping the watermark strength δ. Figure 2(a) illustrates the performance trade-offs of CDMArk under varying message lengths $m \in \{6, 12, \ldots, 30\}$. We observe that the Pareto curves for different message lengths are similar, indicating the scalibility of CDMArk, which contrasts sharply with other TDMA-based approaches. We further evaluate the generalization of CDMArk on different backbone models and decoding configurations. Figure 2(b) compares the performance on two distinct architectures: the MoE (LLaDA-MoE) and the dense model (LLaDA-1.5). In both cases, CDMArk consistently achieves the MPAC, demonstrating that our method is model-agnostic and effective across different architectural paradigms. Similarly, Figure 2(c) examines the impact of the semi-autoregressive block length, a critical hyperparameter in diffusion generation. Whether using a fine-grained or coarse-grained generation strategy, CDMArk maintains a significant performance margin over MPAC.

Empirical Validation of Proposition 4.2. We empirically verify the statistical behavior of our detector derived in Proposition 4.2. We randomly generated 8k samples under the null hypothesis and computed the p-values using both MPAC and CDMArk detectors. Figure 3 visualizes the resulting distributions across different message lengths. For CDMArk, the p-values align perfectly with the uniform distribution $U[0, 1]$ across all settings, empirically demonstrating Proposition 4.2, confirming that our detection score provides an unbiased statistical estimate with controllable Type-I errors. In contrast, MPAC exhibit a distribution skewing towards zero. This bias of detector exacerbates as the message length increases, providing an explanation for the rapid performance decay of MPAC observed in our main experiments.

6 Conclusion

In this paper, we identified the fundamental limitations of prevailing multi-bit watermarking schemes from a perspective of TDMA paradigm. We further introduced **CDMArk**, a novel framework that shifts the paradigm to Code Division Multiple Access (CDMA), which holographically encoding the entire watermarking message across all token positions. Empirical experiments and theoretical analysis demonstrate that CDMArk establishes a superior Pareto frontier in terms of effectiveness, imperceptibility, and robustness, particularly under high-capacity payloads. We hope this work inspires further exploration into signal-processing-inspired multi-bit watermarking schemes.

Limitations

While **CDMArk** is theoretically formulated as a generic watermarking framework applicable to both Auto-Regressive Models (ARMs) and Diffusion Language Models (DLMs), this work primarily focuses on the latter. Our experimental validation is exclusively conducted on DLMs to address the critical gap in multi-bit watermarking for non-causal generation, where existing TDMA-based methods fundamentally fail. Consequently, we did not benchmark CDMArk against state-of-the-art ARM watermarking schemes in their native setting. We leave the exploration of CDMArk's potential in ARMs, as well as its extension to other modalities such as image and video diffusion models, for future work.

7 Ethical considerations

This work utilizes Large Language Models (LLMs) to assist in writing refinement and code generation for result visualization. All models and datasets employed in this study are publicly available and widely used within the research community. There are no ethical concerns regarding data privacy or copyright infringement associated with the resources used in this research.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2023. Structured Denoising Diffusion Models in Discrete State-Spaces. *arXiv preprint*. ArXiv:2107.03006 [cs].

S.P. Boyd and L. Vandenberghe. 2004. *Convex Optimization*. Number 1 in Berichte über verteilte messysteme. Cambridge University Press.

Miranda Christ, Sam Gunn, and Or Zamir. 2023. Undetectable Watermarks for Language Models. *arXiv preprint*. ArXiv:2306.09194 [cs].

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. *Preprint*, arXiv:1907.09190.

Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. 2023. Three Bricks to Consolidate Watermarks for Large Language Models. *arXiv preprint*. ArXiv:2308.00113 [cs].

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. 2025. Watermarking Diffusion Language Models. *arXiv preprint*. ArXiv:2509.24368 [cs].

Abe Bohan Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. 2023. SemStamp: A Semantic Watermark with Paraphrastic Robustness for Text Generation. *arXiv preprint*. ArXiv:2310.03991 [cs].

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. 2023. Unbiased Watermark for Large Language Models. *arXiv preprint*. ArXiv:2310.10669 [cs] version: 2.

Ya Jiang, Chuxiong Wu, Massieh Kordi Boroujeny, Brian Mark, and Kai Zeng. 2025. StealthInk: A Multi-bit and Stealthy Watermark for Large Language Models. *arXiv preprint*. ArXiv:2506.05502 [cs].

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023a. A Watermark for Large Language Models. In *Proceedings of the 40th International Conference on Machine Learning*, pages 17061–17084. PMLR. ISSN: 2640-3498.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. 2023b. On the Reliability of Watermarks for Large Language Models. *arXiv preprint*. ArXiv:2306.04634 [cs].

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. *arXiv preprint*. ArXiv:2303.13408 [cs].

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. 2023. Robust Distortion-free Watermarks for Language Models. *arXiv preprint*. ArXiv:2307.15593 [cs].

Taehyun Lee, Seokhee Hong, Jaewoo Ahn, Ilgee Hong, Hwaran Lee, Sangdoo Yun, Jamin Shin, and Gunhee Kim. 2023. Who Wrote this Code? Watermarking for Code Generation. *arXiv preprint*. ArXiv:2305.15060 [cs].

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2024. Discrete Diffusion Modeling by Estimating the Ratios of the Data Distribution. *arXiv preprint*. ArXiv:2310.16834 [stat].

Spencer Mateega, Carlos Georgescu, and Danny Tang. 2025. Financeqa: A benchmark for evaluating financial analysis capabilities of large language models. *Preprint*, arXiv:2501.18062.

Francesco Mezzadri. 2007. How to generate random matrices from the classical compact groups. *Preprint*, arXiv:math-ph/0609050.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025. Large Language Diffusion Models. *arXiv preprint*. ArXiv:2502.09992 [cs].

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. Your Absorbing Discrete Diffusion Secretly Models the Conditional Distributions of Clean Data. *arXiv preprint*. ArXiv:2406.03736 [cs].

Wenjie Qu, Dong Yin, Zixin He, Wei Zou, Tianyang Tao, Jinyuan Jia, and Jiaheng Zhang. 2024. Provably Robust Multi-bit Watermarking for AI-generated Text via Error Correction Code. *arXiv preprint*. ArXiv:2401.16820 [cs].

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. *arXiv preprint*. ArXiv:2401.16820 [cs].

T. Rappaport and an O'Reilly Media Company Safari. 2001. *Wireless Communications Principles and Practice, Second Edition*. Prentice Hall.

Shangqing Tu, Yuliang Sun, Yushi Bai, Jifan Yu, Lei Hou, and Juanzi Li. 2024. Waterbench: Towards holistic evaluation of watermarks for large language models. *Preprint*, arXiv:2311.07138.

Yihan Wu, Zhengmian Hu, Hongyang Zhang, and Heng Huang. 2023. DiPmark: A Stealthy, Efficient and Resilient Watermark for Large Language Models. *arXiv preprint*. ArXiv:2310.07710 [cs].

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. Dream 7B: Diffusion Large Language Models. *arXiv preprint*. ArXiv:2508.15487 [cs].

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, Toronto, Canada. Association for Computational Linguistics.

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, Mexico City, Mexico. Association for Computational Linguistics.

Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. 2023. Provable Robust Watermarking for AI-Generated Text.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025a. LLaDA 1.5: Variance-Reduced Preference Optimization for Large Language Diffusion Models. *arXiv preprint*. ArXiv:2505.19223 [cs].

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, and 7 others. 2025b. Llada-moe: A sparse moe diffusion language model. *Preprint*, arXiv:2509.24389.

A Pseudo-Code of Algorithms

Here, we present the pseudo-code for both watermarking generation and detection algorithms in Algorithm 2 and 1.

Algorithm 1 Watermark Injection

Input: Pre-trained DLM p_θ; Message $\mathbf{b} \in \{-1,+1\}^m$; Predefined signal constellation $\{v_w\}_{w\in\mathcal{V}}$; Watermark strength λ; Total diffusion steps T; Total sequence length N.
Output: Watermarked sequence $\boldsymbol{x}^0$
Target signal $\mathbf{s} \leftarrow \sum_j b_j \mathbf{e}_j$
`// Start from a fully masked sequence.`
$\boldsymbol{x}^T \leftarrow$ [`[M]`, ..., `[M]`]
for $t \leftarrow T$ **to** 1 **do**
  for *each token position* i *where* $x_i^t =$ `[M]` **do**
    `// Get original distribution from DLM`
    $p_{\text{orig}}(\cdot) \leftarrow p_\theta(x_i^t | \boldsymbol{x}^t)$
    `// Derive watermarked distribution`
    foreach $w \in \mathcal{V}$ **do**
      $u_w \leftarrow \mathbf{v}_w^\top \mathbf{s}$
      $\tilde{q}(w) \leftarrow p_{\text{orig}}(x_i = w) \cdot \exp\left(\frac{u_w}{\lambda}\right)$
    end
    $q(w) \leftarrow \frac{\tilde{q}(w)}{\sum_{w'\in\mathcal{V}} \tilde{q}(w')}$
    `// Sample next token from watermarked distribution`
    Sample $x_i^{t-1} \sim q(\cdot)$
  end
  `// Apply re-masking strategy`
  $\boldsymbol{x}^{t-1} \leftarrow \text{ReMask}(\boldsymbol{x}^{t-1}, t)$
end
return $\boldsymbol{x}^0$

Algorithm 2 Statistical Detection and Message Decoding

Input: Candidate text sequence $\boldsymbol{x} = [x_1, \ldots, x_N]$; Predefined signal constellation $\{\mathbf{v}_w\}_{w\in\mathcal{V}}$; Message length m.

Output: Detected message $\hat{\mathbf{b}}$, P-value p_{val}

```
// Signal Accumulation
```
$\hat{\mathbf{s}} \leftarrow \sum_{i=1}^{N} \mathbf{v}_{x_i}$

```
// Multi-bit Message Decoding
```
$\hat{\mathbf{b}} \leftarrow \text{sign}(\hat{\mathbf{s}})$

```
// Zero-bit Statistical Detection
```
$z \leftarrow \|\hat{\mathbf{s}}\|^2$

$p_{\text{val}} \leftarrow 1 - F_{\chi^2(m)}(z/N)$ `// ` $F_{\chi^2(m)}$ `is the CDF of` $\chi^2(m)$

return $\hat{\mathbf{b}}, p_{val}$

B Proof of Proposition 4.1

Problem Restatement. Recall that due to the factorized nature of the Diffusion Language Model (DLM) distribution (Eq. 1), the global optimization problem decomposes into independent subproblems for each token position. For a specific position, let $p(w)$ denote the original probability of token $w \in \mathcal{V}$, and $q(w)$ denote the target watermarked probability. We aim to maximize the expected alignment score $u_w = \mathbf{v}_w^T \mathbf{s}$ subject to a KL divergence constraint.

The optimization problem is formulated as:

$$\begin{aligned} \underset{\mathbf{q}}{\text{maximize}} \quad & \sum_{w\in\mathcal{V}} q(w) u_w \\ \text{subject to} \quad & \sum_{w\in\mathcal{V}} q(w) \log \frac{q(w)}{p(w)} \le \epsilon \end{aligned} \tag{6}$$

Lagrangian Formulation. Since the objective function is linear (concave) and the feasible set defined by the KL divergence (convex) and linear constraints is convex, this is a convex optimization problem. Furthermore, there exists a feasible solution (e.g., $q = p$), satisfying Slater's condition. Thus, the Karush-Kuhn-Tucker (KKT) conditions are necessary and sufficient for optimality.

We define the Lagrangian $\mathcal{L}(q, \lambda)$ as:

$$\mathcal{L} = \sum_w q(w) u_w - \lambda \left(\sum_w q(w) \log \frac{q(w)}{p(w)} - \epsilon \right)$$

where $\lambda \ge 0$ is the Lagrange multiplier associated with the KL divergence constraint.

Derivation. To find the stationary point, we take the partial derivative of $\mathcal{L}$ with respect to $q(w)$ and set it to zero:

$$\frac{\partial \mathcal{L}}{\partial q(w)} = u_w - \lambda \left(\log q(w) - \log p(w) + 1 \right) = 0$$

We can easily derive the format of optimal q^*.

$$q^*(w) \propto p(w) \exp\left(\frac{u_w}{\lambda}\right) \tag{7}$$

C Proof of Proposition 4.2

Problem Setup and Assumptions Let $\mathcal{V}$ be the vocabulary of size $|\mathcal{V}|$. Let $\{\mathbf{v}_w\}_{w\in\mathcal{V}}$ denote predefined composite signal sets. We define the matrix $\mathbf{V} = \text{concat}(\{\mathbf{v}_w^T\}_{w\in\mathcal{V}}) \in \mathbb{R}^{|\mathcal{V}|\times m}$ constructed as described in Section 4.4. Each row vector $\sqsubseteq_w$ corresponds to a token $w \in \mathcal{V}$.

Consider a candidate text sequence $\mathbf{x} = (x_1, x_2, \ldots, x_N)$ of length N. We define the accumulated signal vector $\mathbf{z} \in \mathbb{R}^m$ and the detection statistic z:

$$\mathbf{z} = \sum_{i=1}^{N} \mathbf{v}_{x_i}, \quad z = \|\mathbf{z}\|_2^2 \tag{8}$$

As shown in Section 4.4, for each message bit position j, our constructed $\mathbf{V}$ exhibit the following statistical properties:

1. **Zero Mean:** $\sum_w \mathbf{v}_w^T \mathbf{e}_j = 0$.
2. **Identity Variance:** $\sum_w (\mathbf{v}_w^T \mathbf{e}_j)^2 = |\mathcal{V}|$.

Under the null hypothesis, the distribution of the candidate text sequence $p(\mathbf{x})$ is independent of $\mathbf{V}$. We assume that the token sequence exhibits sufficient diversity such that the appearance of each token w is nearly the same, i.e $E_{p\in\Delta(\mathcal{V})}[p(w)] = \frac{1}{|\mathcal{V}|}$. Therefore, we have:

$$\begin{aligned} &\forall j \in \{1, 2...m\}, \\ &E_p[E_{w\sim p(\cdot)}[\mathbf{v}_w^T \mathbf{e}_j]] = 0, \\ &E_p[E_{w\sim p(\cdot)}[(\mathbf{v}_w^T \mathbf{e}_j)^2]] = 1 \end{aligned}$$

Applying the multidimensional **Central Limit Theorem (CLT)**, as the sequence length $N \to \infty$, the distribution of the j-th element of $\mathbf{z}$ converges to a normal distribution:

$$\mathbf{z}_j = \sum_{i=1}^{N} \mathbf{v}_{x_i}^T \mathbf{e}_j \sim \mathcal{N}(0, N) \tag{9}$$

Since each dimension of $\mathbf{z}$ is independent, the magnitude of accumulated signal

$$\frac{z}{N} = \frac{1}{N} ||\mathbf{z}||_2^2 = \sum_{j=1}^{m} \left(\frac{\mathbf{z}_j}{\sqrt{N}}\right)^2 \sim \chi^2(m) \tag{10}$$

D Details about Experiments

Implementation Details. We conduct our experiments using LLaDA-MoE-Instruct, a state-of-the-art DLM utilizing a Mixture-of-Experts architecture. Following the evaluation protocol of WaterBench (Tu et al., 2024) and Gloaguen et al. (2025), we employ two distinct datasets to simulate diverse generation scenarios: **ELI5** (Fan et al., 2019) for open-domain long-form Question Answering, and **FinanceQA** (Mateega et al., 2025) for domain-specific knowledge generation. Unless otherwise specified, we generate sequences with a maximum length of $N = 512$ using $T = 512$ diffusion steps. To ensure high-quality generation, we adopt a low-confidence semi-autoregressive remasking strategy with a block size of 64, consistent with standard DLM inference practices.

Baselines. Since existing multi-bit watermarking schemes are predominantly designed for Auto-Regressive Models (ARMs), we adapt them to the DLM setting to ensure a fair comparison:

1. **CyclicShift**: Adapted from Fernandez et al. (2023), this method partitions the vocabulary based on a message-dependent shift. To accommodate the non-causal nature of DLMs, we implement a global-hashed variant similar to Unigram (Zhao et al., 2023), where the Green/Red list partition is fixed globally rather than being context-dependent. It involves two hyperparameters γ and δ, which control the green list ration and watermark strength correspondingly. We evaluate it with message length $m = 6$, as its computational complexity scales exponentially (2^m).
2. **MPAC**: A representative TDMA-based multi-bit scheme (Yoo et al., 2024). It divides the message into chunks and allocates them to tokens using a pseudo-random mapping. It involves two hyperparameters. Radix r implies the length of each message chunk, while δ controls the watermark strength.
3. **CDMArk** (Ours): Our proposed CDMA-based framework. It involves a single hyperparameter δ that controls the injection strength (watermark logits bias).

Evaluation Metrics. We evaluate the methods based on three primary objectives:

- **Effectiveness:** For multi-bit watermarking, we report **Bit Accuracy (BitAcc)**, defined as the percentage of correctly decoded bits under varying message lengths $m \in \{6, 12, 18, 24, 30\}$. For zero-bit detection, we report the **AUC-ROC** score to measure the separability between watermarked and non-watermarked texts.
- **Imperceptibility:** We measure the statistical quality of the watermarked text using **Perplexity (PPL)**. We report two variants: (1) *PPL*, computed by the LLaDA model itself, to quantify the divergence from the original model distribution (i.e., the watermarking cost); and (2) *Oracle-PPL*, computed by a stronger external ARM (**Qwen2.5-32B-Instruct**), to assess the absolute fluency and coherence of the generated text.
- **Robustness:** To evaluate resilience against attacks, we subject the watermarked text to two forms of distortion: (1) *Word Substitution*, where tokens are randomly replaced with synonyms; and (2) *Paraphrasing*, using the Dipper paraphraser (Krishna et al., 2023) to rewrite the text while preserving semantics.

E Additional Experimental Results

We present other experimental results in this section:

- Table 2 presents the performance of different watermarking schemes with LLaDA-1.5 as backbone. The results exhibit a similar conclusion as in Table 1.
- Table 3 and 4 present the latency during generation and detection process. Our proposed CDMArk barely include additional costs in comparison with the other two baselines.
- Figure 4 presents the Pass@1 performance of CDMArk on MATH500. Given that math reasoning is known to have low-entropy distribution, which is challenging to watermarking (Lee et al., 2023), CDMArk still achieves satisfactory performance.

Table 2: Comparison between different watermarking schemes with LLaDA-1.5 as backbone model.

Method	AUC-ROC	Bit Accuracy	Likelihood	PPL	AUC-ROC (Dipper)	Bit Accuracy (Dipper)
No Watermark	–	–	2.66	2.50	–	–
Message Length 6						
CyclicShift(gamma: 0.25, delta: 1.5)	62.97	63.96	2.77	2.63	59.07	55.58
CyclicShift(gamma: 0.25, delta: 2)	73.50	76.54	2.82	2.70	63.62	62.79
CyclicShift(gamma: 0.5, delta: 1.5)	64.25	66.88	2.77	2.61	63.48	57.00
CyclicShift(gamma: 0.5, delta: 2)	72.96	76.50	2.89	2.73	65.69	62.50
MPAC(delta: 1, radix: 1)	62.39	71.79	2.73	2.55	49.69	61.54
MPAC(delta: 1, radix: 2)	60.92	68.50	**2.69**	**2.52**	59.05	59.54
MPAC(delta: 2, radix: 1)	82.35	86.46	2.89	2.74	61.51	**73.46**
MPAC(delta: 2, radix: 2)	**84.64**	**88.33**	2.91	2.76	**66.92**	70.54
CDMArk(delta: 1.25)	82.62	84.33	2.90	2.75	63.75	71.12
Message Length 12						
MPAC(delta: 1, radix: 1)	61.45	66.25	**2.74**	2.58	52.46	60.12
MPAC(delta: 1, radix: 2)	58.59	62.21	2.75	**2.57**	52.02	57.50
MPAC(delta: 2, radix: 1)	77.28	78.23	2.88	2.70	58.24	66.40
MPAC(delta: 2, radix: 2)	76.03	76.98	2.93	2.77	56.48	64.92
CDMArk(delta: 1.25)	**82.37**	**79.42**	2.89	2.76	**67.82**	**69.54**
Message Length 18						
MPAC(delta: 1, radix: 1)	54.17	63.12	2.74	2.57	50.81	56.17
MPAC(delta: 1, radix: 2)	57.85	61.10	**2.67**	**2.49**	54.78	55.46
MPAC(delta: 2, radix: 1)	70.99	73.26	2.88	2.73	57.24	63.79
MPAC(delta: 2, radix: 2)	72.71	73.83	2.92	2.76	59.30	61.96
CDMArk(delta: 1.25)	**83.04**	**76.61**	2.86	2.73	**62.69**	**66.06**
Message Length 24						
MPAC(delta: 1, radix: 1)	55.81	61.35	2.74	2.55	47.33	55.60
MPAC(delta: 1, radix: 2)	54.59	58.95	**2.71**	**2.51**	54.76	55.68
MPAC(delta: 2, radix: 1)	70.70	70.30	2.89	2.74	51.00	60.81
MPAC(delta: 2, radix: 2)	70.08	71.01	2.91	2.75	**57.41**	60.84
CDMArk(delta: 1.25)	**82.84**	**73.78**	2.84	2.68	56.48	**67.25**
Message Length 30						
MPAC(delta: 1, radix: 1)	57.26	60.32	2.73	2.56	50.49	56.01
MPAC(delta: 1, radix: 2)	54.53	58.21	**2.72**	**2.53**	54.48	53.17
MPAC(delta: 2, radix: 1)	67.86	68.36	2.95	2.79	51.17	60.38
MPAC(delta: 2, radix: 2)	66.32	67.48	2.90	2.76	57.63	58.37
CDMArk(delta: 1.25)	**85.76**	**72.38**	2.91	2.76	**78.47**	**64.02**

Table 3: Comparison of **Generation Latency** (sec per sample). Lower values indicate higher efficiency.

Method	**m=6**	**m=12**	**m=18**	**m=24**	**m=30**
No Watermark			32.18		
CyclicShift	32.41	32.64	32.30	32.65	33.40
MPAC	61.53	61.81	63.56	65.02	66.98
CDMArk (Ours)	32.03	31.94	31.90	31.84	32.17

Table 4: Comparison of **Detection Latency** (sec per sample). Lower values indicate higher efficiency.

Method	**m=6**	**m=12**	**m=18**	**m=24**	**m=30**
CyclicShift	10.14	–	–	–	–
MPAC	0.069	0.055	0.062	0.054	0.057
CDMArk (Ours)	0.0010	0.0014	0.0012	0.0013	0.0011

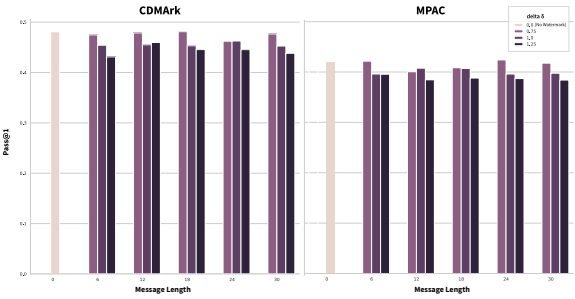

Figure 4: Pass@1 performance of CDMArk under different message lengths.